Incorporating Nodal and Zonal Room Air Models into Building Energy Calculation Procedures

by

Brent T. Griffith

B.S. Mechanical Engineering
University of California, Berkeley 1989

SUBMITTED TO THE DEPARTMENT OF MECHANICAL ENGINEERING IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF

MASTER OF SCIENCE IN MECHANICAL ENGINEERING
AT THE
MASSACHUSETTS INSTITUTE OF TECHNOLOGY

JUNE 2002

Signature of Author:_____
Department of Mechanical Engineering
May 10, 2002

Certified by:_____
Qingyan Chen
Associate Professor of Architecture
Thesis Supervisor

Certified by:_____
Leon Glicksman
Professor of Mechanical Engineering and Architecture
Faculty Reader

Accepted by:_____
Ain Sonin
Professor of Mechanical Engineering
Chairman, Committee for Graduate Students

Incorporating Nodal and Zonal Room Air Models into
Building Energy Calculation Procedures

by

Brent T. Griffith

Submitted to the Department of Mechanical Engineering
On May 10, 2002 in Partial Fulfillment of the Requirements of the Degree of Master of Science in Mechanical Engineering

ABSTRACT

This research focuses on developing a framework and computer code for coupling detailed air models with building energy and load calculations that can be distributed as a toolkit. The complete-mixing model for room air has long been used for such calculations although there are concerns that it might be deficient when air temperature is not uniform because of buoyancy-driven airflow. Nodal models developed by Mundt (1996) and Rees (1998) were implemented, and a new momentum-zonal model developed, for testing the framework with load-calculation routines based on Pederson's (2001) implementation of the Heat Balance Model. The Heat Balance Model has been reformulated to use an array of zone air temperatures. The coupling framework defines near-surface air temperatures for surface convection calculations and determines system flow rates using model predictions for temperature at the air system returns and a room air control location. The toolkit facilitates incorporating room air modeling into building simulation software and provides a convenient method of testing different air models with Pederson's load routines. The toolkit contains a versatile test program that performs detailed, hourly load calculations for a single thermal zone where both network and three-dimensional air flow models have been tightly coupled to the load calculation routines. In running the test program, the effect of air models on sensible load was found to be minor except where modeling involved aggressive diurnal thermal mass strategies or operative temperature controlling. Displacement ventilation nodal models, such as Rees and Haves, appear practical to implement in load and energy programs and should improve results for air system flow rate and return air temperatures. Results show increases of about a factor of four in computing time for nodal models compared to the mixing model. A momentum-zonal model was developed based on a finite-volume formulation of the Eüler equation for inviscid flow and found to produce reasonable results with coupled computing time increases of about a factor of 100 or more. This research also investigates pressure-zonal air models by Lin (1999) and Inard et al. (1996) and finds that the non-linear formulations appear problematic for building simulation.

Thesis Supervisor: Qingyan Chen
Title: Associate Professor of Architecture

TABLE OF CONTENTS

SUMMARY

This thesis forms a report for Research Project –1222 funded by the American Society of Heating Refrigeration and Air-Conditioning Engineers, or ASHRAE. The report will serve as documentation and content for a computer media publication with a working title of "Air Model Toolkit." This research project focuses on developing a framework for coupling models that predict air temperature variations to building energy and load calculations. The toolkit is intended as a helpful extension to the Loads Toolkit (Pederson 2001) that facilitates testing/evaluating different models for room air. In addition to this report, the computer files of code, data dictionaries, and test cases provide more direct and complete documentation of how the models have been implemented in fortran90. The toolkit contains a versatile test program (called AirModelLoads.exe) that performs detailed, hourly load calculations for a single thermal zone where both network and three-dimensional air flow models have been tightly coupled to the load routines. The original request for proposals for this project focused on simpler nodal models, however the project scope was expanded to include three-dimensional zonal models so that the framework would be more versatile.

A reformulation of the Heat Balance Model is presented that allows for simple coupling of loads calculations and room air models. The air in the room is assumed to be a collection of separate, essentially well-mixed, control volumes where each are modeled as having,

- Uniform state conditions such as temperature and pressure
- Constant properties such as density and specific heat
- Transparency to radiation
- Uniform distributions of heat and mass transfer at control volume boundaries

The surfaces of the room are modeled as having,

- Uniform surface temperatures
- Uniform irradiation
- Diffusely emitted radiation
- One-dimensional heat conduction within

In the same way that individual surfaces are modeled as separate objects in the Heat Balance Model, it is possible to divide the room air into regions and model them as separate parts of the room air. An air model accounts for movement of air between these regions, or control volumes as it predicts a distribution of air temperatures. The surfaces of the thermal zone are treated in the usual time-dependent manner except that instead of all of them interacting with a single air control volume they each interact with a specific nearby control volume of air -- termed the "adjacent air control volume." This project uses an adjacent air method to determine the reference temperature used in calculating surface convection heat transfer rather than the room average or supply air. Furthermore,

it is suggested that values should correspond to the air temperature at 0.1 m from the surface (or the modeled temperature that most closely corresponds to such a location) and future convection correlations for buildings use this reference location.

In order to test the framework, air models from the literature were selected for implementing and one new model developed. The following table summarizes the room air models investigated in detail for this project.

Components of the Air Model Toolkit

Toolkit Components	Reference	Non-mixing Applications	
Mundt model	Mundt 1996	Displacment Ventilation for Cooling Load	
Rees and Haves nodal model	Rees and Haves 2001	Displacment Ventilation for Cooling Load	
(Inard pressure-zonal model)	Inard 1996	Natural Convection, various	
POMA Pressure-Zonal Model	Lin 1999	various	
Momentum-Zonal Model	(RP-1222)	various	

A summary is provided of efforts to validate and verify that program results and internal variables are physical, however suitable measured data for a true validation of the coupled air and loads models are not available. Results of air models and the coupled models compare well to steady state measurements.

The Mundt model was found to be simple to implement and to do an adequate job of predicting floor air temperature, but it generally under predicts air temperatures at a control location of 1.1 m from the floor. The Rees and Haves model was found to be straightforward to implement using non-linear equation solving library routines and behaves well in situations where most of the load is from internal sources. The Inard pressure-zonal model was not successfully implemented and found to be overly difficult for off-the-shelf non-linear solvers to obtain sufficiently converged solutions. Similarly the POMA model was found to function but usually the solver did not converge. From this effort it was determined that a linearized formulation of pressure-zonal model might have been a better selection. The momentum-zonal model is suggested as useful for generating temperature fields in a room air that is not well mixed. The model is based on finite-volume, staggered-grid solution to the Euler equation with Boussinesq approx.

Application test cases for cooling and heating design-day calculations were run in order to explore results and the effects of some algorithm control parameters for both the room air models and their coupling to loads calculations. With constant room air temperature set points, air models were found to have only a minor affect on the overall cooling loads. But for displacement ventilation systems, there was about a 25% effect on air system

flow rates. The results for individual surfaces do vary with the additional modeling detail but in aggregate losses and gains often even out. However when intentionally scheduling room air temperatures to take advantage of diurnal thermal mass in the building surfaces, the air models were found to have a more affect on the overall cooling loads than when room air setpoints are steady. It appears practical to use room air models to predict the air temperature at the location of the thermostat and where it enters the returns in order to provide more detail on how the thermal zone is represented to plant models. The time required for computations increased by about a factor of 4 for the nodal models compared to the mixing model. Nodal models could be added to whole building simulations with moderate additional computation time required and could be expected to improve predictions of flow rate and return air temperatures experienced by the plant. For the momentum-zonal model with a grid number of 216, the increase in computation time was about a factor of 100 longer than the mixing model.

NOMENCLATURE

h_c	[W/m^2·K]	Surface Convection heat transfer (film) coefficient
T_{a_i}	[ºC, K]	Air drybulb temperature for near-surface
DT_a	[ºC, K]	Difference from reference air temperature for near-surface
T_{supply}	[ºC, K]	Air drybulb at air system inlet
$T_{leaving}$	[ºC, K]	Air drybulb at return air extract location
T_{return}	[ºC, K]	Air drybulb in return after lighting system's split
$T_{exhaust}$	[ºC, K]	Air drybulb at ventilation exhaust (not back to air handler)
$\dot{Q}$	[W]	Heat flow rate
$\dot{Q}_c$	[W]	Heat flow rate from surface to air from convection
$\dot{Q}_{sys}$	[W]	Air system sensible heat load, negative indicates cooling
$\dot{Q}_{Infil}$	[W]	heat load from infiltration air
$T_{sysDiff}$	[ºC, K]	Difference between air system inlet and outlet
A	[m^2]	Surface area
T_{StatDB}	[ºC, K]	Air drybulb predicted by model at "thermostat"
T_{op}	[ºC, K]	Operative temperature for comfort
$T_{Setpoint}$	[ºC, K]	Air drybulb (desired) control set point value
$\dot{V}$	[m3/s]	Air system flow rate
ρ	[kg/m^3]	Air density
c_p	[J/kg·K]	air specific heat at constant pressure
$\dot{m}$	[kg/s]	mass flow rate
$\dot{Q}_{conv,s}$	[W]	Convection heat gain from internal sources
$Q_{rad,s}$	[W]	Radiation heat gain from internal sources
T_{node}	[ºC, K]	Air drybulb at a particular node
T_s	[ºC, K]	Surface Temperature at Inside Face
T_{so}	[ºC, K]	Surface Temperature at Outside Face
T_{outBC}	[ºC, K]	Surface Boundary Condition for Outside Face
W	[kg-H_20/kg-Air]	Humidity ratio
C_d	[ ]	Discharge Coefficient
F	[varies]	Vector of functions to be evaluated
J	[varies]	Jacobian matrix
i	[]	individual surfaces
j	[]	current time step
k	[]	time history steps
Y_i	[]	cross CTF coefficients
X_i	[]	outside CTF coefficients
Φ_i	[]	flux CTF coefficients
Z_i	[]	inside CTF coefficients
q''_{ko}	[W/m^2]	conduction heat flux on outside face
$q''_{\alpha\, sol}$	[W/m^2]	absorbed direct and diffuse solar radiation

q''_{LWR}	[W/m^2]	net long wavelength radiation flux exchange with air/surroundings
q''_{conv}	[W/m^2]	surface convection flux with outside air
h_{co}	[W/m^2·K]	outside face surface convection coefficient
q''_{ki}	[W/m^2]	conduction heat flux at inside face

Acronyms

VAV	Variable Air Volume
CFD	Computational Fluid Dynamics
RANS	Reynolds-Averaged Navier Stoke equation
ASHRAE	American Society of Heating, Refrigerations, and Air-Conditioning Engineers, Atlanta, GA
RP	Research Project (ASHRAE)

1. INTRODUCTION

Energy and load calculation procedures have relied on the assumption that zone air is thoroughly mixed for some thirty years. The application of a single control volume with a uniform zone air temperature at any point in time is reasonable for typical forced air system configurations where relatively good mixing is a design intent. More detailed room air modeling is interesting because the "well-stirred" zone model might be inadequate for some system designs or operating modes including:

- Air system off
- Displacement ventilation
- Under-floor air distribution
- Chilled beams
- Natural ventilation
- Mixed-mode ventilation
- Baseboard and convective heating
- Large or tall spaces, such as atria, auditoria and stairwells.

In such systems, the design may explicitly rely on the non-uniformity of the zone air temperature to improve energy efficiency and/or indoor air quality. Or conventional forced-air systems may be switched off when unoccupied or serve large, difficult-to-mix spaces so that the natural buoyancy-driven flow of warm/cold air causes non-uniform zone air temperatures. If a room is not uniform, then perhaps additional detail on room air temperature will alter the result of load and energy calculations. Or perhaps models will deliver better data on diurnal dynamic behavior with regard to thermal storage in the building mass, HVAC system return air dry bulb temperatures, and the spatial distribution of parameters affecting thermal comfort. Building rating systems need fair and accurate methods of simulating designs that are materially different from conventional forced-air systems yet also meet needs for space conditioning. Comfort data throughout a zone can be used to assess the cost/performance tradeoffs between different design options such as using perimeter zone equipment or improving the façade components (better windows) and eliminating perimeter equipment. Current European trends using natural and hybrid ventilation may influence wider adoption of such design measures into some North American buildings thereby creating a need for ASHRAE engineers to model them with confidence.

The potential advantages of combining detailed air modeling with load calculations go both ways. From the point of view of airflow prediction, it is extremely helpful to have good values for important boundary conditions such as wall temperatures, internal loads, inlet flow rates as well as a framework that automatically updates these boundary conditions for different times of the day.

1.1 Research Project Goals

The term *Heat Balance Model* refers to a procedure put forth by the ASHRAE community for calculating cooling load and energy performance characteristics of buildings. Our research has three goals related to extending the Heat Balance Model and expanding the modeling detail in programs from the ASHRAE Loads Toolkit (Pederson 2001),

1. revise the Heat Balance Model to move away from the complete-mixing assumption for zone air and develop interface mechanisms that ease efforts to couple the envelope heat balance models to more detailed models for air conditions.
2. document and implement demonstration programs that couple one or more nodal models for displacement ventilation to the routines in the Loads Toolkit,
3. document and implement demonstration programs that couple one or more zonal models for convective heating to the routines in the Loads Toolkit.

Thus, the focus is on documenting models and developing fortran90 routines to allow expanding the number of predicted zone air conditions from order of 1 up to order of 1000 for each thermal zone in a building load or energy simulation. The fortran 90 programming language was stipulated by ASHRAE in order to be consistent with previous toolkits. One starting point for this effort is the body of research termed the Heat Balance Model that has been developed over the years through the ASHRAE research projects (RP-664, RP-875, and RP-987). The other starting point for the project is the similarly large body of work that has developed a variety of methods of modeling room air conditions. By publishing modern Fortran code and supporting documentation, it is hoped that ASHRAE engineers will eventually be able to access more detailed design information if the building has a unique design or operating mode that is not well mixed. The product of this effort is another "toolkit," termed the *Air Model Toolkit,* which is a collection of documentation, source code, programs, input files, and related items. The intent is that this toolkit become an addition to a series of toolkits published by ASHRAE that are related to building energy and load calculations.

This project develops a framework for coupling air models to loads calculations and demonstrates the framework using air models selected from the literature and a model that is a new contribution. Together the different air models offer a wide range of complexity, programming technique, and numerical methods. It is hoped that the coupling mechanisms are set up clearly so that other developers could use the framework to simplify either including the Heat Balance Model into his or her new room air model or including a new surface model to one of the air models. Implementing a set of different air models in the same framework also allows collecting data on execution times in order to base conclusions on whether or not such air models are suitable for load or annual energy calculations.

1.2 Toolkit Organization

This report is organized around the different interests of the toolkit user and the fact that it is likely to be published as a series of linked electronic files. Section 2 presents theory and background that provides key mathematical formulations for the air models and solution techniques. Section 3 continues with theory and background for coupling air models to the surface models of the Heat Balance Method. Section 4 presents and compares model results from efforts to verify, validate, and otherwise characterize them in terms of accuracy and speed. Section 5 explores the consequences of using the various detailed room air models in practice by applying the coupled air and surface models to load calculations of realistic building thermal zones. Chapter 6 summarizes the findings. Appendix A presents information on program source code in the Air Model Toolkit and is meant for a software developer. Appendix B is a User's Guide for the researcher or savvy practitioner who wants to run single-zone simulations using the demonstration programs. Appendix B also contains an encyclopedic reference for the input objects required to run the air model demonstration programs.

The Air Model Toolkit consists of a collection of computer files in addition to this report. The files are contained in the top-level directory `\AirModelToolkit\` as diagrammed in Figure 1.1. Computer files are organized into important sub-directories for documentation, components, and sample programs. Source code for each of the air models is collected in the components group. The coupled air models and surface models are in the sample program group. There are numerous test cases serving as functioning examples of input files. Test case directories contain batch files that assist executing compiled programs located in different directories.

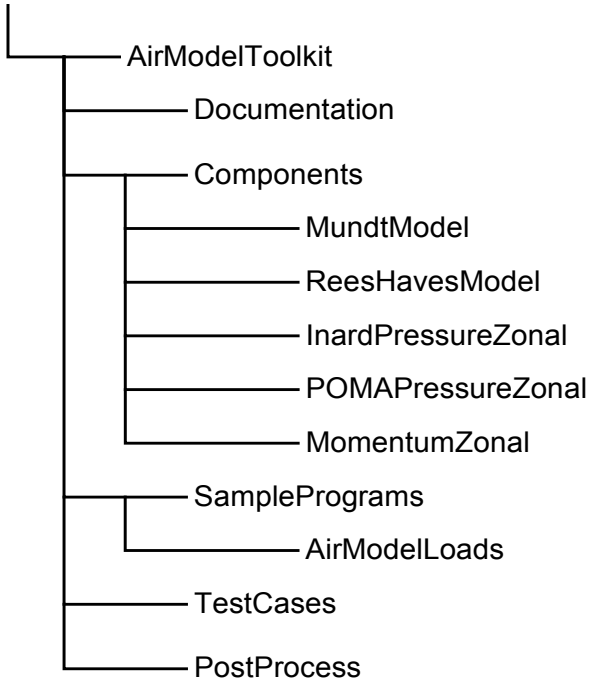


Figure 1.1 Air Model Toolkit Directory Structure

2. ROOM AIR MODELS

This section presents background and theory related to nodal and zonal room air models. Section 3 will discuss the subject of coupling them to loads calculations, but what we going to be interested in obtaining from the air models is a set of predicted values for effective bulk-air temperature, T_{a_i}, adjacent to the *i*th surface of the thermal zone as well as additional temperature values for the air going into the returns and at the control point, or thermostat. The first part of Section 2 presents background information and a discussion of the literature on nodal and zonal room air models for predicting temperature distributions. The following sections present key mathematical formulations for five such models currently selected for the Air Model Toolkit.

2.1 Background

It is necessary to narrow the scope of a review because of the wide variety of fluid flow models in math, physical sciences, and engineering. The decision to focus on so-called nodal and zonal models from the building science literature was made because of the perception that they would offer methods efficient enough for routine annual energy simulation of entire buildings. The range of complexity is between that of the fully mixed model and Computational Fluid Dynamics models, or CFD. Simpler models are appealing because they offer shorter computing times and better numerical stability while more detailed models can be expected to return more accurate temperature distributions.

Building simulations can involve other types of air modeling such as: zone-to-zone air exchanges, naturally-driven ventilation and infiltration, outdoor wind pressure, distribution of indoor air pollutants, convection inside surface constructions, lighting fixture operation, and HVAC equipment and its air handling and distribution systems. All these different categories of air modeling are relevant to buildings, however our focus is on room air models known as nodal or zonal which apply to the air in a single room. These other aspects of air modeling present opportunities to extend the current modeling framework into increasingly detailed building simulation. This is especially true for the popular multi-zone models for room-to-room air movement (Walton 1989, Feustel 1998, Dols 2000). These models determine flow between spaces by solving a linked system of pressure drop (Bernoulli) equations. This effective approach for simultaneous analysis of HVAC system, infiltration, and multi-room airflow problems has been implemented in several building load and energy simulation programs (Hensen 1991, Moser 1992, Kendrick 1993). Axley (2001) reports that it should be possible to apply the methods used in multi-zone programs to flow within a single space. This variation of a zonal model lacked energy balances since the underlying multi-zone program used was arranged for mass balances rather than heat balances. Negaro (1995) coupled CFD to multi-zone air modeling methods by using a pressure drop relation to switch from pressure to mass-flow boundary conditions. Indoor air quality is also outside the scope of this project, but it should be noted that in most instances the air-modeling framework developed in this research could be extended to include additional balances for conservation of air pollutant species and be extended to allow modeling the removal of

gaseous pollutants. This discussion of what we are *not* focusing on is provided to clarify that what we are focusing on -- models for room air in a single space.

Figure 2.1 shows a classification of room air models. The literature in this area uses a variety of terminology for classifying air models. Although the terms *nodal* and *zonal* are used interchangeably, for the purposes of this report, a distinction is made between them. The distinction is basically one of how strictly and how resolved the geometry of the control volumes is defined. A "nodal" model treats the building room air as an idealized network of nodes connected with flow paths. A "zonal" model uses Cartesian grid of well-defined control volumes. In both cases energy balances are solved, but zonal models typically have many more fluid balance relations.

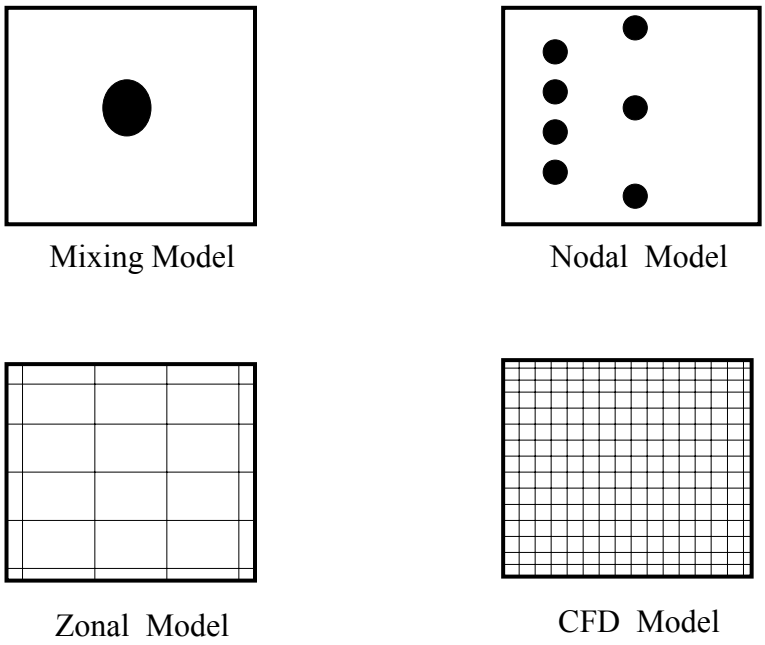


Figure 2.1 Classification of room air models

Such models have been developed for more than 30 years and are plentiful. This makes an effort to categorize them both useful and challenging. Different sub-categories of nodal models exist depending on the application (convective heating, displacement ventilation, natural ventilation) and number of nodes (2-node, 4-node, etc.), and flow configuration (prescribed or modeled). Different sub-categories of zonal models stem from the variables solved such as: temperature-zonal, pressure-zonal, and momentum-zonal. Zonal models are distinguished from CFD in that they are generally quite coarse, do not necessarily attempt an accurate prediction of the flow *field,* and utilize more reduced forms of the governing differential equations. Historically, nodal models were first developed for convective heating applications and so we first discuss these before moving on to cooling applications.

2.1.1 Nodal Models for Convection Heating

Lebrun (1970) was apparently the first to propose a nodal model for providing an estimate of thermal stratification in the context of building energy use. This work focused on modeling how heat gets convected around a room by a heater situation under a cold window. Lebrun's model used 6 nodes connected in a network with nine flow paths using prescribed flow rates. This early work began a series of nodal models that have been summarized by Allard and Inard (1992). Laret (1980) developed an analytical model with four control volumes and a linear distribution of flow in the vertical direction between the core control volume of air and the control volume associated with the heater's plume. This model also allowed subdividing vertical walls when determining surface convection heat transfer. Van der Kooi and Bedeke (1983) used measured airflows to develop a nine-node model for improving room load calculation. Lebrun and Ngendakumana (1987) fixed air flow patterns, and used various empirical laws for different flow components, such as jets, plumes, etc.

Howarth (1980) developed a two-node model of rooms with buoyancy driven flow from convective heaters. The thermal zone is divided into an upper and lower zone and separate heat balances are formulated for each. Air exchange between the upper and lower zones is driven by the upward plume from the heater and is balanced by the downward plume along colder walls as diagrammed in Figure 2.2. Empirical correlations were developed to describe the mass flow in the wall boundary layers and convective surface heat transfer between the heater, room air and walls.

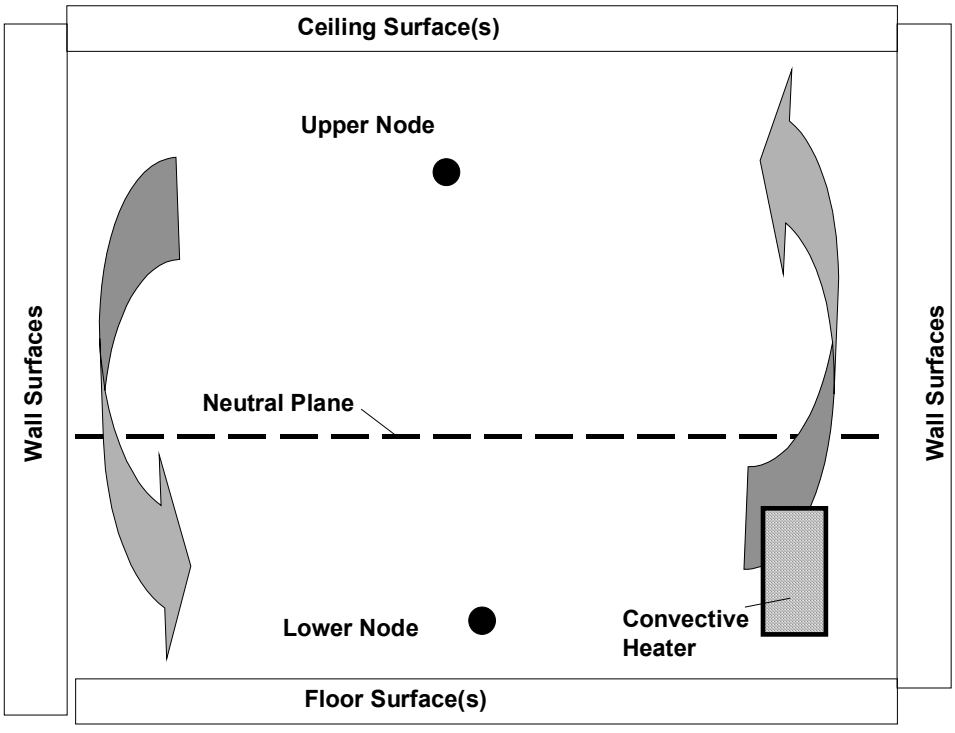


Figure 2.2 Schematic of Howarth 2-Node Model

Inard and Buty (1991) compared Howarth's two-node to a five-node model they developed and found both compared well to CFD results for one case. This five-node model is characteristic of nodal air models for applications where air movement is driving by convection heaters such as baseboard or radiators and is diagrammed in Figure 2.3. Correlations are used do describe the mass flow in the network that is driven by buoyancy forces. The correlations may depend on the specific heater and room geometry and so are difficult to develop for general use.

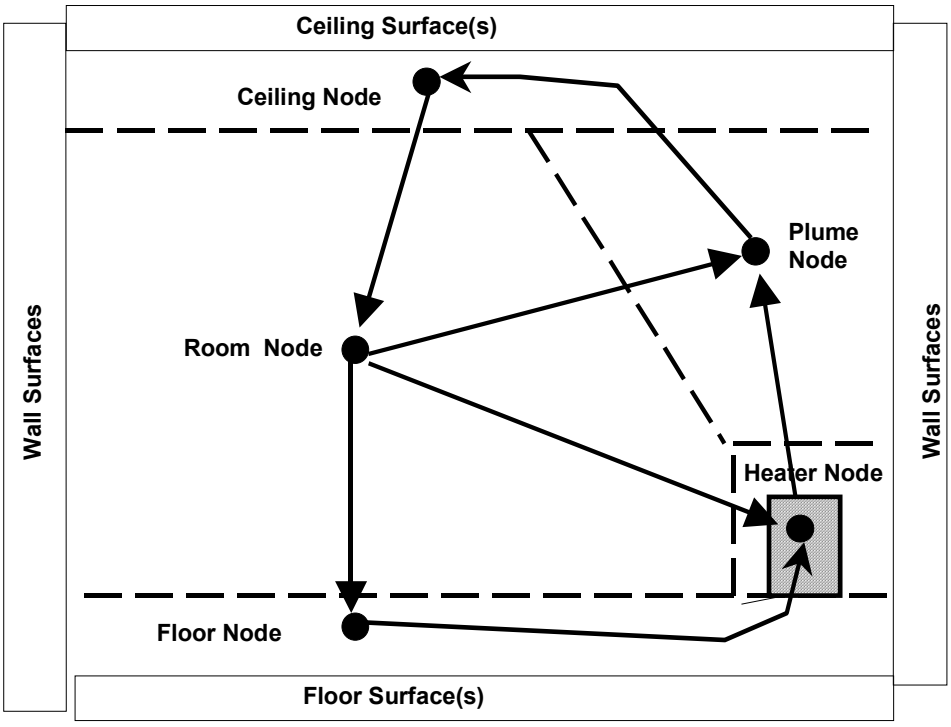


Figure 2.3 Schematic of Inard 5-Node Model

2.1.2 Displacement Ventilation Nodal Models

Displacement ventilation has received considerable attention by researchers in the last decade and models have been developed to describe room air conditions. Here the focus shifts from situations where heating equipment causes buoyancy driven flow to cooling situations where internal loads generate buoyant plumes and the space is conditioned by an air system. Mundt (1996) provides models for predicting the vertical temperature gradient in rooms with displacement ventilation. Mundt's model follows from making these assumptions: (1) the supply airflow conditions are such that supply air is heated to a floor air temperature, $T_{AirFloor}$, by convection surface heat transfer at the floor, (2) there is no entrainment of air from above, and (3) room air temperature can be modeled with linear distribution in the vertical direction. Figure 2.4 diagrams the Mundt model. An idealized displacement ventilation situation is to have supply air evenly distributed just at the floor. The air is warmed by the floor and then migrates only upward. Li et. al. (1993) describes a "four node" model that is closely related to Mundt's extended model and the formulation here. surfaces.

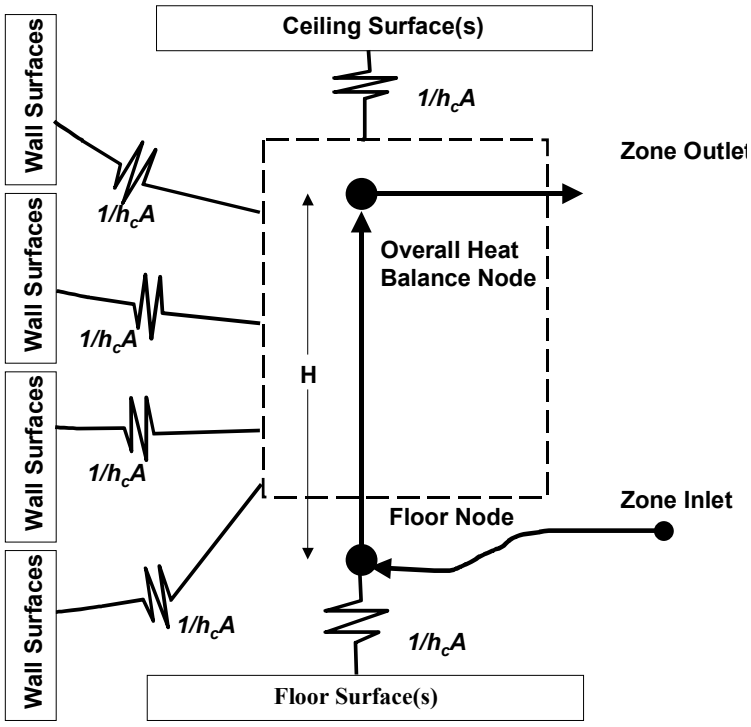


Figure 2.4 Schematic of Mundt model

Rees and Haves (1995) introduced and later revised (Rees and Haves 2001) a nodal model for displacement ventilation and chilled ceilings. This model is more elaborate

than the Mundt model consisting of network of nodes as diagrammed in Figure 2.5. The room is subdivided into a total of nine regions. Three of the regions represent the thermal plume(s) where air is being transported upward by buoyancy from the internal heat sources. Six of the regions represent the bulk air of the room as its various parts interact with the plume and enclosing zone surfaces.

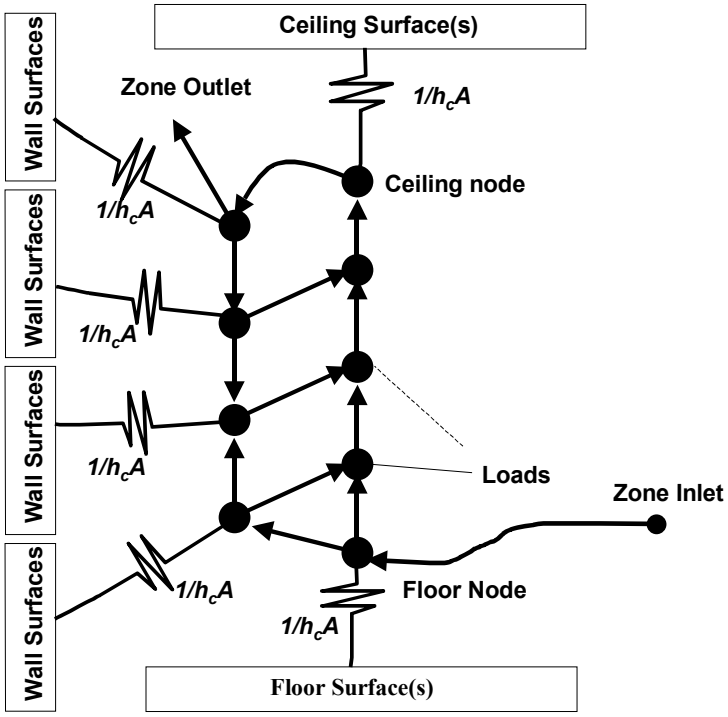


Figure 2.5 Schematic of Rees and Haves Nodal Model

The lower plume air nodes include terms for internal loads. The room air nodes include terms for surface convection. A network describes airflow between the nodes with nodes as the vertices and flow paths as edges. The user prescribes convection coefficients, loads, and mass flow rates.

A limitation of nodal models is that prior knowledge of the airflow patterns is needed in order to specify mass flow in the thermal network. While the flows rates in a nodal network might scale with overall system flow (or switch between various modes with different networks, flow rates, etc.), we generally consider the relative mass transfer along the edges of the network to be fixed for a given application of a nodal air model. Harrington (2001) developed a nodal model for natural ventilation where the model selects between five different airflow patterns based on the Archimedes number. This example shows the possibility of developing a library of flow configurations, and rules for switching between them, in order to expand the applicability of nodal models. Researchers who develop nodal models typically tune the mass flow patterns using experimental data and/or results from more complex models such as CFD. This points to the potential for using more general/detailed room air models to automate generating a

library of nodal model airflow configurations for a specific thermal zone. Another method of expanding the utility of nodal network models would be to adjust path flows by incorporate analytical and empirical relations from fluid mechanics that have been developed for assessing natural ventilation (Linden 1999). But for fixed network flow, an analyst faced with a real project may have good reason to suspect that the zone being modeled for load or energy requirements is very different from the small laboratory chambers or cases used to develop the models. Such model airflow configurations may have been developed for certain combinations of internal source and envelope loads and not apply as well when other ratios of these loads are present. The prior determination of the airflow patterns can be difficult even for an experienced designer.

The original request for proposals focused on models referred to here as nodal, however during the review it was decided to also investigate more complex models because of a lack of generality in nodal models for any room or airflow configuration. An air model may be more useful if it doesn't detract from the wide-ranging applicability enjoyed by the Heat Balance Model. Therefore we also investigated zonal models that are more complex than nodal models (but remain simpler than RANS-CFD) and included them in the framework for coupling to loads calculations.

2.1.3 Zonal Models

Zonal modeling introduces more dynamics into the prediction of mean flows compared to nodal models. In this report, a "zonal" model is considered to use a Cartesian grid to sub-divide the room air into control volumes in the horizontal direction as well as the vertical. In zonal modeling, equations are formulated that attempt to account for how flow rates might change based on temperature differences, length scales, and initial momentum. Each control volume in the grid, or cell, is used to formulate balance equations. Some of the cells are associated with a special driving mechanism because walls, jets or plumes directly affect them. These are termed *special cells*. Special cells have "flow laws" associated with them which are generally simple correlations chosen or generated by the model developers so that model predictions match experiments. Special cell laws can be drawn from the general engineering literature correlations and analytical fluid mechanics and recast in a suitable form. Figure 2.6 diagrams a classification of zonal models where we distinguish between temperature-zonal, pressure-zonal, and momentum-zonal.

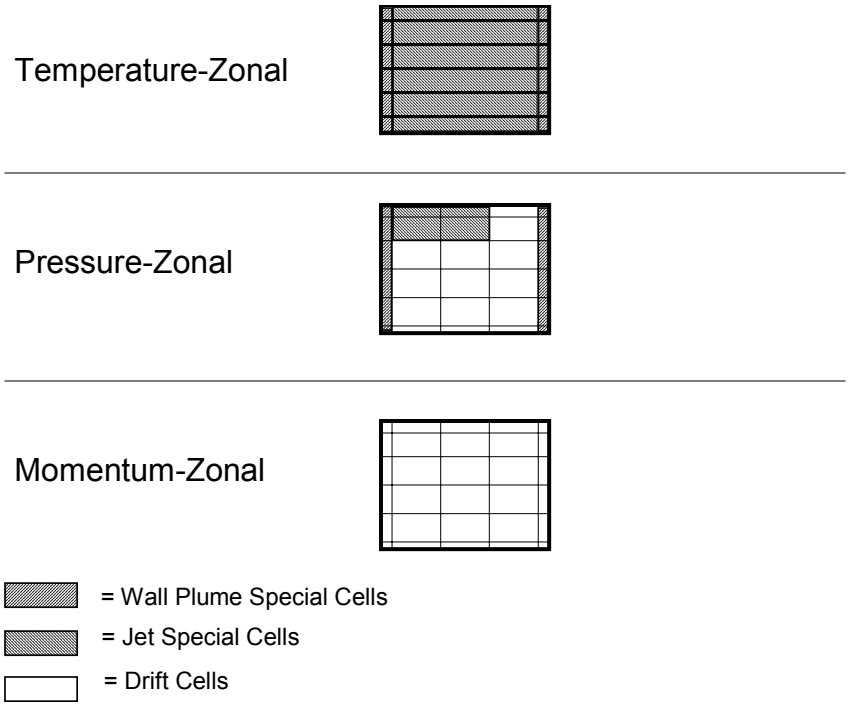


Figure 2.6 Classification of Zonal Models

Temperature-zonal models use empirical correlations based on temperature differences in combination with special laws for flows like jets and plumes. Pressure-zonal models use pressure drop (Bernoulli) relations to compute mass flows between special flows. Momentum-zonal model conserve linear momentum as well as mass and energy and don not necessarily require special cells.

An example of a temperature-zonal model was presented by Togari et al. (1993). This model is intended for use in HVAC applications for large vertical spaces such as atriums. This air model uses all special cells where the main variable is temperature. The space is subdivided into vertical layers and energy balance equations are drawn for each layer. Terms in the balances account for mass flows that vary based on temperatures and account for non-isothermal supply air jets and wall plumes. Togari doesn't explicitly consider a separate control volume near surfaces but presents an adaptive model for wall plumes that can adjust direction and entrainment characteristics. Geshwind et. al. (1996) and Fracastoro et. al. (2001) have also formulated temperature-zonal models. The main problem with such models is that is difficult to arrive at general-purpose coefficients for convective heat transport terms (e.g. $\dot{Q}_i = \alpha(T_{a_1} - T_{a_2})$) that have been developed from a small set of experiments.

Dalicieux and Bouia (1991) were apparently the first to publish what is termed a *pressure-zonal* model that uses pressure as a state variable and solves energy and mass balance equations in the context of building room air modeling. This model began as a two-dimensional model but appears to have initiated considerable research effort over the last decade by a number of researchers in France. Inard, Bouia and Dalicieux (1996)

demonstrated a functional three-dimensional pressure zonal model with special cells for walls, jets and plums that we refer to as the Inard Model. This is a basic non-linear formulation that has been at the heart of much research. The Inard pressure-zonal model is discussed in the Section 2.4. Wurtz and colleagues have published extensively on the subject of pressure-zonal model (Wurtz 1995, Wurtz et al. 1999, Wurtz et. al 2001). Wurtz's earlier formulations were similar to the Inard pressure-zonal model but added second-order resolution of these horizontal flows. For a given control volume the pressure is known to vary hydrostatically and this linear distribution can be modeled rather than assuming a single value for pressure applies to the entire control volume. For control volume faces that are vertical there may be a neutral plane, Z_n, at the height where pressure difference between both sides of boundary is zero. Considering that the mass flow above and below Z_n could be in opposite directions, models can use linear flow distribution across the face rather than a uniform, "top-hat" distribution. This approach is also used by Lin in the POMA program (Lin 1999, Haghigat et al. 2001). The POMA pressure-model is included in the Air Model Toolkit and discussed in Section 2.5.

Development of zonal models and coupling with energy simulation is the subject of ongoing research. Recently, developments to automate the process of applying special laws to cells that drive the flow have been reported by Gagneau and Allard (2001) and Deque et al. (2001) and are implemented in a program called Sim_Zonal. Wurtz et. al. (2001) presented the formulations in this program as linear-pressure zonal models that use sequential solution techniques rather than the direct solutions of the non-linear pressure-formulation. Clearly such research has demonstrated that pressure-zonal models can be applied to energy calculation procedures.

Zonal models offer increased generality compared to nodal models but they are more complicated and therefore more difficult to implement in computer programs. A difficulty in applying pressure-zonal models to general building simulation is the requirement of using special laws to describe flows in certain control volumes. This creates a situation where some characteristics of the flow need already be known prior to the simulation so that special cells can be set up appropriately. The code must also contend with organizing computations that vary for one cell to the next. Thus, expertise is needed by the user to describe the thermal zone. It would also appear that developing comprehensive programs would require a sizeable programming effort that is beyond what can be achieved in a small research project.

These concerns led to formulating a new model that has enough physics in it so that each cell can be treated in the same manner eliminating the need to manage and predetermine application of special cell laws. Section 2.6 presents this variation of a zonal model which is termed "momentum-zonal" because it balances/conserves linear momentum in addition to the mass and energy balances. The momentum-zonal model doesn't necessarily require the use of special cells but may over predict flows by not incorporating turbulence models and the empirical "knowledge" incorporated by the flow laws of special cells.

2.1.4 Models Selected for Toolkit

The wide variety of models available for prediction room air temperatures makes it impractical to implement and test each formulation. The goal of this research is not necessarily to identify "the best" models but rather to develop a framework for using them with the Heat Balance Model. In the case that a particular model includes surface modeling, we are only interested in the air model portion since the Heat Balance Model will already account for surface phenomena. Recognizing that in practice such detailed modeling would most likely to be applied to commercial buildings, efforts focus on cooling load calculations with non-mixing air system designs such as displacement ventilation. Other important criteria applied when selecting models from the literature included whether or not the documentation provided sufficiently complete information on mathematical formulations, values for constants, and verification test cases with results. It is also desirable to implement models with a broad range of complexity.

Five air models were selected for developing code in order to have working examples to couple to the loads routines. These are listed in Table 2.1. The mixing model is preserved and used for baseline comparisons. The Mundt model was selected because its simplicity allows for direct algebraic computation of air temperature distributions. The Rees and Haves model was selected as an example of a reasonably complex nodal and well-documented network model. The Inard model was selected as the classic formulation of a non-linear, pressure-zonal model. The POMA pressure-zonal model was selected because it appears to have a more capable formulation and program code was available. The momentum-zonal model is a new development that is suggested as useful for prediction temperature field without the need to manage special cells within the air domain. The remaining sections will discuss the details of the room air models.

Table 2.1 Summary of air models selected for toolkit

Name	Reference	Main Assumptions
Mixing	Pederson 2001	Complete mixing
Mundt model	Mundt 1996	ideal displacement ventilation, no entrainment ,linear gradient
Rees and Haves nodal model	Rees and Haves 2001	Mass flows pattern
Inard pressure-zonal model	Inard 1996	Inviscid, neglect momentum, special cell laws
POMA pressure-zonal model	Lin 1999	Inviscid, neglect momentum, special cell laws
momentum-zonal model	(this report)	Inviscid, No diffusion

2.2 Mundt model

Mundt (1996) points out that a floor air heat balance provides a simple and reasonably accurate (Mundt 1996, Li et al. 1993) method of modeling the temperature near the floor surface. The slope of a linear temperature gradient can then be obtained by adding a second upper air temperature value that comes from the usual overall air system cooling load heat balance. Figure 2.7 diagrams the temperature distribution versus height being calculated by the model. Mundt also presents an extended model that includes surface nodes, wall heat transfer, radiation in addition to air nodes. As will discussed in Chapter 3, the surface temperatures reside in the surface domain and are to be modeled using the Heat Balance Model's treatment of the inside face of surfaces. Therefore we recast Mundt's model using only room air nodes, but after coupling to loads calculations the capabilities included in Mundt's extended model will be preserved (and improved). Mundt's floor air heat balance is extended to include convection heat gain from equipment and by ventilation or infiltration that may be introduced near the floor in order to maintain all the terms currently in the air heat balance of the Heat Balance Model. This yields the following heat balance for a floor air node,

$$\rho c_p \dot{V}(T_{AirFloor} - T_{SUPPLY}) = h_{cFloor} A_{Floor}(T_{Floor} - T_{AirFloor}) + Q_{ConvSourceFloor} + Q_{InfilFloor} \qquad (2.1)$$

where

ρ is the air density
c_p is the air specific heat at constant pressure
$\dot{V}$ is the air system flow rate
T_{supply} is the air system's supply air drybulb temperature
h_{cFloor} is the convection heat transfer coefficient for the floor
A_{floor} is the surface area of the floor
T_{floor} is the surface temperature of the floor
$Q_{convSourceFloor}$ is the convection from internal sources near the floor (< 0.2 m)
$Q_{InfilFloor}$ is the heat gain (or loss) from infiltration or ventilation near the floor

"Floor splits" are the fraction of total convective or infiltration loads that are dispersed so as to add heat to the air located near the floor. The user prescribes values for floor splits as input. No guidance is known to be available to use in recommending floor splits, but the user could for example account for equipment known to be near the floor, such as tower computer cases, or supplementary ventilation designed to enter along the floor. Equation 2.1 can be solved directly for $T_{AirFloor}$ and is used in this form in the demonstration program implementing Mundt's model. We also allow the floor to be subdivided into an arbitrary number of surfaces in the following equation

$$T_{AirFloor} = \frac{\rho c_p \dot{V} T_{SUPPLY} + \sum h_{cFloor} A_{Floor} T_{Floor} + Q_{ConvSourceFloor} + Q_{InfilFloor}}{\rho c_p \dot{V} + \sum h_{cFloor} A_{Floor}} \qquad (2.2)$$

The upper air node temperature is obtained by solving the overall air heat balance for the entire thermal zone for the temperature of the air leaving the zone and going into the air system returns, $T_{leaving}$.

$$T_{Leaving} = \frac{-\dot{Q}_{sys}}{\rho c_p \dot{V}} + T_{Supply} . \tag{2.3}$$

where $\dot{Q}_{sys}$ is the air system heat load with negative values indicating a positive cooling load. The vertical temperature gradient or slope, *dT/dz*, is obtained from,

$$\frac{dT}{dz} = \frac{T_{Leaving} - T_{AirFloor}}{H_{return}} . \tag{2.4}$$

where H_{return} is the distance between the air system return and the floor air node assumed to be 0.1 m from the floor and *z* is the vertical distance.

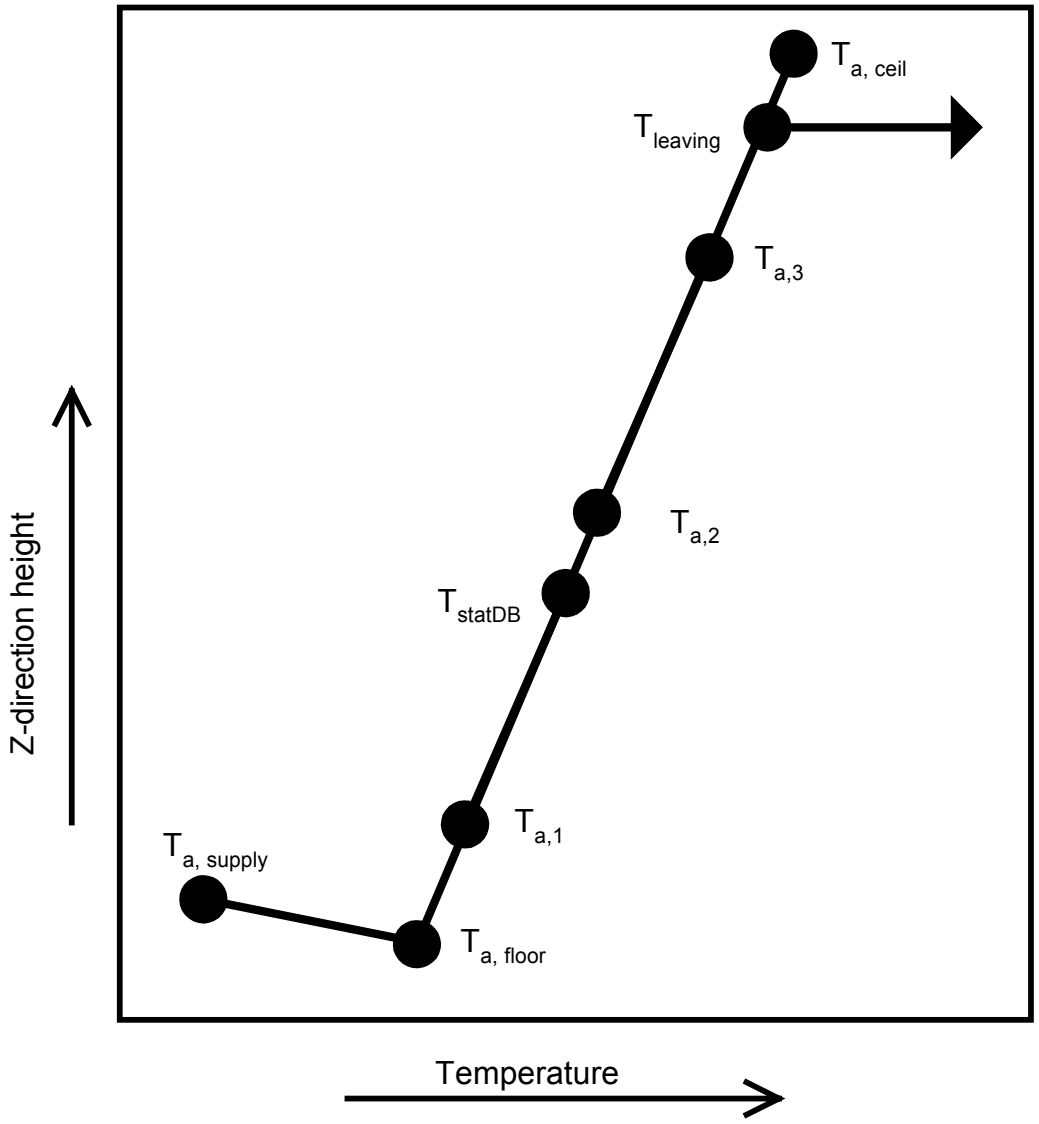


Figure 2.7 Height versus temperature schematic for Mundt model

The constant slope allows obtaining temperatures at any vertical location using,

$$T_{a_i} = T_{leaving} - \frac{dT}{dz}(z_{leaving} - z_i) \tag{2.5}$$

So for example the temperatures near the ceiling can easily be determined. Accounting for the location of the thermostat inside the zone (e.g. 1.1 m) is accomplished by returning the temperature for the appropriate height to the appropriate air node used for control. If the walls are subdivided in the vertical direction as shown in Figure 2.7 then the air model can provide individual values for each surface based on the height and slope. However, no additional heat balances are necessarily made (in the air domain) at these points as all the surface convection is passed to the model in the totaled value for

Q_{sys}. There are schemes to add additional hat can be used to introduce More detailed nodal network models do write additional heat balances as discussed in the next section.

2.3 Rees and Haves model

The Rees and Haves (2001) present a model for room air for displacement ventilation. A network of nodes is used to formulate a system of heat balance equations with one equation for each of nine nodes. The general balance equation is

$$0 = \sum_{in} \dot{m}_i c_p T_{neigh,i} - \sum_{out} \dot{m}_i c_p T_{node} + \sum_{surfs} h_c A(T_{surf} - T_{node}) + Q_{conv,s} \qquad (2.6)$$

where,

$\dot{m}_i$ is the mass flow rate of air along a path in or out of a node
$T_{up,i}$ is the temperature of the upstream node or air system inlet
$Q_{conv,s}$ is the portion of the convection from internal sources for that node

The "last" node in the network is connected to the return air. For this node, equation 2.6 would apply but rather an overall energy balance is written to ensure that it holds,

$$0 = \dot{m} c_p (T_{leaving} - T_{Supply}) + \sum_{j,i} h_c A(T_{j,surf} - T_{i,node}) + \sum_{sources} Q_{j,conv,s} \qquad (2.7)$$

It is difficult to obtain accurate/generalized values for $\dot{m}_i$ along each path of the network. Rees used CFD to generate flow rate values for specific cases and found some general rules to apply (see Table B.6 in Appendix B for values). For convenience, the path flow rates can be organized as non-dimensional factors to be multiplied by the air system flow rate and density (to obtain $\dot{m}_i$). Flow is a one-way coordinate during a given application of this model. The direction of flow in leg is the difference between Rees's "Model A" and "Model B." The balance equations from Equation 2.6 will differ for the two affected nodes for the two models.

Equation 2.6 is written for eight of the nine nodes and with Equation 2.7 form a system of nine equations with nine unknown node air temperatures, T_{node}. The roots of the equations are a set of node temperature values where the system of equation is satisfied. Nodes that are associated with surfaces the provide zone air temperatures as shown schematically in Figure 2.8. The system of is not necessarily non-linear but it desirable to use a non-linear equation solver to allow the use of temperature dependent correlations for convection coefficients, h_c. The system of equations does not appear to be particularly challenging and most multi-dimensional, root-finding solvers would be expected to perform satisfactorily. The demonstration programs have implemented two types of Newton-Raphson solvers. One of the solvers was from *Numerical Recipes* (Press et. al. 1996) and is discussed in detail in Section 2.2.2. The other is the library routine NEQNF from IMSL based on the public domain code from MINPACK known as HYBRID1. This second solver uses a modified Powell technique that is beyond the scope of this research.

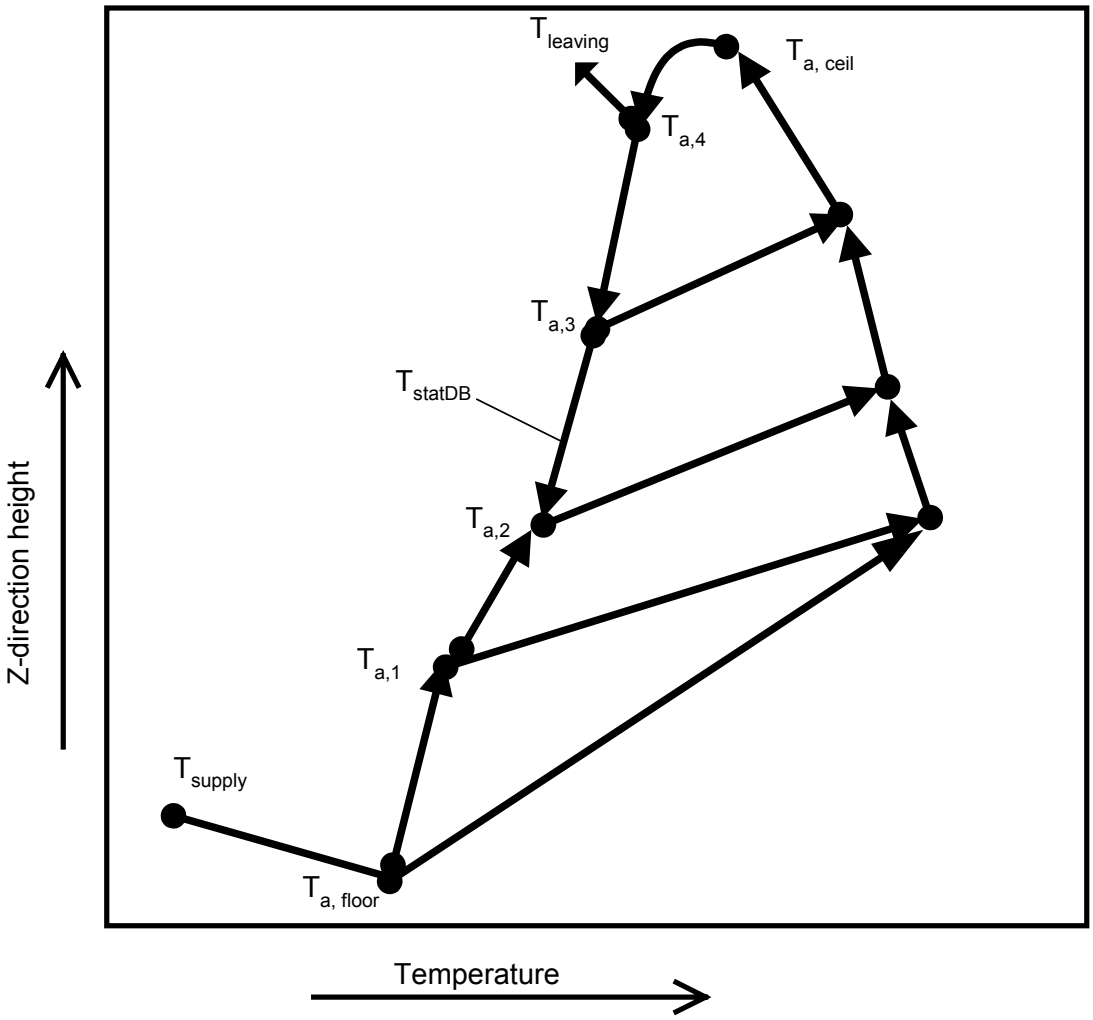


Figure 2.8 Height versus temperature schematic for Rees and Haves model

2.4 Inard Pressure-Zonal

The model presented by Inard, Bouia, and Dalicieux (1996) is referred to in this report as the Inard pressure-zonal model. The room air is subdivided into a structured Cartesian grid of control volumes or "cells" diagrammed in Figure 2.9. Cells that are directly affected by features such as inlet jets, plumes from heat sources, or natural convection wall plumes, are considered "special cells" and use empirical laws to model flow. The intervening cells are termed "drift cells" in this report but have a variety of names in the literature. It is assumed that within a cell the properties are uniform.

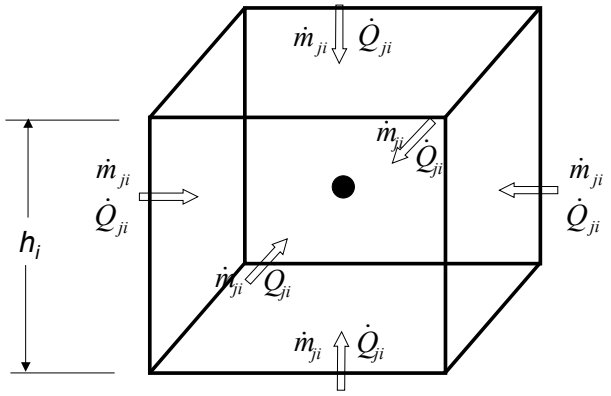


Figure 2.9 Control volume cell

Mass and energy balances are then implemented for each control volume, subscript *i*, considering steady state situations,

$$\sum_{j=1}^{n} \dot{m}_{ji} + \dot{m}_{source} = 0 \tag{2.8}$$

$$\sum_{j=1}^{n} \dot{Q}_{ji} + \dot{Q}_{conv,s} + \dot{m} c_p (T_{a_i} - T_{Supply}) + \sum_{k} h_c A_k (T_k - T_{a_i}) + = 0 \tag{2.9}$$

Where,

ji subscript indicates flux from neighbor cell *j* into this cell *i*
k subscript indicates surfaces that might be in contact with cell
$\dot{m}_{source}$ is inlet or outlet that might be present
T_{a_i} is the temperature of cell *i*

The model computes mass and energy flux terms for each face of a control volume. The mass flow into or out of cell, $\dot{m}_{ij}$, is modeled using pressure drop relations that use a discharge coefficient, C_d. Values for C_d were originally obtained from studies of flow through open doorways (large area openings) and are usually taken as 0.8 or 0.83. Axely (2001) questioned this formulation and showed that such modeling introduces non-physical and grid-dependent pressure drops. Different relations are used depending on the orientation of the cell's control volume face. For vertical faces with horizontal flow the mass flux terms for use in Equation 2.8 are calculated using,

$$m_{ji} = \pm \sqrt{2\rho_i} C_d A_{ji} \left| P_i - P_j \right|^{\frac{1}{2}} \tag{2.10}$$

where,

P is the cell pressure

For horizontal faces with vertical flow the pressure term must account for the hydrostatic pressure difference and the mass flux term is calculated using,

$$\dot{m}_{ji} = \pm\sqrt{2\rho_i} C_d A_{ji} \left| P_i - P_j - \frac{1}{2}(\rho g h_j + \rho g h_i) \right|^{\frac{1}{2}} \tag{2.11}$$

Where,

g is the acceleration due to gravity
h is the size of the control volume in the vertical direction

The heat flow in or out of a cell face, $\dot{Q}_{ji}$, is the needed change in enthalpy and is calculated using,

$$\dot{Q}_{ji} = \dot{m}_{ji} c_p (T_j - T_i) \tag{2.12}$$

Inard et. al. (1996) present flow correlations for three types of special cells: (1) horizontal non-isothermal jet adjacent to a wall, (2) thermal plume from a heat source adjacent to a wall, and (3) wall boundary layer flow arising from surface convection heat gain or loss. For example the wall boundary layer flow is calculated using,

$$\dot{m}_{ji} = 0.004 l_i H_{ji} (T_{surfIn} - T_i)^{1/3} \tag{2.13}$$

Where,

l_i is the depth of cell parallel to the wall
H_{ji} is the distance from the top or bottom of the wall to the cell face

The balance equations (2.8 and 2.9) form a system of equations where the goal is to find a set of values for temperature and pressure that satisfy them. The square root dependence on pressure difference and the large number of unknowns makes this numerically challenging. Once a solution is obtained the temperature values for cells adjacent to surfaces are used to generate effective bulk air temperatures, or $T_{a,i}$'s, for use in surface convection calculations. Also the cell associated with the location of a thermostat is used to provide a prediction of the thermostat reading, T_{statDB} and the cell with an air system outlet provides a prediction for the temperature of air entering the returns, $T_{leaving}$.

2.5 POMA Pressure-Zonal

Lin (1999) presents a variation of the pressure-zonal model called POMA, for **P**ressurized z**O**nal **M**odel with **A**ir diffuser (Haghighat et al. 2001). The model is similar to the Inard model and uses the same basic power law formulation for airflow as a function of pressure difference. POMA can be distinguished by its use of more refined horizontal flow, different solution techniques, and different special flow laws. The main

assumptions are that the air is inviscid, each control volume has uniform temperature and density. POMA however models pressure by using a reference at the bottom of each control volume and allowing it to vary hydrostatically in the vertical direction.

The flow across a control volume face is determined by the pressure difference using a power law,

$$\dot{V} = C_d \Delta P^n A \tag{2.14}$$

where,

$\dot{V}$ volume flow rate, (m^3/s)
ΔP: pressure difference, (Pa)
C_d: coefficient of power law, usually taken as 0.83, ($m/s\ Pa^n$)
A: area of boundary, (m^2)
n: flow exponent, usually taken as 0.5.

The application of Equation 2.14 varies depending on the orientation of the control volume face, the presence a neutral plane, and any association with special cells. Lin (1999) describes the various possibilities for the ΔP that lead to a variety of equations based on Equation 2.14.

POMA introduces more detail in the determination of horizontal flow between two drift cells. For a given control volume a linear pressure distribution is assumed. For control volume faces that are vertical there may be a neutral plane, Z_n, at the height where pressure difference between both sides of boundary is zero. Considering that the mass flow above and below $\mathbf{Z}_n$ could be in opposite directions, the model uses a linear flow distribution across the face rather than a uniform, "top-hat" distribution as shown in Figure 2.10.

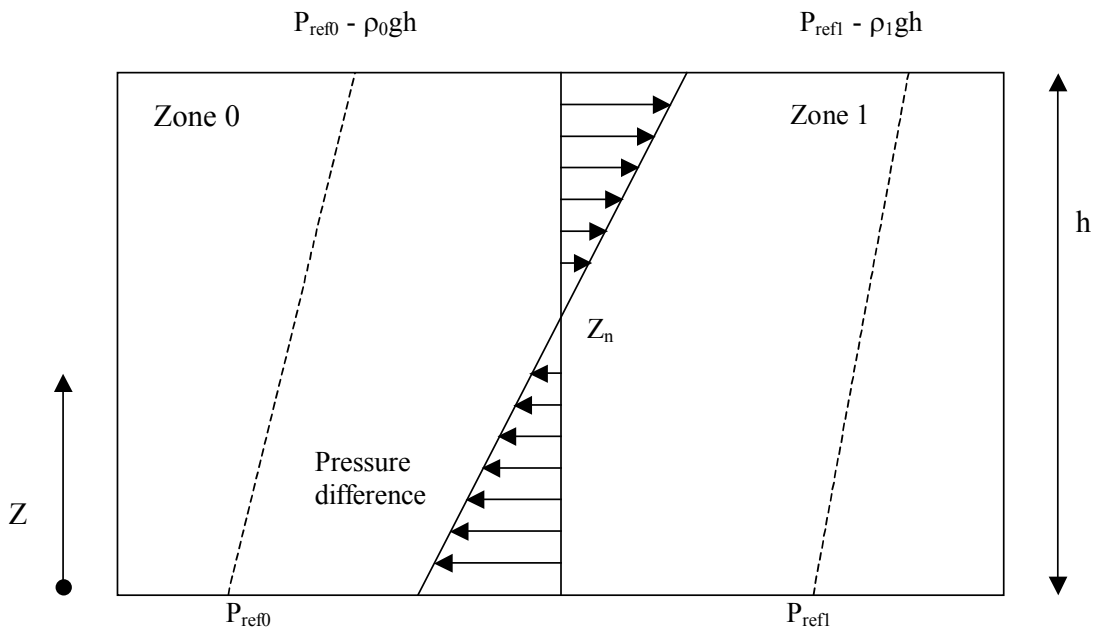


Figure 2.10 POMA modeling of vertical boundaries

Applying Equation 2.14 and integrating along the cell face yields,

$$\begin{aligned} m_{0-Zn} &= \int_0^{Zn} C_d \, L \, \rho_{0-Zn} \left|\Delta P\right|^n dz \\ &= \int_0^{Zn} C_d \, L \, \rho_{0-Zn} \left|\Delta\rho \, g \, (Z_n - Z)\right|^n dz \\ &= C_d \, L \, \rho_{0-Zn} \left|\Delta\rho \, g\right|^n \left[\frac{\left|Z_n\right|^{n+1}}{n+1}\right] \end{aligned} \tag{2.15}$$

$$\begin{aligned} m_{Z_n-H} &= \int_{Zn}^{h} C_d \, L \, \rho_{Zn-H} \left|\Delta P\right|^n dz \\ &= \int_{Zn}^{h} C_d \, L \, \rho_{Zn-H} \left|\Delta\rho \, g \, (Z_n - Z)\right|^n dz \\ &= C_d \, L \rho_{Zn-H} \left|\Delta\rho g\right|^n \left[\frac{\left|Z_n - H\right|^{n+1}}{n+1}\right] \end{aligned} \tag{2.16}$$

where, $m_{0\text{-}Zn}$: mass flow rate from 0 to Z_n on vertical boundary, (kg/s),
$m_{Zn\text{-}h}$: mass flow rate from Z_n to h on vertical boundary, (kg/s)
$\rho_{0\text{-}Zn}$: density of airflow from 0 to Z_n, (kg/m^3)
$\rho_{Zn\text{-}h}$: density of airflow from Z_n to h, (kg/m^3)
L: depth of zone, (m)

The total mass flow rate across the vertical boundary ($m_{o\text{-}H}$) is:

$$m_{o-h} = m_{0-Zn} + m_{Zn-h} \tag{2.17}$$

For control volume faces located at walls there is no airflow across the face and heat flow is modeled using the usual expression for surface convection,

$$\dot{Q} = hA(T_{S,i} - T_i) \tag{2.18}$$

The movement of air across control volume faces convects energy which is determined using Equation 2.12.

POMA includes special cell flow laws for heat source plumes and jets but not wall boundary layer plumes (Lin 1999)

Newton-Raphson with Line Search and Backtracking

A Newton-Raphson solver from Numerical Recipes (Press et. al. 1996) is used in POMA. POMA implements a double precision version and changed argument passing. Press et al. (1992) present the following discussion of their Newton-Raphson method.

A typical problem gives N functional relations to be zero, involving variables x_i, i=1, 2, ..., N:

$$F_i(x_1, x_2, ..., x_N) = 0 \qquad \text{i} = 1,2,..., \text{N}. \tag{2.19}$$

In the neighborhood of x, each of the functions F_i can be expanded in Taylor series

$$F_i(\mathbf{x} + \delta\,\mathbf{x}) = F_i(\mathbf{x}) + \sum_{j=1}^{N} \frac{\partial F_i}{\partial x_j} \delta x_j + \text{O}(\delta \mathbf{x}^2) \tag{2.20}$$

Where, **X** is the entire vector of values x_i,

In matrix notation, Equation (22) is expressed as:

$$\mathsf{F}(\mathbf{x} + \delta\,\mathbf{x}) = \mathsf{F}(\mathbf{x}) + \mathsf{J} \cdot \delta\mathbf{x} + O(\delta\mathbf{x}^2) \tag{2.21}$$

Where, **F** is the entire vector of functions F_i, and

J is the Jacobian matrix:

$$J_{ij} = \frac{\partial F_i}{\partial x_j} \tag{2.22}$$

By neglecting terms of order δx^2 and higher, and by setting $\mathbf{F}(\mathbf{x}+\delta\mathbf{x})=0$, a full Newton step, which moves each function closer to zero, can be obtained as:

$$\delta \mathbf{x} = -\mathbf{J}^{-1} \cdot \mathbf{F} \tag{2.23}$$

The corrections are then added to the solution vector,

$$\mathbf{x}_{new} = \mathbf{x}_{old} + \delta \mathbf{x} \tag{2.24}$$

and the process is iterated to convergence. However, as well-known, this general Newton-Raphson method will fail if the initial guess is not sufficiently close to the root. Moreover, numerical tests of the Newton-Raphson method solution indicated occasional instances of very slow convergence as the iterations almost oscillate between two different sets of values (Walton, 1993). Thus, a line search and backtracking method was selected to find a proper step of variations.

$$\mathbf{x}_{new} = \mathbf{x}_{old} + \lambda \delta \mathbf{x} \ , \qquad 0 < \lambda \leq 1 \tag{2.25}$$

where, λ is the coefficient for the acceptable step.

The aim is to find λ so that $F_i(\mathbf{X}_{old}+\lambda\delta\mathbf{X})$ has decreased sufficiently. A useful strategy (Press et al, 1992) is: First try the full Newton step, λ=1. This will lead to quadratic convergence when $\mathbf{X}$ is sufficiently close to the roots. If $F_i(\mathbf{X}_{new})$ does not meet the acceptance criteria, we backtrack along the Newton direction, trying a smaller value of λ, until a suitable point is found. Here, the criterion for the acceptable step is:

$$f(\mathbf{x}_{new}) \leq f(\mathbf{x}_{old}) + \alpha \nabla f \cdot (\mathbf{x}_{new} - \mathbf{x}_{old}) \tag{2.26}$$

The choice of λ is based on the following rules. First, try λ = 1, if this step is not acceptable, try

$$\lambda = -\frac{g'(0)}{2[g(1) - g(0) - g'(0)]} \tag{2.27}$$

Where:

$$g(\lambda)=f(\mathbf{X}_{old}+\lambda\delta\mathbf{X}) \tag{2.28}$$

On the second and subsequent backtracks,

$$\lambda = \frac{-b + \sqrt{b^2 - 3ag'(0)}}{3a} \tag{2.29}$$

Where,

$$\begin{bmatrix} a \\ b \end{bmatrix} = \frac{1}{\lambda_1 - \lambda_2} \begin{bmatrix} 1/\lambda_1^2 & -1/\lambda_2^2 \\ -\lambda_2/\lambda_1^2 & \lambda_1/\lambda_2^2 \end{bmatrix} \bullet \begin{bmatrix} g(\lambda_1) - g'(0)\lambda_1 - g(0) \\ g(\lambda_2) - g'(0)\lambda_2 - g(0) \end{bmatrix} \tag{2.30}$$

Where,

λ_1 and λ_2 are the previous and recent value from Equation (31)

More detail of linear searches and backtracking of λ can be found in (Press et al, 1992).

2.6 Momentum-Zonal

Pressure-zonal models assume velocities in drift cells are so low that momentum can be ignored. In formulating a new model this assumption is changed to assert that velocities and viscosity are low enough that the flow is inviscid and we can consider conservation of linear momentum. This project formulates a so-called "momentum-zonal" model by using coarse-grids with finite-volume numerical techniques to solve the Eüler equation. Inard et al. (1996) describe the pressure-zonal model as a simplification of the discretization equation by Patankar (1980) that is still used for numerical solution of the Reynolds Average Navier-Stokes (RANS) equation. The momentum-zonal model uses this discretization equation to solve the steady-state Eüler equation for inviscid flow which is,

$$(\mathbf{V} \cdot \nabla)\mathbf{V} = \frac{1}{\rho} \nabla P + \mathbf{g} \tag{2.31}$$

where, $\mathbf{V}$ is the velocity vector
∇ is the differential operator "del"
$\mathbf{g}$ is the gravity force vector

The momentum-zonal model can be thought of as being situated between Bernoulli-based, pressure zonal models and RANS-based CFD. A simplified "zonal model" is asked to generate a coarse estimate of zone air temperatures. It is not necessary that it completely resolve the flow field but rather give a rough account of mass flows for the benefit of the energy balances. Turbulence is common in room airflow and modeling the air as inviscid neglects viscosity and turbulence. If the airflow has significant turbulence then it should mix rather well and the complete mixing model is likely to be reasonable for building simulation. While not suitable for general modeling (such as with forced air systems) we suggest that momentum model can be used with coarse grids and buoyancy dominated flow situations to produce data that is comparable to the results of pressure-zonal models. At a minimum the model allows testing the coupling framework for zonal models.

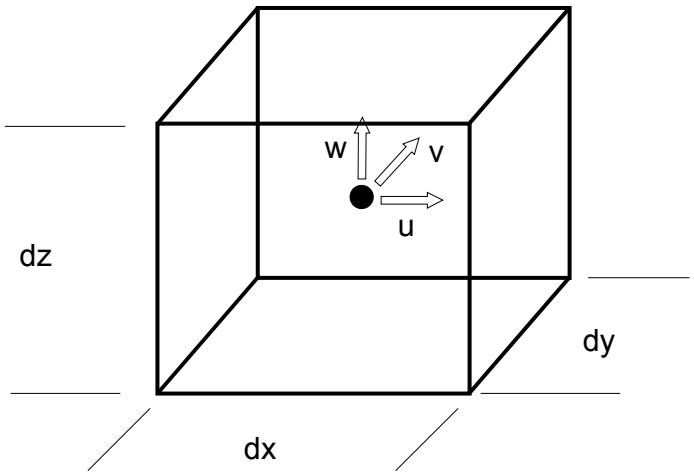


Figure 2.11 Differential element for room air

Figure 2.11 diagrams a differential element of the type used in calculus. The three components of **V** are *u* in x-direction, *v* in y-direction, and *w* in z-direction. Writing the differential form of the Euler equation for inviscid flow in Cartesian coordinates in three-dimensions, along with conservations of energy and mass, forms a system of partial differential equations that are the basis of the momentum-zonal model.

$$
\begin{aligned}
u\frac{\partial u}{\partial x}+v\frac{\partial u}{\partial y}+w\frac{\partial u}{\partial z} &= -\frac{1}{\rho}\frac{\partial P}{\partial x} \\
u\frac{\partial v}{\partial x}+v\frac{\partial v}{\partial y}+w\frac{\partial v}{\partial z} &= -\frac{1}{\rho}\frac{\partial P}{\partial y} \\
u\frac{\partial w}{\partial x}+v\frac{\partial w}{\partial y}+w\frac{\partial w}{\partial z} &= -\frac{1}{\rho}\frac{\partial P}{\partial z}-\beta(T_{ref}-T_{a_i})g \qquad \text{(2.32 a-e)} \\
u\frac{\partial T_{a_i}}{\partial x}+v\frac{\partial T_{ai}}{\partial y}+w\frac{\partial T_{a_i}}{\partial z} &= \frac{1}{c_p}\dot{Q}_{s,conv}+\frac{hA}{c_p}(T_s-T_{a_i})+\dot{m}(T_{ai}-T_{Supply}) \\
\frac{\partial u}{\partial x}+\frac{\partial u}{\partial y}+\frac{\partial u}{\partial z} &= 0
\end{aligned}
$$

where β is the expansion coefficient of air and
T_{ref} is a reference temperature somewhere in the domain

The main assumptions and approximations include: incompressible and inviscid flow, Boussinesq approximation for buoyancy, and fluid at thermodyamic equilibrium. The momentum-zonal model finds solutions to the system of differential equations in Equation 2.32 by using of finite-volume numerical techniques. An iterative and sequential approach is as an important alternative to the direct, simultaneous solutions

sought by the non-linear solvers in the pressure-zonal models discussed in Section 2.5. Equation 2.32 is recast into a simpler form using the generic flow property ϕ and Einstein's notation,

$$\rho u_i \frac{\partial \phi}{\partial x_j} = S \qquad (2.33)$$

Where,

f is u, v, w, or T

S is a source term as for the right-hand sides in Equation 2.32

The flow domain is descretized using a structure grid and flow properties are solved for each grid point. The temperature is assumed to apply to the entire control volume centered on that grid point. The velocities apply to the control volume faces because of the staggered grid formulation (but can be reported at cell centers). The usual nomenclature uses subscript "P" to refer the current grid point being evaluated, subscript "E" to refer to the point to the east, "W" to the west, "N" to the north, "S" to the South, "T" to the top and "B" to the bottom.

After integrating Equation 2.33 over the chosen control volume, the descretization equation in three-dimensions can be obtained,

$$a_p \phi_p = a_E \phi_E + a_W \phi_W + a_N \phi_N + a_S \phi_S + a_T \phi_T + a_B \phi_B + b \qquad (2.34)$$

where,

$$a_E = MAX(-F_E, 0)$$
$$a_W = MAX(F_W, 0)$$
$$a_N = MAX(-F_N, 0)$$
$$a_S = MAX(F_S, 0)$$
$$a_T = MAX(-F_T, 0)$$
$$a_B = MAX(F_B, 0)$$
$$b = source\ terms + false\ timestep\ term$$
$$a_P = a_E + a_W + a_N + a_S + a_T + a_B - source\ term$$

The formulation expressed in Equation 2.34 reflects purely convective transfer using the upwind technique.

The flow rates are calculated from,

$$
\begin{aligned}
F_E &= (\rho u)_E \Delta y \Delta z \\
F_W &= (\rho u)_W \Delta y \Delta z \\
F_N &= (\rho v)_N \Delta z \Delta x \\
F_S &= (\rho v)_S \Delta z \Delta x \\
F_T &= (\rho w)_T \Delta x \Delta y \\
F_B &= (\rho w)_B \Delta x \Delta y
\end{aligned}
\qquad \text{(2.35 a -f)}
$$

The source term, *b*, includes pressure forces arising from buoyancy using the Boussinesq approximation. The so-called "false time step" is a form of relaxation that gets its name from having units of time and is used to improve model performance with buoyancy driven flows. A staggered grid pressure correction technique is used to solve for a pressure distribution that is consistent with mass continuity and the momentum conservation. The contribution of normal diffusion is neglected in the formulation thereby saving computation. False numerical diffusion is expected with coarse grids.

Boundary conditions are applied to eliminate flow along the domain boundary (at the outer walls) and at blockages inside the flow domain (if desired). These enclosures also provide surface convection to the cells using temperature boundary conditions. Inlet and outlets may also be defined as well as heat flux boundary conditions. The SIMPLE (**S**emi-**I**mplicit **M**ethod for **P**ressure-**L**inked **E**quations) algorithm is used. The overall computation strategy is shown in Figure 2.10. Section A.8 (in Appendix A) provides more information on the momentum-zonal code developed for the Air Model Toolkit.

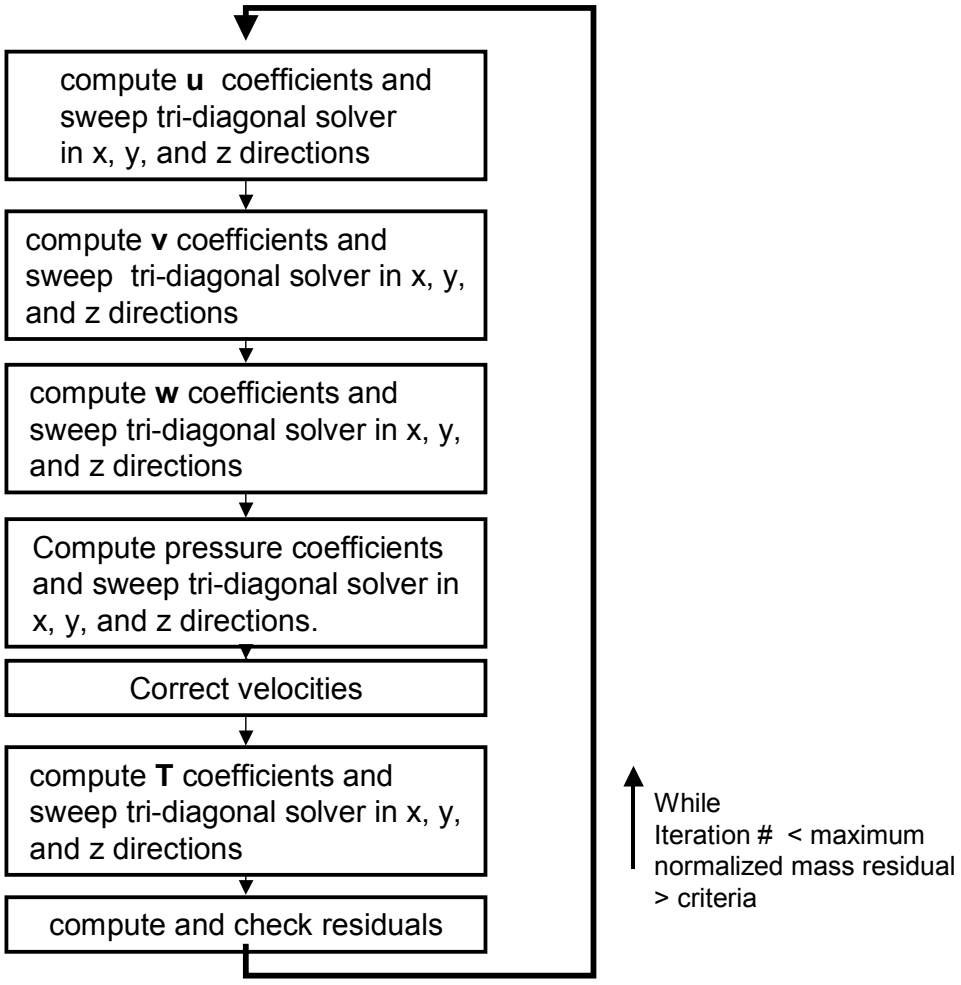


Figure 2.10 Computation strategy in momentum-zonal model

3. COUPLED AIR AND LOADS MODELS

This chapter discusses using air models in the context of load and energy calculations for buildings. The expression *coupled air and loads model* refers to combining separate room air models with the Heat Balance Model. Section 2 (above) discussed theory related to various air models; here the focus shifts to providing boundary conditions for the air models. This coupling is of interest because it might improve load and energy calculations by accounting for variation in zone air temperatures. Or on the other hand, in might improve air model predictions for comfort and air quality by providing more realistic boundary conditions for surface temperatures and air system flow rates.

A modular approach is used and recommended for building simulation (Sowell 1991, Crawley 2001). It is desirable to reduce the overall problem into smaller modules and connect them with manageable interfaces in order to simplify the construction and maintenance of computer code. The research reported here uses the air-surface boundary at the inside face of the enclosing walls, floors, ceilings as the point to separate two physical domains. Air modeling takes place in a separate part of the program than does the surface modeling. A different approach would be to combine surfaces and air models into one larger modeling framework such as conjugate CFD or thermal networks and find solutions in both domains at the same time. The separating of air and surface domains is done to facilitate developing a modeling framework and because ASHRAE requested we use the existing capabilities of the *Loads Toolkit* (Pederson 2001) with minimal changes.

This chapter is organized in the following manner. The first section notes guidance obtained from the work of other researchers who coupled building load and energy calculation with detailed room air models. The second section discusses the (original) Heat Balance Model and then presents a reformulation suitable for coupling to room air models. The third section discusses implementing the combined models using iterative algorithms. The fourth section discusses surface convection at the inside face. The fifth section discusses air system flow rates. The final section discusses some implications of the coupled modeling approach and assumptions.

3.1 Prior Coupling of Loads/Energy and Airflow Calculations

This section mentions some of the research that has been conducted in this area. Such efforts are not unique and methods consistent with the Heat Balance Model have already been coupled to room air models. The discussion is presented to show that such coupling is feasible and that prior research provides useful guidance.

Detailed zone models of the thermal network type are available with both mass and energy balances and already offer the capabilities envisaged when coupling of nodal room air models to the *Loads Toolkit* (Sowell 1991, Walton 1993). These models represent the thermal zone with both surfaces and air nodes in a single network. The models present a single representation of the thermal zone to an HVAC component simulation. While probably uncommon in practice, researchers and engineers have long

had the ability to formulate detailed network models of a thermal zone and solve them using a variety of software tools such as SPARK (1997) or IDA (EQUA 2002).

Computational fluid dynamics, or CFD, has been used to model building room airflow for more than 25 years. CFD has been applied in research-class building simulations aimed at improving the treatment of surface convection. While earlier research did not actually combine the models into one computer program, more recent research has demonstrated the feasibility of coupling modules that implement room air models to the other models for building load/energy simulation. Negaro et. al. (1995) coupled CFD to the building load and energy simulation program ESP-r and later Beausoleil-Morrison (2000) expanded these capabilities. The ESP-r/CFD implementation also couples with multi-zone air models. ASHRAE sponsored research project 927 (Chen et al. 1999) that coupled CFD to the load and energy program ACCURACY (Chen 1988). Both of these efforts implemented so-called "zero-equation" (Prandtl mixing-length) turbulence models in the interest of improving the computational efficiency of the CFD models. Coupling CFD and building energy/loads calculations remains an active area of research.

Zhai (2001) coupled CFD to the EnergyPlus computer program (Crawely 2001). Zhai examined implications for stability and convergence of four different methods of coupling in terms of which variables are passed between the air and surface domains. Variables considered include: surface temperature, effective surface convection coefficient, distribution of effective air temperatures, surface convection heat flux, and effective surface convection coefficient for average room air temperature. Zhai (2002) concludes that stable solutions do in fact exist and the advantages of coupling are preserved for the following coupling method. The CFD model generates values for the surface heat transfer coefficient and effective air temperature distribution and passes these to surface models. The surface models pass values for the surface temperature to the CFD model. Therefore, this method is selected as a guiding principle for the Air Model Toolkit. However generating values for the surface heat transfer coefficient remains problematic and this subject is discussed further in Section 3.4.

3.2 Formulation of Heat Balance Model with Room Air Models

This section discusses the main assumptions for an extended version of the Heat Balance Model that allows including additional detail derived from room air models. It is desirable to formulate a combined model that is simple and applicable to both nodal and zonal models. We first review the original Heat Balance Model and then present the assumptions suitable for coupling air and surface models

3.2.1 Original Heat Balance Model

The term *Heat Balance Model* refers to a procedure put forth by the ASHRAE research community for calculating load and energy performance characteristics of buildings. The name draws from the method's application of the first law of thermodynamics and (presumably) its inclusion of all the important energy transport modes. The method uses

control volume/integral formulations described in numerous publications (Pederson 1997, Rees et al 2000). Attributes of this model as presented in the *ASHRAE Fundamentals 2001* are summarized here for completeness.

The main assumptions of the Heat Balance Model are:

1. Mixed zone air at uniform temperature
2. Uniform surface temperatures
3. Uniform long-wave and short-wave irradiation at surfaces
4. Diffuse radiating surfaces
5. One-dimensional heat conduction within surfaces.

The following processes are considered distinct in terms of modularizing the problem:

1. Outside Face Heat Balance
2. Wall Conduction process
3. Inside Face Heat Balance
4. Air Heat Balance.

Our research focuses on adding substantial detail to the fourth process while minimizing changes in modeling the first three processes. The well-mixed assumption for the air in the thermal zone has been described as "the most fundamental" assumption in the Heat Balance Model (ASHRAE 2001). The implication of this assumption is that the air is modeled using one control volume and a single value for air temperature is assigned to the zone. The Air Heat Balance is assessed for a control volume filling the room up to, but not including, the surfaces modeled by the Inside Face Heat Balance. The control volume's face lies just to the side of the air at the inside face (so that only the surface convection heat transfer term crosses the boundary).

3.2.2 Reformulated Heat Balance Model for Air Models

In reformulating the model to account for more detailed room air models the well-mixed assumption is altered but not eliminated. The air heat balance of the original model is retained (and should be enforced) and considered an overall air system heat balance. This heat balance amounts to an aggregate assessment of the air system's change in enthalpy and (in our nomenclature) referred to as the "$\dot{Q}_{sys}$ -equation." Thus the features of the original model are retained as we add resolution to the modeling of zone air temperature. The consequence of this change will affect the Heat Balance Model wherever its model equations include variables for the zone air temperature.

In addition to the single control volume for room air, additional subdivisions of this control volume are allowed for the purpose of modeling heat transport within the room and the resulting distribution of air temperatures within the room air of the thermal zone. Part of applying control volume theory is to account for all instances of flux across control volume boundaries. Instances of mass, energy, and information that might cross air control volume faces include: (a) surface convection heat transfer, (b) air inlets and outlets (air system, infiltration, and exhaust), (c) radiation (solar, short-wave, and long-

wave), and (d) information (control data, load schedules). This list guides what phenomena need to be addressed when coupling.

The term "inside face" refers to the inside face of an enclosure surface (such as walls, floor, ceiling) that faces the room air. The inside face is an important point of coupling and is discussed in Section 3.4. Often the goal of a load calculation is to determine the air inlet conditions required to meet the loads and maintain thermal comfort. As in the Heat Balance Model the inlet psychometric conditions are fixed, but more detailed room air modeling has important consequences for the outlet conditions as discussed in Section 3.5. As in the Heat Balance Model the air and zone contents are assumed to be transparent to radiation. The internal heat sources are modeled using convection/radiation splits.

We are now ready to restate the main assumptions for coupled surface and air models. Note that the assumptions for room air control volumes parallel those for surfaces. The air in the room is assumed to be a collection of separate, essentially well-mixed, control volumes where each are modeled as having:

1. Uniform state conditions such as temperature and pressure
2. Constant properties such as density and specific heat
3. Transparency to radiation
4. Uniform distributions of heat and mass transfer at each control volume boundary

In aggregate, the room air control volumes must agree with an overall air system heat balance. The surfaces of the room are modeled as having:

1. Uniform surface temperatures
2. Uniform irradiation
3. Diffusely emitted radiation
4. One-dimensional heat conduction within

The surfaces of the zone are treated in the same manner as before except that instead of all of them interacting with a single air control volume they each interact with a specific control volume of air. This near-surface air is referred to as the *adjacent air control volume*.

There are five distinct processes:

1. Outside face heat balance
2. Wall conduction heat transfer
3. Inside face heat balance
4. Air system heat balance
5. Air convective heat transport

The additional fifth heat transfer process distinguished here accounts for convective energy transport arising from the movement of air between control volumes. Note that

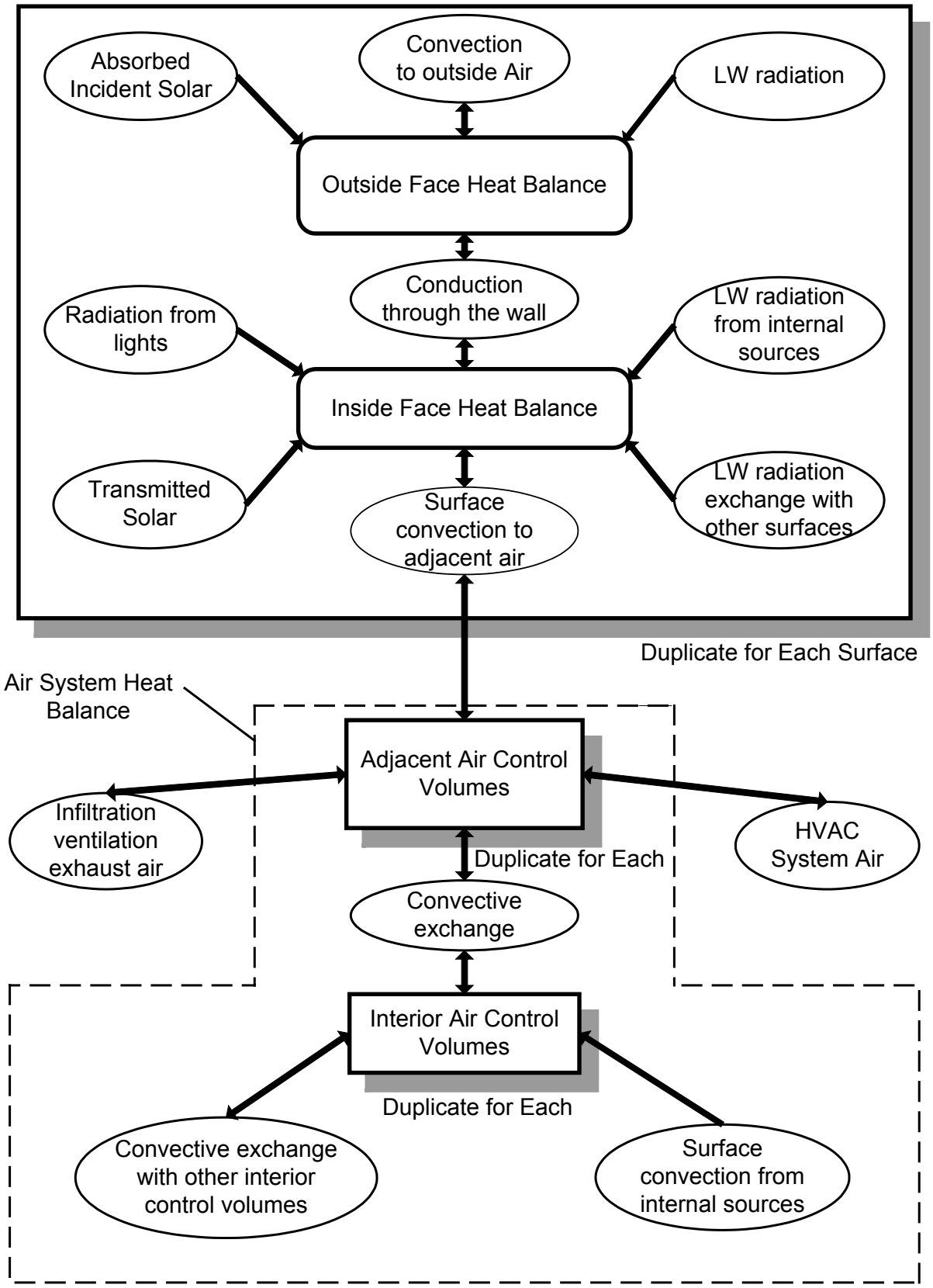


Figure 3.1 Schematic of Heat Balance Processes in a Zone with Air Models

surface *convection* heat transfer is a different phenomenon than *convective* heat transport in the room air. Figure 3.1 is an adaptation of the "Heat Balance Diagram" showing the added detail of the present reformulation.

3.3 Implementing the Coupled Surface and Air Models

The heat balance processes for a thermal zone can be formulated in manner consistent with the formulation described in Section 2.2 of the Loads Toolkit. The equations for the primary variables of outside face temperature, inside face temperature, and the system load are repeated for the sake of completeness. These models calculate transient heat transfer through the surface construction using a time series method called Conduction Transfer Function, or CTF, for computational efficiency at hourly time steps (Pederson 2001). The outside face heat balance is solved for its surface temperature,

$$T_{so_{i,j}} = \frac{-\sum_{k=1}^{nz} T_{s_{i,j-k}} Y_{i,k} + \sum_{k=1}^{nz} T_{so_{i,j-k}} X_{i,k} - \sum_{k=1}^{nq} \Phi_{i,k} q''_{ko_{i,j-k}} + q''_{\alpha sol_{i,j}} + q''_{LWR_{i,j}} - T_{si_{i,j}} Y_{i,o} + T_{o_j} h_{co_{i,j}}}{-X_{i,o} + h_{co_{i,j}}} \quad (3.1)$$

where,

i	indicates individual surfaces
j	indicates current time step
k	indicates time history steps
Y_i	are the cross CTF coefficients
X_i	are the outside CTF coefficients
Φ_i	are the flux CTF coefficients
T_s	is the inside face temperature
T_{so}	is the outside face temperature
q''_{ko}	is the conduction heat flux on outside face
$q''_{\alpha sol}$	is the absorbed direct and diffuse solar (short wavelength) radiation
q''_{LWR}	is the net long wavelength radiation flux exchange with air/surroundings
q''_{conv}	is the surface convection flux with outside air
h_{co}	is the outside face surface convection coefficient

We now introduce a modification to the Loads Toolkit's equations where the zone air temperature, T_a, is rewritten with an additional subscript, i, for the surface index ($T_{a_j} \rightarrow T_{a_{i,j}}$ or $T_a \rightarrow T_{a_i}$). The inside face heat balance is solved for its surface temperature,

$$T_{s_{i,j}} = \frac{T_{so_{i,j}} Y_{i,o} + \sum_{k=1}^{nz} T_{so_{i,j-k}} Y_{i,k} - \sum_{k=1}^{nz} T_{s_{i,j-k}} Z_{i,k} + \sum_{k=1}^{nq} \Phi_{i,k} q''_{ki_{i,j-k}} + T_{a_{i,j}} h_{c_{i,j}} + q''_{LWS} + q''_{LWX} + q''_{SW} + q''_{sol}}{Z_{i,o} + h_{c_{i,j}}} \quad (3.2)$$

where,

Z_i	are the inside CTF coefficients

q''_{ki} is the conduction heat flux at inside face

h_{c_i} is the surface convection heat transfer coefficient

The zone air temperature is represented by an array of values that are of "primary" interest in a coupling surface and air models. The equations to generate values for T_{a_i} are part of the air model itself and therefore depend on which air model is going to be used in the coupled models.

The overall air system heat balance leads to the $\dot{Q}_{sys}$ -equation,

$$\dot{Q}_{sys} = \sum_i A_i h_{c_i} \left(T_{s_{i,j}} - T_{a_{i,j}} \right) + \dot{Q}_{conv,S} + \dot{Q}_{Infil} \tag{3.3}$$

The iterative, or successive substitution heat balance solution method is selected to find solutions to these equations. Figure 3.2 diagrams how to implement such a coupled model in the context of a design-day calculation. Here the outer-most loop is for finding a steady periodic solution by repeating the same design-day environmental conditions. The next loop moves through the time steps per day where the toolkits use 24 time steps per day for hour-by-hour simulations. The iteration loop runs at a single time step and is used to allow sequentially computing each of the primary variables. After revisiting each calculation many times the effect is to find a solution that satisfies all the relationships. A fourth loop is shown in Figure 3.2 that is optional and referred to as the secondary air system loop as discussed in Section 3.5.

Figure 3.3 diagrams the general steps involved in the "Call Air Model" step shown in Figure 3.2. This encompasses the steps involved in passing data to and from the air model as well as evaluating the entire model itself. Some models may also receive other types of data not indicated. For example, a call to the Mundt model will also receive the current value for $\dot{Q}_{sys}$.

In this project the focus is on "tightly coupled" models where the air model is computed with the same frequency as the primary variables in the surface domain. This could be considered computationally intense and so numerous other schemes can be considered. One method computes the air model only once per heat balance time step. There are also a variety of library/lookup schemes that can be used with DT-coupling where the results from an air model are stored and the distribution applied for subsequent iterations or time steps without necessarily re-computing them very often.

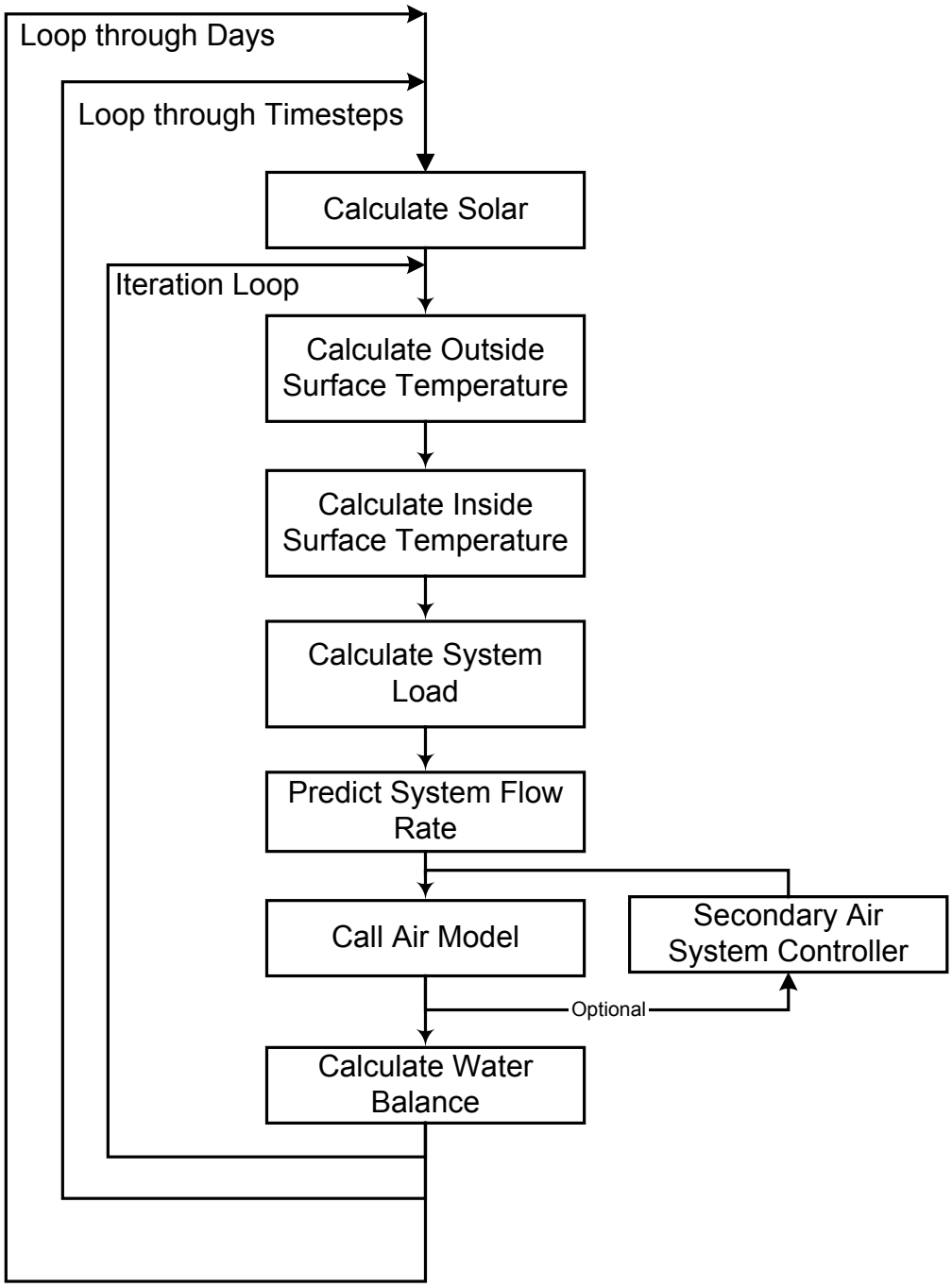


Figure 3.2 Iterative calculation strategy for coupled air and heat balance models

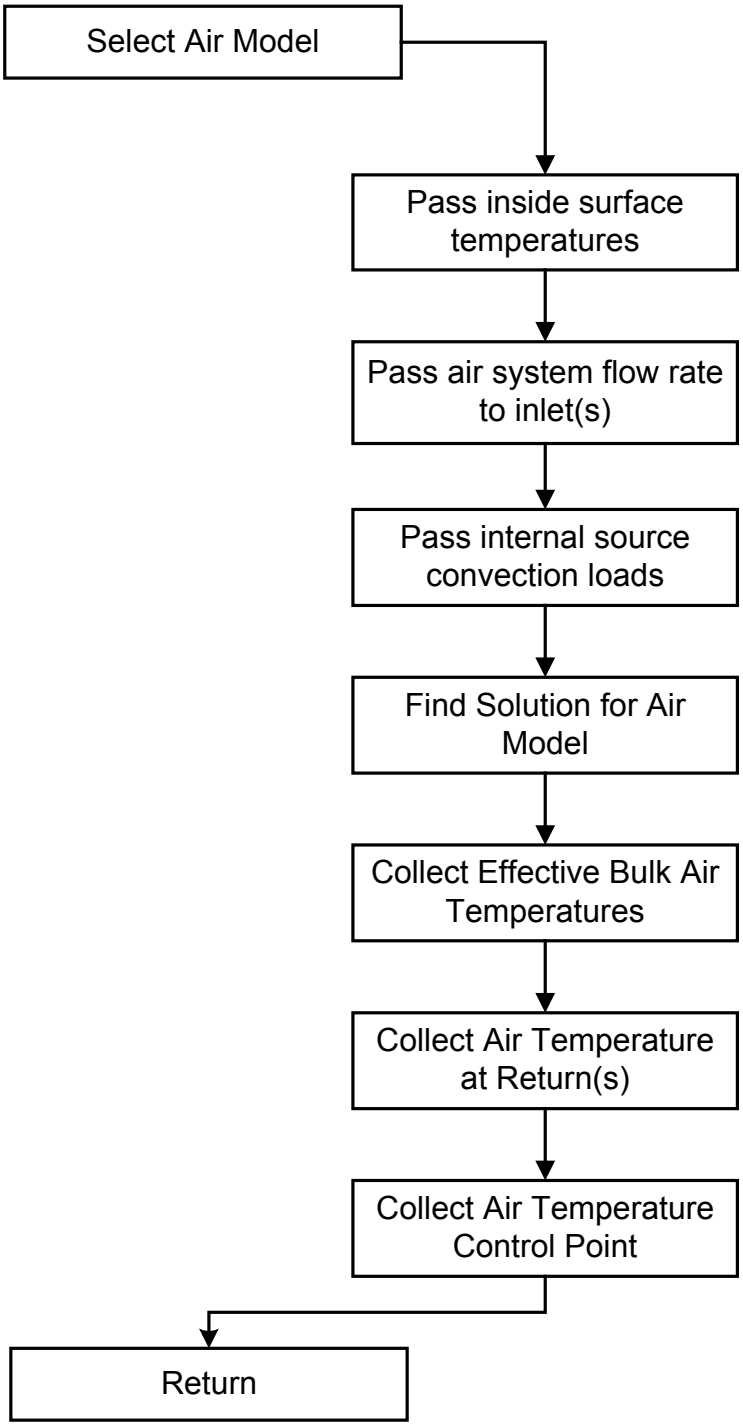


Figure 3.3 General steps in calling an air model

3.4 Inside Face Convection Heat Transfer

One of the primary reasons to consider more refined air models is to do a better job of predicting surface temperatures and rates of convection surface heat transfer. Beausoleil-Morrison (2000) noted that in certain situations inaccuracies in modeling convection at the inside face can cause errors of 20% or more in results for load or energy calculations. The term inside face refers to the inside face of a surface being modeled. An infinitesimally thick control volume is used at this surface to perform a heat balance that allows calculating the inside face's temperature. The basic situation is diagrammed in Figure 3.4. There are many other terms for the inside face heat balance, but only the one affecting the air model is shown in Figure 3.4.

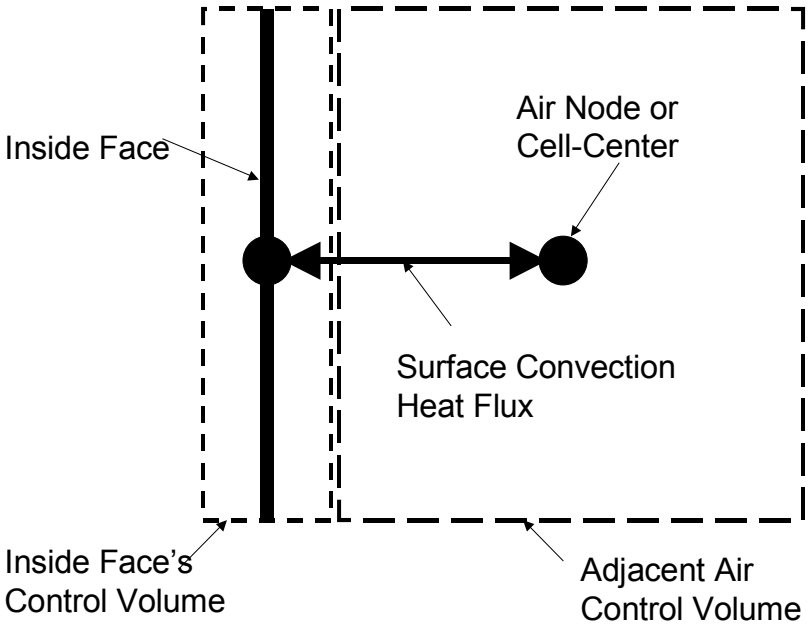


Figure 3.4 Inside face and adjacent air control volumes

Figure 3.4 diagrams the modeling construct termed *adjacent air control volume* that distinguishes a particular region of air from the rest of the zone air in that it exchanges heat by convection mechanisms with a particular surface. The adjacent air control volume is used to produce a temperature value for the effective bulk air temperature experienced by that surface.

Surface convection appears in heat balances in both the air and surface modeling domains and a coupled model should require that they be determined in a consistent manner. This introduces Patankar's (1980) "First Rule" which requires that when a face is common to two control volumes the same expression be used for calculating flux at that face for both control volumes. To follow this rule, the model equations used for surface convection terms in the heat balances should be the same for shared boundary of the inside face

control volume (surface domain) and for the adjacent air control volume (air domain). As long as this rule is met the convection calculation can be made in both domains and need not be confined to one or the other.

The usual model for convection surface heat transfer is,

$$\dot{Q}_c = A\, h_c\, (T_s - T_a) \qquad (3.4)$$

The area, A, and surface temperature, T_s, are straightforward; however the film coefficient, h_c, and the effective air temperature, T_a, are perhaps deceptively simple and are discussed in more detail. The sign convention here is that positive surface convection heat transfer, $\dot{Q}_c$, indicates net heat flow from the surface to the air and therefore adds to the cooling load. Equation 3.4 is a *mean* relation for an individual surface so values for T_a, T_s, and h_c are averages appropriate for the surface. In the event that the resolution of the air model is finer than the surface model, the data should be aggregated so that they conform to the surface's area. This averaging is necessary since the underlying surface's heat flow is modeled as one-dimensional, but does incorporate significant modeling assumptions since locally these values are expected to vary. One implication is that buoyancy-driven airflow may be augmented by localized thermal anomalies such as metal structural components. In heating conditions, the colder perimeter of insulated glazing units will expose air to a much larger temperature difference, and hence airflow, than that which would be obtained by the window average surface temperature.

3.4.1 T-Couple vs. DT-Couple

Equation (3.4) models surface phenomena and as such it can be written for each surface and there are two approaches,

$$\dot{Q}_{c_i} = A_i\, h_{c_i}\, (T_{s_i} - T_{a_i}) \qquad (3.5)$$

$$\begin{aligned} \dot{Q}_{c_i} &= A_i\, h_{c_i}\, (T_{s_i} - T_{RoomAir}) - A_i\, h_{c,i} (T_{a_i} - T_{RoomAir}) \\ &= A_i\, h_{c_i}\, (T_{s_i} - T_{RoomAir} - \Delta T_{a_i}) \end{aligned} \qquad (3.6)$$

where,

i denotes individual surfaces

$T_{RoomAir}$ is a reference air temperature (average, point of control)

$\Delta T_{a_i} = (T_{a_i} - T_{RoomAir})$

While mathematically equivalent, the choice between Equation 3.5 and Equation 3.6 leads to two different methods for coupling air and surface models. The first method is referred to as *T-couple* and is based on Equation 3.5 and the second method is *DT-couple* based on Equation 3.6.

In T-coupling the result from the air model is used directly and there is no assumption that the thermal zone is under controlled conditions. The air temperature in Equation 3.5 is a variable and no constant reference temperature is brought in the equation. This makes it an important method to consider for modeling non-mechanical designs or conditions when the loads are not being completely met. The method requires that the air model produce reasonably accurate values for air temperatures, overall air system heat balance, and the thermostat.

In DT-coupling the results from the air model are not used directly but are shifted by substituting $T_{setpoint}$ for $T_{roomAir}$ in Equation 3.6. In DT-coupling, if the air model's result for $T_{RoomAir}$ does not agree with the desired condition then this deviation is applied to the distribution. The intent with this method is that after iteration the two domains are "dragged" together so that the actual shift is small (< 0.01ºC). Therefore, if the thermal zone is being controlled to a steady temperature, the final result is the same as it would be with T-coupling. Care should be taken in implementing this method so that Patankar's first rule is not violated at either the inside face or the overall air system temperature difference. DT-coupling appears that it should be more stable. DT-coupling is an important method in the event that the air model has relatively poor accuracy with respect to absolute temperature but still produces useful information about the distribution of air temperatures (above or below setpoint).

3.4.2 Spatial Location for Determining Adjacent Air Temperature

Building rooms are enclosures and not semi-infinite fluid regions. Considering that T_a is a variable and not a constant, a framework for coupling air models to surface models requires clear specification of how values for T_{a_i} are to be selected. This air temperature is also known as the reference temperature for convective heat transfer calculations, but the term "reference temperature" is avoided here because it implies fixing the value. For the complete mixing model, one obvious selection is that T_a should to match the one available value which could be considered an air average temperature model of the physical situation. Fisher and Pederson (1997) concluded that the supply air temperature was better because it reduced uncertainty in correlations, but they did not consider the potential for room air models to generate near-surface temperatures . In nodal models, each surface is associated with a particular node and the result for temperature at that node can be used for T_a . Here the effective bulk air temperature for the surface convection can be considered the average for a control volume associated with that node. Although all surfaces need have a node associated with them, some nodes can interact only with other nodes (and do not affect surfaces). For the more structured and resolved geometry of a zonal model, the basic question is what distance scale to use when determining the effective bulk air temperature. From a classical heat transfer engineering point of view, the far field bulk air is assumed to be isothermal however this is not the situation in a closed room. Figure 3.5 diagrams the issue for a ceiling in a room with thermally stratified air. The point "A" is where the distribution matches the zone average, point "B" is the temperature at some arbitrary distance from the ceiling, and point "C" is the temperature at inside face of a ceiling. At point "B" the air temperature is higher than the surface and so the air is loosing heat to the surface. Were the zone average at point "A" used to assess convection it would appear that the ceiling is heating

the zone air, a completely opposite result (unless one resorts to negative values for $h_{c)}$. This discussion points to the need to select an arbitrary yet appropriate distance scale to use in determining T_a.

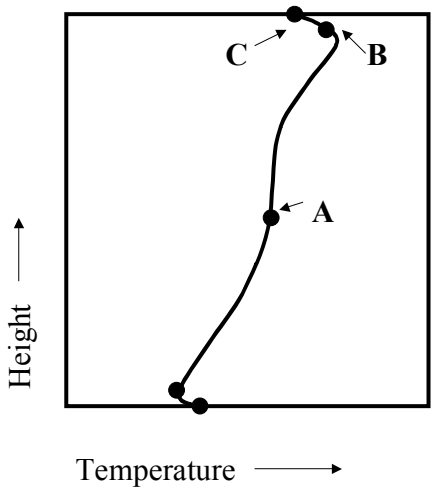


Figure 3.5 Temperature distribution schematic in stratefied room

A distance of 0.1 m (4 inch) into the air away from a building surface's inside face is selected as an appropriate geometrical scale for determining T_a. This value is chosen in view of the following points:

- The point must be outside the thermal boundary layer
- The point should not be too far outside thermal boundary layer
- Hot box thermal test facilities for measuring component U-factors measure bulk air temperatures a distance greater than 0.075 m (3 in.) (ASTM 976)
- 0.1 m. is often used for floor air temperature at ankle height.
- Zhai et al. (2002) uses 0.1 m for coupling CFD with EnergyPlus

The implication of a "0.1 m rule" for T_a is that for zonal models the adjacent air control volume should be 0.2 m thick so that the control volume center is at 0.1 m. If finer-grid air models are used, the value for T_a passed to the surface domain is not necessarily from the first grid but should be determined from the grid(s) near 0.1 m. The value should also be for averaged for a plane of air at 0.1 m that conforms to the surface's area. We suggest that adopting a 0.1 m rule as a standard method of selecting near-surface temperatures will help establish consistent application of equations 3.5 or 3.6. It will be helpful if experimental data reported air temperatures at 0.1 m and used such referencing in developing correlations for surface convection heat transfer coefficient h_c.

3.4.3 Surface Convection Coefficient

Having discussed the framework in which the film coefficient, h_c, will be used, what remains is to select a method of determining appropriate values. Beausoleil-Morrison developed a comprehensive, yet pragmatic, methodology for selecting appropriate correlations to use for calculating h_c within a building simulation program. While the

general approach remains valid there is reason to suspect that correlations for h_c that have been developed for building simulation may not apply well to our framework. The value for h_c needs to be consistent with the spatial location for T_{a_i} or ΔT_{a_i}. This is because these correlations were developed for the mixing model's choice of T_a. Values from nodal and zonal models for T_{a_i} or ΔT_{a_i} will often be closer to the temperature of the wall requiring higher values of h_c to obtain the same heat flow rate. This issue arises often in fine-grid CFD modeling where the first grid point can be inside the thermal boundary layer. A convection correlation for building simulation may have been developed for use with the mixing assumption and so may have built-in dependence on both air movement and temperature near the wall. Developing a new suite of correlations for h_c is beyond the scope of this research project. Therefore, our interim approach is to rely on the recommended values for h_c developed for use with specific models, ASHRAE defaults, and user prescribed values. For nodal models, the adjacent air control volume is larger and so convection correlations are probably suitable. For example, the general rules for the Rees and Haves nodal model uses a correlation for convection for the lowest part of the walls. Ultimately a set of convection correlations should be developed/tested for use in nodal and zonal modeling. Such correlations can be a function of mass flow rate (mean velocity) in the adjacent air control volume as well as the usual temperature difference, orientation, and length scales.

Values for h_c can be computed directly using CFD with turbulence models that are able to resolve flow well near a surface. This can be problematic in the context of buildings leading Beausoleil-Morrison (2000) among others to concluded that CFD does a poor job at the air domain boundaries and that the surface convection heat transfer coefficient, h_c, for use in the surface domain still need to be determined using empirical correlations. On the other hand, Chen et al. (1999) and Zhai et al. (2001) directly determine local values for h_c from the model's computation of effective turbulent viscosity, μ_{eff}, using,

$$h_{c,i} = \frac{c_p \mu_{eff}}{\mathrm{Pr}\,\Delta x} \tag{3.7}$$

where,

Pr is the Prandtl number

Dx is the distance between the control volume center and the wall

Equation 3.7 can be derived from basic principles of turbulent fluid mechanics but the accuracy of its result depends on the turbulence model used to compute μ_{eff} and there is a dependence on grid size. Equation 3.7 is not useful for the much simpler room air models because they are not expected to generate accurate values for μ_{eff}. Using very-fine grids and modeling inside the near-wall laminar boundary layer allows computing good results. Although it is still an open question whether or not such methods are useful for routine building simulations, they are good candidates for use by researchers in developing h_c correlations for use in the current framework.

3.5 Air System Flow Rate

An important convenience afforded by the historical use of the complete mixing assumption is the ease with which air system flow rates can be determined for a certain combination of load, control set point, and supply air temperature. The mixing assumption provides that three important temperature values are all the same, the average of entire zone air, $T_{RoomAvg}$, the air leaving the zone, $T_{leaving}$, and the air at the location of the thermostat or point of control, T_{statDB}. If all three of these are equal and equal to the desired temperature, $T_{setpoint}$, then there is an implicit assumption that the thermal zone is also comfortable. For a steady-state VAV system that meets the load, $\dot{Q}_{sys}$, the air system flow rate, $\dot{V}$, is calculated using the mixing model by,

$$\dot{V} = \frac{Q_{sys}}{\rho c_p (T_{Supply} - T_{RoomAvg})}. \tag{3.8}$$

Equation 3.8 is a simple rearrangement of the system energy balance (negative Q_{sys} indicates positive cooling load) where ,

$$\rho c_p \dot{V} (T_{Supply} - T_{Leaving}) = \dot{Q}_{sys} \tag{3.9}$$

With additional detail from room air models we can rewrite Equation 3.8 by exchanging $T_{leaving}$ for $T_{roomAvg}$.

$$\dot{V} = \frac{\dot{Q}_{sys}}{\rho c_p (T_{Supply} - T_{Leaving})} \tag{3.10}$$

Note that for the mixing model, the value for $T_{setpoint}$ is used for $T_{RoomAvg}$ in Equation 3.8 but there is no analogous situation for Equation 3.10 since with air models we expect $T_{leaving}$ to have values that differ from $T_{setpoint}$. For this reason, Equation 3.10 for use in a poorly mixed thermal zone does not have the same predictive capability as Equation 3.8 for a well mixed zone. If we allow for spatial temperature gradients within the zone then a unique and comfortable solution for $\dot{V}$ is not ensured. The system flow could be too much or too little for T_{statDB} to equal $T_{Setpoint}$. In DT-coupling, the previous value for $T_{leaving}$ is used and model results are dragged together after iteration by applying deviations between T_{statDB} and $T_{Setpoint}$. In T-coupling, we consider an additional controller to find $\dot{V}$ as discussed in the following section.

3.5.1 Secondary System Air Iteration Loop

The so-called secondary system air iteration loop models air system flow rate as a function of T_{statDB} or some quantity representing comfort conditions. A secondary air system loop maintains room temperature control as a "real thermostat" would by adjusting $\dot{V}$ up or down depending on deviations between $T_{Setpoint}$ and T_{statDB}. A simple air model could be inverted to find a value for $\dot{V}$ that is comfortable, but in general

iteration is helpful to find an air system flow rate that meets $\dot{Q}_{sys}$ and comfort criteria. The toolkit implements such a loop that is conceptually similar to how a VAV controller/thermostat might operate. This secondary loop runs inside the main iteration loop and uses constant values for parameters from the surface domain while making additional calls to the air model. Before entering the loop, an initial prediction of $\dot{V}$ is made using Equation 3.10 and the current value for $T_{leaving}$. The basic idea for a cooling situation is to increase system flow if T_{statDB} is too high and decrease flow if T_{statDB} is too low. The toolkit implements a simple proportional controller. Equation 3.11 shows a method of determining the controller gain by differentiating Equation 3.10 with respect to the system air temperature difference. This allows scaling the controller response based on the overall scale of the zone's cooling load. A user selected thermostat tolerance (e.g. 0.05 K) governs when the control condition has been met.

$$\text{(Proportional Gain)} = -\frac{2Q_{sys}}{\rho c_p (T_{Supply} - T_{Leaving})^2} \tag{3.11}$$

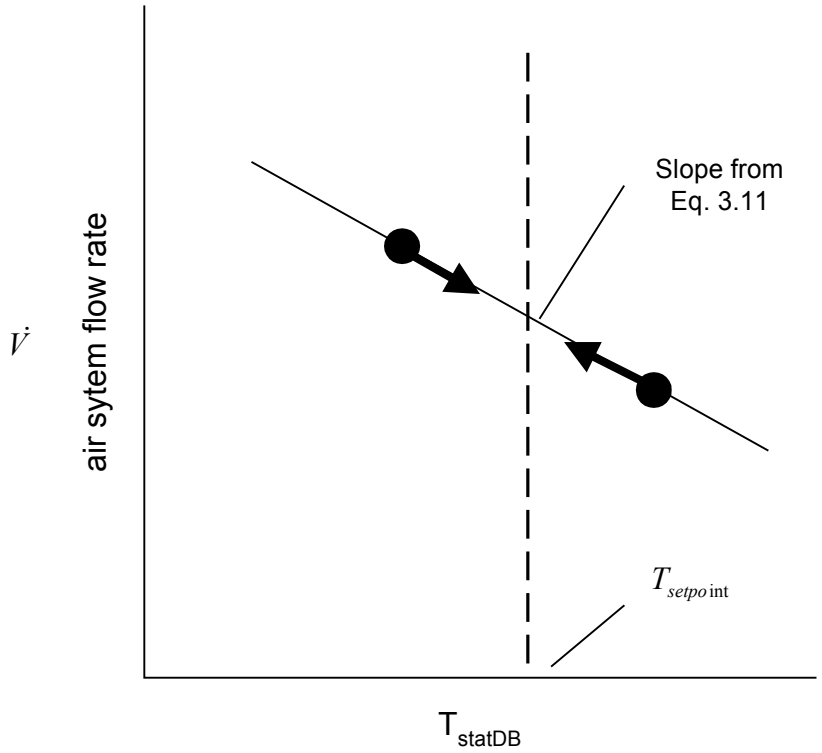


Figure 3.6 Simple control diagram for secondary air system loop.

Air system control can also be a function of additional comfort parameters such as radiant temperatures, air velocity, or humidity. Since a physical thermostat will likely be on a wall and humans sense more than just dry bulb air temperature, it is also of interest to control the air system based on some operative temperature, T_{op}. Equation 3.12 shows one method of modeling a value for the operative temperature, T_{op}, where the mean radiant temperature, T_{MRT}, is included, though clearly others could be used.

$$T_{op} = \frac{1}{2}\left(T_{MRT} + T_{statDB}\right) \tag{3.12}$$

If there is reason to distrust the capability of the air model to accurately predict T_{statDB}, or if the air model is not sensitive to variations in the system flow rate, then the secondary system iteration loop discussed above is likely to fail. This extra loop also requires extra calls be made to the air model and this may be computationally expensive.

3.5.2 Modified Air System Temperature Difference

A simpler method than the secondary iteration loop may be useful when using the DT-coupling method. In this case assuming that the zone is at the desired controlled condition and the air model is only used to obtain a distribution of how air temperatures deviate from the thermostat's reading. Zhai (2002) suggests that convergence can be accelerated by determining $\dot{V}$ using Equation 3.10 with an adjusted value for the air leaving the zone, $T'_{leaving}$ that has been modified by the deviation between $T_{Setpoint}$ and T_{statDB}.

$$T'_{Leaving} = T_{Leaving} - (T_{statDB} - T_{Setpoint}) \tag{3.13}$$

3.6 Discussion

This section provides comments concerning geometry and additional discussion related to the quasi-steady and radiation assumptions.

3.6.1 Geometry Considerations

There are at least two treatments for expressing the geometry of a thermal zone when representing it in a computer program. The first is a somewhat free geometry where the program knows the areas and orientations of all the surfaces but not necessarily exactly how they are arranged with respect to each other. The second is to more thoroughly describe the locations of surfaces throughout space. The former requires input for area and direction and is used in the Loads Toolkit. The later might use input for the vertices for each surface using a three-dimensional Cartesian coordinate system, as is done in EnergyPlus. Focusing on implementing models for the air domain in combination with surface models of the freer type, leads to the creation of additional geometry specification where the air domain is leading the surface domain. The nodal air models retain the free geometry characteristic but lead to sub-dividing surfaces in the vertical direction. The zonal air models specify a grid of air control volumes using a three-dimensional Cartesian coordinate system. The surface area segments need to be selected so that they conform to the geometry in the air models. More elaborate, full-featured programs would likely automate a surface reprocessing capability so that surfaces can be described in the usual way, but for now the user may need to adopt a new perspective when specifying surfaces so that they are consistent with what is most useful for the air models. Section B.2 in the Appendix provides more detail on developing grids for program input.

3.6.2 Quasi-Steady Air Model Assumption

In the Heat Balance Model the capacitance of the air is assumed to be negligible in the formulation of the Air Heat Balance. Maintaining this assumption allows room air models to use quasi-steady balances and provides the useful simplification that mean flow rates for mass and heat transfer are modeled as instantaneously steady. The overall heat capacity of the room air is considered low and neglecting its effect allowable for load and energy calculations. This assumption may be poor for other types of simulations, such as modeling HVAC controls, or where time steps of less than about 0.1 hour are being used. Note that air heat capacity remains an important parameter in a room air models since airflow convects thermal energy around the room.

The unsteady term being neglected can be investigated by including it (temporarily) in the air energy balance,

$$\rho V c \frac{dT}{dt} = Q_{surfConvTot} - Q_{ConvGains} - Q_{sensInfil} - Q_{sys} = \sum Q \tag{3.14}$$

The quasi-steady assumption sets the term on the left hand side to zero. Historically, building simulation programs for load calculation use room air temperature as a fixed input and are organized around predicting the parameters required to maintain that temperature. If the room temperature is being maintained at a constant value then the dT/dt term is zero and so can be neglected.

Equation 3.14 can be integrated and solved to show the order of magnitude of the time constant for room air is

$$\tau \propto \frac{\rho V c_p}{\left| \sum Q \right|} \tag{3.15}$$

Thus a room with a volume of 100 m^3 that experiences an abrupt change in convection heat load of 1000 Watts will change about one Kelvin in 120 seconds. This is argued to be a short time period compared to a one-hour time step used in load and energy simulation. Note that the considerably higher heat capacity of solids leads to time constants for the surfaces that are much longer that this. This provides an important physical basis for separating air and surface modeling domains. For example, in thermal network models (Sowell 1991, Walton 1993) the air is modeled with nodes that are "mass-less." The framework used in the Air Model Toolkit can be justified based on the differences in the temporal physical behavior between the air and surfaces (in addition to the desire to implement traditional Loads Toolkit algorithms for surfaces).

If room air temperatures are floating, then the dT/dt term amounts to a source or sink of energy that is being neglected. In order to put the error in perspective we can compare the magnitude of the source/sink to the heat terms that are modeled. If air temperatures of 100 m^3 room are changing at an overall rate of 1 K per hour then the unsteady term in Equation 3.14 contributes about 30 W to the instantaneous energy balance. Such a room

might have a nominal cooling load (the sum of the other terms in Equation 3.14) of 40 W/m^2 or 1400 W so it can be argued that the quasi-steady assumption should do little to alter load or energy calculation results. The driver program that calls the quasi-steady air model could keep track of changes in temperature over time and pass the information to the air model for use as a volumetric source/sink term. Air models for use with shorter time steps or for simulating HVAC controls should consider adding this term.

It is also worth noting that considerable efficiency is gained in the modeling of room air by assuming steady-state flow. Modeling transient conditions within the air domain requires short time steps that would substantially increase the already serious computational requirements of the more advanced room air models.

3.6.3 Radiation

The Air Model Toolkit makes no direct contribution in the area of adding detail to radiation modeling since the air models neglect radiation. Glicksman and Chen (1998) studied the assumption of transparent air with regard to infrared absorption of water vapor and concluded that it can be significant for larger spaces. This absorption amounts to a volumetric source term that depends on the humidity ratio and surrounding wall/gas temperature differences. The rate of heat gain being neglected is of order of 1 W/m^3 for typical wall/gas temperature differences.

Since the Air Model Toolkit focuses on increasing the resolution of models within the air domain, no change is made to the way radiation is modeled in the surface domain. Though not a requirement of the Heat Balance Model, the implementation demonstrated in both the *Loads Toolkit* and the Air Model Toolkit use the Mean Radiant Temperature method (Walton 1983). Sowell (1991) describes a different model partitioning scheme where the more complex zone model includes nodes at the inside surfaces and allows adding detail to the determination of radiation exchanges.

The descretization of room air and can lead to finer resolution of the surfaces and offers a framework that might be helpful for implementing more rigorous calculation of radiation heat transfer. The geometry of sub-surfaces may be obtained from a zonal air model grid. The contents of the room, such as people, lights, computers, furniture, and the classic "thermal mass" walls, quite naturally lie in the room and so the details of their spatial distribution are modeled in the air domain. The radiation/convection split techniques used in the Heat Balance Model determine the proportions of the internal load that add heat to the air domain and to the surface domain. Model simplicity is attained by extending the assumption of no radiation absorption in the air to the contents of the zone. A more detailed model for radiation exchanges between inside faces could also interact with zone contents. Such expanded modeling would probably lead to modeling surface temperatures of zone contents. This would have implications for air models since they would need to allow temperature boundary conditions at internal cells rather than the currently implemented heat flux boundary conditions.

4. AIR MODEL TOOLKIT

This report is part of a "toolkit" that includes source code for computer programs that implement air models discussed in Sections 2 and 3. This section presents a brief summary of the toolkit and focuses on efforts to verify, validate and otherwise investigate the behavior. Here we discuss the model implementations in a general fashion without referring to the specifics of code, routines, and variables. Appendix A provides detail on the fortran90 source code. Appendix B discusses how to run and create input for programs.

This section presents results from model predictions along with published results and measured data in an effort to verify and validate the routines distributed in the toolkit. Verification refers to the reasonable reproduction of the original or expected results of a particular model. Program verification was performed at many stages during code construction to check that the computer data were physically meaningful and in agreement with other's results (and our expectations). Validation refers to comparing model predictions to experimental measurements or different (preferably more-detailed) models. The process involves developing test cases as input files to the model, running the model, and then comparing the outputted results. Results are "compared" graphically by plotting combined data from published results and measurements along with the new results. Determining "agreement" remains somewhat subjective, but for the purposes of this report "good" agreement is where deviations are within about 0.5 K and "bad" agreement is beyond 1.0 K. We are also able to compare results from different models although not all air models are applicable to all test cases. There are typically two versions of the air models, the stand-alone or component version and the version coupled to the entire loads calculation. Applying different boundary conditions allows using some test cases for both versions. The stand-alone air models receive boundary conditions for the supply air temperature, T_{supply}, the air system flow rate, $\dot{V}$, the inside face surface temperature, T_{si}, and the convective portion of the internal load source(s), $Q_{conv,s}$ Both the radiative and convective portions of the total internal load are boundary conditions for validating coupled surface and air modeling, along with T_{supply}, $T_{Setpoint}$ and "outside" conditions such as the surface temperature of the outside face, T_{os}.

This section is organized in the following manner. The first section presents a brief summary of the Air Model Toolkit in order to introduce the demonstration programs being evaluated. The second section presents examples of verification exercises for each of the air models taken from the literature. The third section presents results and discussion of validation exercises for the momentum-zonal model. The fourth section presents steady-state validations of the coupled air and surface models.

4.1 Summary of the Air Model Toolkit

"Air Model Toolkit" refers to the collection of documentation, source code, programs, input files, and related items produced by ASHRAE Research Project-1222 (RP-1222). The intent is that this toolkit become the fourth in a series of toolkits published by

ASHRAE that are related to building energy and load calculations. The first toolkit focused on secondary HVAC systems (Brandemuehl 1993). The second toolkit focused on primary HVAC systems (Lebrun 1999). The third toolkit focused on load calculations and demonstrates the Heat Balance Model (Pedersen 2001). This third toolkit is referred to here as the "Loads Toolkit" and it has a much stronger link to the Air Model Toolkit than the first two toolkits. All the toolkits contain computer program subroutines and documentation for models and algorithms to implement them. Source code is in the FORTRAN family of programming languages. Fortran90 is used in the Air Model Toolkit at the request of ASHRAE and in order to be consistent the three previous toolkits. The Air Model Toolkit also uses the same data input utilities as the Loads Toolkit and the program EnergyPlus (Crawely 2001). Historically toolkits have focused on subroutines and any larger programs that perform modeling have been put together in order to test subroutines rather than to develop a complete program for building simulation. This introduces the notion of a "demonstration program" where code is meant to demonstrate using the subroutines in the process of showing methods of computing the model equations. While efforts are made to create code that is error-free, a demonstration program is not intended to be a full-featured program with all the robustness expected of commercial software. For example, the level of error checking of user input does not meet modern programming practice. Rather the focus is on simple, clearly arranged code that is readable and informative as it attempts to document how to conduct such calculations. The developer should also consult the Software Developer Guide in Appendix A and the User Guide in Appendix B for more information on how the models are implemented.

The fortran90 programs and modules have been organized into two different groups, sample programs and components. The sample programs demonstrate coupling air models to a load calculation. The individual air models are components of the toolkit and are also referred to as "stand-alone" air models. These components have been developed as stand-alone programs in order to test the subroutines and modules that have been written for coupling to the loads models combined program. Four models from the literature have been selected for implementing in the toolkit. The first two models are by Mundt (1996) and Rees and Haves (2001) and apply to designs with displacement ventilation. The third model is by Inard, Bouia, and Delicieux (1996) which applies to many designs but we focus on natural convection. The forth model is by Lin (1999).
The next section present results from testing component air models.

4.2 Verification of Air Model Components

Table 4.1 lists the cases selected to include in this report to illustrate our efforts to verify air model implementations. Verification refers to the reasonable reproduction of the original or expected results of a particular model. Models approximate reality and so in order to verify a model we compare new results to published predictions from the model as well as measured results. Numerous other cases have been run during the process of developing code, but in the interest of brevity we select four representative cases that correspond to laboratory thermal test chamber experiments from the literature. The main

criterion when selecting measured cases was the availability of high-resolution data for the distribution of room air temperatures in the vertical direction.

Table 4.1 Test Cases selected for Verification of Air Models

Test Case Identifier	Description	Reference	Measurement Location
DisplaceVent_B3	Displacement Ventilation	Li et. al. 1993	National Swedish Institute for Build Research, Gävle, Sweden
DisplaceVent_12	Displacement Ventilation	Rees 1998	Loughborough University, UK
ColdWindow	Sealed Natural Convection	Allard et. al. 1987 Inard et. al. 1996	CETHIL's MiniBat, France,
ConvectHeat	Convective Heater	Chen et al. 1999	MIT , Cambridge, MA, USA

CFD results were also generated for this research project in order to supplement the measured data and show results using more detailed models. Many developers of nodal and zonal models used CFD to generate reference data for validation purposes (e.g. Lin 1999, Rees 1998) . CFD data offers more spatial resolution than the measured data available for the test cases. Some of the test cases were modeled using a commercial CFD program (CHAM 1999) to compute mean room air conditions using a standard k-ε turbulence model.

4.2.1 Mundt Model

The Mundt model works with the assumption of a linear temperature gradient and formulates one additional heat balance at the floor. From the floor air temperature and the temperature at the return the slope is obtained and a distribution of air temperatures can be easily modeled. Section 2.3 discusses the model.

Verification of the Mundt model involved comparing model predictions to those published by Mundt (1996). The test case referred to as "DisplaceVent_B3" corresponds to measurements conducted by Li et al. (1993) at the National Institute for Building Research in Gävle, Sweden. (The case is called "II" by Mundt.) Case DisplaceVent_B3 is a steady-state laboratory chamber measurement characterized in Tables 4.2 and 4.3. The overall inside size of the chamber is 4.2 m by 3.6 m by 2.75 m in height. Air temperatures were measured at different heights. The total load was 300 W and we modeled this with a split of 75% convection and 25% radiation since the source contained

some low-emittance surfaces. (Note that Rees (1998) modeled as 100% convection.) The heat source is a single porous container with light bulbs and aluminum and measures 0.4 x 0.3 x 0.3 m. For this test case it was positioned low with its lower face 0.11 m from the floor.

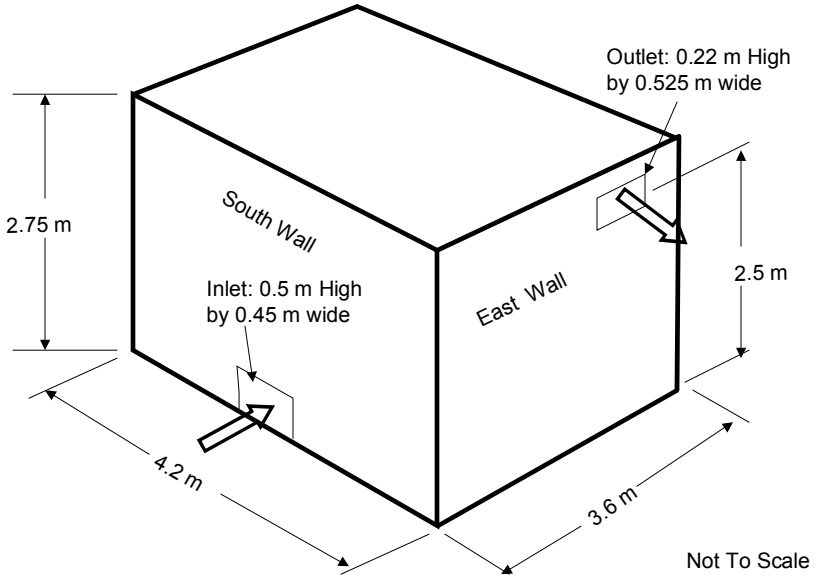


Figure 4.1 Thermal Test Chamber for Li et al. Case B3

The chamber is guarded by temperature-controlled regions "outside" in order to establish steady-state conditions. The experiment had a constant air system flow rate and supply air temperature. For verifying an air model's implementation (in a computer program) it is important to check it in isolation from the surface models. By fixing the inside face surface temperatures and internal load convection rates, the air model is presented with what can be thought of as a static snapshot of one time step during a load calculation. Table 4.2 lists boundary conditions for use in testing a stand-alone air model. The air model may also need values for surface heat transfer coefficients. Mundt used h_c values of 5.0 ($W/m^2 \cdot K$) for the floor and ceiling surfaces. Rees and Haves (2001) also studied this case and developed h_c values of 8.51 $W/m^2 \cdot K$ for the ceiling, 6.06 $W/m^2 \cdot K$ for the floor, 1.3 $W/m^2 \cdot K$ for the lower walls, and 6.3 $W/m^2 \cdot K$ for the upper walls. This case will be discussed again in Section 4.3 in the context of validating the coupled air models and loads routines using different boundary conditions.

Table 4.2 Stand-Alone Air Model Boundary Conditions (Case DisplaceVent_B3)

	Value	Units
$Q_{conv,s}$ internal load	225	[W]
$\dot{m}$	0.0414	[kg/s]
T_{supply}	18.0	[ºC]
T_{surfIn}, Ceiling	25.16	[ºC]
T_{surfIn}, Floor	22.13	[ºC]
T_{surfIn}, walls: 0 to 1/4 H	22.35	[ºC]
T_{surfIn}, walls: 1/4 to 1/2 H	23.05	[ºC]
T_{surfIn}, walls: 1/2 to 3/4 H	23.63	[ºC]
T_{surfIn}, walls: 3/4 to H	23.88	[ºC]

A verification exercise was conducted in order to test the subroutines implementing the Mundt Model. Table 4.3 summarizes the overall results and compares results from the toolkit subroutines versus Mundt's published predictions and the measurements. Figure 4.2 plots the results and shows how the linear air temperature model compares to the measured distribution of air temperatures. This is a sample of the result used to conclude that that the subroutines implementing Mundts' floor air heat balance produce the expected output.

Table 4.3 Mundt Model Verification Overall Results (Case DisplaceVent_B3)

	$\dot{Q}_{sys}$ [W]	$T_{airFloor}$ [ºC]	$T_{leaving}$ [ºC]
Measurment (Li et. al. 1993)	-285	20.9	24.8
Mundt (1996) extended model prediction	-256	20.6	24.4
Toolkit Stand-alone Mundt	-256	20.9	24.4

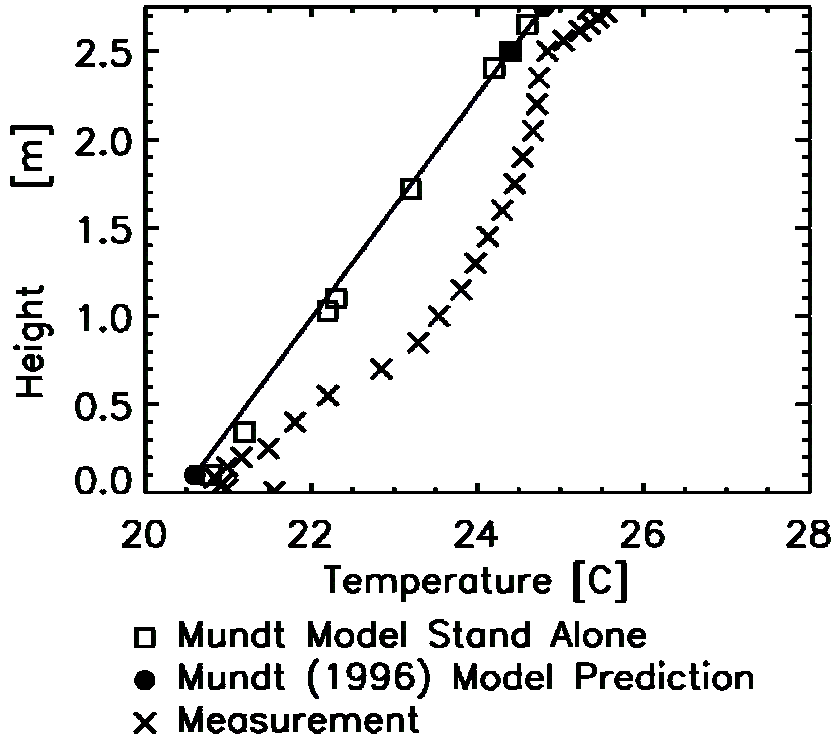


Figure 4.2 Verification Results for Stand Alone Mundt Model

4.2.2 Rees and Haves Nodal Model

The Rees and Haves nodal model uses a prescribed flow pattern in a network of nodes to model a vertical temperature distribution. Section 2.3 discusses the model formulation and A.5 discusses the implementation.

Verification exercises compared toolkit results using the Rees and Haves model to those published by Rees (1998). The test case referred to as "DisplaceVent_12" corresponds to laboratory chamber measurements conducted by Rees that are characterized in Figure 4.3 and Tables 4.4 and 4.5. These experiments are similar in nature to those of Li et al. (1993) and our case DisplaceVent_B3. The overall inside size of the chamber is 5.43 m by 3.09 m by 2.78 m in height. Detailed measurements were made of air temperatures at different heights. Heat sources were simulated using a stack of boxes with light bulbs inside sitting on 0.6 x 0.6 m tables at a height of 0.72 m. Four separate tables/heat sources were distributed evenly around the middle of the room (see Rees 1998). Rees gives internal load splits of 53% radiation and 47% convection that were developed from radiometric measurements. The total internal load was 300 W.

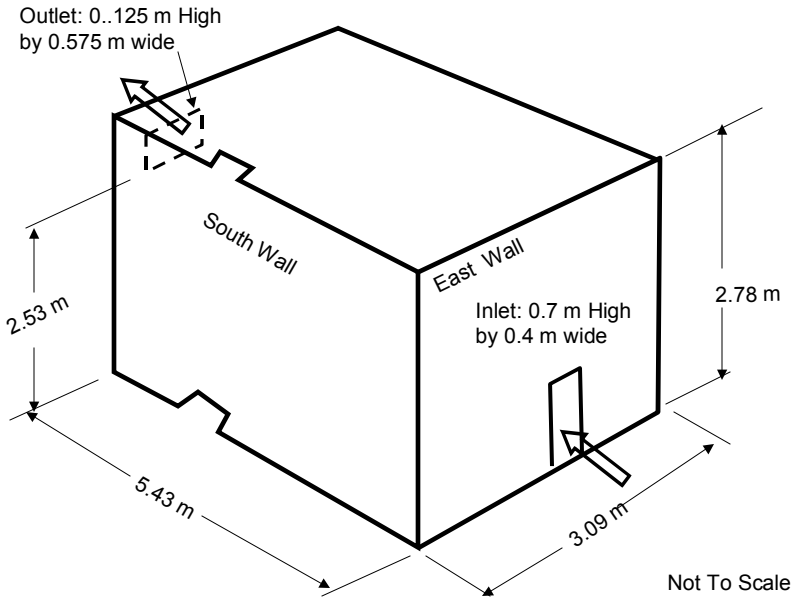


Figure 4.3 Test chamber for Rees's case DisplaceVent_12

Table 4.4 Case DisplaceVent_12 stand-alone air model boundary conditions

	DV12	Units
$Q_{conv,s}$	188	[W]
$\dot{m}$, air system mass flow	0.0451	[kg/s]
$T_{supply,}$ supply air temp.	18.39	[ºC]
T_{surfIn}, Ceiling	23.26	[ºC]
T_{surfIn}, Floor	21.97	[ºC]
T_{surfIn}, walls: 0 to 1/4 H	21.19	[ºC]
T_{surfIn}, walls: 1/4 to 1/2 H	21.84	[ºC]
T_{surfIn}, walls: 1/2 to 3/4 H	22.40	[ºC]
T_{surfIn}, walls: 3/4 to H	22.52	[ºC]

A verification exercise was conducted in order to test the subroutines and modules implementing the Rees and Haves component air model. Table 4.5 summarizes the overall results and compares results from the toolkit subroutines versus Rees's published predictions and measurements. Figure 4.4 plots the results and shows how the model compares to the measured distribution of air temperatures. This is a sample of results used to conclude that that the subroutines, modules, and non-linear solver package implementing the Rees and Haves model produce the expected output.

Table 4.5 Verification results for toolkit Rees and Haves Model (Case DisplaceVent_12)

DV12	$\dot{Q}_{sys}$ [W]	$T_{sysDiff}$ [ºC]	$T_{leaving}$ [ºC]	T_{statDB} [ºC]
Measured (Rees 1998)	-212	4.6	23.0	21.5
Rees and Haves nodal model, published (Rees 1998)	-223	4.8	23.2	21.5
Rees and Haves model Air Model Toolkit component	-221	4.76	23.15	21.7

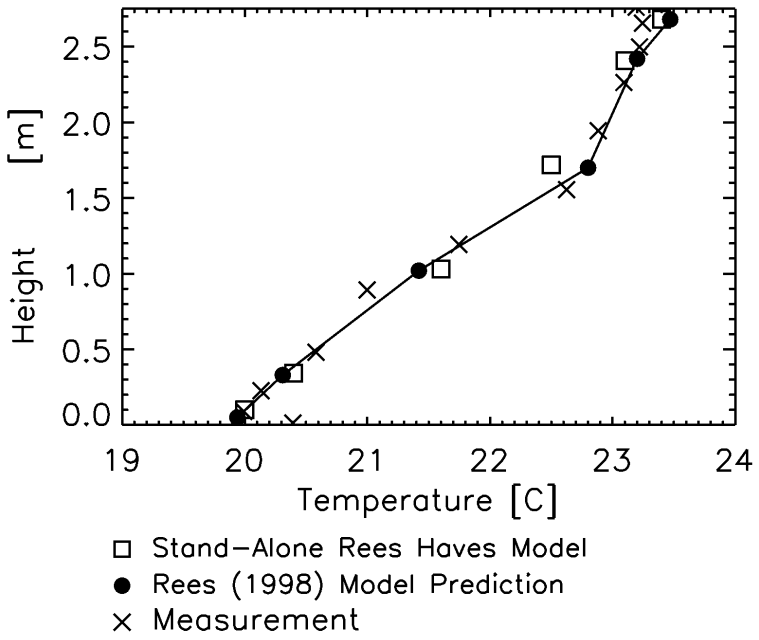


Figure 4.4 Air Temperature results for toolkit Rees and Haves nodal model, published predictions, and measurements (Case DisplaceVent_12)

4.2.3 Inard Pressure-Zonal model

The Inard pressure-zonal model uses a grid of control volumes to model movement and temperature distribution of air using a non-linear formulation. Section 2.4 discusses the model formulation and section A.6 discusses the implementation.

Thus far little success has been achieved in obtaining well-converged solutions with the toolkit's programs for the Inard pressure-zonal model. The Air Model Toolkit demonstration programs have used (or attempted to) a number of different solvers

including: (1) Newton-Raphson with line search and backscatter (see section 2.2.2), (2) secant method using Broyden's update technique (Press et. al. 1996), (3) IMSL Math Library routine DNEQNF which is based on the MINPACK subroutine HYBRD1, and (4) IMSL Math Library routine DNEQNJ which is based on the MINPACK subroutine HYBRIDJ. The latter use a modified Powell technique that is another variation on Newton-Raphson of the type known as double dog-leg. Continuing efforts include trying to find coding errors, different numerical solvers, and user-defined Jacobian matrix, etc. Unless additional progress is made we cannot conclude that our implementation of the Inard-model is valid. This effort showed that developing a pressure-zonal model probably requires a larger-scale research/programming project than was possible in the context of this research project. It might have been a better choice of models to pursue linearized formulations and sequential solution techniques such as presented by Wurtz et. al. (2001).

4.2.4 POMA

This section presents verification exercise results from the POMA pressure-zonal model. POMA's calculation engine was originally written in the C family of programming languages; POMA was rewritten in fortran90 for this toolkit. The actual conversion of the program was in two stages. POMA was first rewritten in fortran90 to use the original input method and test files. At this stage, fifteen different cases were tested. Comparing the two implementations (C and Fortran90) found that temperatures agreed almost exactly ($\pm 10^{-6}$ K). Figure 4.5 shows temperature distributions from both the C and fortran90 versions. These input files for POMA included user input for an initial guess of the temperature field and *successful* initial guesses had already been worked out for them. The second stage of revising POMA involved adding routines for coupling to the loads models and using the input framework of the Air Model Toolkit. (This was not as trivial as expected because the programs use different coordinate systems.) In the context of coupling to a load calculation, it seems impractical to require the user formulate an initial temperature guess for each call to POMA. Several attempts at implementing automatic temperature guessing of initial temperature fields were made but a general method was not found.

Zonal models can be applied to the problem of modeling natural convection in a sealed room. This has been called the "cold window problem" (Wurtz 1999) with the idea that under heating conditions, low performance windows get cold and may cause natural convection driven flow to develop in a room. Our case called "ColdWindow" refers to the second case in Inard et al. (1996). The measurements were performed in the MINIBAT laboratory thermal test chamber (Allard et. al. 1987). The chamber internal size is 3.1 x 3.1 x 2.5 m. The apparatus attempts to achieve isothermal inside face wall temperatures. The boundary conditions for this case consist of the following inside face surface temperatures: south wall at 16.9ºC, north wall at 33.0ºC, east wall at 26.9ºC, west wall at 27.3ºC, ceiling at 28.5ºC, and floor at 25.9ºC. The chamber is sealed; there is no air system. All of the airflow is from natural convection. The main flow of heat is from the hotter north wall to the cooler south wall.

Figure 4.5 plots the distribution of air temperatures within the test chamber from the measurement (Inard 1996) and POMA results. In all the figures for the ColdWindow case, the left side is the cooler south wall and the right side is the warmer north wall.

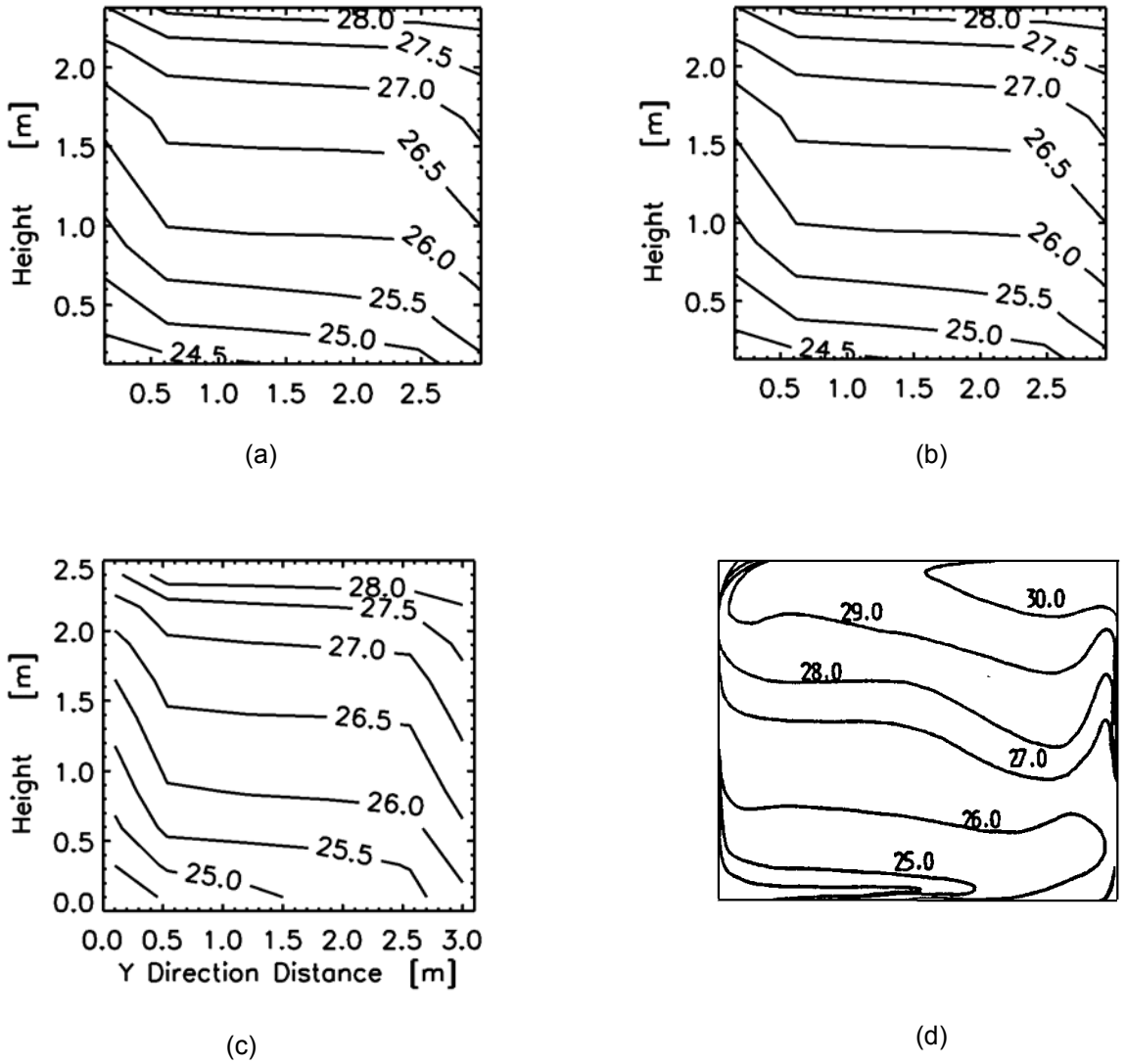


Figure 4.5 Air temperature results for case ColdWindow from POMA: (a) original POMA in C/C++, (b) POMA in Fortran90 with original input, (c) Toolkit POMA with auto-generated temperature guess, (d) measurement (Inard 1996).

The validation exercise has allowed us to conclude that POMA has been successfully converted from C to fortran90. The new input routines have also been verified to work with POMA. Therefore we can conclude that the toolkit implementation of POMA is valid. However, in developing new input files and methods for automatically generating initial guesses we have found that fully converged models are difficult to obtain. Often the solver gets "stuck" and cannot be accurately described as "globally convergent." This caused serious problems for coupling to the loads models and so this model was not helpful for testing the framework.

4.3 Validation of Momentum-Zonal Model

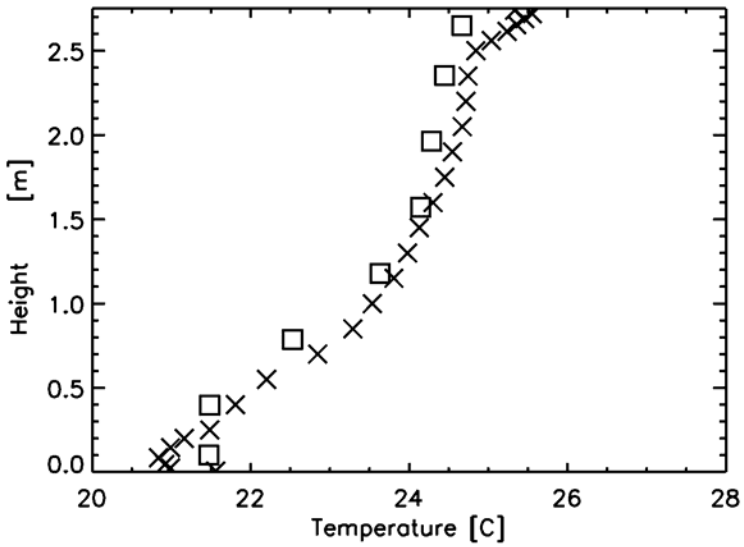
This section presents a summary of efforts to validate and characterize the behavior of the momentum-zonal model. The developers of the other models have validated their models, but since this model is a new contribution we seek to validate the application of such a model to certain types of room airflow. The grids used in this testing were intentionally somewhat crude in that the mesh was distributed uniformly rather than fine-tuned to obtain a custom grid that "fits" the problem. Better performance might be obtained with manual tuning of the grid, but we wanted to test the model in the context of very simple and coarse implementation. The momentum-zonal model is capable of fine-grid modeling as well as coarse-grid modeling envisaged for zonal models. Therefore models were run using both low resolution and moderate grid numbers. The less coarse grids allow representing room contents as blockages.

The following three sections present examples of validation exercises. The first section presents case related to displacement ventilation. The second section presents a natural convection case. The third section presents a convective heater case. The fourth section discusses the findings.

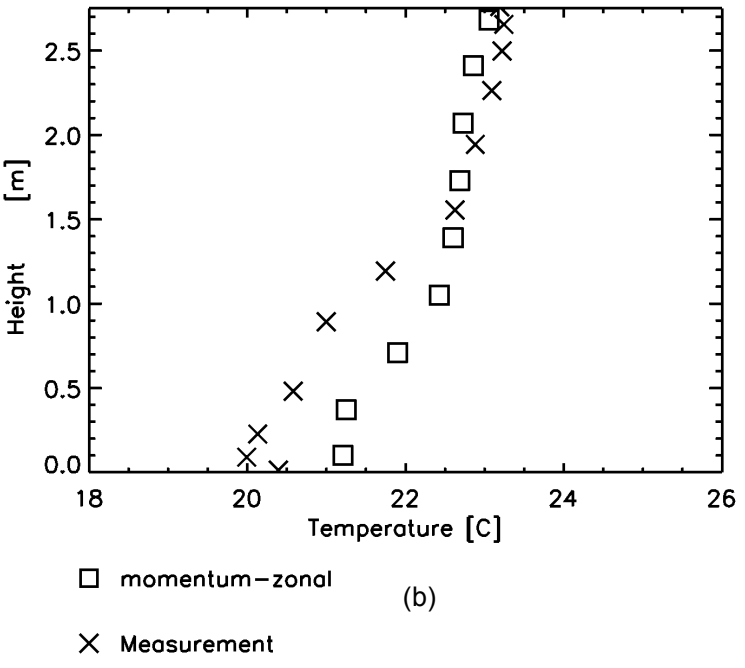
4.3.1 Displacement Ventilation

The two displacement ventilation cases described above were used in a validation exercise for the momentum-zonal model. Figure 4.6 shows the results where data for a single vertical "column" of cells have been extracted to compare to the measured distribution. Agreement is good for the DisplaceVent_B3 case, yet for the DisplaceVent_12 case agreement is not very good for the lower portion of the room however the errors are on the side of the mixing model.

Although more researchers studying displacement ventilation systems have focused on side-wall diffuser designs, currently more new buildings use under-floor air distribution, or UFAD, rather than side-wall. A numerical validation exercise was performed to test the capability of the momentum-zonal model to predict the air temperature of such systems. A case was developed to model a block of eight office cubicles with individual swirl-type flow diffusers for each. The thermal zone is an interior zone with only internal loads. Kobayashi (2001) studied such systems and tested methods of modeling the diffusers using CFD. These techniques were used here and represent the flow from one inlet using a square block of eight patches. Each patch has a velocity vector arranged so that flow is directed at 60º from horizontal and with the remaining horizontal component directed tangentially. The CFD simulation used a 72 x 52 x 28 grid, standard k-ε turbulence modeling, and required many days of computing. A complete description of the cubicle partitions, people, computers, and table tops was included in the CFD model but not the zonal model. The zonal model used an 10 x 8 x 10 grid and represented each diffuser with one cell (1.2 m. by 1.1 m.) and initial momentum of 2.0 m/s.



(a)



(b)

Figure 4.6 Air temperature results for stand-alone momentum zonal model (a) case DisplaceVent_B3 and (b) case DisplacVent_12 case

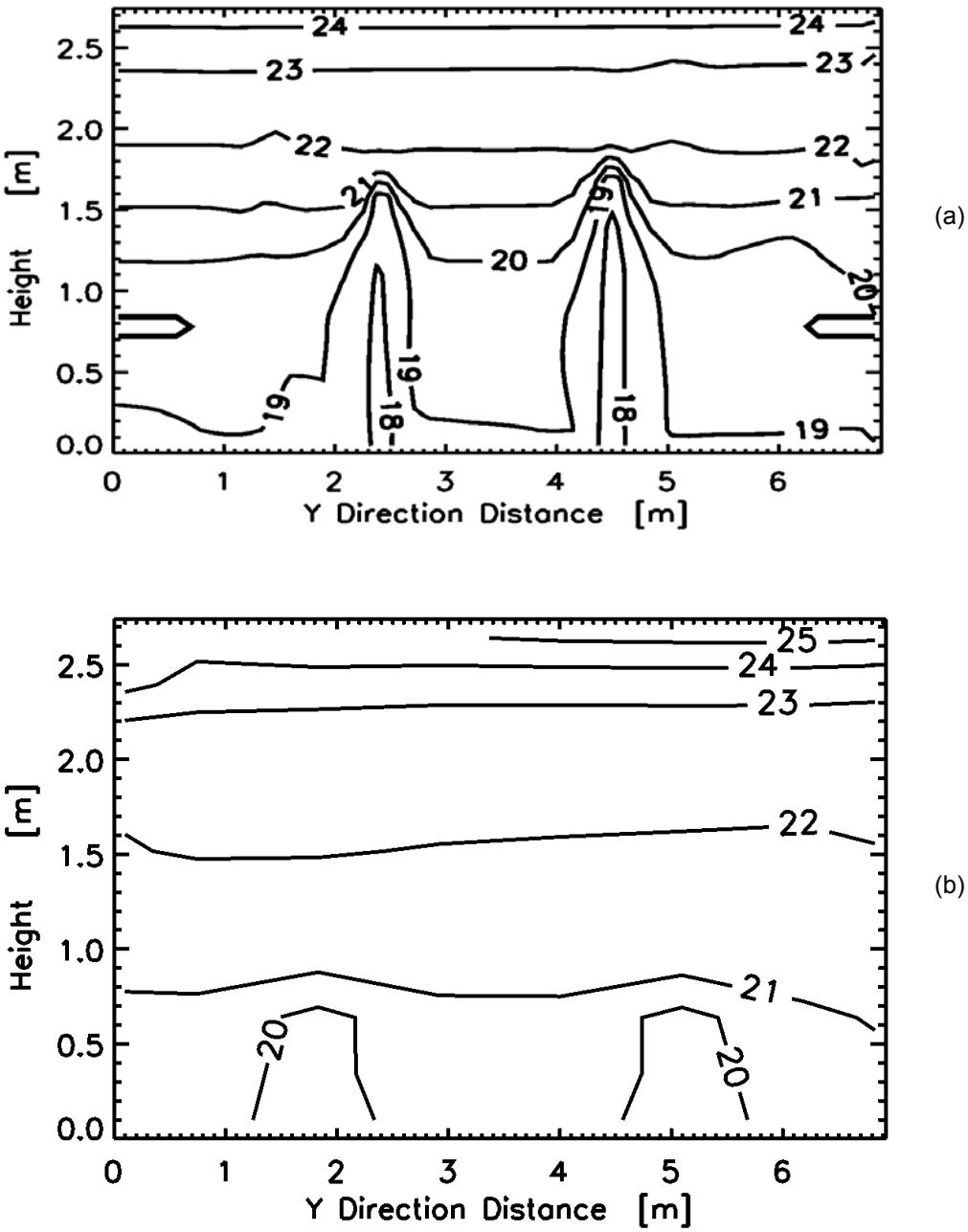


Figure 4.7 Air temperatures (ºC) distributions in cubicles with Under-Floor Air Distribution (a) k-ε CFD grid of 72x52x28 *, (b) momentum-zonal model grid of 10x8x10.*

Figure 4.7 shows the results for temperature distributions for the zonal model compared to the CFD model. The results show that such a model could capture much of the air temperature distribution characteristics of UFAD. In using the model on this case it was found that the buoyancy forces appear very strong in the model.

4.3.2 Natural Convection

The ColdWindow case (Inard's case 2) was selected as representative and presented here (see Section 4.2.3 for case description). Figure 4.8 shows the results from the momentum model for air temperatures using two different grids. The computed airflows are shown in Figure 4.9 for various grid resolutions along with the computation results by Inard (1996) and from POMA. The momentum-zonal model probably shows exaggerated buoyancy-driven flow and slightly errs on the side of mixing.

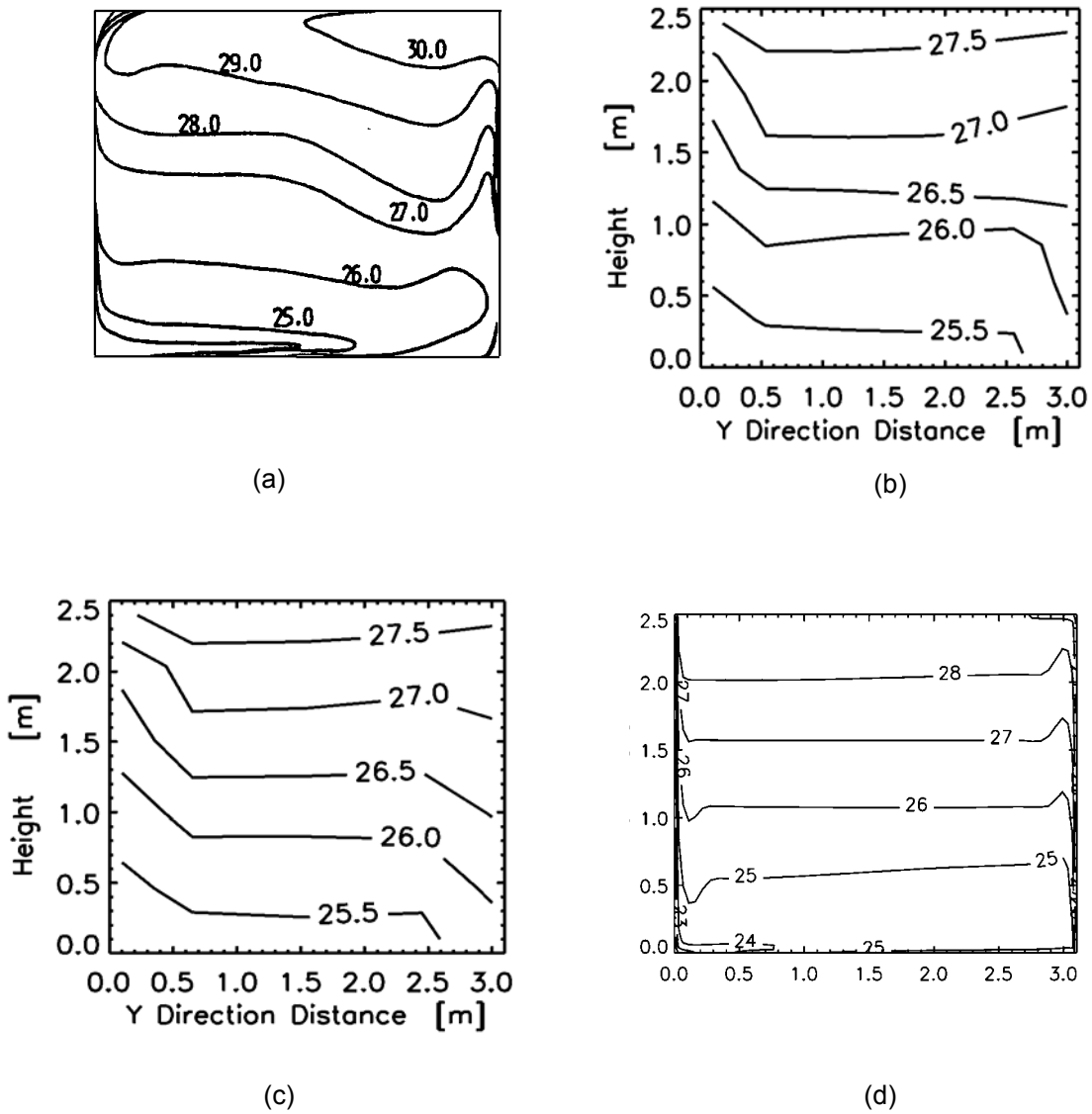


Figure 4.8 Air temperatures for ColdWindow Case (a)measurement (Inard 1996), (b) momentum-zonal model (2D-1x6x10), (c) momentum-zonal (3D-5x5x6), (d) k-ε CFD

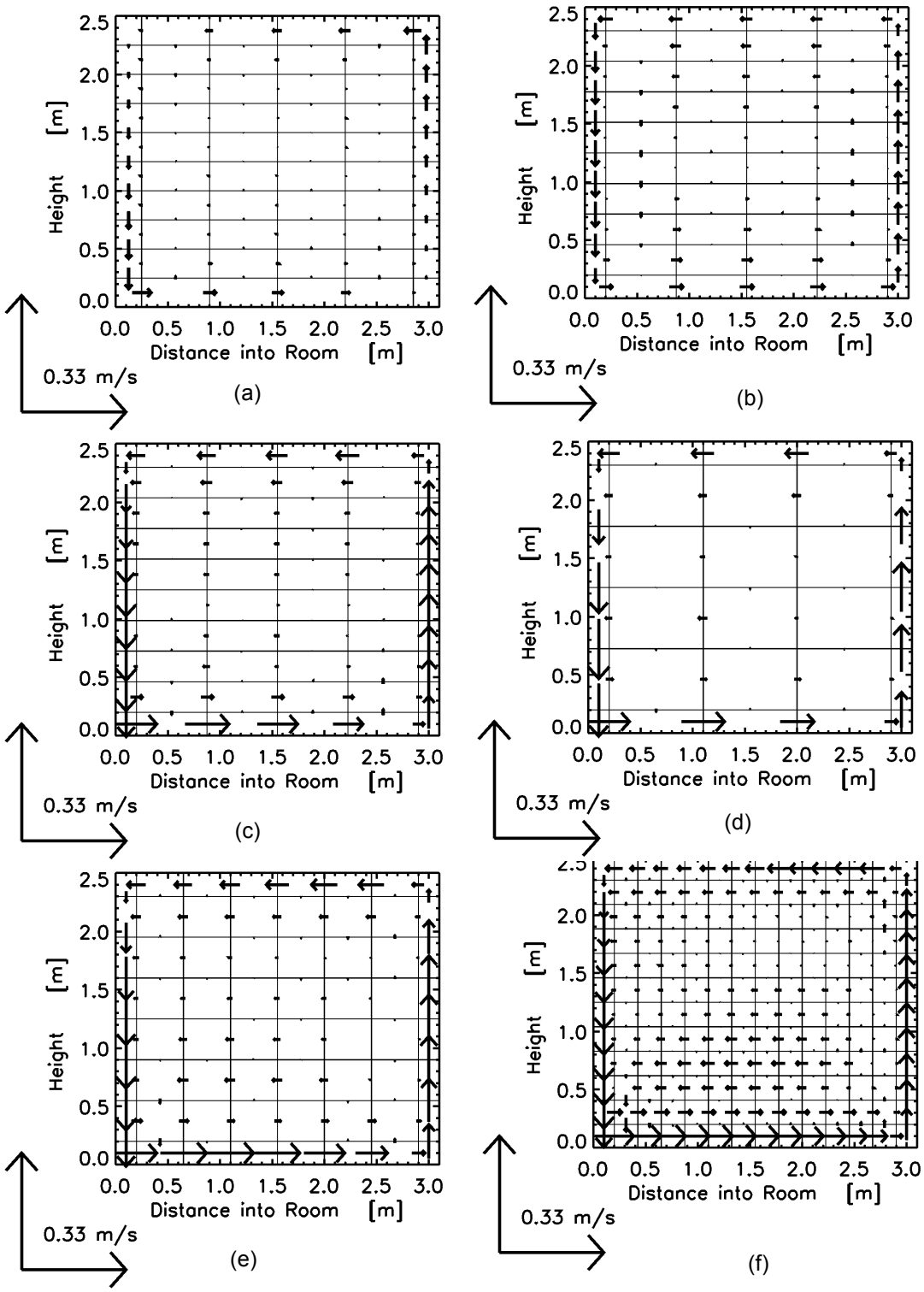


Figure 4.9 Airflow patterns for ColdWindow Case (a)Inard (1996) published model prediction (2D-1x6x10), , (b) toolkit POMA model prediction (2D-1x6x10), (c) momentum-zonal model (2D-1x6x10), (d) momentum-zonal (3D-5x5x6), (e) momentum-zonal (3D-8x8x8), (f) momentum-zonal (3D-14x14x12)

4.3.3 Convective Heating

The momentum-zonal model was tested using several cases corresponding to measurements conducted in the room airflow test chamber located at the Massachusetts Institute of Technology. The case referred to as "ConvectHeat" was selected as representative and is diagramed in Figure 4.12. In this test case, a baseboard heater drives most of the flow with a low-velocity, large-area inlet air simulating infiltration (3 ACH). This case was used for validation of CFD models for room airflow and was documented as the "infiltration case" for ASHRAE Research Project-927 (Chen et. al. 1999). The chamber is described by Yuan et. al. (1999b). More complete description of the cases and measured results are available in Appendix C of the report for RP-927 (to be included in toolkit). These data are useful for validating an airflow simulation but do not provide "outside" boundary conditions for validating coupled models.

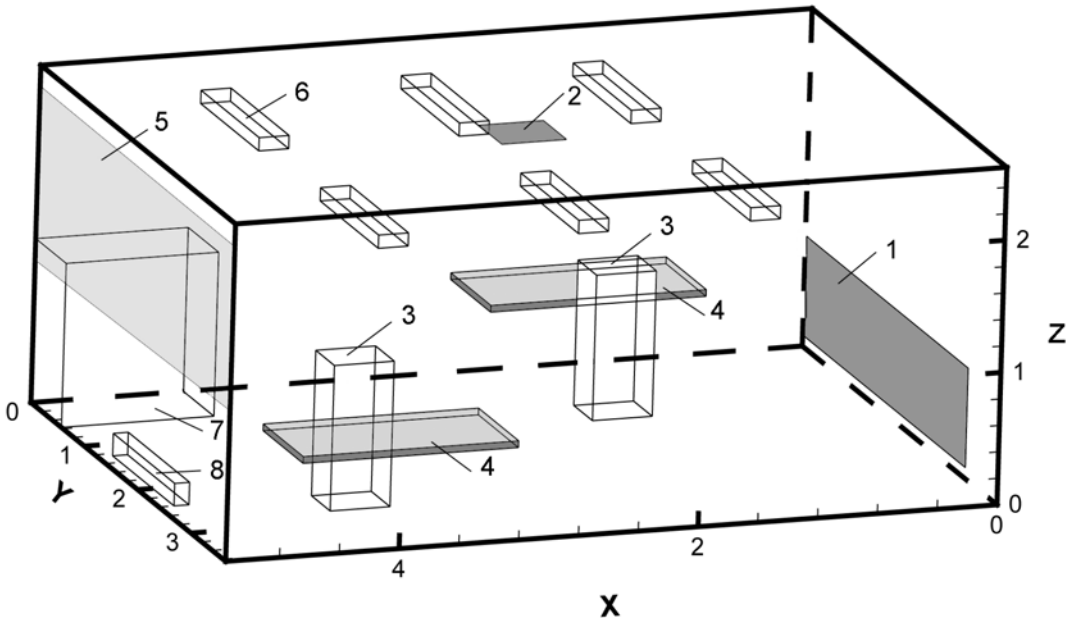


Figure 4.12 Test chamber layout for convective heater case(inlet -1, outlet -2, person -3, table with computer-4, window -5, fluorescent lamps -6, cabinet -7, baseboard heater -8).

Although experimental data are shown as contour plots, they were actually gathered from 5 poles along the slice (y=1.83 m) with 8 probes on each (in the vertical direction). Therefore the plots show measured data with considerable extrapolation between the poles. The measurements also gathered mean air velocities at six heights on the poles; the highest velocities were near the inlet (0.1 m/s) and near the ceiling in the half of the room closer to the heater (0.16 m/s). All of the velocities at probe locations are within the uncertainty of the probes. No measured data for the plume are available.

Table 4.7 presents a summary of the measured (Chen et al. 1999) and simulated results for the ConvectHeat case . The results are plotted in Figures 4.13 as temperature contour

plots. Three different runs of the momentum-zonal model are shown corresponding to different options in using the program. Coarse grid and fine grid models were run in order to explore their effect on the predictions of the model. The coarse grid run in Figure 4.13 used 12 x 7 x 6 or 504 cells. The finer grid runs in Figure 4.13 used 27 x 22 x 20 or 11880 cells. For the coarse grid, heat sources that are close together are combined and no blockages are used. The two finer grids model each of the internal contents separately and in one case the contents are blockages and in the other case they do not block flow. Results are mixed, but the overall results indicate that the coarse-grid solution is probably as useful as the fine-grid model.

Table 4.7 Momentum-zonal model overall results for the ConvectHeat case

ConvectHeat Baseboard Heater, MIT	$\dot{Q}_{sys}$ [W]	$T_{sysDiff}$ [ºC]	$T_{leaving}$ [ºC]	T_{statDB} [ºC]
Measured	491	10.8	24.0	24.2
Coarse-Grid (504 cells) no blockages	574	10.9	25.7	24.1
Fine-Grid (118800 cells) with blockages	503	9.7	24.1	22.9
Fine-Grid (118800 cells) no blockages	610	10.9	26.4	24.1

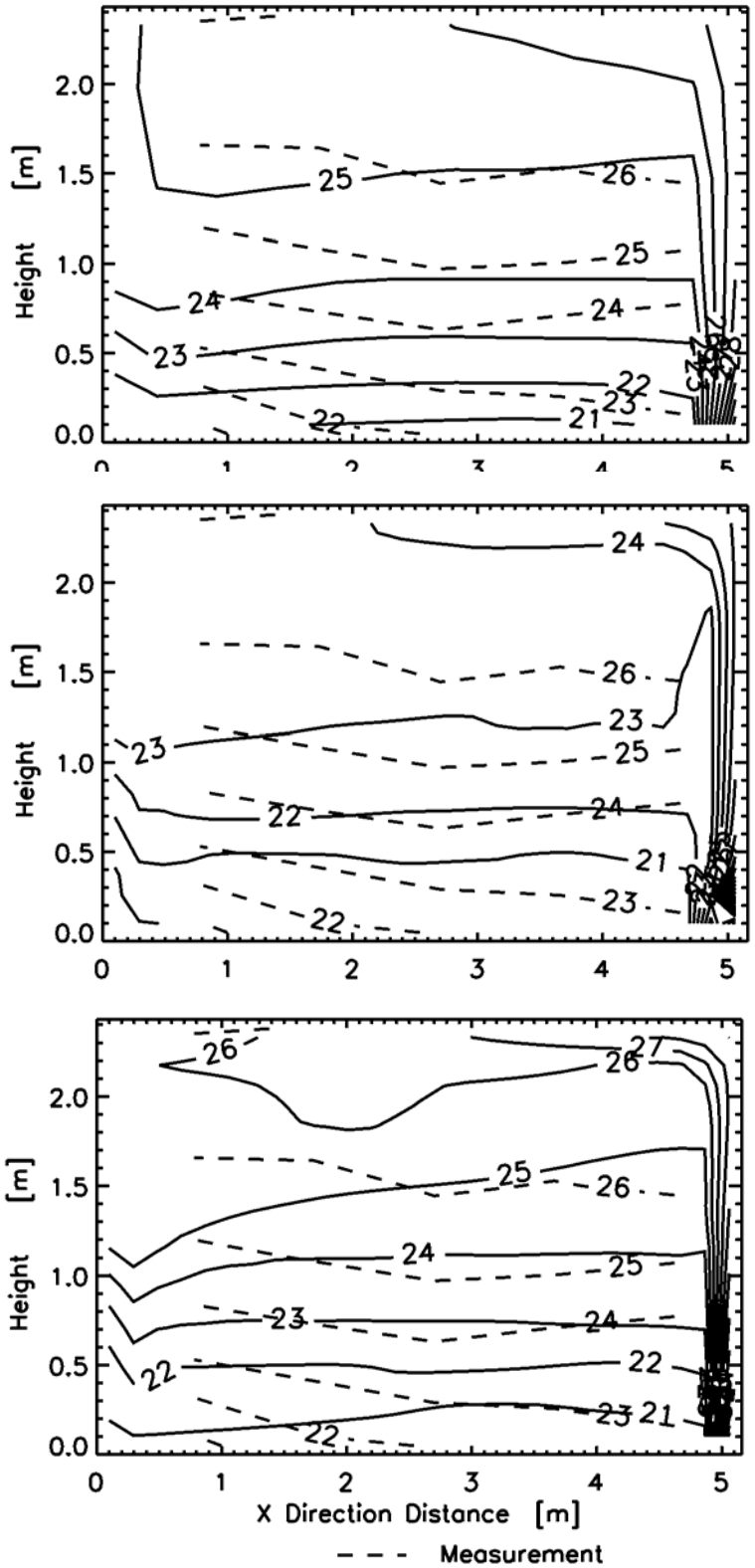


Figure 4.13 Momentum-zonal model results for ConvectHeat case (a) 504-cell grid no blockage, (b)11880-cell grid with blockages, (c) 11880-cell grid no blockages

The results show that the momentum-model is able to compute air temperature distributions that agree reasonably well with measurements and other models. It was helpful to pre-run the air model in stand-alone operation in order to find relaxation parameters that provided relatively fast convergence through a process of trial and error. Non-physical or wildly varying temperatures were not encountered. We are mainly interested in computing temperature distributions in the context of load and energy calculations, but the flow field predictions appear reasonable for low mass flow rates. In some cases significant deviations were found between the momentum model and measurements and other models. However, it appears that the model tends to be conservative in terms of erring in the direction of the mixing model. This is likely because of numerical diffusion that is expected with use of coarse grids in such a numerical scheme. The model does not handle turbulence and therefore allows some exaggerations where high velocity flows are being treated as laminar. In reality, turbulence dampens out the strength of thermal plumes and this might not be captured in the momentum model. The model appears to over-predict buoyancy. Coarse grids appear to counteract the exaggerated plume problem probably because the momentum source is spread out over larger control volumes. While additional testing is required, it appears that neglecting blockages does not have a large affect on the temperature results for the small zones and contents in the test cases. The model appears useful for generating zone air temperature data at the coarse grid resolutions used in zonal modeling for testing the coupling framework under certain types of room airflow situations. These situations are where thermal stratification is present and room mean air velocities are low (<0.2 m/s).

4.4 Validation of Coupled Air and Loads Models

The component air models have been coupled to models for calculating loads. The air models were coupled to a collection of routines from the Loads Toolkit that iteratively calculate the daily cooling load for a steady periodic design day. The coupled model formulation is discussed in Section 3 and the implementation is discussed in Section A.2.

This section presents examples from validation exercises conducted in order to validate the coupled air and loads models. Adequate validation is always a problem for building simulation and the problem is certainly not easier with air models. Adding detailed data on the distribution of air temperatures to the usual requirements for validating dynamic building simulations makes high-quality data all the more scarce. Because heat flow conditions through zone surfaces are not always well characterized in room airflow experiments, data that are suitable for checking the accuracy of stand-alone air models are not necessarily applicable to coupled surface and air models. In view of the measured data that are available, it was decided that validation under steady-state situations is the only option available. Note that it is easier to obtain high-quality data from steady-state experiments. Since the air models are quasi-steady, this seems appropriate. The dynamic surface and environment models used in the Air Model Toolkit are the same as those in the Loads Toolkit and are assumed (and appear) to be well verified and validated

elsewhere. Therefore we proceed with steady-state validation exercises in order to test the framework for combining air models and load calculation routines.

4.4.1 Displacement Ventilation Models

For validating coupled surface and air modeling, the boundary conditions are T_{os}, T_{supply} and $T_{Setpoint}$ along with the internal loads. The value for $T_{Setpoint}$ (see Table 4.8) is then considered an input for the desired room air temperature. For a cooling load calculation, we can imagine the simulation is of a variable air volume system and the model "finds" the flow rate that was actually fixed in the experiment. For the coupled models, values for outside face boundary conditions are fixed and we are interested in the predictions for inside face surface temperatures which are now freed up for modeling (compared to the stand-alone air models). The system flow rate should be controlled to meet $T_{Setpoint}$. A value for $T_{Setpoint}$ was extracted from the measured air temperature data by interpolating between measured air temperature locations to obtain a value at 1.1 m from the floor. We are interested in the values obtained for $T_{leaving}$.

Steady-state validation checks have been performed for two cases drawn from the measurements by Li et al. (1993) and two cases from the measurements by Rees (1998). We continue using the test case DisplaceVent_B3 (already described in Section 4.2.1) as an example for couple models under steady state. The surfaces are modeled as resistance constructions with Li's values for U-factor of 0.36 ($W/m^2 \cdot K$) for all surfaces except the west wall which had U-factor 0.15 ($W/m^2 \cdot K$). Table 4.8 shows the "outside" boundary conditions for testing the coupled air and loads models.

Table 4.8 Coupled Air and Surface Model Boundary Conditions
Case DisplaceVent_B3

	Value	Units
$Q_{conv,s}$ (estimated splits)	225	[W]
$Q_{rad,s}$ (estimated splits)	75	[W]
T_{supply}	18.0	[ºC]
$T_{Setpoint}$	23.8	[ºC]
Outside Face ("TG"), East, North, West, and Floor	22.63	[ºC]
Outside Face Air ("TB"), South and Ceiling	19.9	[ºC]

Table 4.9 lists overall results for the DisplaceVent_B3 case using the coupled air and loads models. An important modeling assumption in this exercise is the split between radiation and convection for the internal load. The mixing model performs adequately. In general we find that the coupled models have only a small affect on the result for $\dot{Q}_{sys}$ but do have a significant affect the result for $\dot{V}$ since the overall air system heat balance depends strongly on the air system temperature difference ($T_{leaving}$ - T_{supply}). The results indicate that (for this case) the Mundt model over predicted the temperature at the outlet leading to a system flow rate that is too low. The Rees and Haves model used a special

flow pattern developed by Rees for this specific case rather than the general rules. The Rees and Haves model agreed with the momentum-zonal model. It is encouraging that the models are able to obtain nearly the same air flow rate as used in the experiment. Figure 4.14 compares the results from three different models using the DT-coupling method and no secondary air system loop.

Table 4.9 Overall results for coupled air and loads model (Case DisplaceVent_B3) *

DisplaceVent_B3	$\dot{Q}_{sys}$ [W]	$\dot{V}$ [m^3/s]	$T_{sysDiff}$ [C]	$T_{leaving}$ [C]	T_{statDB} [C]
Measured Li et. al. (1993)	-285	0.035	6.8	24.8	23.77
Mundt (1996) Extended Model prediction	-256	(0.035)	6.4	24.4	N/A
Toolkit Mixing, coupled	-247	0.036	5.8	(23.77)	(23.77)
Toolkit Mundt Model DT couple, no Tstat Loop	-237	0.0269	7.4	25.44	23.29
Toolkit Mundt Model T couple, Tstat Tol. 0.05C	-232	0.023	8.5	26.5	(23.77)
Toolkit Rees Haves, stand-alone, prediction	-274	(0.035)	6.6	24.6	23.5
Toolkit Rees and Haves Coupled, prediction	-253	0.034	6.3	24.3	23.77
Momentum-Zonal , DT couple, no Tstat Loop	-253	0.035	6.2	24.2	23.77
Momentum-Zonal , T couple, Tstat Tol. 0.05ºC	-252	0.035	6.2	24.2	23.72

* Values in parenthesis are model inputs rather than results

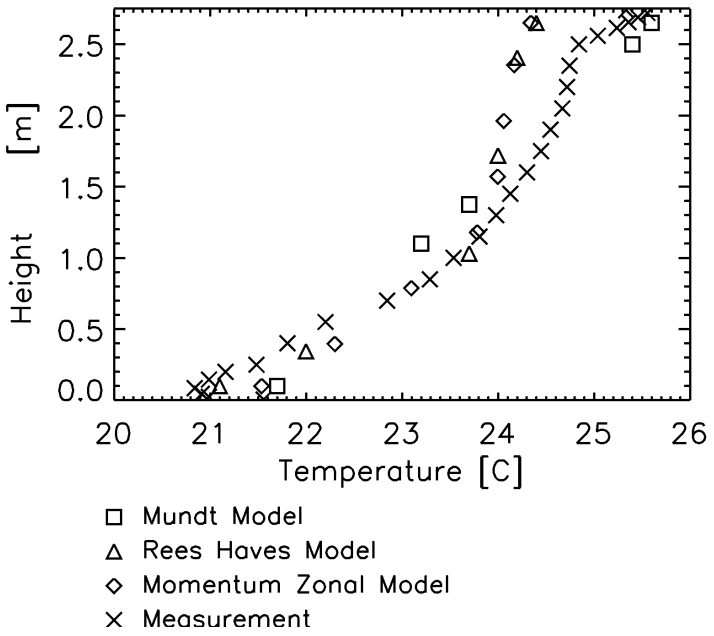


Figure 4.14 Coupled air and loads model predictions for air temperatures case DisplaceVent_B3

4.4.2 Zonal Models

The zonal models are applicable to designs other than displacement ventilation and so different validation exercises are warranted. However, adequate experimental data with outside boundary conditions have not been located. Unless additional measured data are located or created, validation efforts are limited to separate checking of the air models (in stand-alone) and loads routines (traditional building simulation).

Coupling the Inard pressure-zonal has not yet been tested since converged solutions were not obtained in stand-alone operation. Many well-converged solutions have been obtained using the stand-alone POMA model, however obtaining converged solutions was too problematic for the model to be coupled to the loads calculation routines in an effective manner. Other researchers have reported developing computer programs based on combining pressure-zonal modeling techniques with wall heat transfer models (Wurtz e al. 2001, Gagneau and Allard 2001, Mohammedi et. al. 2001). The reported success clearly indicates such an approach is reasonable. However, our finding is that it would require a larger scale project to succeed in developing (1) a very robust method of solving pressure-zonal models, (2) automating the process of applying special cell flow laws, (3) initial guess/error/convergence checking control structures, and (4) linearized formulations that will allow successful coupling to loads calculation. The coupling framework appears to function with the momentum-zonal model and would be expected to function with other zonal models. One basic problem is the difficulties encountered in solving large systems of non-linear equations with many unknowns. Without either just

the right (lucky) initial guess or a very sophisticated equation solving method the solver "settles" at some local minima and stays there.

4.4.3 Mixing Model

Verification exercises compared the mixing model using AirModelLoads.exe to the results provided with the Loads Toolkit (Pederson 2001). The previous toolkit provides input files and output files for 10 zone models of a Los Angeles office building. Figure 4.15 shows a sample of the results. This exercise provides assurance that new results for the baseline loads calculations using the mixing model are reasonable and that the output and post-processing operations are not introducing errors. The deviations are likely because of different convection coefficients selected at compile time.

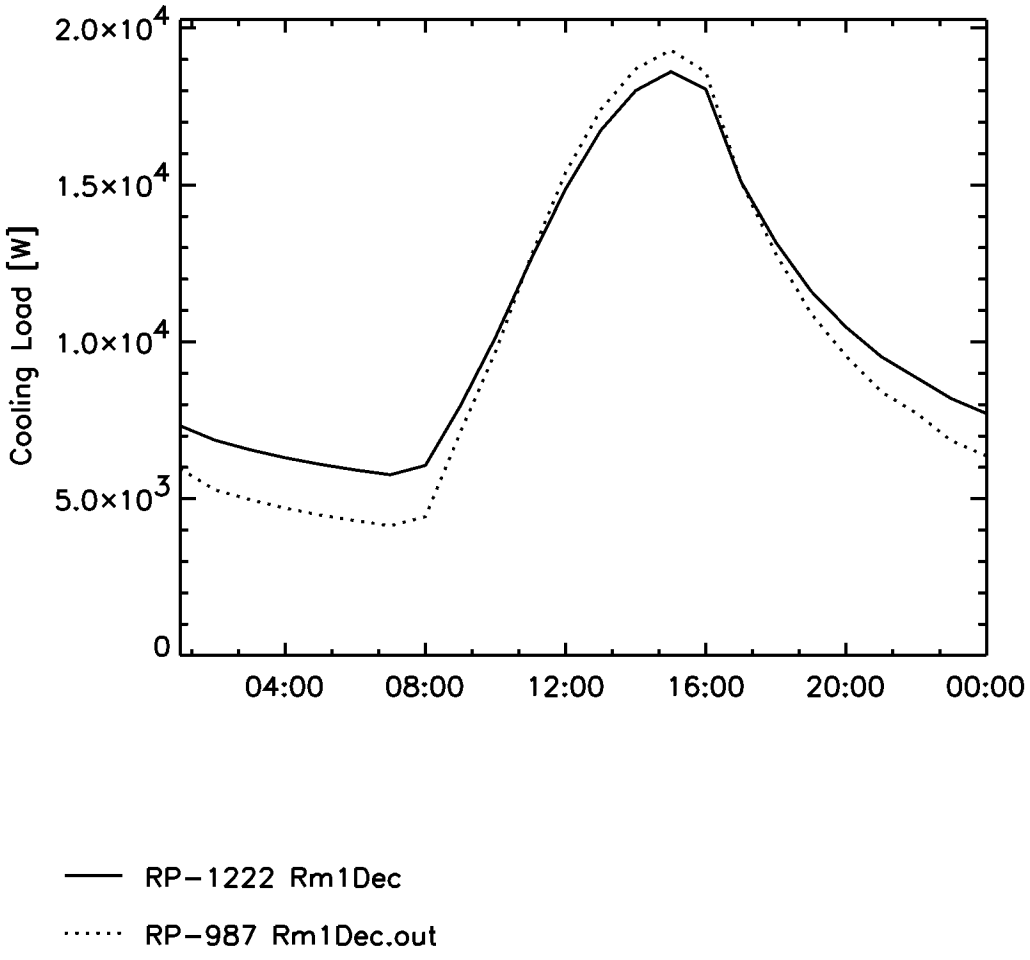


Figure 4.15 Comparison of AirModelLoads results with mixing model to Loads Toolkit results, Case Room 1 December

5 APPLICATIONS

This section presents example results from coupled air and loads models using the program called *AirModelLoads.exe* distributed as part of the Air Model Toolkit. Our intent is to (1) demonstrate that the program and component models are functional, (2) compare results from different air models to each other and the mixing model, and (3) characterize how incorporating air models may affect the results of load calculations. In this section we run the coupled air and surface models in transient problems where internal loads and environmental conditions fluctuate in a steady periodic fashion as for design-day cooling load calculations.

Most of this section presents an office under cooling loads served with displacement ventilation. A second section presents a heating load calculation with a baseboard heating and a "cold" window.

5.1 Office with Displacement Ventilation

The first numerical exercise presented is a test case referred to simply as "Office." A zone description was developed with the intent of modeling a medium-sized office space at cooling design-day conditions for Sacramento, CA. The zone was patterned after a new state government building (Department of Education)that is a fairly energy-efficient (30% better than code). Figure 5.1 diagrams the single zone which models one representative space with a west-facing glazed façade. The room is 8 m by 8 m with an interior height of 2.74 m. The intent is to model an open-plan office with overhead lighting that is for seven people (each with a computer). The actual building uses under-floor air distribution but it is modeled as having side-wall displacement ventilation because the nodal models were developed for side-wall distribution rather than under-floor. The internal load schedules are shown in Figure 5.2 and are patterned after energy modeling practice for a day-shift schedule. Internal loads are model with splits of 50% convection and 50% radiation. The wall constructions and other model inputs are documented in the input files included with the toolkit. The south wall is typical stone veneer on metal framing and was exposed to outside air but has no windows. The west wall has stone on the lower portion and is entirely glazed above with low shading coefficient insulated glazing units.

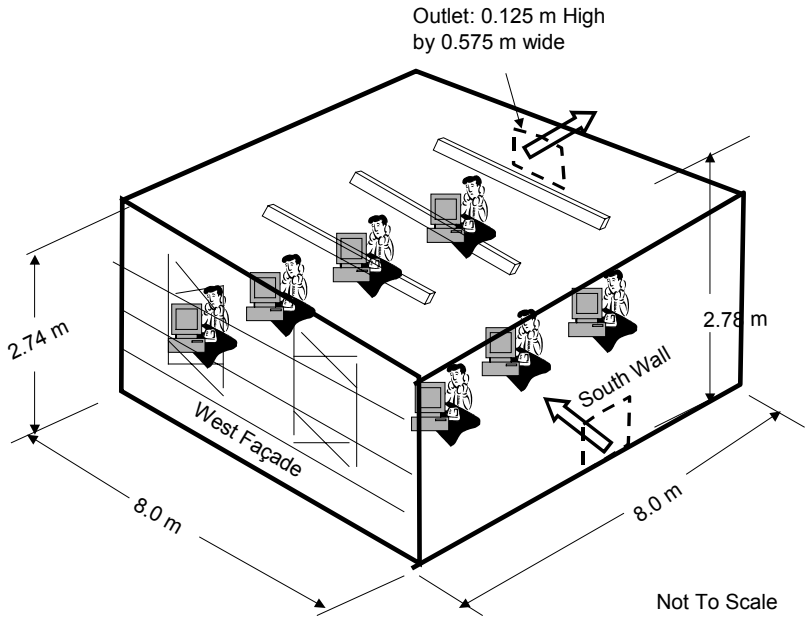


Figure 5.1 Schematic of Office Case, modeled for cooling design day in Sacramento, CA

This test case was used with four different air models: complete mixing, Mundt's linear model, Rees and Haves nodal model, and the momentum-zonal model. For the Mundt model, no portion of the internal sources adding heat to the floor air (floor convection splits were zero). The Rees and Haves model used the suggested "rules" and the h_c correlation for the lowest walls, and 3.0 W/m^2·K for upper vertical wall surfaces, and 5.9 W/m^2·K for ceiling surfaces. Default ASHRAE surface convection film coefficients were used for the other model of 4.68 W/m^2·Kfor the vertical walls, 1.25 for the ceiling, and 4.37 for the floor. The momentum-zonal model used non-blockage interior objects with internal sources distributed uniformly among the objects. Radiation distributions used the default methods rather than user-defined distributions.

5.1.1 Constant Room Air Set Point

Figure 5.3 shows overall cooling load calculation results for the Office case using a constant room air set point of 22.8˚C and supply air (deck) temperature of 17.2˚C.

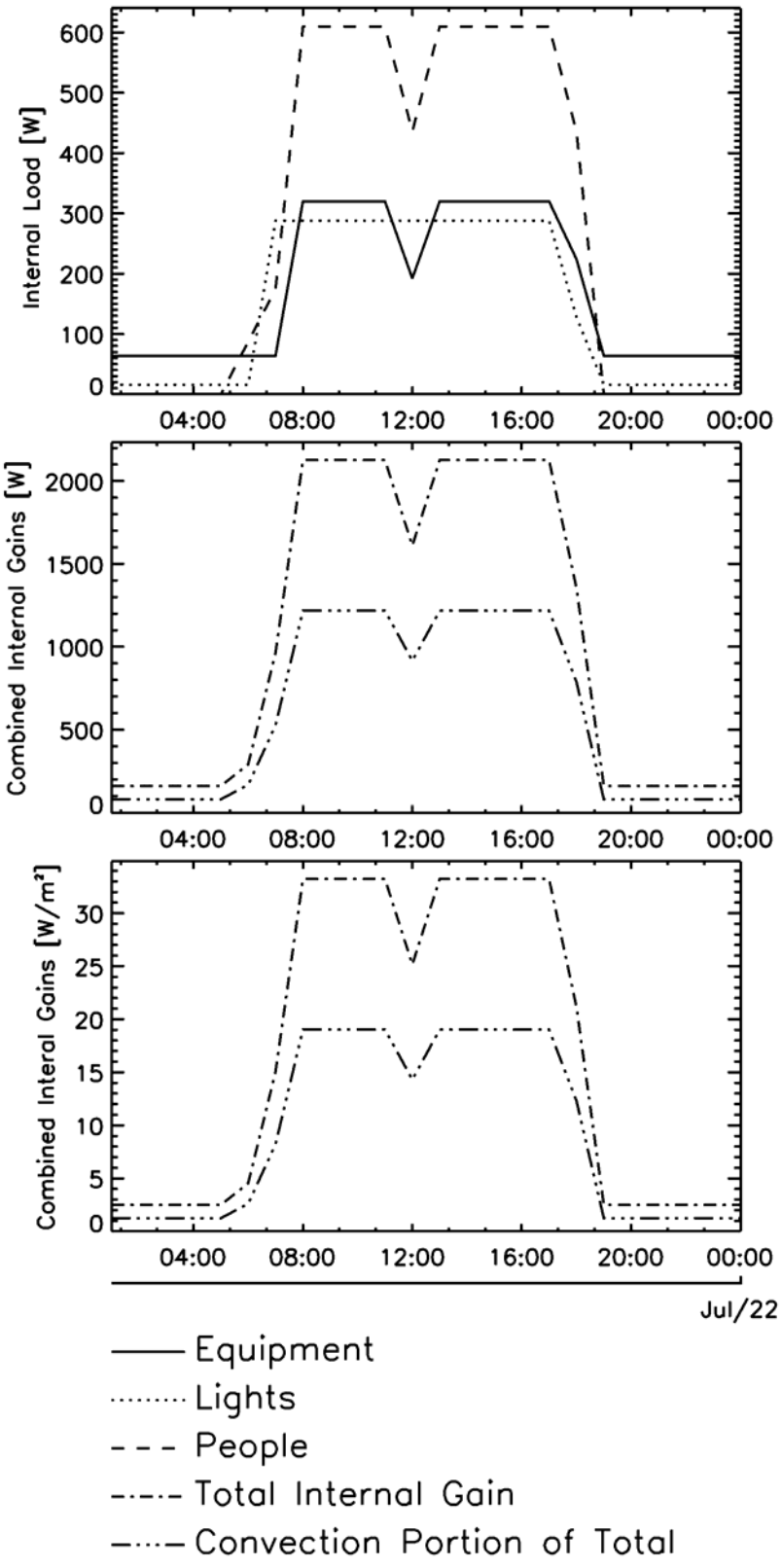


Figure 5.2 Internal load profiles for Office case

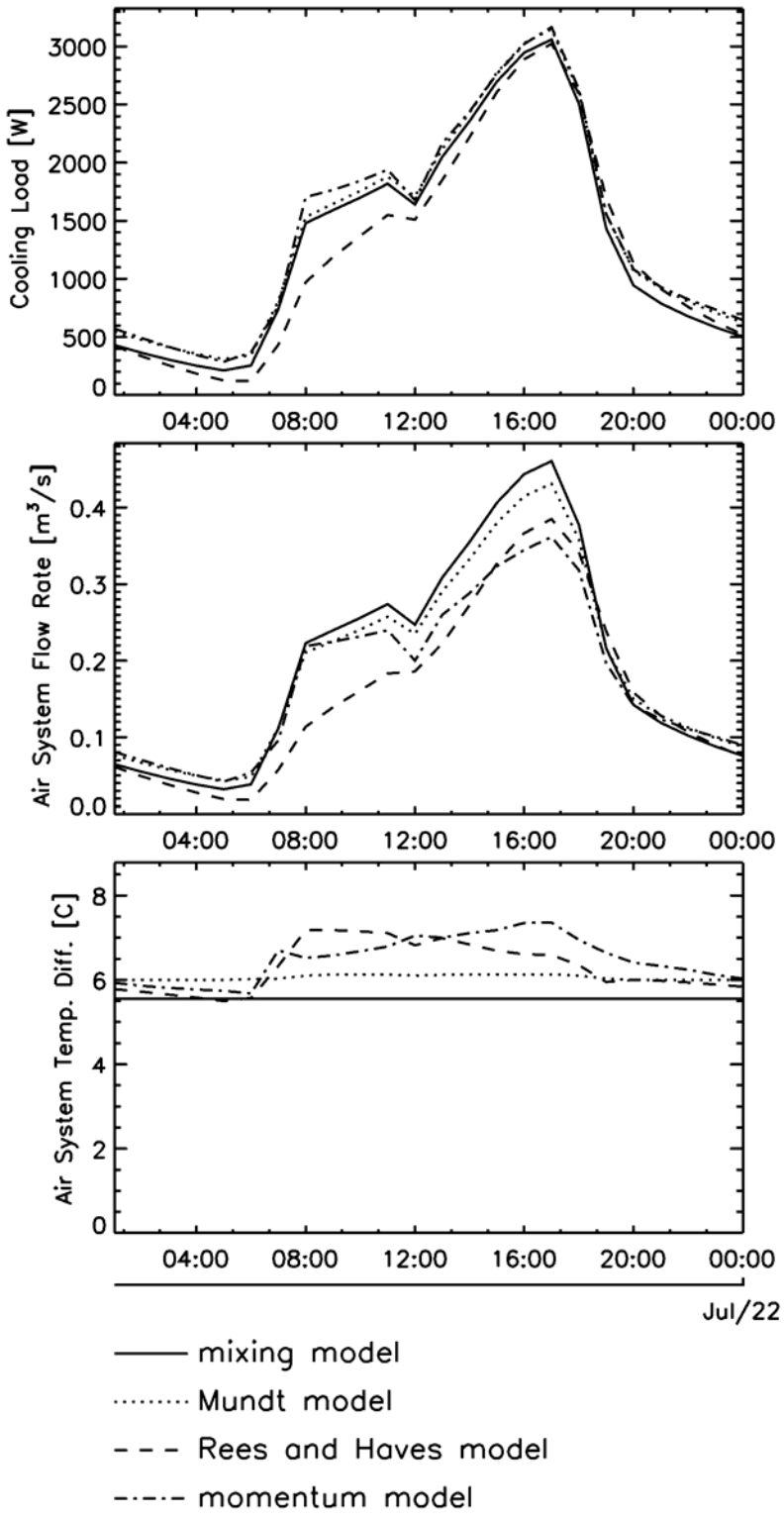


Figure 5.3 Office load calculation results for $\dot{Q}_{sys}$, $\dot{V}$, and $T_{sysDiff}$ for different air models and constant room air set point

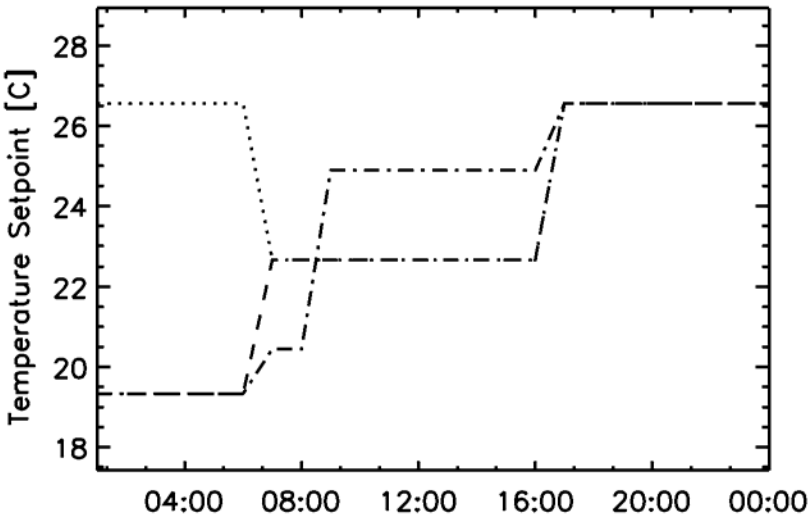
5.1.2 Room Air Set Point Strategies

Non-constant room air temperature set point strategies are also investigated in order to assess how coupled air and loads models respond when simulating diurnal thermal mass storage strategies. The idea is that air models allow predicting how the convective part of internal sources interact with building fabric thermal mass. Is heated air being extracted before it has a chance to warm the building mass? Similar concerns lead to modeling return air splits for lighting loads. Braun et al. (2001) studied possible cooling load reductions using various room set point control strategies. The central idea is that pre-cooling the building thermal mass during off peak hours can reduce the total load during on-peak hours. (This sort of load leveling is extremely important to electricity generation interests; Braun found reductions in total cooling costs of up to 40%.) Three set point strategies were draw from Braun's research, "moderate pre-cool," "max discharge," and the more conventional "night setup." These set point strategies are shown in Figure 5.4. As discussed in Section 3.6, a fluctuating room air temperature will introduce errors associated with the real heat capacity of the air and the use of the quasi-steady modeling assumption. The magnitudes of these errors have been calculated and are shown in Figure 5.4.

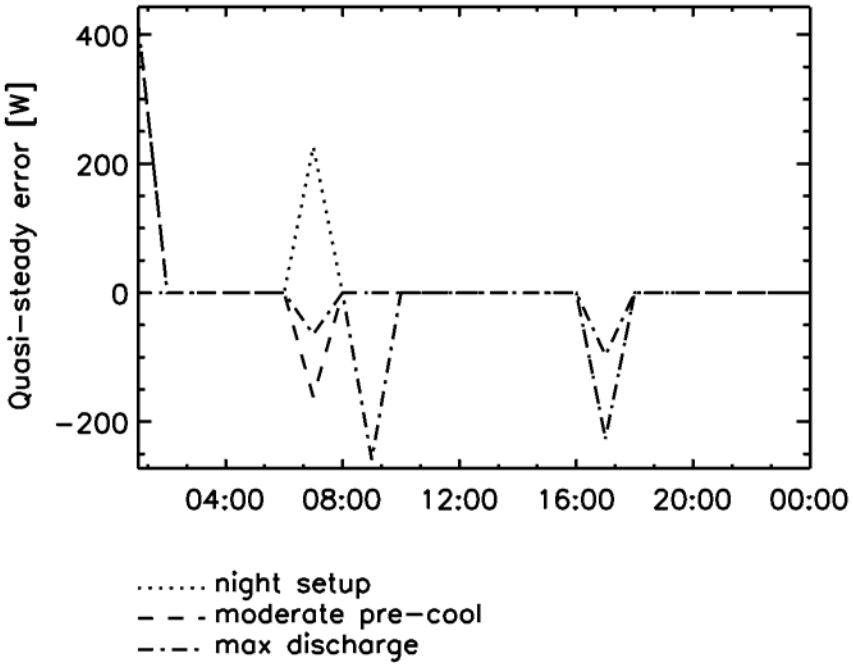
Result for the office case cooling load using room air set point strategies are shown in Figures 5.5. Results for the moderate pre-cool strategy are also shown in Figure 5.6 along with air system flow rate and temperature differences. The load profiles are similar to Braun's however different zones are being modeled so it is not possible to make a direct comparison. Figure 5.6b shows that relatively high supply air temperature of 17.2˚C (needed for displacement ventilation) requires excessive amounts of air system flow in order to meet the early morning pre-cooling set point of 19.3˚C.

A simulation exercise ran the office case with and without carpet installed in order to explore how the models might capture any benefits from more exposed thermal mass. Figure 5.7 shows the results with and without carpet for the mixing model and the momentum model using the moderate pre-cool set point strategy. For the mixing model the afternoon peak cooling load reduction by removing the carpet was 17% and for the momentum model the reduction was 22%. Figure 5.8 plots data from the same simulations as Figure 5.7 but in a much different way.

Figure 5.8 shows how selected air reference temperatures changed when using the air model. These deviations are plotted against the ratio of the overall cooling loads at the same time step. Figure 5.8a can explain why air models do not always affect cooling loads since the effective bulk air temperatures are shifted both up and down with respect to those of the mixing model. This appears to roughly balance out in many cases resulting in little change in the net cooling load. Figure 5.8b shows qualitative differences in that the data cluster further below 1.0 than do the data of Figure 5.8a. This supports the assertion that improved detail in thermal zone modeling is helpful for strategies that make better use of the building thermal mass.



(a)



(b)

Figure 5.4 Room air temperature control strategies (a) set point values, (b) associated magnitudes of errors from quasi-steady assumption ($\rho c_p V \frac{dT}{dt}$)

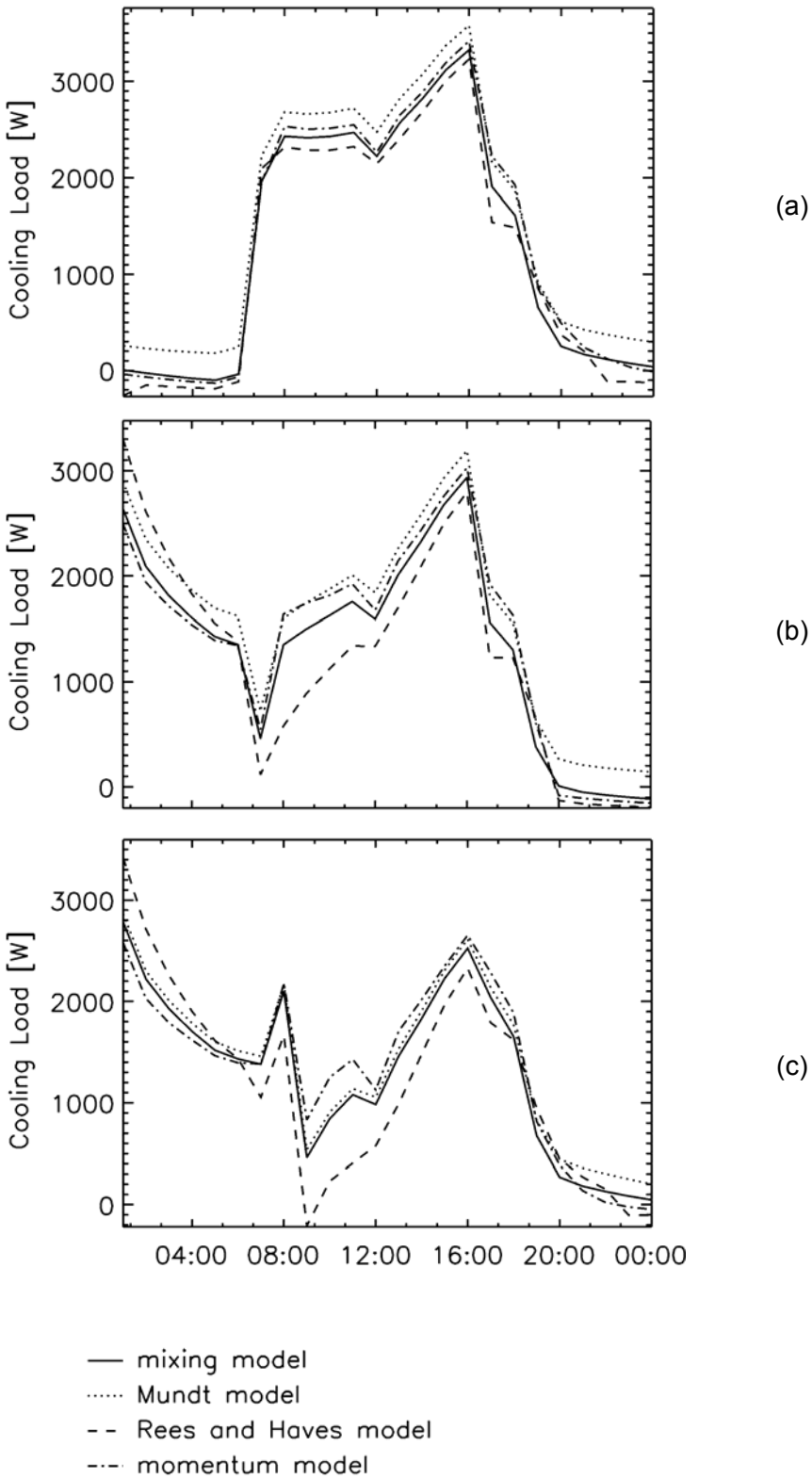


Figure 5.5 Cooling loads for Office for different models and set point strategies, (a) night setup, (b) moderate pre-cool, and (c) max discharge.

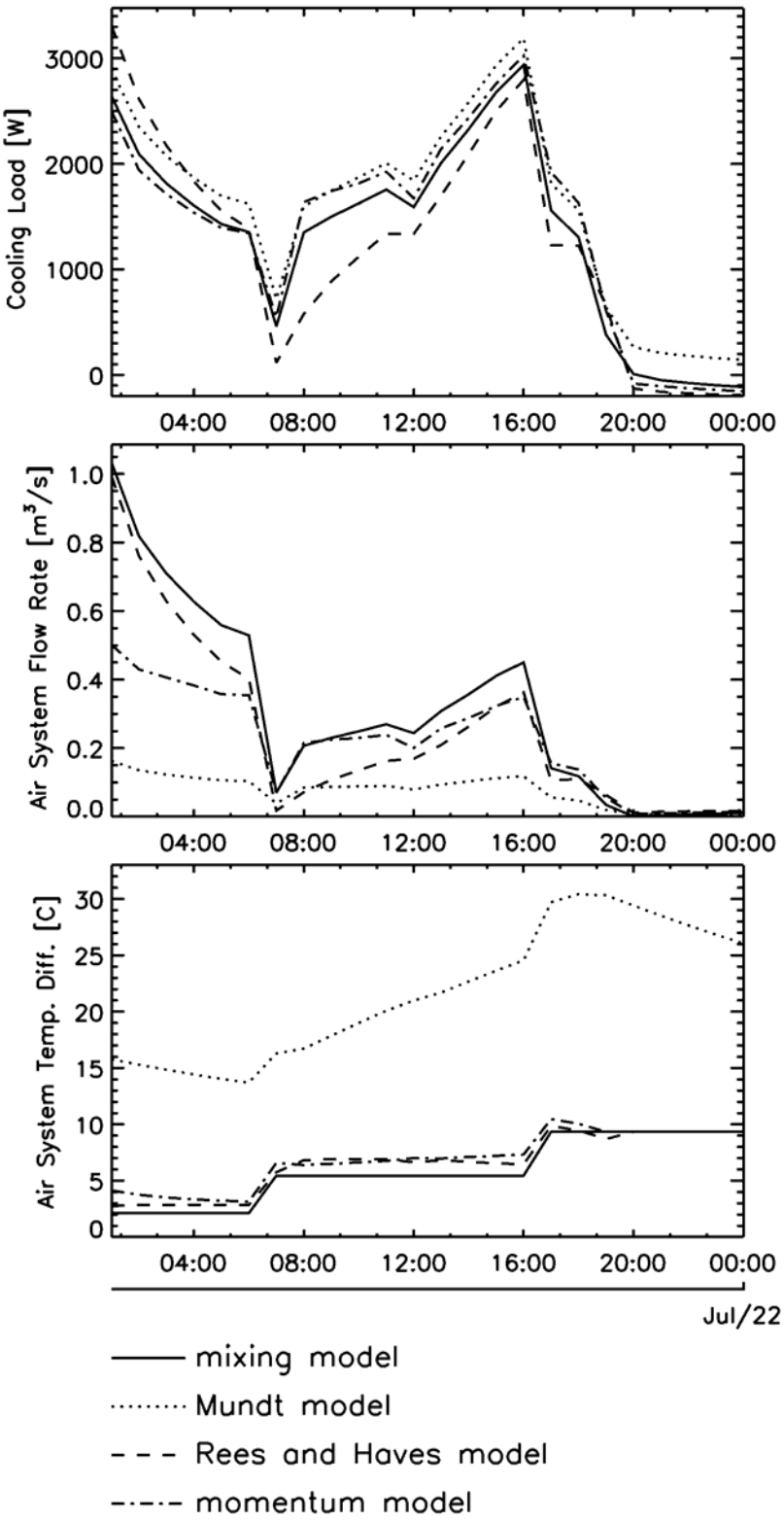


Figure 5.6 Office load calculation results for different air models and moderate pre-cool set point strategy (a) $\dot{Q}_{sys}$, (b) $\dot{V}$, and(c) $T_{sysDiff}$

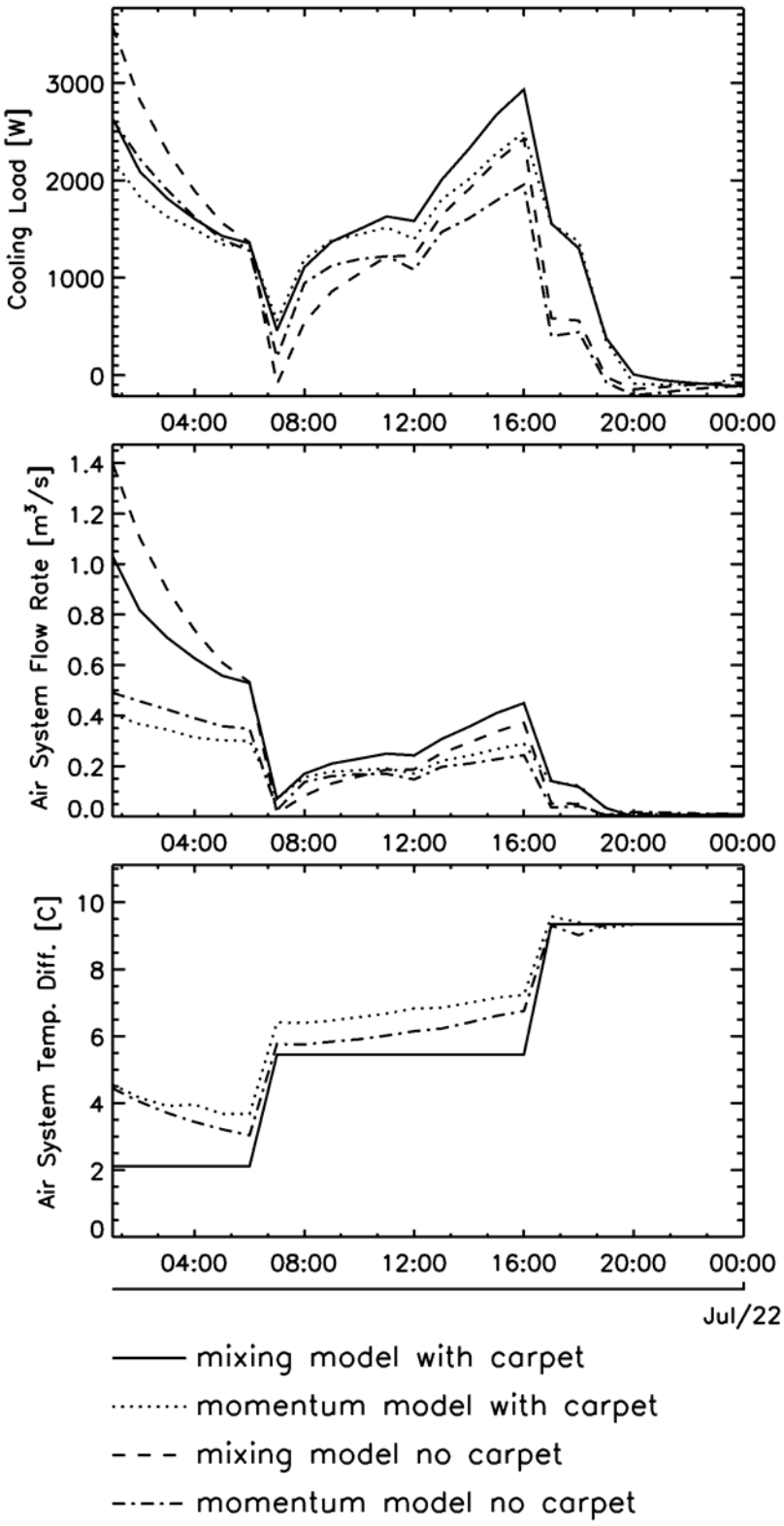
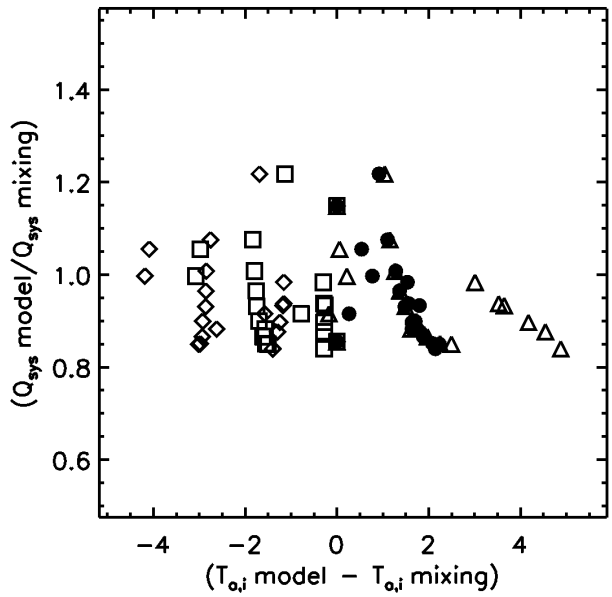
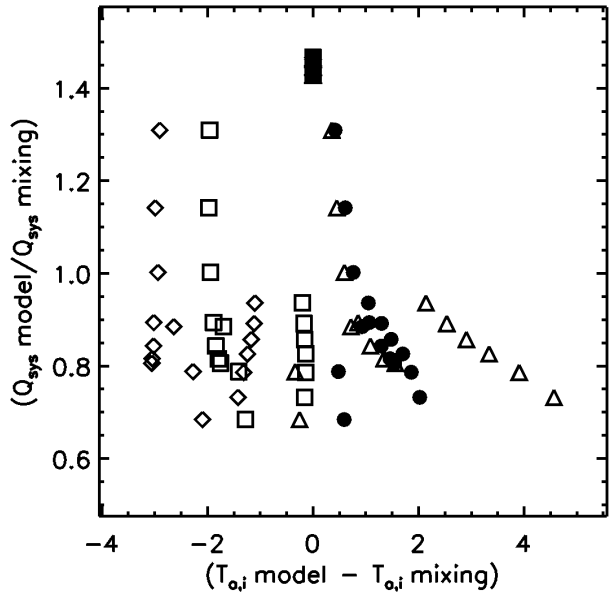


Figure 5.7 Office load calculation results for $\dot{Q}_{sys}$, $\dot{V}$, and $T_{sysDiff}$ for mixing and momentum air models with and without carpet, moderate pre-cool set point strategy



(a)



(b)

◇ Air near floor

△ Air near ceiling

□ Air near lower wall

● Air near upper wall

Figure 5.8 Ratio of change in $\dot{Q}_{sys}$ with momentum-zonal air model to $\dot{Q}_{sys}$ with mixing model as a function of deviations in $T_{a,i}$ (a) with carpet and (b) without carpet

The results for cooling load show both increasing and decreasing loads depending on the air model. Comparing the load curves in Figure 5.5 to those in Figure 5.3a shows that with the more complex set point strategies there is more variation in results for total loads. The air models are capturing something different, but without adequate and comparable validation it is impossible to say which model is more accurate. Because of the serious potential for diurnal thermal mass strategies to save cooling costs and the strong interaction with architectural design, further investigation and validation are warranted so that forward building load and energy models can properly credit such strategies.

5.1.3 Air System Control Loop

With out the complete mixing assumption there is not necessarily a unique solution for air system flow rate. The option of using an additional iteration loop to determine air system flow rate with feedback from a thermostat was discussed in Section 3.5. The basic idea is to run a proportional controller for $\dot{V}$ based on the air model's prediction of temperature at the thermostat and the desired set point. This iteration loop is considered optional with DT-coupling but probably required for T-coupling. The many coupling options are grouped into two categories: (1) DT-couple and (2) T-couple. In DT-couple there is no secondary loop but a corrective shifting occurs in the loads side. This was found to be effective as solutions did get "dragged together" so that after a few iterations the magnitude of such shifting was typically around 0.02 K or less. In T-coupling the system flow rate is found using a secondary air system loop.

The secondary air loop did not perform well with the Mundt model because the linear model does a poor job of predicting temperature at the thermostat. More recent correlations by Yuan (1999a) might be used to correct this deficiency. The secondary air system control loop completely failed with the Rees and Haves model apparently because the prediction of the thermostat reading was insensitive to flow rate.

Both coupling strategies performed well with the momentum-zonal model. Figure 5.9 shows the results and very little difference was found for this case between the two coupling methods. The computing time is roughly a factor of 1.6 higher for the secondary air loop with final a thermostat tolerance of 0.05. The added computation time is because of the additional calls to the air model. For example, the DT-couple method with no secondary air loop made 720 calls to the air model while the T-couple method with the secondary air model made 1316 calls to the air model. The proportional gain calculated with Equation 3.11 appears to work well offering quick response.

The two coupling methods, T-couple and DT-couple, can provide essentially the same results if the air model is able to predict a good value for the temperature at the thermostat. The DT-couple model was found to be more stable and faster and is therefore recommended if the thermal zone is being controlled. However T-coupling method can be considered if the model is intended to capture conditions where it does not make sense

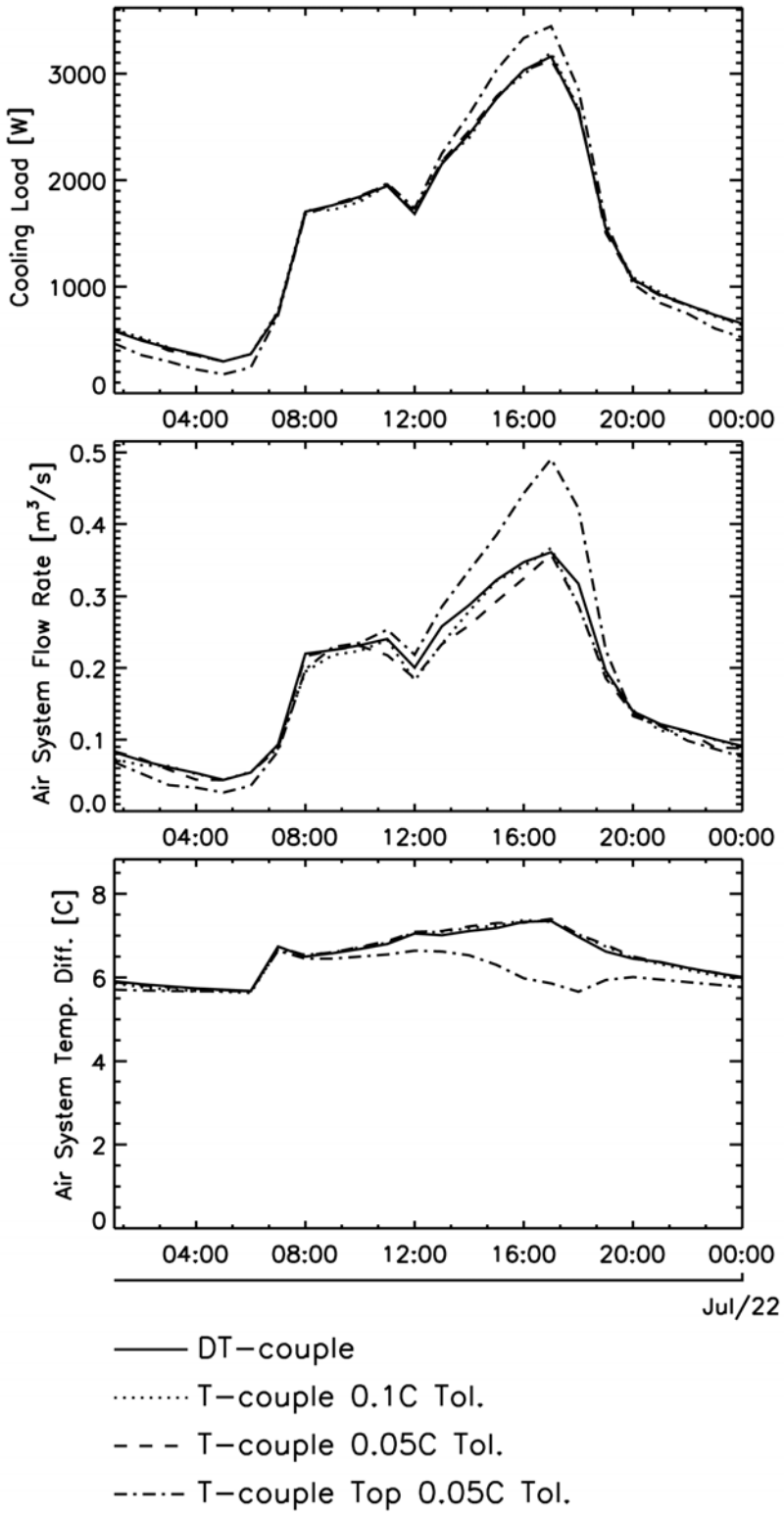


Figure 5.9 Momentum-zonal model results for $\dot{Q}_{sys}$, $T_{sysDiff}$, and $\dot{V}$ with three air system control strategies

to shift by the deviation between the model's prediction at the control point and the room set point (e.g. not controlled).

The secondary air loop also allows controlling system flow rate based on other criteria besides a dry bulb air temperature. Section 3.5.1 discussed using an operative temperature for control that includes the mean radiant temperature. Figure 5.9 also shows results for T-coupling with the secondary air loop but control based on T_{op} as defined in Equation 3.12. This control situation leads to a quite different overall solution with 10% higher cooling loads and a 37% increase in air system flow rate. Such controlling schemes can probably be more realistic and are important because of the large affect on both load and air system flow rate. The increases can be understood by considering that if surfaces are warmer than the setpoint (as they are likely to be because of radiative gains) the air system will cool the air below setpoint resulting in large air temperature differences. Most of the test runs in this report do not use T_{op} criterion nor has this been implemented for the mixing model, but this preliminary investigation of shows such modeling is important.

5.1.4 Zonal Grid Resolution.

Figure 5.10 shows the result for the office case with moderate pre-cool strategy using two grid resolutions: 6x6x6 and 10x10x11. There are slight difference in the result but again it is impossible to say that one is better than the other. The computing time is a factor of 2.6 higher for the finer grid resolution.

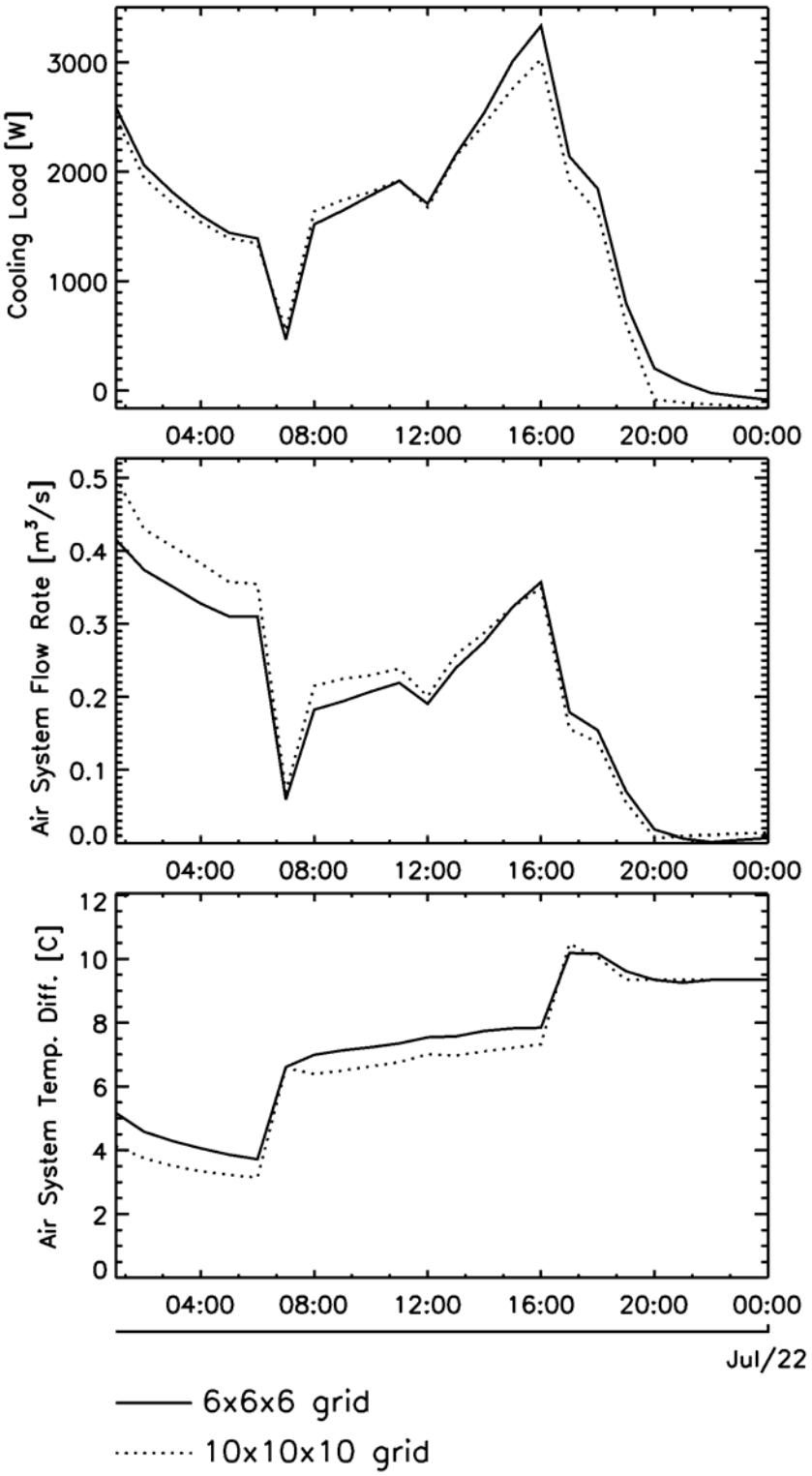


Figure 5.10 Momentum-zonal results for two grid resolutions for moderate-precool.

5.1.5 Computational Requirements

This section compares the computational expense of some of the air models in terms of the time it takes for the entire program to execute a design-day calculation. Table 5.1 summarizes the results. All of these runs are the Office case. For the air models, two warm-up days using the mixing model preceded six days using the air models. The inner iteration loop was set to both 5 cycles and 10 cycles per heat balance time step. Exiting the inner iteration loop would be good practice, but in these runs this iteration loop was fixed for fairness in comparing different models. Adaptive criteria could be based on the progress of inside face surface temperatures. The program uses the mixing model if heating loads are encountered, but will otherwise call the air models a total of 6 x 24 x 5 or 720 different times for the 5 iteration setting and 6 x 24 x 10 or 1440 times for 10 iterations. The momentum-zonal model was run at two different grid resolutions, 6 x 6 x 6 and 10 x 10 x 11. All simulations were run on the same 800 MHz personal computer. This is a dual-processor Windows 2000 machine and so the single-processor program was typically able to use one entire processor. The momentum-zonal model had the highest memory requirements at about 38 Megs (fixed array sizes for 60x60x60 grid).

The Mundt model increased run times by a factor of about 4. Much of this is probably because of the computing overhead involved in the coupling framework that creates an additional layer of interface between the air model and the loads routines. The Rees and Haves model increased run times by a factor of 3 to 7. The momentum zonal model increases run times by a factor of 100 to 300 depending on the grid and the level of dynamics in the thermal situation. With the momentum zonal model, simulation times are longer when there is more dynamic behavior in the thermal zone because the model converges much faster when boundary conditions have not changed much and the previous solution is still quite good. The average time for each call to the momentum-zonal model ranged from 0.5 to 7 seconds. Annual energy simulations have not been performed but the computing time could be estimated by multiplying the values in Table 5.1 by a factor of sixty (minutes become hours). This indicates annual energy calculations might take 30 minutes using a nodal-network model and 10 to 45 hours using the momentum-zonal model.

Table 5.1 Simulation run times for load calculations with different air models

Air model	$T_{setpoint}$ Strategy	5 Iterations per time step (hr:min:sec)	10 Iterations per time step (hr:min:sec)
mixing	Constant	0:00:05	0:00:10
	Moderate Precool	0:00:09	0:00:17
Mundt	Constant	0:00:19	0:00:28
	Moderate Precool	0:00:38	0:00:45
Rees Haves	Constant	0:00:37	0:01:13
	Moderate Precool	0:00:30	0:00:59
momentum-zonal 6x6x6	Constant	0:09:47	0:13:08
	Moderate Precool	0:17:12	n/a
momentum-zonal 10x10x11	Constant DT-couple	0:29:16	0:39.05
	Constant T-couple	0:48:15	1:11:41
	Moderate Precool	0:44:55	0:54:11

5.2 Convective Heating

A second numerical exercise is presented to explore heating applications using a test case referred to as "BaseBoard." The BaseBoard case zone description was developed with the intent of modeling a baseboard-style convective heater situated underneath a cold window. This is a design-day calculation using heating conditions for location Minneapolis, MN. The geometry and configuration matches that of the test case ConvectHeat discussed in Section 4.3 and Diagrammed in Figure 4.11. The room is 5.16 m by 3.65 m with an interior height of 2.43 m and two people, overhead lights and a convective heater. The BaseBoard case adds mythical wall constructions and cold outdoor conditions that are not part of the ConvecHeat case's experiment. The steady outdoor air temperature of –26.5C corresponds to 99.6% heating design condition (ASHRAE 2001). The wall constructions and other model inputs are documented in the input files included with the Air Model Toolkit. The north wall and roof are exposed to the outdoors but have no glazing. The floor, south, west walls are interior partitions. The south wall is typical stone veneer on metal framing and was exposed to outside air but has no windows. The east wall has stone on the lower portion and is entirely glazed above with low shading coefficient insulated glazing units. The values for h_c were set so that they would be same for both the mixing model the momentum-zonal model. These were ASHRAE default values of 4.68 ($W/m^2 \cdot K$) for the walls, 4.37 for the floor, and 1.25 for the ceiling.

The results are given in Figure 5.11 and show that the momentum-zonal model is functional for convective heating applications. True validation is difficult, but the results appear reasonable. The change in heating load is small but shows higher heating loads. Surface temperatures are altered by using the air model with deviations from the mixing model ranging from – 0.5 to 1.0ºC.

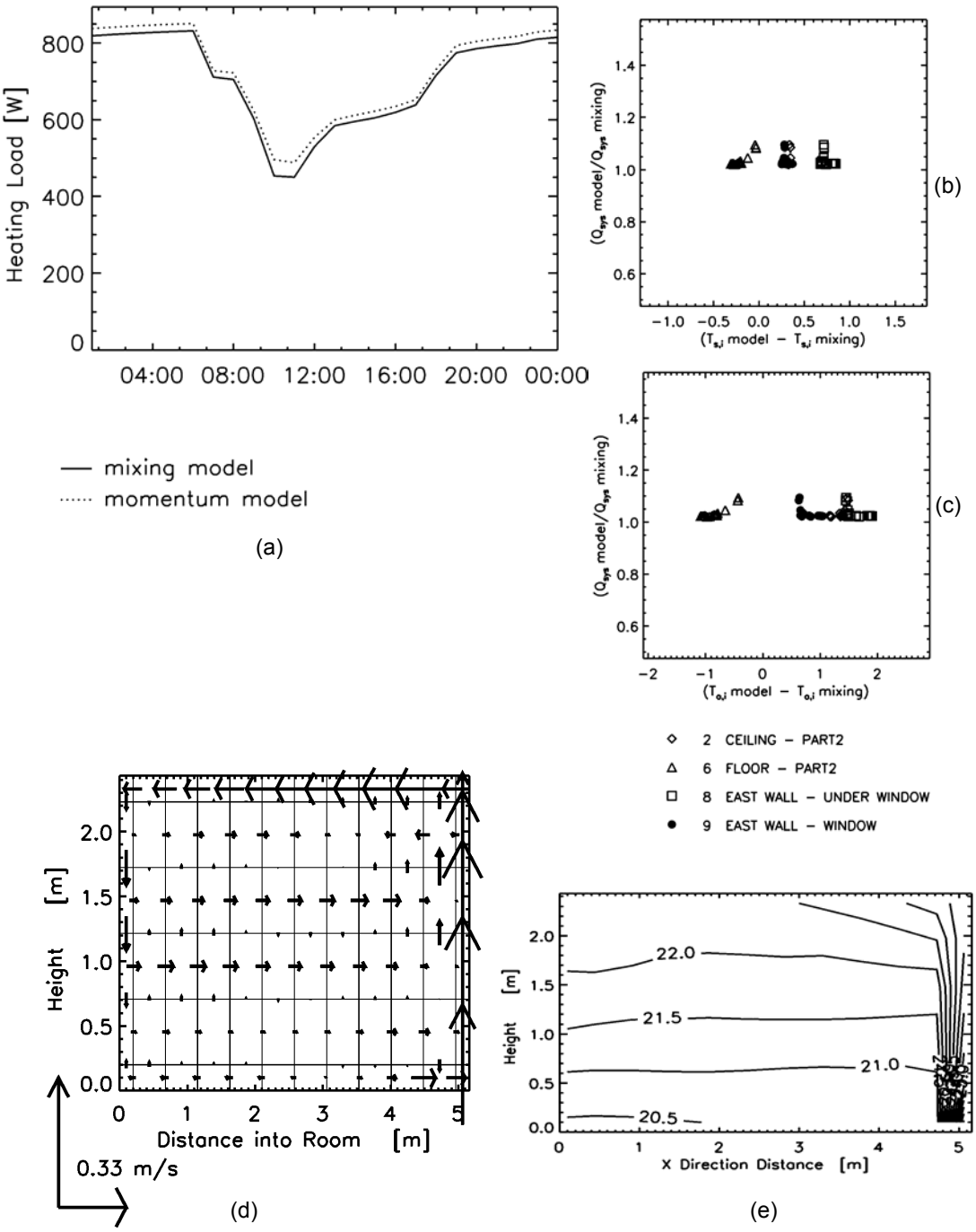


Figure 5.11 Heating load results for case base-board heating case (a) heating load, (b) change in $\dot{Q}_{sys}$ versus T_s (c) change in $\dot{Q}_{sys}$ versus T_a (d) computed flow field at 9:00 am (e) computed air temperatures at 9:00am

6. CONCLUSIONS

This research project developed a framework and implemented computer code for coupling room air models and heat balance based load calculations. The load calculations used the successive substitution iterative technique implemented by Pederson et al. (2001). The code and model formulation were altered to accept zone air temperature as an array of values so that surfaces, returns, and a thermostat can take on values that differ from the room set point. Applications for such modeling are where the room air is not well mixed such as a stratified condition with warmer air at the top of room. The temperature distributions are generated using room air models of the type known as nodal and zonal. Four models were selected from the literature and implemented (with varying success) in computer programs in order to demonstrate and test the coupling framework. These models are (1) the Mundt (1996) model, (2) the Rees and Haves (2001) nodal model, (3) the Inard, Bouia, and Dalicieux (1996) pressure-zonal model, and (4) POMA (Lin 1999). A fifth model termed "momentum-zonal" was developed that uses finite-volume techniques to solve the Eüler flow equation.

The model-coupling framework and code appears to perform well although true validation data are not available. Program validation and verification exercises showed that the air models produce reasonable results by themselves and the load calculation routines have been verified elsewhere. The toolkit contains a versatile test program (called AirModelLoads.exe) that performs detailed, hourly load calculations for a single thermal zone where both network and three-dimensional air flow models have been tightly coupled to the load routines. This research focused on sensible load calculations, but room air modeling is also considered very useful for evaluating thermal comfort and indoor air quality. For these purposes, the toolkit's coupling framework provides a thermal envelope calculation engine that generates useful boundary conditions for surface temperatures and system flow rates that should aid detailed modeling of the indoor environment.

Probably the next level of detail beyond the complete mixing model is to model the room air temperature variation as linear in the vertical direction. For displacement ventilation air system cooling, Mundt suggests a simple method of dynamically determining a slope for a linear model. This simple model provides an air temperature near (0.1 m) the floor that becomes the lower point in determining a linear gradient. The upper point is at the location of the return air duct and is obtained from the current value for cooling load and air system flow rate from the load calculations. These two points providing a convenient method of adjusting the slope of a linear model that easily yields an array of zone air temperatures. The Mundt model was found to do an adequate job of predicting floor air temperature, but generally under predicts air temperatures at a control location of 1.1 m from the floor. When coupled to the loads models this error can cause system flow rates to be modeled as too low and return air temperatures to be too high. Overall cooling loads were only slightly affected by the model. An implementation of the Mundt model would appear to benefit from a two-piece linear formulation where the thermostat reading is provided by the correlations developed by Yaun et. al. (1999a).

The nodal model suggested by Rees and Haves accomplishes a similar task but with better resolution at head-height using a piecewise linear model of the vertical temperature distribution. Room air temperatures are assigned to nodes connected by a network of prescribed airflow rates. Nine unknown air temperature values are found by solving for the roots of balance equations using a non-linear equation solver. The model behaves well in situations where most of the load is from internal sources that add heat to interior plume nodes.

Pressure-zonal models were found to be quite challenging to the numerical solvers and/or extremely sensitive to the initial guess for values of the unknowns (temperatures and pressures). A functioning pressure-zonal model called POMA has been converted to fortran90 and interfaced to the loads toolkit routines. Although many instances of fully converged solutions were obtained with POMA, this was not the typical outcome. When coupled, the POMA solver generally remained "stuck" at local minima and failed to find roots to the equations. The initial guess problem is closely tied to the numerical solver and appears generally problematic for implementing non-linear pressure-zonal model into time marching building simulation programs. Other researchers report better results (Mohammedi et. al. 2001, Wurtz 1999) where sophisticated mathematical modeling packages have been used to assist in solving the systems of equations. A better selection for a pressure-zonal model might have been the linearized formulation of Musy and Wurtz (Wurtz et al. 2001).

A so-called momentum-zonal model has been implemented and coupled to the loads routines. This model solves the steady Euler equation using finite-volume numerical techniques. This model is suggested as a "zonal" model when used with coarse grids. Our research in the area of CFD made this sort of code easier to implement than the pressure-zonal models. However, our experience in conducting this research project leads to the conclusion that the *direct* solutions (for temperature and pressure) sought for non-linear pressure-zonal models are more difficult to obtain in a robust manner than the *sequential* iterative solutions (for temperature and mean flow components) sought with the momentum-zonal model. Our intent with this contribution to the Air Model Toolkit it is to make a very simple (and fast) code available testing the coupling framework. The momentum-zonal model was validated against measured data and found to predict temperatures well in situations where room air is stratified. Additional validation would be necessary before apply the model to conventional forced ventilation although it could be used to compute flow in drift cells between special-cells with flow laws.

It appears practical to use room air models to predict the air temperature at the location of the thermostat and where it enters the returns in order to provide more detail on how the thermal zone is represented to plant models. Results from cooling load calculations using the complete mixing have been compared to those using the room air models. When room set points were constant, predictions showed only minor changes to the overall cooling load indicating that the mixing model does an adequate job of determining system loads. For displacement ventilation, the air models showed consistent and significant changes to the predicted air system flow rate and return air temperatures. This allows simulations to account for the higher heat extraction efficiency offered by

displacement ventilation. In one test case, system flow rate reductions of around 25% were obtained which could have important implications for fan energy and even sizing air handlers. The altering of return air temperatures will also affect plant/economizer thermodynamics. The results for individual surfaces do vary with the additional modeling detail but in aggregate losses and gains often even out. However when intentionally scheduling room air temperatures to take advantage of diurnal thermal mass in the building surfaces, the air models were found to have a larger affect on the overall cooling loads. An alternative strategy for air system control based on an operative temperature that includes mean radiant temperature showed a 10% affect on cooling load and a 37% affect on system flow rate.

The time required for computations increased by about a factor of 4 for the nodal models compared to the mixing model. Nodal models could be added to whole building simulations with moderate additional computation time required and could be expected to improve predictions of flow rate and return air temperatures experienced by the plant. For the momentum-zonal model with a grid number of 216, the increase in computation time was about a factor of 100 longer than the mixing model. Incorporating a three-dimensional, coarse-grid, air-modeling package into building simulation is feasible with contemporary computers and but would require users to be very patient.

Future research should consider directing additional modeling detail to the areas of surface convection heat transfer coefficients and thermal radiation. Convection correlations should be revisited in order to determine appropriate relations when using the adjacent air method (e.g. 0.1 m rule) rather than the complete mixing method for determining the bulk air temperature. The thermal interactions between internal loads and the building thermal mass are by both radiative and convective processes with each comprising roughly half of the heat transfer. The results of air models depend strongly on the convection loads from internal sources and so cannot be accurate if the convection/radiation splits are inaccurate. This leads to direct modeling of the surface temperatures of the contents.

There is a critical need for measured data to use in validating coupled air and loads models. Although challenging and expensive, controlled laboratory experiments with diurnal design-day fluctuations and detailed air measurements are needed to validate the detailed buildings simulations discussed here. Future measurements related to such studies should report the shapes, locations, and surface temperatures of heat source simulators used in experiments as well as distributions of air temperatures at 0.1 m away from enclosure surfaces.

ACKNOWLEDGEMENTS

I would like to thank Dr. Qingyan Chen for his patience, guidance, and support that allowed performing this research.

The POMA pressure-zonal model implemented in the toolkit is based on fortran90 code contributed by Yi Lin and some POMA documentation is taken from Lin's thesis (Lin 1999). Brent Griffith prepared the remainder of the program code and this report.

ASHRAE Technical Committee 4.7 sponsored this research as RP-1222 where a slightly different title is used, "Incorporation of Nodal Room Heat Transfer Models into Energy Calculation Procedures." ASHRAE is the American Society of Heating, Refrigeration and Air-conditioning Engineers, Atlanta, GA.

REFERENCES

Allard, F. 1998. *Natural Ventilation in Buildings: A Design Handbook*. London, James and James.

Allard, F., Brau, J., Inard, C., and Pallier, J.M. 1987. Thermal experiments of full-scale dwellings cells in artificial conditions. *Energy Building,* Vol. 10, pp. 49-58

Allard, F. and Inard, C. 1992. Natural and mixed convection in rooms: Prediction of thermal stratification and heat transfer by zonal models. Proc. of ASHRAE conf. Room Air Convection and Ventilation Effectiveness, Tokyo, Japan *ISRACV, 1992*, pp. 335-342

Arai, Y., Togari, S. and Miura, K. 1994. Unsteady-state thermal analysis of a large space with vertical temperature distribution. ASHRAE Transactions,Vol. 100, Part 2, pp.396-411.

ASHRAE, 2001. *2001 ASHRAE Handbook Fundamentals*. American Society of Heating, Refrigeration and Air-Conditioning Engineers, Inc. Atlanta, GA.

Axley, J.W. 2001. Surface-drag flow relations for zonal modeling. Building and Environment 36 (2001) pp. 843-850.

Awbi, H.B. 1998. Calculation of convective heat transfer coefficients of room surfaces for natural convection. *Energy and Buildings* 28(2) 219-227

Bauman F, Gadgil, A., Kammerud, R., Altmayer, E. and Nansteel, M. 1983. Convective heat transfer in buildings: recent research results. ASHRAE Transactions 89 (1)

Beausoleil-Morrison, I. 2000. The adaptive coupling of heat and air flow modeling within dynamic whole-building simulation. Thesis. University of Strathclyde, Glasgow U.K.

Bouia, H. and Dalicieux, P. 1991. Simplified modeling of air movements inside dwelling room. *Proceedings of the Building Simulation Conference,* August, 1991. (now IBPSA)

Brandemuehl, M.J. 1993. HVAC2 toolkit algorithms and subroutines for secondary HVAC systems energy calculations. American Society of Heating Refrigeration and Air-conditioning Engineers. Atlanta, GA

Braun, J., Montgomery, K. and Chatervedi, N. 2001. Evaluating the performance of building thermal mass control strategies. Intl. J. of HVAC&R Research. Vol. 7 (4) pp. 403-428. American Society of Heating Refrigeration and Air-conditioning Engineers. Atlanta, GA

Castanet, S., Bouia, H., Duta, A. and Inard, C. 1998. Ventilation efficiency in dwelling cells: contribution to the validation of a zonal model. *ROOMVENT'98*, Vol. 3, pp.976-981

CHAM. 1999. PHOENICS Version 3.3. Concentration, Heat and Momentum Limited. London, UK.

Chen, Q. 1988. Indoor airflow, Air Quality and Energy Consumption of Buildings. Thesis. Delft.

Chen, Q., Glicksman, L., and Srebric, J. 1999. Simplified Methodology to Factor Room Air Movement and the Impact of Thermal Comfort into Design of Radiative, Convective and Hybrid Heating and Cooling Systems. Final Report for ASHRAE RP-927. ASHRAE Special Publications

Clarke, J.A., Hensen, J.L.M. and Negrao, C.O.R., 1995, Predicting indoor air flow by combining network approach, CFD and thermal simulation, *Proc. of 16th AIVC Conf. Implementing the results of ventilation research*, Palm springs, Sept. 1995, pp. 145-154.

Crawely, D., Lawrie, L., Winkelmann, F., Buhl, W., Huang, Y., Pedersen, C., Strand, R., Liesen, R., Fisher, D., Witte, M., and Glazer, J. 2001. EnergyPlus: creating a new-generation building energy simulation program. Elsevier Science. *Energy and Buildings* 33. pp 319-331.

Deque, F., F. Evin, E. Wurtz, and L. Mora. 2001. Sim_Zonal: software evaluating indoor temperature distributions and air movements for rapid appraisals – first application to an cell. Proceedings of Seventh International IBPSA Conference, Rio de Janeiro, Brazil, August 13-15, 2001

Dols, W., Walton, G. Denton, K. 2000. CONTAMW 1.0 User Manual. Building and Fire Research Laboratory. NISTIR 6476. National Institute of Standards and Technology. U.S. Department of Commerce

EQUA. 2002. IDA Simulation Environment. Sundyberg, Sweden.

Fisher, D., and C.O. Pederson. 1997. Convective heat transfer in building energy and thermal load calculations. ASHRAE Transactions Vol. 103(2).

Feustel, H. 1998. COMIS – An International Multizone Air-Flow and Contaminant Transport Model. LBNL-42182. Lawrence Berkeley National Laboratory.

Fowler, M., Scott, K. 2000. *UML Distilled Second Edition*. Addison Wesley.

Fracastoro, G.V., Mutani, G., Perino, M., 2001. Experimental and theoretical analysis of natural ventilation by means of windows opening. Clima2000/Napoli 2001 World Congress – Napoli (1).

Gagneau, S., J.M. Nataf, and E. Wurtz. 1997. An illustration of automatic generation of zonal models. IBPSA Building Simulation '97. Prague, Czech republic

Gagneau, S., and F. Allard. 2001. About the construction of autonomous zonal models. Energy and Buildings 33 (2001) pp. 245 –250.

Glicksman, L., and Q. Chen. 1998. Interaction of radiation absorbed by moisture in air with other forms of heat transfer in an enclosure. Proceedings of Roomvent '98, Stockholm, Sweden June 14 –17, 1998.

Gschwind, M., J.J. Bezian, T. Hasebe, G. Fonzes, S. Fujita, P. Loiseau, and I. Takeda. 1996. A zonal model to simulate a room heated by a gas heat pump (GHP) BTHEBES", *ROOMVENT'96*, Vol. 1, pp. 61-68

Harrington, L., 2001. Computer Modelling of Night-time Natural Ventilation. Thesis. Loughborough, UK.

Haghighat, F., Y. Lin and A.C. Megri. 2001. Development and validation of a zonal model – POMA. Building and Environment 36 pp. 1039-1047

Hensen, J., 1991. On the thermal interaction of building structure and heating and ventilating system. Thesis. Eindhoven. Holland

Howarth, A.T. 1980. Temperature distributions and air movements in rooms with a convective heat source. Ph.D. Thesis, University of Manchester.

Howarth, A.T. 1985. The prediction of air temperature variations in naturally ventilated rooms with a convective heating, *Build. Serv. Eng. Res. Technol.*, 6 (1985) pp. 169-175

Inard, C. and Buty, D. 1991. Simulation of thermal coupling between a radiator and a room with zonal models, *Proceeding of 12th AIVC Conference*, Ottawa, Canada, 1991, Vol. 2, pp. 125-131.

Inard, C., Bouia, H. and Dalicieux, P. 1996. Prediction of air temperature distribution in buildings with a zonal model. *Energy and Buildings*, Vol. 24, pp. 125-132

Kendrick, J. 1993. An overview of combined modeling of heat transport and air movement. AIVC Technical Note 40. February 1993. Oscar Faber PLC.

Kobayashi, N. 2001. Floor-supply displacement ventilation system. M.Sc. Thesis, 174 pages, Department of Architecture, Massachusetts Institute of Technology, Cambridge, MA.

Laret, L. 1980. Contribution au développement de modèles mathématiques du comportement thermique transitoire de structures d'habitation, *Ph.D. Thesis, University of Liège.*

Lebrun, J. and Ngendakumana, J. 1987. Air circulation induced by heating emitters and corresponding heat exchanges along the wall: test room results and modeling. *Proceeding of ROOMVENT'87*, Stockholm, Sweden, Session 2a, Paper 6

Lebrun, J. 1970. Exigences physiologiques et modalités physiques de la climatisation par source statique concentrée. Ph.D. thesis, University of Liège,

LeBrun, J., Bourdouxhe, J., Grodent, M. 1999. HVAC 1 Toolkit: A toolkit for primary HVAC system energy calculation. American Society of Heating Refrigeration and Air-conditioning Engineers. Atlanta, GA

Li, Y., Sandberg, M. and Fuchs, L. 1993. Vertical temperature profiles in rooms ventilated by displacement: Full scale measurement and nodal modeling, Indoor Air, Vol. 2, pp.225-243.

Li, Y. and Holmberg, S. 1994. General flow and thermal condition in indoor airflow simulation. *Building Environment*, Vol. 29, pp. 275-281

Liesen, R.J., Pedersen, C.O. 1997. An Evaluation of Inside Surface Heat Balance Models for Cooling Load Calculations. ASHRAE Transactions Vol 103(2). (RP-875).

Lin, Y., 1999. POMA – A Zonal Model for Airflow and Temperature Distribution Analysis. Masters Thesis. Concordia University. Montreal, Quebec, Canada

Linden, P.F. 1999. The fluid mechanics of natural ventilation. *Annual Review of Fluid Mechanics*. 31:201-238.

McClellan, T.M., Pedersen, C.O. 1997. Investigation of Outside Heat Balance Models for Use in a Heat Balance Cooling Load Calculation Procedure. ASHRAE Transactions Vol 103(2). (RP-875).

McConnell, S. 1993. Code Complete. Microsoft Press

Mohammedi, M., F. Lainault, f. Milcent, J.-P. Petit. 2001. CFD and Zonal approaches: comparison and validation. Proceedings IBPSA conference Building Simulation 2001, Rio de Janeiro, Brazil Aug. 13-15, 2001

Moser, A., et al 1992. A new method for linking results of detail air flow pattern calculation with multizone models, *13th AIVC conference*, Nice, France pp. 63-77

Mundt, E., 1996, The performance of displacement ventilation systems-experimental and theoretical studies, Ph. D. Thesis, Royal Institute of Technology, Stockholm.

Musy, M., E. Wurtz, A. Sergent. 2001. Buildings air-flow simulations : automatically-generated zonal models. Proceedings of Seventh International IBPSA Conference, Rio de Janeiro, Brazil, August 13-15, 2001

Musy, M., E. Wurtz, F. Winkelmann, and F. Allard. 2001. Generation of a zonal model to simulate natural convection in a room with a radiative/convective heater. *Building and Environment* 36 (2001) pp. 589-596.

Natif, J.-M., E. Wurtz, 1993. Application of the SPARK environment to 3d air flow problems. In Proceedings of Building Simulations '93, Adelaide, Australia. IBPSA, August 1993.

Negrao, C., 1995. *Conflation of Computational Fluid Dynamics and Building Thermal Simulation*. Thesis. University of Strathclyde, Glasgow, UK.

Patankar, S.V. 1980. *Numerical Heat Transfer and Fluid Flow*. Taylor and Francis, Hemisphere Publishing Corporation.

Pederson, C.O. 1997. Development of a heat balance procedure for cooling loads. ASHRAE Transactions Vol 103(1).

Pederson, C.O. 2001. Toolkit for building load calculations. American Society of Heating Refrigeration and Air-conditioning Engineers. Atlanta, GA

Press, W., Teukolsky, S., Vetterling, W., and Flannery, B. 1992. *Numerical Recipes in Fortran 77 The Art of Scientific Computing.* Second Edition. Cambridge University Press.

Press, W., Teukolsky, S., Vetterling, W., and Flannery, B. 1996. *Numerical Recipes in Fortran 90 The Art of Parallel Scientific Computing*. Second Edition. Cambridge University Press.

Rees, S.J., and P. Haves. **1995.** A model of a displacement ventilation system suitable for system simulation. Proceedings of "Building Simulation-95", vol.1 Madison, WI, Aug. 14-18, 1995. pp 199-205

Rees, S.J. 1998. Modeling of displacement ventilation and chilled ceiling systems using nodal models. PhD Thesis, LoughBorough University. 1998.

Rees, S. J., Spitler, J. D., Davies, M. G., and Haves, P. 2000. Qualitative Comparison of North American and U.K. Cooling Load Calculation Methods. ASHRAE HVAC&R Research Vol. 6, No. 1, January 2000.

Rees, S.J. and Haves P., 2001, A nodal model for displacement ventilation and chilled ceiling systems in office spaces, *Building and Environment 36 (2001)* pp. 753-762.

RSI. 2000. Using IDL: IDL Version 5.4. Research Systems, Inc., Boulder. CO.

Sowell, E. 1991. A General zone Model for HVACSIM+ User's Manual. Report No. OUEL 1889/91, University of Oxford.

SPARK, 1997. SPARK 1.0.1 Reference Manual: simulation problem analysis and research kernel. Lawrence Berkeley National Laboratory, Ares Sowell and Associates.

Togari, S., Arai, Y. and Miura, K. 1993. A simplified model for predicting vertical temperature distribution in a large space. *ASHRAE Transactions 99.*

Walton, G.N. 1983. Thermal Analysis Research Program Reference Manual. NBSIR 83-2655. National Bureau of Standards. (now NIST).

Walton, G. 1993. Computer Programs for Simulation of Lighting/HVAC Interactions. NISTIR 5322. National Institute of Standards and Technology.

Wurtz, E., Nataf, J-M, and Winkelmann, F., 1999, Two- and three-dimensional natural and mixed convection simulation using modular zonal models in buildings, *Int. J. of Heat and Mass Transfer*, Vol. 42, pp. 923-940.

Wurtz, E. 1995. "Modélisation tridimensionnelle des transferts thermiques of aéraauliques dans le bâtiment en environnement orienté object", Ph.D. thesis, Ecole Nationale des Ponts et Chausseés, 1995

Wurtz, E., F. Deque, M. Musy, and L. Mora. 2001. A thermal and airflow analysis tool using simplified models based on the zonal method. In Proceedings of CLIMA 2000 7th REHVA World Congress held in Napoli September 2001

Van der Kooi, J. and Bedeke, K. 1983. Improvement of cooling load programs by measurements in a climate room with mass. Proceedings of the XVIth International Congress of Refrigeration. Vol. E, pp. 54-60.

Voeltzel, A. Carrie, F.R. and Guarracino, G., 2001, Thermal and ventilation modeling of large highly-glazed spaces, *Energy and Buildings*, Vol. 33, pp. 121-132.

Yuan, X., Chen, Q., and Glicksman, L.R. 1999a. Models for prediction of temperature difference and ventilation effectiveness with displacement ventilation. ASHRAE Transactions, 105(1), 353-367.

Yuan, X., Q. Chen, L. Glicksman, Y. Hu and X. Yang. 1999b. Measurements and computations of room airflow with displacement ventilation. ASHRAE Transactions Vol 105(1). (RP-949)

Zhai, Z., Q. Chen, J.H. Klems, and P. Haves. 2001. Strategies for coupling energy simulation and computational fluid dynamics programs. IBPSA Conference. Rio de Janeiro, Brazil.

Zhai, Z., Chen, Q., 2002. Coupling energy simulation and computational fluid dynamics programs. Final report for project Bild-IT for Lawrence Berkeley National Laboratoy. Berkeley, CA 94720. LBNL-

APPENDIX A: SOFTWARE DEVELOPER GUIDE

This appendix provides information for the programmer-user of the Air Model Toolkit developed for ASHRAE Research Project-1222. The first section presents an overview of the contents of the Air Model Toolkit including programming standards and input and output methods. The second section discusses how the framework for coupling air models to the surface models is implemented. The third section of Appendix A documents some of the changes made to Loads Toolkit's routines that lie outside the main focus of Air Model Toolkit but were needed to broaden the applications with which we could use the mixing model for baseline comparisons. Then the following five sections discuss each of the separate air model components. The last section provides recommendations for how a developer might go about using the provided framework to couple a different air model to the surface models.

The main body of this report avoids program source code details, however in the appendices we discuss the implementations in the Air Model Toolkit and use terms that refer to fortran90 statements, modules, subroutines and variable names. The following typographical conventions are used in Appendix A.

text	General document text (Times New Roman 12pt)
filename.idf	the name of a computer file (Italic Times New Roman 12pt)
`CodeFont`	actual Fortran code or general input (Courier New 10pt)

Some of the figures in this section are software diagrams used to show the organization and interactions of program units. Two types of diagrams are used: (1) simple calling tree and looping diagrams, and (2) UML inspired USE diagrams. The later are based on Unified Modeling Language or UML (Fowler and Scott 2000). UML is a standardized graphical notation used in object-oriented programming. UML syntax is used here informally to describe modularity and `USE` relationships by considering a fortran90 module as analogous to a "Class". A module that accesses another module by `USE` statement will have a navigability arrow pointing from it to the module it is `USE`'ing.

A.1 Air Model Toolkit Overview

The goals of Research Project-1222 are being met by developing Fortran90 code for distribution as part of a new ASHRAE toolkit termed the *Air Model Toolkit*. This toolkit is intended to be the fourth in a series of toolkits the latest (and closely related) being the ASHRAE Loads Toolkit. The Fortran programs and modules have been organized into two different groups, sample programs and components. The sample programs demonstrate coupled surface and air models (as well as complete mixing air model). The separate air models are components of the toolkit. Sections 2 and 3 in the main body of

this report provide details of the theory. The developer should also consult the User Guide in Appendix B for more information on how the models are implemented.

The components are collections of Fortran90 files that demonstrate implementations of the room air models. The components have stand-alone programs that allow testing the modules in isolation from the rest of the Loads Toolkit, except for the Mundt model which is very simple. Table A.1 provides a summary of the room air models implemented

Table A.1 Summary of Air Models

Type of Model	Reference	Variables Solved	Geometric Range	Main Assumptions	Numerical Technique
Mixing		T (average)	none	Complete Mixing	Trivial Direct Formulae
Nodal Model	Mundt 1996	$T_{airfloor}$ $\frac{dT}{dz}$	Z (Height)	Ideal Displ. Vent. Floor/supply HB Linear T Gradient	Trivial Direct Formulae
Nodal Model	Rees and Haves 2001	T	Z (network)	Mass Flows One-way connect.	Newton. with line search and backscatter (Press, et. al 1996)
Pressure-Zonal Model	Inard 1996	P and T Simult	X, Y, Z Structured Cartesian Grid	Special models Cd = 0.83 N < ~(350)	Newton-Raphson various kinds Broydens Method (Numerical Recipes, IMSL)
Pressure-Zonal Model	Lin 1999	P and T Simult.	X, Y, Z Structured Cartesian Grid	Special models Cd = 0.83 N < ~(350)	Newton. w/ linesearch and backscatter (Press, et. al 1996)
Momentum-Zonal Model	New	P, T, U, V, W … Sequent.	X, Y, Z Structured Cartesian Grid	No Diffusion Laminar/Cd = ?	Upwind, SIMPLE, sequential iteration Patanker CV Formulation

The Air Model Toolkit includes computer files of various types. The directory (folder) organization is diagrammed in Figure A.1.

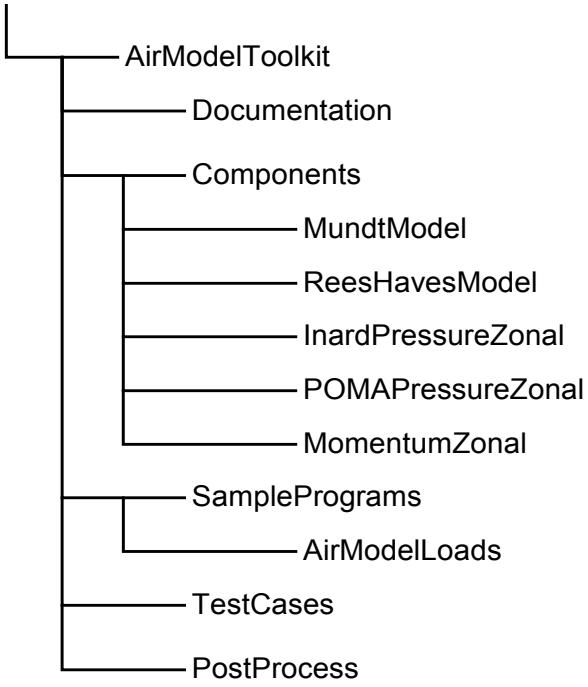


Figure A.1 Air model toolkit directory structure

The rest of this section provides information on the programming standard and general input and output issues.

A.1.1 Toolkit Programming Standard

ASHRAE Special Publications publishes a series of "toolkits" with computer media containing program source code and documentation. The ASHRAE Loads Toolkit documentation includes a programming standard that is also adopted for the Air Model Toolkit. Section 1.2.4 from the Loads Toolkit is quoted here for completeness.

> "1.2.4 Programming Conventions
>
> This section contains all of the rules and conventions that were applied to the program source code in the Toolkit. It is intended to be a coding guideline both for programmers as well as reviewers. It is also hoped that this will help the Toolkit user become familiar with the code quickly.
>
> Code Development Philosophy
>
> The following coding criteria were prioritized in descending order of importance:
>
> Level 1: Readability, Testability, Maintainability, Robustness, Reliability
>
> Level 2: Portability, Reusability

Level 3: Speed, Size

In general, clarity of the Toolkit code took precedence over computational efficiency.

Format and Comments

All code is placed in "free-format" as opposed to the fixed format used by Fortan 77 and other versions of FORTRAN. In addition, the following guidelines were followed for all free-formatted code in the Toolkit:

Try to confine each line to 80 characters. This allows most code to be seen on a standard size screen and be printed without resorting to micro-font or landscape mode.

Column 6 is not associated with a continuation line in free format. To continue one line onto the next line, and ampersand character "&" is placed at the end of a line to be continued.

To help visually distinguish between FORTRAN 90 keywords and other code elements such as comments and names, FORTRAN 90 keywords are in all uppercase while other elements are in mixed case.

Tab characters are not allowed in any source code file.

Only "!" is valid for indicating comments. Endline comments are allowed and encouraged. Guidelines from Code Complete (McConnell 1993) should be followed for endline comments. Several suggestions are repeated here:

Use endline comments to annotate data declarations.

Use endline comments for maintenance notes (bug fixes, for example).

Use endline comments to mark ends of blocks.

Avoid endline comments that merely repeat the code.

Avoid endline comments for multiple lines of code. In other words, avoid using a comment at the end of one line that applies to several lines of code.

No lines should extend past 132 characters.

Programming Safety

FORTRAN 90 supports many safety features such as argument checking across procedures, `IMPLICIT NONE`, and `INTENT` attributes that enable the compiler to find many programming errors. The following rules were followed for Toolkit code:

Use `IMPLICIT NONE` to require all variables to be explicitly declared.

Use `INTENT` to specify how a procedure uses a function argument. `INTENT` can be one of the following:

> `IN`: the argument is an input to the procedure. It cannot be redefined or become undefined while the procedure runs.

OUT: The argument is an output of the procedure. The procedure must define the argument before it uses the argument. Any actual argument associated with the argument must be definable. On invocation of the procedure, the argument is always undefined.

INOUT: The argument can both receive data from and return data to the invoking program unit.

Use ALLOCATED to check if the data is [sic] already allocated.

Use PUBLIC or PRIVATE to specify the access attribute of the data objects within a module.

Use SAVE attribute to preserve the value of a variable after the execution of a RETURN or END statement in a procedure.

Naming Conventions

In all of the naming conventions listed below, it is assumed that the limit on the length of names is 31 characters and that spaces are not allowed as valid characters in any name. Also, underscore characters "_" should be avoided.

Variable Naming Convention

In general, variable names are selected to form identifiable, mnemonic descriptions of the variables. Since the variable names tend to appear in the code much more often than subroutine or module names, logical abbreviations are thus encouraged. Typically, lengthy words should be shortened to between 3 and 5 characters to make a logical yet concise name for the various program variables. Common state or process variables may begin with a one-character letter to designate the variable. For example, temperatures begin with a "T." Although some FORTRAN 90 compilers may be case insensitive, both upper and lower case characters are used in the Toolkit to form indentifiable words within variable names, for example, "ToutsideAir." Variable names should be consistent among the Toolkit module routines.

Subroutine Naming Convention

Subroutine names should be constructed using the verb-predicate rule. Every subroutine performs some action (the "verb") on a particular item or data set (the "predicate"). The subroutine name is thus constructed using the verb-predicate combination to arrive at a unique name for a particular algorithm. The verb should be the first part of the name followed by the predicate. For example: CalculateViewFactor.

Module and Source Code File Naming Convention

Since modules typically are associated with objects or data groups, the name which is selected for a module should refer to the object or data grouping. Data-only modules should use "Data" in its name after a logical descriptive term or terms. All modules have a suffix "Mod." For example: ExteriorConvectionMod and InputDataMod.

Since the module is the basic program unit (except the Toolkit driver main program), source code files should use the name of the module as the base name with ".f90" as the file extension. **"**

It was clearly specified that this research project (RP-1222) should follow the same programming standard and use routines from the Loads Toolkit as much as possible in developing demonstration program code. Therefore our research has involved reviewing and using much of the code constructed for the Loads Toolkit. This activity and our own

program code development efforts led to formulating (and abiding by) additional programming guidelines for this and future such "toolkits."

RP-1222 Programming Standard

In addition to the programming conventions from the Loads Toolkit quoted above, the following guidelines were followed for the Air Model Toolkit.

- Focus on *domain*-oriented code where the domain is the body of engineering knowledge currently being used in the building science. Avoid highly abstract and heavily nested data structures. Use terminology from the ASHRAE community when developing names.
- Minimize loops by using conforming arrays and array indexing. This can make code more compact and understandable and should increase the future value of code that may be compiled for parallel execution on multi-processor computers. (Press et. al 1996)
- Modularize large components using a consistent methodology. This can provide natural orientation when moving between different air models. For example, each air model has a separate input manager module which collects routines that handle transferring data into the model's working variables (e.g. *ReesHavesInputMgr.f90* and *ZonMoInputMgr.f90*).
- Use consistent methodologies when coupling air models to surface models. This is expected to augment code reusability and to lead to a more versatile coupling mechanism for future adaptation of new air models.
- Organize array indexing and control such that the actual order of input object definitions in the input file is irrelevant.
- Revise FORTRAN77 code so that it compiles as free-format ".f90, " includes `IMPLICIT NONE`, `PRIVATE`, and `INTENT(in/out)` statements, and eliminates common blocks by converting them to data-only modules. These are only the minimum criteria for determining when older code is "converted." Additional re-engineering should be performed including: (1) modularizing code by grouping similar routines into fortran90 modules, (2) eliminate GOTOs and alternate subroutine entry points, and (3) reedit comment and line continuation operators.

A.1.2 ToolKit Input

The Air Model Toolkit uses the same method of inputting data as the Loads Toolkit. This section provides a brief summary of the input process; see the User Guide in Appendix B for descriptions of input required for each of the air models. The interested reader/programmer can also obtain introductory and more detailed information on the toolkit input process in the Loads Toolkit documentation in the file *ToolkitInformation.pdf* (sect 1.3, page 10) and EnergyPlus documentation in *InputOutputReference.pdf* and/or *ModuleDeveloper.pdf*. The input routines are in the module *InputProcessor.f90*. The code originates from the EnergyPlus program and uses

the same 2-text-file technique but with a different input data dictionary. The input processor is a robust set of utilities for reading the files and making the information available to other parts of the program. Input data for running a simulation are contained in a text file that needs to be called *in.idf*. The input processor learns what it needs to know about how to process the input file at run time by also reading a second text file that acts as a data dictionary. For the Toolkit, this file must be called *ToolkitTest.idd*. The object definitions contained in *.idd* files are commented in the file so as to explain the purpose of the character and numeric input fields. The `InputProcessor` does very little the way of altering the floating point and character data and in this one sense is "dumb." Note that there are usually a large number of object definitions in any given .idd file that have no supporting code. The user should understand that not every object present in the .idd file is really implemented in any one compiled program.

Including a `USE InputProcessor` statement in a subroutine allows making subroutine calls to access data from the user input file. The most commonly used subroutine calls are `GetObjectItem` and `GetNumObjectsFound`. The first call returns the numeric and character data for a particular input object and the second returns the number of objects of a particular type that are in the input file. One of the features of the input method is that it does not, or at least *should* not, matter in which order the user places the object definitions in the input file. Therefore a policy in constructing code is to make sure that code does not inadvertently require order to the input objects. A call to `GetObjectItem` will return an individual object by an index number that is determined by placement in the input file. It is also recommended that user input be accessed in one place rather than making repeated calls to the `InputProcessor` for the same data.

A.1.3 Toolkit Output

The programs can produce a potentially large amount of data and separate data files. The widely varying spatial and temporal domains and switching between different air models makes collecting results into a single file a daunting project. Fairly comprehensive output is made available but it is organized for ease in reading into a post-processing program rather than a spreadsheet. Table A.2 shows the important files created by running the programs along with information on where in the code the files are created and/or written. Both dynamic and fixed file units are being used. All data output are formatted ASCII text. The most important files are *Toolkit.out*, *NodeData_.out*, and *ZonalData_.out*.

Table A.2 Output data files from the toolkit programs

filename	File unit	modules	subroutines
Toolkit.out	ReportFile	*HeatBalanceSimMgr.f90*	CalcHeatBalance ReportHBResults ReportSimulationResults
Nodedata_.out	NodeOutFileUnit	*AirDataManager.f90*	WriteNodeData
ZonalData_.out	ZonalOutFileUnit	*AirDataManager.f90*	WriteZonalData
Debug.out	DebugOutputFileUnit	*(many)*	(many)_
Audit.out	EchoInputFile	*InputProcessor.f90*	ProcessInput
Toolkit.err	StandardErrorOutput	*InputProcessor.f90*	ShowErrorMessage
ReesHavesModel_.out	RHModelFileUnit	*ReesHavesOutputMgr.f90*	WriteReesHavesOutput
output_room.txt	20	*Room.f90 (POMA)*	output
Result.Dat	13	*ZonMoOutWriter.f90 ZonalMomentumSimMgr.f90*	PRCASE, PRVARI, ManageZonMoModel
Balance.Dat	11	*ZonMoOutWriter.f90 ZonalMomentumSimMgr.f90*	PRSOUR ManageZonMoModel
TECP___x.out	14	*ZonMoOutWriter.f90*	PLOTVT

The *Toolkit.out* file is the basic output file from the loads domain and applies to all coupled air model simulations as well as the mixing model. Table A.3 shows the data contained in the lines of the *Toolkit.out* text file. A results reporting subroutine called `ReportHBResult` was developed to allow fixed-column reporting at every iteration, timestep, and day of the simulation. This report file is generated if requested in the input file by including the input object "`ToolKit Output`" with the first parameter set to "2" (e.g `toolkit output, 2;`). Otherwise the original toolkit.out file from the Loads Toolkit is generated. The number of records, or individual lines varies with the size of the problem.

Table A.3 Data Organization for Output File Toolkit.out

Record, line of text file	Space separated data	Datum per record
1	Number of surfaces, number of timesteps/day, number of days, number of iterations	4
Next # of surfaces	Number and name of surface	2
Next (# of timesteps * # of days * # of iterations)	day, timestep, iteration, Q_{sys}, $\dot{V}$, $T_{sysDiff}$, T_{StatDB} , TairOut, T_{supply}, # secondary air system iteration loops, `AirModelShift, AirModelQsys`	12
Next (# of surfaces) X (# of timesteps)	Temperature at inside face, effective bulk air for surface, h_c , temperature at outside face	6

The output file *NodeData_.out* is generated by the module `AirDataManager` when running a nodal model and the subroutine `WriteNodeData` is called. The `AirDataManager` module is discussed below (Section A.2.2.) This file contains the locations and names of air nodes and their temperatures at each time step. Table A.4 shows the data contained in each type of record in the *NodeData_.out* text file.

Table A.4 Data Organization for Output File NodeData_.out

Record, line of text file	Space deliminated data	Datum per record
1	# of Air Nodes, # of Time steps, # of Variables (in this output file)	3
Next # of air nodes	node number , node name	2
Next # of air nodes	X, Y, Z (locations in zone, only Z coordinate used)	3
Next (# of air nodes) X (# of timesteps)	Node number, Timestep, Node drybulb temperature (ºC), magnitude of massflow, Humidity Ratio	5

The output file *ZonalData_.out* is generated by the module `AirDataManager` when running a zonal model and calling the subroutine `WriteZonalData`. Table A.5 shows the data in its file records.

Table A.5 Data Organization for Output File ZonalData_.out

Record, line of text file	Space deliminated data	notes
1	# of Cells, # of Surfaces, # of Timesteps, # cells in X-direction, # of cells in Y-Direction, # of cells in Z-direction	
next # of cells	I, J, K, Name of cell (usually "default")	
next # of cells	I, J, K, X-coordinate, Y-coordinate, Z-coordinate, Size in X-direction, Size in Y-Direction, size in Z-direction	Geometry
Next (# of Cells) X (# of timesteps)	I, J, K, Timestep, Pressure, drybulb temperature	Cell center state variables
Next (# of Cells) X (# of timesteps)	I, J, K, Timestep, $\dot{m}$ direction-1, $\dot{m}$ direction-2, $\dot{m}$ direction-3, $\dot{m}$ direction-4, $\dot{m}$ direction-5, $\dot{m}$ direction 6	Cell face Mass flows

These files contain a lot of data. More than is easily digested with a spreadsheet. Fairly extensive data processing and visualization routines were written in order to check that data being produced was *physical*. A data visualization application with a GUI was written in an interpreted language called IDL (RSI 2000) that reads these output files for the purpose of processing and plotting results. A commercial, third party software package is needed to run the crude application. Running the application called "AirModelBrowser" provided fast and powerful method of visualizing results and was found very helpful for checking the code during development. This application was used to generate results plotted in Sections 4 and 5 of this report. Many other types of plots and 3D visualizations are produced that do not reproduce well as black and white figures. It is not clear if ASHRAE will want to include this IDL code in the toolkit but it is currently included. More information is available in the readme file in the directory *\AirModelToolkit\PostProcess\readme.txt.*]]

A.2 Coupled Loads and Air Models

This section introduces the coupling of the load calculation routines with room air models. The room air models themselves will be discussed in subsequent sections. The Air Model Toolkit includes a program called *AirModelLoads.exe* that demonstrates coupling various air models to the surface models. The program is based on the successive substitution solution technique from the Loads Toolkit. Different air models are included in one program to simplify running test cases. Table A.6 lists all of the fortran90 files that are used in the AirModelLoads program. More detail of the individual air models are provided in later sections of this Appendix.

Table A.6 Fortran files for coupled demonstration program AirModelLoads.exe

Category	Source files
Loads Toolkit Modules	Toolkit.f90, HeatBalanceSimMgr.f90, InputProcessor.f90, ExteriorConvectionMod.f90, ExteriorLWRadiationMod.f90, PsychrometricsMod.f90, GroundTemperatureMod.f90, UtilityMod.f90, CTFMod.f90, EnvrnSurfTemperatureMod.f90, InfiltrationVentilationMod.f90, InteriorConvectionMod.f90, InternalGainsMod.f90, RadiativeGainsDistributionMod.f90, ShadingMod.f90, SkyRadiationMod.f90, SkyTemperatureModelsMod.f90, SolarPosition.f90, ViewFactorMod.f90, VeiwsToGroundAndSky.f90, Windows.f90, ZoneLWRadiationMod.f90
General Air Model Toolkit Modules	AirDataManager.f90, ConvectionCorrelations.f90
Mundt Model Modules	MundtSimMgr.f90, MundtInputMgr.f90, DataMundtModel.f90
Rees and Haves Model Modules	DataReesHavesModel.f90, FCN.f90, ReesHavesInputMgr.f90, ReesHavesOutputMgr.f90, ReesHavesSimMgr.f90, (also using IMSL Numerical Libraries)
POMA Pressure-Zonal Model Modules	POMASimMgr.f90, POMAInputMgr.f90, Room.f90, Zone.f90, VerBoundary, HorBoundary.f90, WallSurface.f90, Jet.f90 nr.f90, nrtype.f90, nrutil.f90, Newt.f90, fdjac.f90, fminln.f90, funcv.f90, lnsrch.f90, lubksb.f90, lucmp.f90
Momentum-Zonal Modules	ZonalMomentumSimMgr.f90, ZonMoCoefficientCalcs.f90, ZonalMomentumData.f90, ZonMoInputMgr.f90, ZonMoReportMgr.f90, TriDiagCalcs.f90

A.2.1 *HeatBalanceSimMgr.f90*

The main level program unit is *Toolkit.f90*. This small driver program calls the top-level module, *HeatBalanceSimMgr.f90*. This module implements the surface heat transfer models along with environmental, solar, other code needed to perform building cooling load calculations for a single zone. It is based on the Loads Toolkit module *SuccessiveSubstitutionSolution.f90*. Table A.7 summarizes the subroutines that were added or significantly altered for the Air Model Toolkit. This module controls the overall program flow for coupled models. The conventional mixing model remains functional and is more versatile than before. Figure A.2 diagrams the loops and calling sequences for a design-day load calculation.

Table A.7 Subroutines in HeatBalanceSimMgr.f90 Affected by Coupling Surface and Air Models

	Routine name	notes
Modified routines from Loads Toolkit	`CalcHeatBalance`	added intialization, and control constructs for using Air Models
	`CalcOutsideHeatBalance`	added new outside boundary conditions
	`CalcInsideHeatBalance`	changed handeling of air temperature and film coeff
	`CalcAirHeatBalance`	Split into `CalcQsys` and `CalcLumpAirHeatBalance`
New routines for Air Model Toolkit	`CallAirModel`	handles data passing and calling air models
	`CalcQsys`	formerly `CalcAirBalance`
	`CalcLumpAirHeatBalance`	formerly `CalcAirBalance`
	`PredictVdot`	Equation 3.10
	`CalcVdotPGain`	secondary air system loop
	`GetOutsideSurfaceBCInput`	"NOD" set up
	`ReportHBResults`	new *toolkit.out*
	`GetIntraZoneModelParameters`	coupling control parameters
	`ResolveSurfIDs`	coupling surface organization

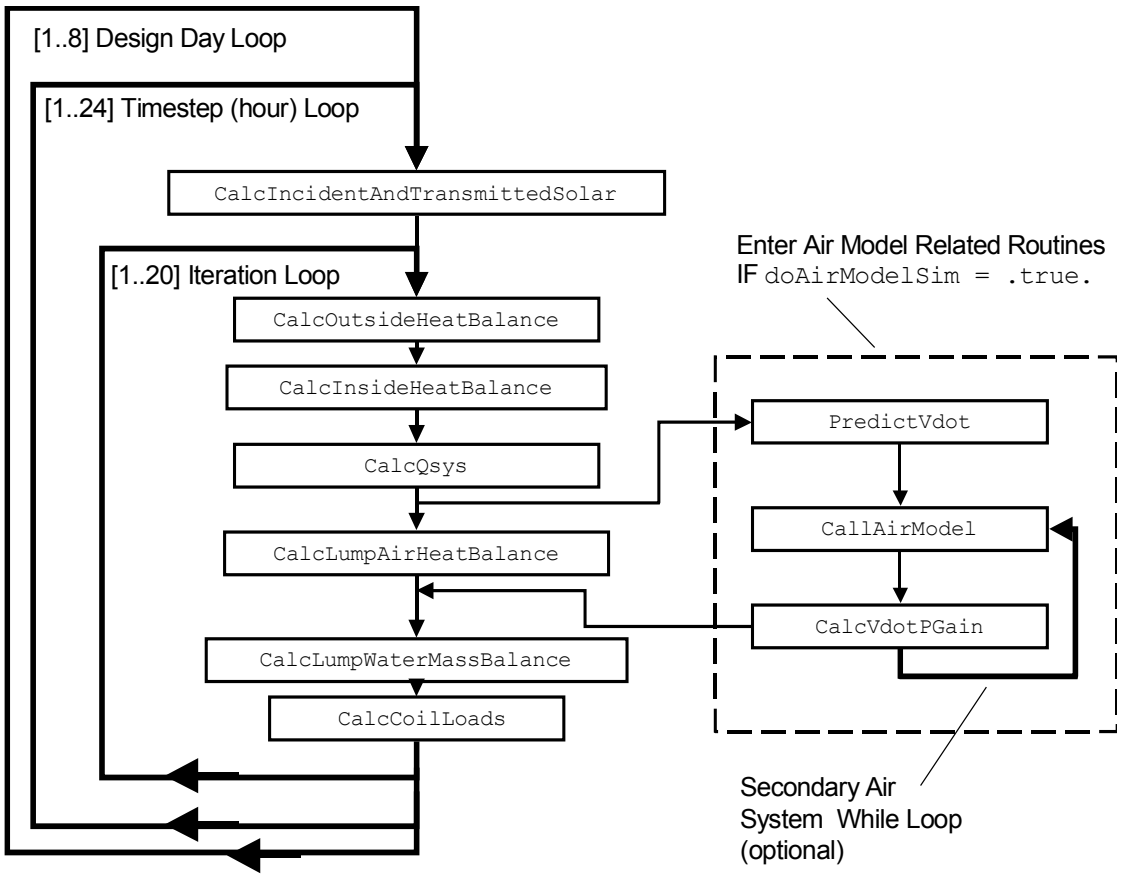


Figure A.2 Subroutine Calling in `CalcHeatBalance` *-- HeatBalanceSimMgr.f90*

The `CallAirModel` routine in *HeatBalanceSimMgr.f90* handles calling, passing values to, and collecting results from the desired room air model. Figures 3.3 and 3.4 diagram the steps involved in calling an air model. The exact calls differ between nodal and zonal models.

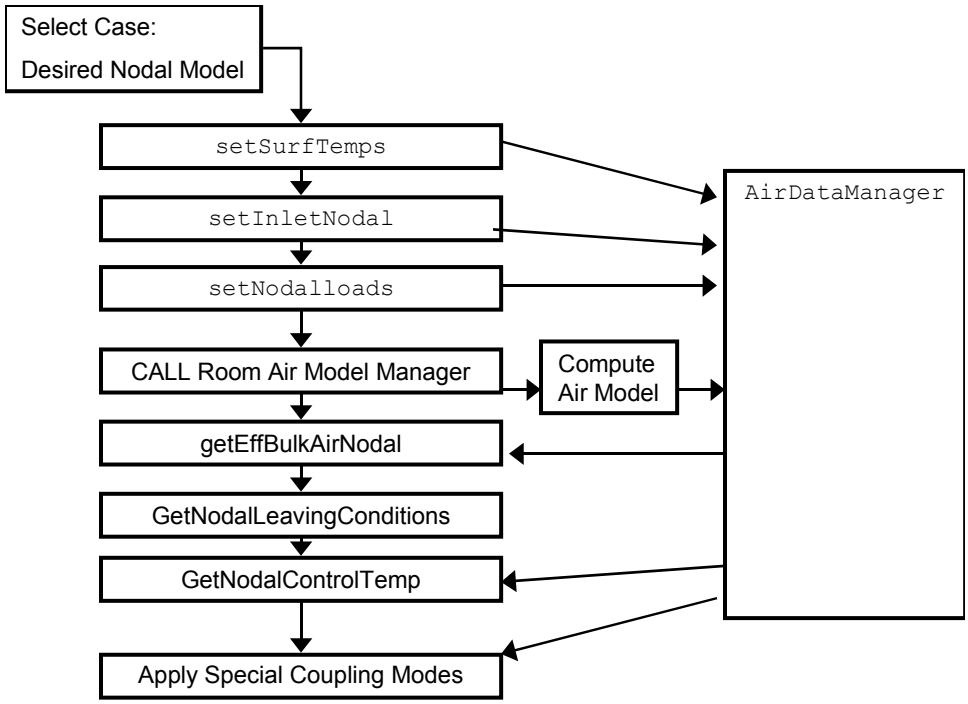


Figure A.3 Subroutine Calling in `CallAirModel` for a Nodal Air Model -- *HeatBalanceSimMgr.f90*

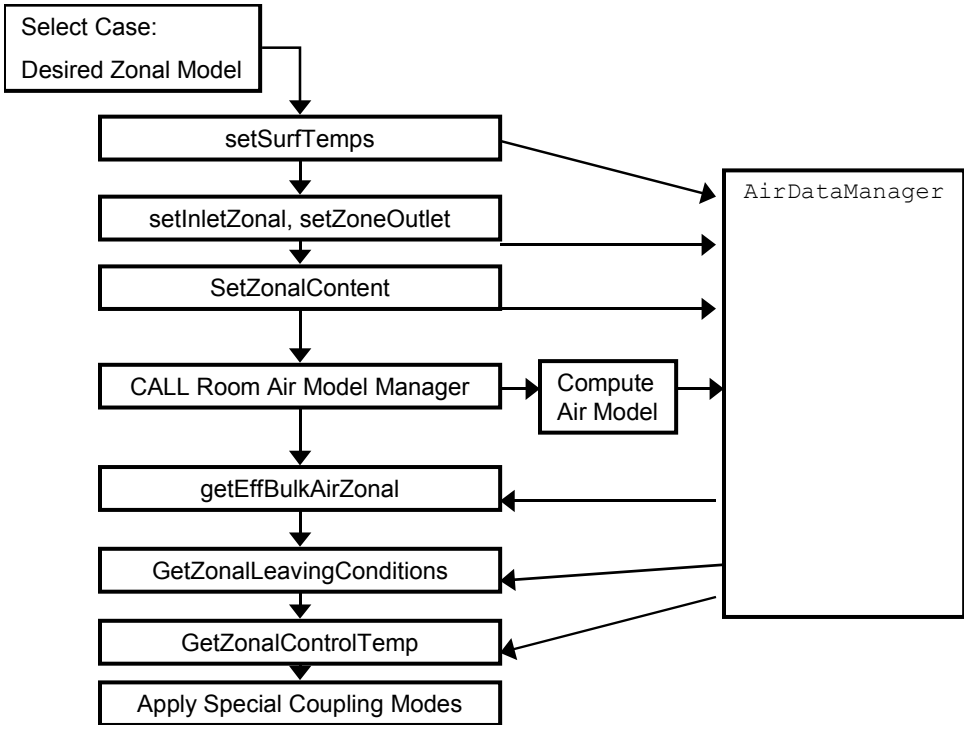


Figure A.4 Subroutine Calling in `CallAirModel` for a Zonal Air Model -- *HeatBalanceSimMgr.f90*

Program flow controls have been implemented to allow altering run behavior during the course of a simulation. The input object "select air model" is set up to control certain aspects of the coupling scheme (see Input Object Reference in Appendix B). The variables used in the code for control structures are listed in Table A.8. The control can be varied from one day to the next. This allows testing various controlling schemes with out as much recompiling. The flexibility is also designed so one model can be used for one day and different model the next day so as to provide better initial conditions.

Table A.8 Program control variables in HeatBalanceSimMgr.f90

Variable	Integer		Comments
`AirModelType`	1	Complete Mixing	
	2	Mundt Nodal	
	3	Rees and Haves Nodal	
	4	POMA Pressure-Zonal	
	5	Inard Pressure-Zonal	
	6	Momentum-Zonal	
`TempCoupleScheme`	1	T-Couple: direct air model	Instable some Air Models
	2	DT-couple: relative air model	assume at setpoint
`HcType`	1	Conventional	Also available for Mixing
	2	Air Model Hc's	get from `AirDataManager`
`VdotControlScheme`	1	Conventional	Use DT model comfort
	2	Secondary system control loop	Air Model must predict Tstat well
	3	in-zone heater	only for momentum model
`TstatTol`	[ºC]	Tolerance on control	e.g. 0.01ºC to 0.2ºC
`ToperativeScheme`	1	Dry Bulb at Thermostat	
	2	Comfort Temperature	½(MRT + Tstat)
`AirUpdateScheme`	1	no timestep shift	
	2	Copy result from prior timestep to start	call `CopytoNewTimestep`

The secondary air system loop is shown in Figure A.2 is discussed in Section 3.5.1. This loop simulates how a VAV controller might operate and runs inside the main iteration loop using constant values for parameters from the surface models including Qsys and inside face surface temperatures. Before entering the loop an initial prediction of system air flow rate is made in the routine `PredictVdot`. A `WHILE` loop calls the routine

`CallAirModel` followed by `CalcVdotPGain` until a thermostat tolerance level is met. The basic idea is to increase system flow if the control temperature is too high and decrease it if the control temperature is too low.

A.2.2 *AirDataManager.f90*

AirDataManager.f90 is an important module in the Air Model Toolkit. It encapsulates the data structures and input/output methods for connecting an air model to the loads routines. The `AirDataManager` module collects reusable code when coupling a number of different air models to the same collection of loads routines. It's metaphor is the connective tissue that allows hanging a net of nodes or a cage of cells onto an arbitrary number of building surfaces. Figure A.3 diagrams the USE relationship using informal UML syntax (Fowler and Scott 2000). The `AirDataManager` works closely with different air models and provides an interface layer between an air model and the surface models from the Loads Toolkit. Typical program execution is to set new values for an iteration or time step in the `AirDataManager`, then call the air model to execute. Once the air model has run (using the boundary conditions it received from `AirDataManager`) it passes its results back to the `AirDataManager`. The surface models then are updated with relevant results by querying the `AirDataManager`. This is done to provide a consistent interface to the surface models so that it is easier to implement different air models. The module is intended to ease a users introduction of his or her own model for air into such simulations.

This collection of routines and data emulates object-oriented programming in that it provides methods of manipulating data in the form of get and set routines. However, `AirDataManager` doesn't really hide its data as it declares `PUBLIC` access for many of its variables. A policy that has been used is that data transfer between the air and surface domains be accomplished using subroutine calls. Therefore it is best to consider using `ONLY` statements for the `AirDataManager` subroutines needed by routines in the surface domain. Within the air domain, the air models generally make use of public access to the data in `AirDataManager` but typically use their own working variables (with copies of the data). This may slow computations but is done to allow separating variables for numerical work. The air model may need to work with double precision variables, different arrays sizes, and need not store data for every time step.

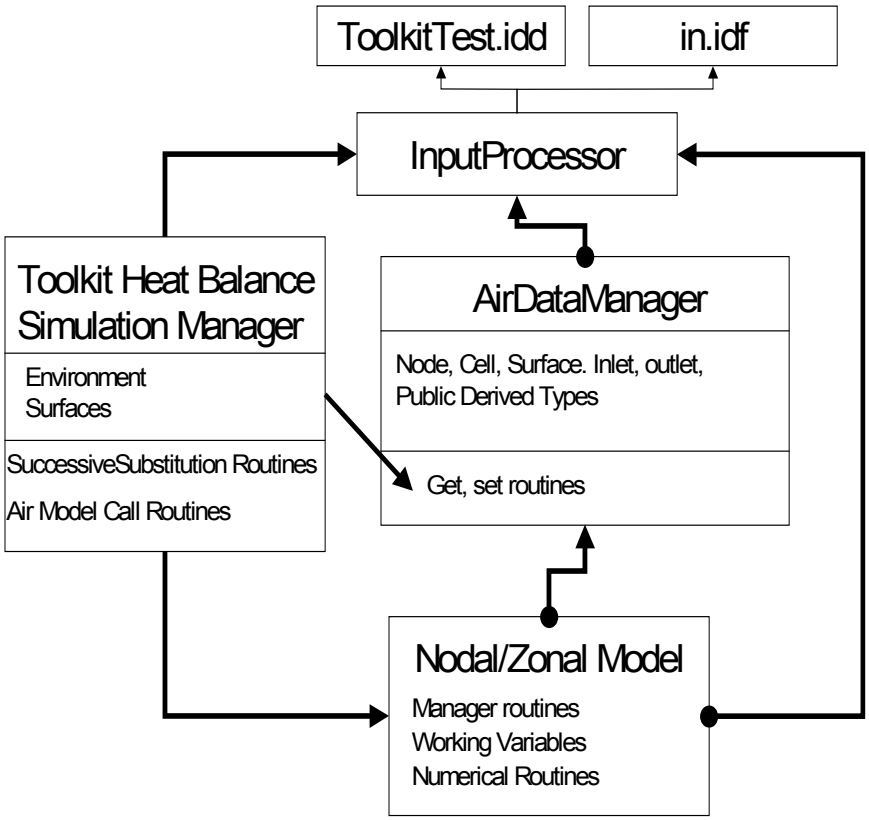


Figure A.5 AirDataManager USE diagram

Table A.9. Subroutines in AirDataManger Module

Category	Nodal Models	Zonal Models
Input, Initialization, and Output	ConstructNodes	ReadIDFZoneGrid
		InitAirDataCells
	WriteNodeData	WriteZonalData
Modeling utilities		GuessAirTs
		SimpleGuessAirTs
		GuessHydrostaticPs
		ApplyHydrostaticDistribution
	CopytoNewTimestep	CopytoNewTimestep
		ModelFaceMdots
Query routines	setSurfTemps	setSurfTemps
	GetAirNodeNum	CheckInletZonalSolution
	GetNodalControlTemp	GetZonalControlTemp
	GetNodalLeavingConditions	GetZonalLeavingConditions
	GetNodeConditions	
	SetNodeConditions	setZoneOutlet
	GetSurfIDs	GetSurfIDs
	setNodalloads	SetZonalContent
	setInletNodal	setInletZonal
	getSurfHc	getSurfHc
	getEffBulkAirNodal	getEffBulkAirZonal
	setBarametricPressure	setBarametricPressure
		setZoneOutlet
		GetTotalWallConv

The data structures, or fortran90 derived types, in AirDataManager are setup for both nodal and zonal models. Different structures are used since these models treat the zone air in different ways. As shown in Table A.8, there are often similar routines that work with either nodal or zonal models. An encyclopedic reference of every subroutine seems unwarranted; check the source code to see how each routine works. The data stored in the module are of essentially everything of interest to simplified air modeling. Section A.9 discusses more details of the AirDataManager and coupling in the context of how to couple a new model to the loads calculations. The following section discusses some of the modifications to the loads routines before we move on to discuss the air models in later sections.

A.3 Modifications to Loads Toolkit Routines

This Air Model Toolkit is an extension of the Loads Toolkit (Pederson 2001) in that it uses many routines from that toolkit in the programs. This section discusses significant modifications that were made to routines from the Loads Toolkit programs. While the focus of our research is on implementing air models with more detail than the mixing model, it is important that comparisons be made to the mixing model and so the complete mixing model is retained. This section details changes made that affect mixing air model simulations so that comparisons can be made to the more detailed air models. Some

variables were moved up from the subroutine level to the module level in order to make them more widely available.

A.3.1 Air System Off Floating Bulk Zone Temperature

It is useful to be able simulate a building that is in the situation of having free floating room temperatures. Therefore the current work made some changes or additions to the sample program from the Loads Toolkit to allow such simulations while still using the single-node, fully-mixed air model.

Fixing the room temperature as a model parameter is reasonable if a mechanical system is present and running and capable of meeting the load. Thus the room operating temperature was fixed as an input variable unless a capacity limit was placed on the system. Considering that single-node results (and speed timings) will need to be compared to the more complex air node models, it was desirable to alter routines so that more flexible scenarios for floating room air temperature can be simulated using baseline techniques. The focus of the new calculations documented in this section is to allow switching off the system completely "off" (nor allow the mathematical equivalent of backwards system air flow). The quasi-steady model implemented here could be improved on by using the transient formulation used in EnergyPlus (Energyplus Developers Guide).

The starting point in developing a system-off free floating bulk temperature for the toolkit is the "Qsys" equation .

$$Q_{sys} = Q_{surfConvTot} - Q_{ConvGains} - Q_{sensInfil} \tag{A.1}$$

If the system is scheduled "off" or the resulting system air flow calculation is negative then Qsys =0. This indicates that the heat transfer by surface convection at the walls must equal the heat transfer by convection from the internal equipment, people, and lights plus the heat transfer by infiltration/ventilation. So we write,

$$Q_{surfConvTot} = Q_{ConvGains} + Q_{sensInfil} \tag{A.2}$$

Where,

$$Q_{surfConvTot} = \sum_{i=1}^{nsurfaces} h_{c,i} A_i (T_{bulkair} - T_{s,i}) = \sum_{i=1}^{nsurfaces} h_{c,i} AT_{bulkair} - \sum_{i=1}^{nsurfaces} h_{c,i} AT_{s,i} \tag{A.3}$$

Solving for the bulk room temperature we obtain,

$$T_{bulkAir} = \frac{(Q_{convgains} + Q_{sensInfil}) + \sum_{i=1}^{nsurfaces} h_{c,i} AT_{s,i}}{\sum_{i=1}^{nsurfaces} h_{c,i} A} \tag{A.4}$$

This equation has been added to the `CalcLumpAirHeatBalance` subroutine that was formerly the `CalcAirHeatBalance` routine from RP-987. This modification is made to allow using test cases that include scheduled system off periods of time and be able to compare simulation runs of the conventional single-node sort to nodal/zonal simulations. In order to activate this mode, include the input object '`Const Vol System, 0.0;`'.

A.2.2 Inside Surface Boundary Conditions

The heat balance technique formulates a fairly complete description of heat flows at the interior surface in order to determine the surface temperature and surface heat flow rates. The routine `CalcInsideHeatBalance` was modified allow using convection coefficient that may be determined from user input to the air model (or from convection correlations processed the air model). This routine also stores a value for mean radiant temperature for the overall space for possible use with control based on a comfort condition that includes radiation. The values for $T_{a,i}$ are stored in `TeffBulkAir`. The values for h_c are stored in `HcIn`.

A.2.3 Outside Surface Boundary Conditions

Additional resolution of the air inside, can also lead to changes on the *outside* surface in the Heat Balance Method. Several new boundary conditions have been developed.

"NOD"Air Node Boundary Condition

Often buildings have walls, floors, and internal furnishings that are modeled as being exposed to other zones that are just like the current zone. Conditioned zone air is on both sides. These internal partitions or thermal mass might have used boundary condition "TA" in the original loads toolkit. With an incomplete mixing model there is no longer a single zone air temperature to apply to the outside surface of the construction. Thus the models have been modified to include identifying "other side" boundary conditions. For example in a commercial building a floor slab could have the ceiling air as the boundary condition for the "outside surface" Internal mass walls may be exposed to the same sort of condition on each side. A self-similar symmetric condition is also enforced by applying the surface to itself on the backside while still model the heat balance for storage. So a self-referring NOD-style outside boundary condition is computed differently than a "TA" boundary condition since locally air temperatures can now change throughout the day even under controlled conditions. The adjacent air temperatures are organized by the surfaces that they are associated with so "NOD" conditions are described by surface associations rather than air node or cell associations.

"TRP" Boundary Condition

A new outside surface boundary condition was developed called "TRP" which stands for Temperature Return Plenum. Displacement ventilation models would be expected to be

used for commercial buildings and will alter predicted temperatures and heat flows for ceilings and floors. But it was found to be difficult to do a good job of simulating ceiling and floor surfaces well with preexisting boundary conditions such as "TA," "TB," and "TG". The idea behind using "TRP" boundary condition is that of simulating one zone in a multistory commercial building where the ceiling and floor surfaces are separated by a continuous return plenum as for a drop ceiling. This ceiling is not (yet) treated as a separate zone) The current implementation of this temperature boundary condition is that it is first initialized to the values input with a TempSpecial object and then changed during computations to equal the return air temperature calculated in the subroutine CalcCoilLoads. Heat gain from lights is included (per LightingSplits) but heat flows from the floor and ceiling surfaces are not.

"TB2" Boundary Conditions

An additional boundary condition and temperature input were made available that area analogous to the "TB" boundary conditions. This allows fixing an additional outside face air temperature without solar. The temperature data source is the input object "Tspecialextra"

A.4 Mundt Nodal Model

The Air Model Toolkit contains a selection of models for determining the distribution of air temperatures. The simplest is the referred to as the Mundt model – see section 2.2 The implementation is centered around one main subroutine, `CalcToolkitMundtModel`. Table A.10 lists the fortran90 source code files for implementing the Mundt model. The main routine has these arguments passed into it: `TimeStep, TAirAvg, SupplyAirTemp, SupplyAirVolumeRate, QsysCoolTot, Wheadheight, Pressure, QventCool, ConvIntGain` . No arguments pass data out, rather the results are stored in the `AirDataManager`. The model solves for the air temperature near the floor and at the returns in order to generate a linear gradient. The slope of the gradient is then used to determine temperatures that depend on the height. The user can select the Z-direction coordinates and number of room air nodes at will. Constraints on the slope have been implemented (in hopes of avoiding unstable situations) so that the slope is set to lie between 0.001 and 5.0. Values are calculated from $T_{leaving}$, using the slope and the Z-direction coordinates of the node.

Figure A.6 shows the relationship of the modules. This simple model would not necessarily need to be modularized in the manner that it is, however this is done so that a common module organization exists for different air models. Diagrams such as Figure A.6 get more complex for the other air models.

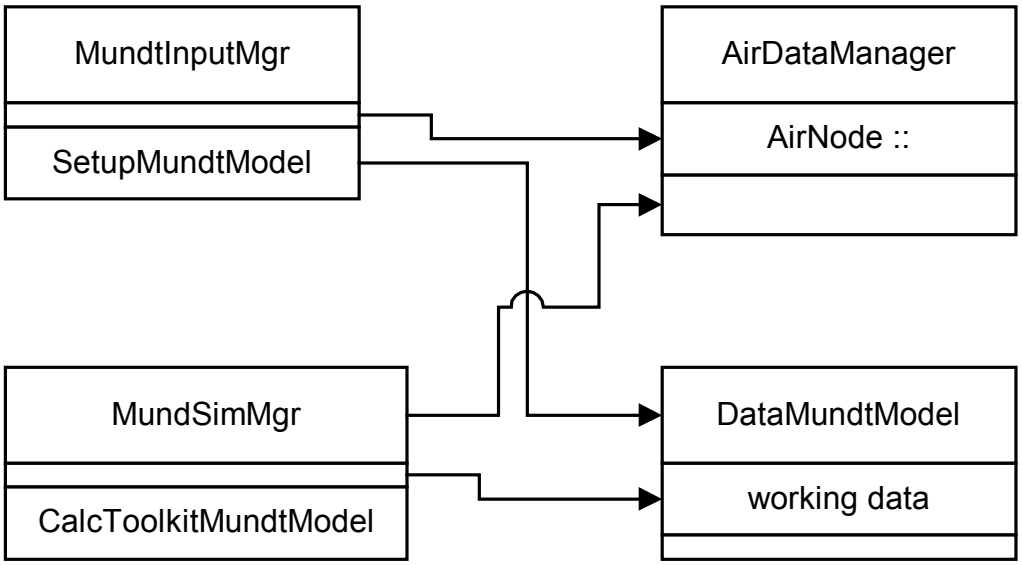


Figure A.6 Mundt model `USE` diagram

Table. A.10 Fortran files for Mundt model

Category	Source file	notes
Mundt Model	MundtSimMgr.f90	entry point
	MundtInputMgr.f90	transfer in data, initialize
	DataMundtModel.f90	Working data for model
Air Model Toolkit	AirDataManager.f90	air model toolkit utility
	InputProcessor.f90	input utility
	PsychrometricsMod.f90	air density utilities
	UtilityMod.f90	Kelvin to C reporting

The stand-alone component version of the Mundt model uses a main program driver called *MundtDriver.f90*. This program tests the routine and is hard-coded to run the case DisplacVent_B3. The main working variables for the model are collected in *DataMundtModel.f90*. Setup, initialization and input are handled by the module in *MundtInputMgr.f90*.

A.5 Rees and Haves Nodal Model

The Rees and Haves nodal model is implemented as an example of a nodal network model. *ReesHavesSimMgr.f90* is the entry point for the collection of routines that comprise the implementation of a nodal model presented by Rees and Haves (2001). This model is provided in two different configuration that use different non-linear solvers. One version of the Rees and Haves model component uses the IMSL library routine NEQNF which is based on the MINPACK routine HYBRD1. The second version uses the routines published by Press et al. (1996) *Numerical Recipes* implementing a Newton-Raphson solver with linesearch and backscatter (see section 2.3). Table A.11 lists the fortran90 files needed for the Rees and Haves Model when using the IMSL library. Table A.12 lists the files needed when using the *Numerical Recipes* solver. The coupled model used in *AirModelLoads.exe* uses the IMSL solver for the simple reason that the POMA model uses (almost) the same *Numerical Recipes* solver and file name conflicts prohibited easily compiling both air models. The source code for the IMSL routines is not provided but is provided in the toolkit for the *Numerical Recipes* routines. Compiled versions in the toolkit used the IMSL libraries that are distributed with Compact Visual Fortran (Professional). If you do not have access to the IMSL libraries, and want to use the *Numerical Recipes* version in the AirModelLoads.exe program then POMA will have to be removed and the new modules replaced. Testing showed both the solvers produce the same results in similar amounts of computing.

Table. A.11 Fortran files for Rees and Haves Nodal Model with IMSL library

Category	Source file	notes
Rees and Haves Model	ReesHavesSimMgr.f90	entry point
	ReesHavesInputMgr.f90	Transfer data into model, initialize
	ReesHavesOutputMgr.f90	transfer data out
	DataReesHavesModel.f90	Working data for model
	FCN.f90	Balance Equation for non-linear solver
Air Model Toolkit	AirDataManager.f90	
	InputProcessor.f90	
	PsychrometricsMod.f90	
	UtilityMod.f90	

Table. A.12 Fortran files for Rees and Haves Nodal Model with Numercal Recipes Solver

Category	Source file	notes
Rees and Haves Model	ReesHavesSimMgr.f90	entry point
	ReesHavesInputMgr.f90	Transfer data into model, initialize
	ReesHavesOutputMgr.f90	
	DataReesHavesModel.f90	Working data for model
	ConvectionCorrelations	
	FUNCV.f90	Balance Equations
Solver	NEWT.f90	Newton-Raphson
	FDJAC.f90	finite diff. jacobian
	FMIN.f90	
	LUBKSB.f90	
	LUDCMP.f90	
	LNSRCH.f90	
	NR.f90, NRTYPE.f90	
	NRUTIL.f90	
Air Model Toolkit	AirDataManager.f90	
	InputProcessor.f90	from E+, Loads Toolkit
	PsychrometricsMod.f90	from Loads Toolkit
	UtilityMod.f90	TempKtoC

The main task of the model is to formulate a series of energy balance equations for room air nodes. These balance equation are in the routines *FUNCV.f90* or *FCN.f90* (depending on the solver). The data structures for the nodes and connecting flow paths are ordered as diagrammed in Figure A.7 which corresponds to "Model B" in Rees and Haves (2001). There is a certain set of nodes (and their integer type codes) and flow path objects that must be specified in order to work with the Rees and Haves model, see Section B.4 for a discussion of input requirements. The flow network model is also considered suitable for use with chilled ceiling applications, so were code developed to model chilled ceilings in the loads calculation routines, this model of the air domain should respond appropriately. But since chilled ceiling is not currently available on the surface side, the model implemented here is currently for displacement ventilation with out chilled ceilings.

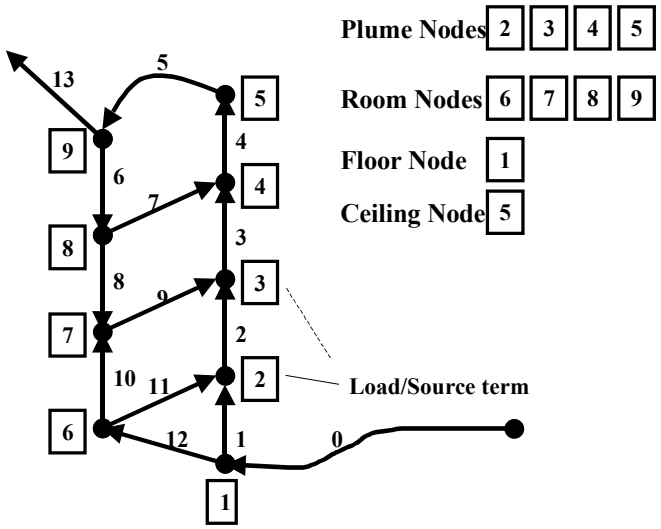


Figure A.7 Rees and Haves Model Node and Flow Path Ordering Scheme

The relationships of the modules are diagrammed in Figure A.8. The main working variables for the model are collected in *DataReesHavesModel.f90*. These variables are allocated and otherwise filled by routines in *ReesHavesInputMgr.f90.* The data are obtained from the `AirDataManager` or the user input file. There are two input routines. `ProcessNodalInput` is called the first time the Rees and Haves model is called and `UpdateNodalInput` is called for subsequent calls to the model. Results are sent to the AirDataManager using the routine `ManageReesHavesOutput` and written to a file using `WriteReesHavesOutput`.

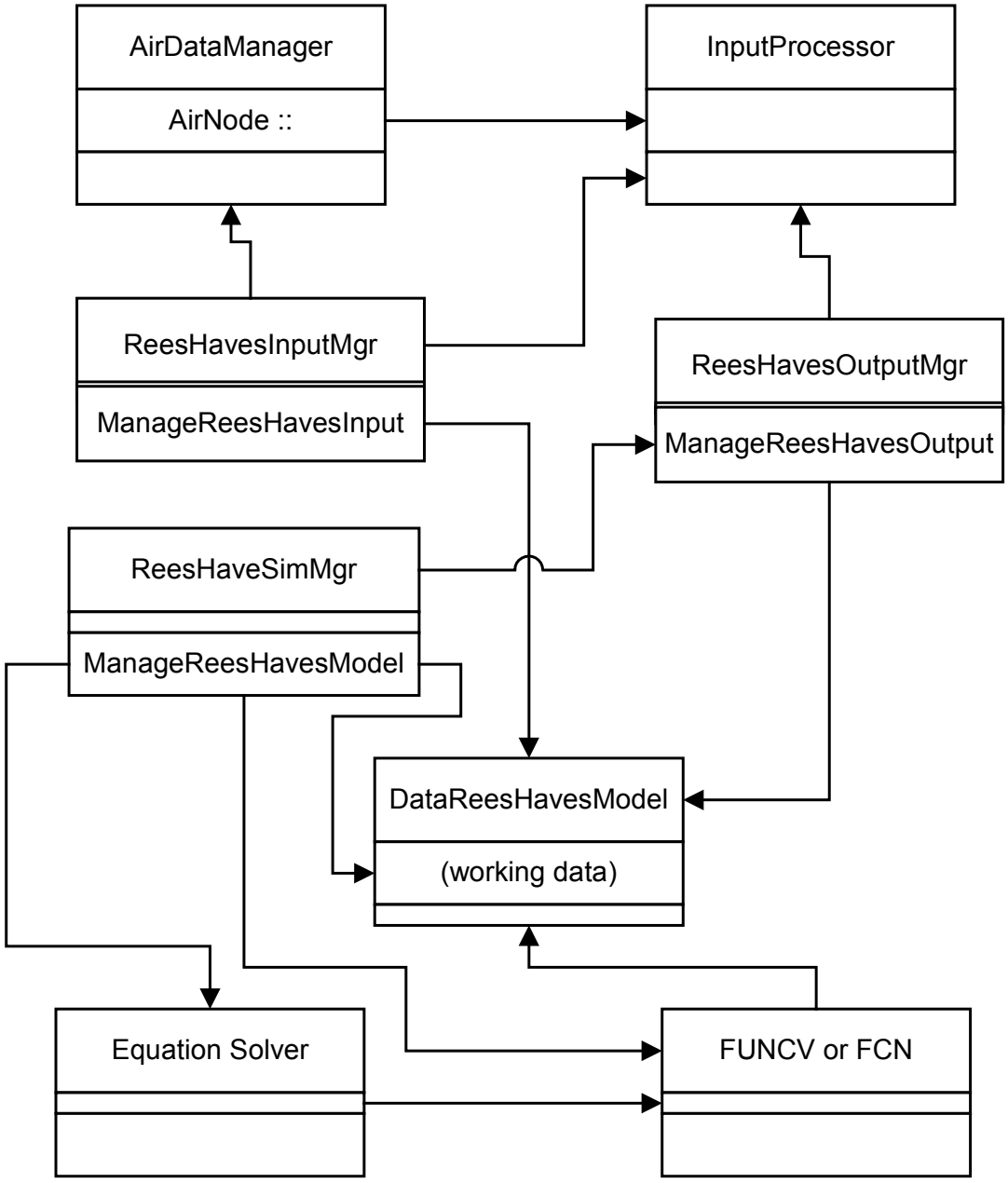


Figure A.8 Rees and Haves USE Diagram

A.6 Inard Pressure-Zonal Model

This project has attempted to implement a pressure-zonal model described by Inard et al. (1996). Unfortunately it has been difficult to obtain well-converged solutions. Unless significant progress is made this model should probably be removed from the toolkit.

This section contains various notes related to the implementation. The Inard model has been used (or attempted to) with a number of different solvers including: (1) Newton-Raphson with line search and backscatter (see section 2.2.2), (2) secant method using Broyden's update technique (Press et. al. 1996), (3) IMSL Math Library routine DNEQNF which is based on the MINPACK subroutine HYBRD1, and (4) IMSL Math Library routine DNEQNJ which is based on the MINPACK subroutine HYBRIDJ. The latter use a modified Powell technique that is another variation on Newton-Raphson of the type known as double dog-leg. Work is ongoing.

The fortran90 files required for the IMSL versions are listed in Table. A.13. As in the Rees and Haves model the balance equations are written in a function called FCN.f90. Most of the involved code that computes the terms of the balance equations resides in a separate module CellFaceRoutines.f90 and the equations in FCN.90 mostly sum the terms. Figure A.9 diagrams how the modules work together. Figure A.10 shows the coordinate system used. Table A.14 shows the integer type codes used to identify coordinates in the code. This coordinate system and integer type codes are the same as elsewhere in the toolkit with the exception of POMA.

Table A.13 Fortran Files for Inard Pressure-Zonal Model

Category	Source file	notes
Inard Pressure-Zonal Model	ZonalInardSimMgr.f90	entry point
	ZonInardInputMgr.f90	Transfer data into model, initialize
	ZonInardOutputMgr.f90	Report to `AirDataManager`
	DataZonalInard.f90	Working data for model
	CellFaceRoutines	Compute main terms
	FCN.f90	Balance Equations
	LSJAC.f90	Jacobian of FCN
Solver	IMSL Numerical Libraries DNEQNF or DNEQNJ	Newton-Raphson MINPACK HYBRID1
Air Model Toolkit	AirDataManager.f90	
	InputProcessor.f90	from E+, Loads Toolkit
	PsychrometricsMod.f90	from Loads Toolkit
	UtilityMod.f90	TempKtoC

Table A.14 Direction Definitions.

Direction Identifier	*Coordinate System*	Cardinal	
1	- X	West	W
2	+ X	East	E
3	- Y	South	S
4	+ Y	North	N
5	- Z	Top	T
6	+ Z	Bottom	B

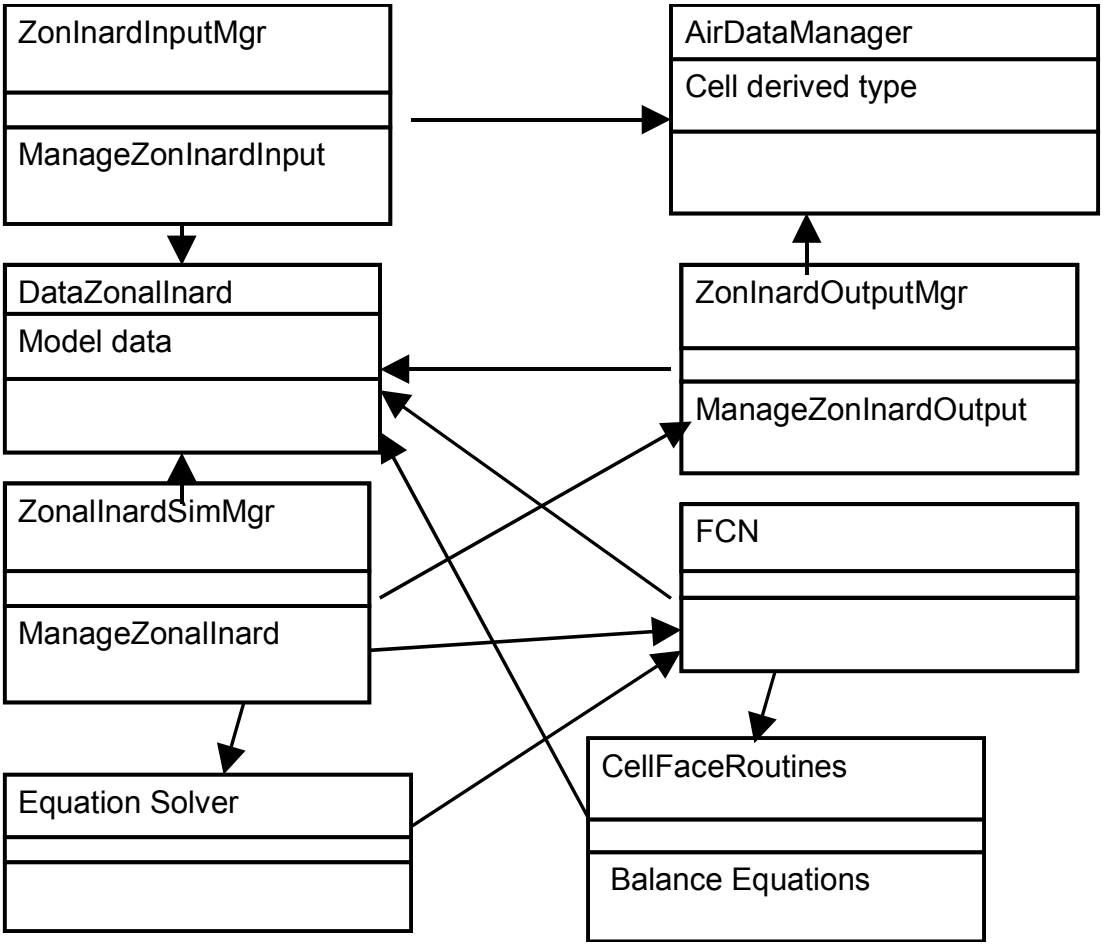


Figure A.9 Inard pressure-zonal model `USE` *diagram*

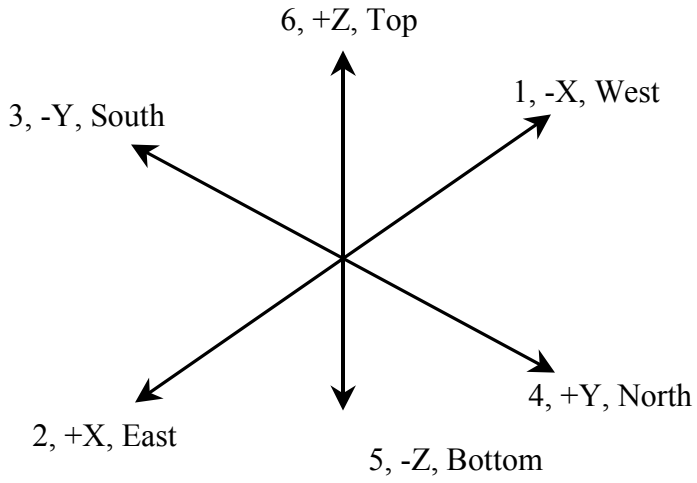


Figure A.10 Inard pressure-zonal model directions and coordinate system

Cell and Face Type Codes

This section presents information on the organization and control constructs used in the Air Model Toolkit implementation of the model by Inard et al (1996). Taxonomy of different types of cells and flows is used to control how the flow and thermal characteristics are computed for a given cell. The organization of cell types and face types is shown in Tables A.15 to A.17.

Table A.15 Type Definitions for Cells

CellType	Description	Acceptable face flow types
1	Drift w/ Reference Presssure	$[\dot{m}3, \dot{m}4, \dot{m}5], [\dot{Q}3]$
2 (default)	Drift , "standard"	$[\dot{m}3, \dot{m}4, \dot{m}5], [\dot{Q}3]$
3	Slip Wall Drift, typically horiz	$[\dot{m}1, \dot{m}3, \dot{m}4, \dot{m}5], [\dot{Q}1, \dot{Q}3, \dot{Q}4]$
5	Wall Plume Special	
6	Jet Special	
7	Heat source Plume Special	

Table A.16 Type Definitions for Cell Faces: Mass Flows

MdotType	Description	Notes
1	Blockage/wall	$\dot{m} = 0$
2	Source/Sink	For inlets and outlets
3 (default ë)	Horizontal Bernoulli Drop	For vertical face
4 (default ñ)	Vertical Bernoulli Drop	For horizontal face
5	Entrainment Balance	
6	Wall Plume Model	
7	Jet Model	
8	Heat Source Plume	

Table A.17 Type Definitions for Cell Faces: Energy Flows

QdotType	Description	
1	Adiabatic	No flux
2	Source/Sink	Heat flux B.C.
3 (default)	Flow Enthalpy (mcT)	Usual flows, Inlet/outlet
4	Surface Convection	Temperature B.C.

Within the program many logical masks are used to control array accessing using `Pack` statements. The default values indicate how control volume faces are initialized. When adding additional objects the program will need to "unset" the logical masks for that type (e.g. `Mdot3Mask(1) = .false.`). MdotType = 3 is for cell faces oriented vertically. MdotType = 4 is for cell faces oriented horizontally. The logical masks are used to control program flow and therefore need to be carefully set when applying new treatments to the cell faces.

A.7 POMA Pressure-Zonal Model

The POMA pressure zonal model is included in the toolkit as an example of a pressure-zonal model. *ZonPressSimMgr.f90* is the entry point for a collection of modules that comprise the model termed *POMA* for "**P**ressurized z**O**nal **M**odel with **A**ir diffusers." The code in the toolkit is by Lin and is a fresh translation into Fortran90 for RP-1222; the original POMA was written in C/C++. POMA in fortran90 emulates Object Oriented Programming. This model uses modified, double-precision versions of numerical routines published by Press et al. 1996 (Numerical Recipes) for Newton-Raphson solver. Table A.18 lists the files needed for using POMA. Figure A.11 shows how the modules interact. Figure A.12 shows the coordinate system inside POMA.

Table. A.18 Fortran files for POMA pressure-zonal model

Category	Source file	notes
POMA Model	POMADriver.f90	Stand-alone POMA
	POMASimMgr.f90	Coupled Entry point (formerly main.f90)
	POMAInputMgr.f90	Transfer data in (formerly input.f90)
	Room.f90	Thermal zone
	Zone.f90	Data for each cell
	VerBoundary.f90	vertical faces
	HorBoundary.f90	Horizontal faces
	WallSurface.f90	Wall heat transfer
	Jet.f90	Jet models
Air Modeling Kit	AirDataManager.f90	Toolkit routine
	InputProcessor.f90	utility
	PsychrometricsMod.f90	utility
	UtilityMod.f90	utility
Equation Solver	newt.f90	NR solver entry point
	fdjac.f90	Finite difference Jacobian
	fminln.f90	NR
	funcv.f90	Balance equations
	lnsrch.f90	NR
	lubksb.f90	NR
	lucmp.f90	ludcmp.f90 ?
	nr.f90, nrtype.f90, nrutil.f90	NR

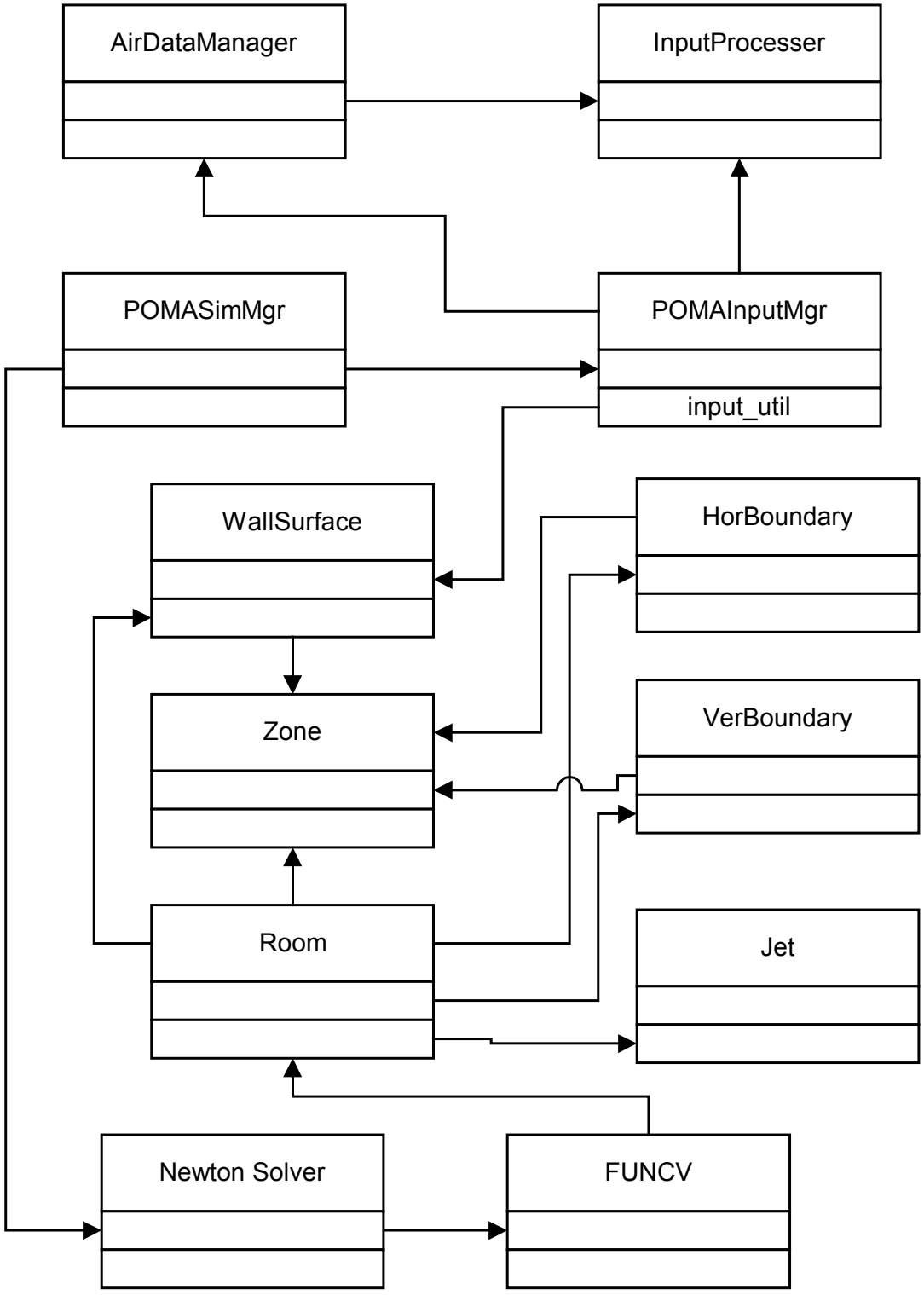


Figure A.11 POMA Model USE *diagram*

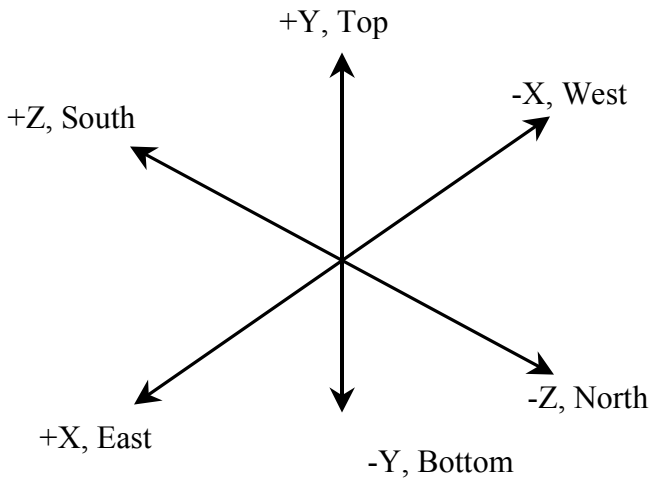


Figure A.12 POMA pressure-zonal model directions and coordinate system

A.8 Momentum-Zonal Model

The Air Model Toolkit includes code that implements the momentum-zonal model described in section 2.2.3. The code is a collection of modules constructed to solve the steady-Euler equation using finite-volume techniques. *ZonalMomentumSimMgr.f90* is the entry point. The underlying code is from Prof. Q. Chen and builds from a legacy CFD code as well as the more recent ASHRAE research project (RP-927) that implemented a simplified CFD program (Chen et al. 1999). Much of the code is based on older code originally written in FORTRAN77. The F77 version was reworked so as to use fortran90 language features such as modules, IMPLICIT NONE, and PRIVATE. It compiles as free format .f90 files. Table A.19 lists the files needed use the momentum-zonal model. Figure A.13 shows how the modules interact.

Table A.19 Fortran90 files for momentum-zonal model

Category	Source file	notes
momentum-zonal model	*ZonalMomentumSimMgr.f90*	Entry point
	ZonMoCoefficientCalcs.f90	compute terms
	ZonalMomentumData.f90	model working variables
	ZonMoInputMgr.f90	initialize and input
	ZonMoOutWriter.f90	write files
	ZonMoReportMgr.f90	report to Airdatamanger
	TriDiagsCalcs.f90	TDMA solvers
Air Model Toolkit	*AirDataManager.f90*	
	InputProcessor.f90	
	PsychrometricsMod.f90	
	UtilityMod.f90	

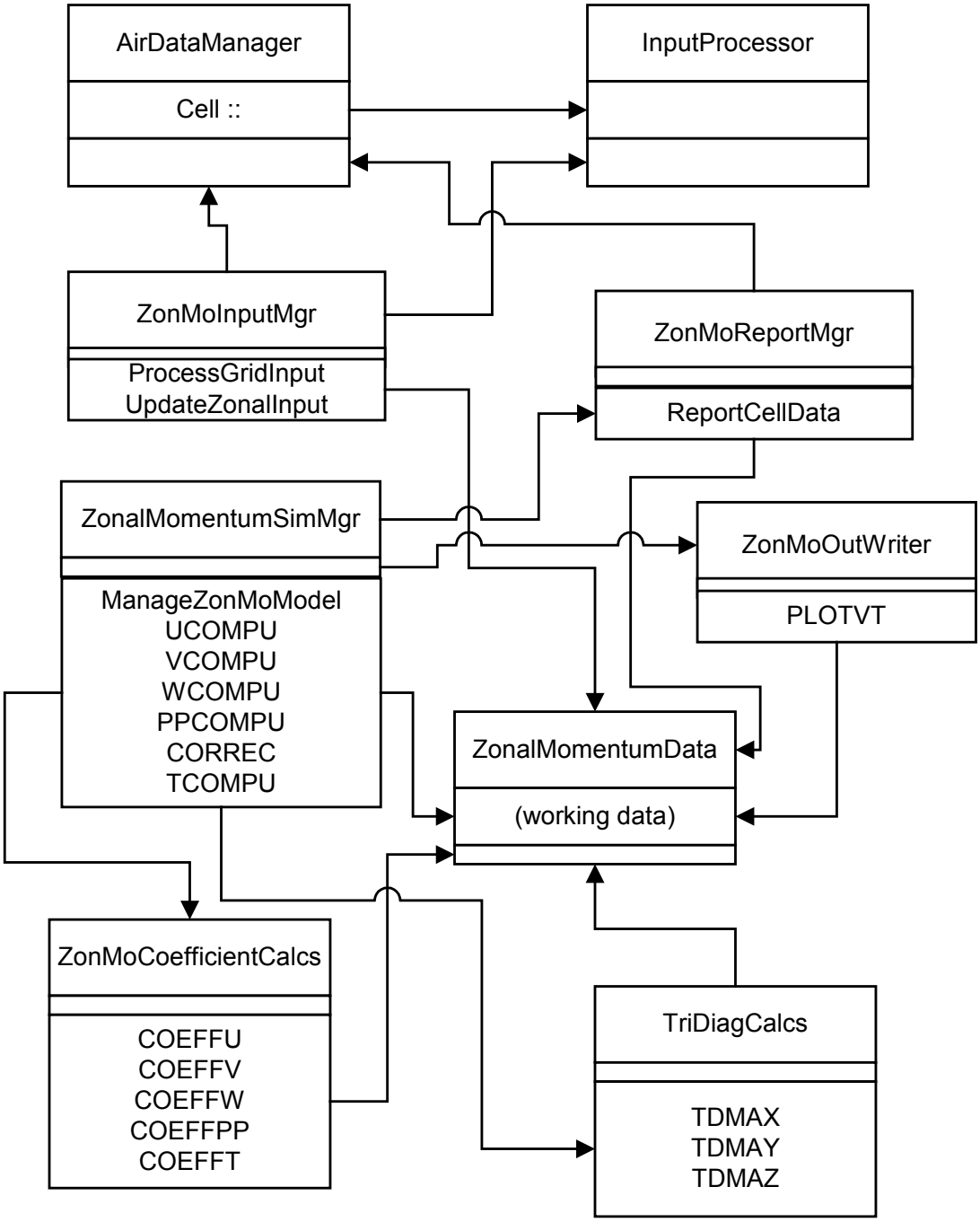


Figure A.13 Momentum-zonal model USE diagram

A.8.1 Notes on Momentum Zonal model

The remainder of the section discusses the inner workings of the momentum-zonal program. Figure A.14 shows the coordinate system and integer type codes used in the code. Figure A.15 shows the calling sequence for main iteration loop and shows the order in which the sequential solutions are made.

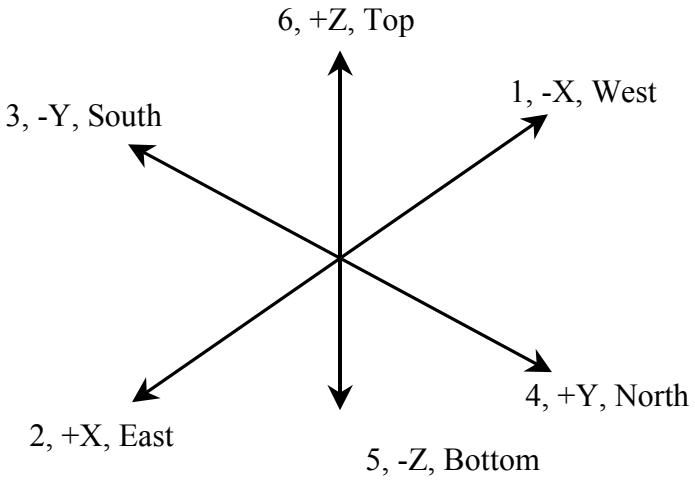


Figure A.14 Momentum-zonal coordinate system

Array sizes

The dimensions of arrays in *ZonalMomentumData.f90* are fixed at compile time by the variables, `NI`, `NJ`, and `NK` designated as `PARAMETER`. The current settings are such that the maximum grid number allowed is 60 cells in any direction. Also the number of boundary conditions that can be specified is fixed by the `PARAMETER`-variable `NB` which is currently set at 50. The dimensions can be increased or decreased. A user needs to change the values for `NI, NJ, NK`, and `NB` and recompile the program. The model typically operates at grid numbers well below these limits. Memory requirements for array sizes (e.g. 60 x 60 x 60) are around 35 Mbytes. At lower grid numbers 10 x 10 x 10 or lower the model is suggested as an alternative "zonal" model however at fine-grid numbers the model should compute potential flow field with Bousinessq buoyancy.

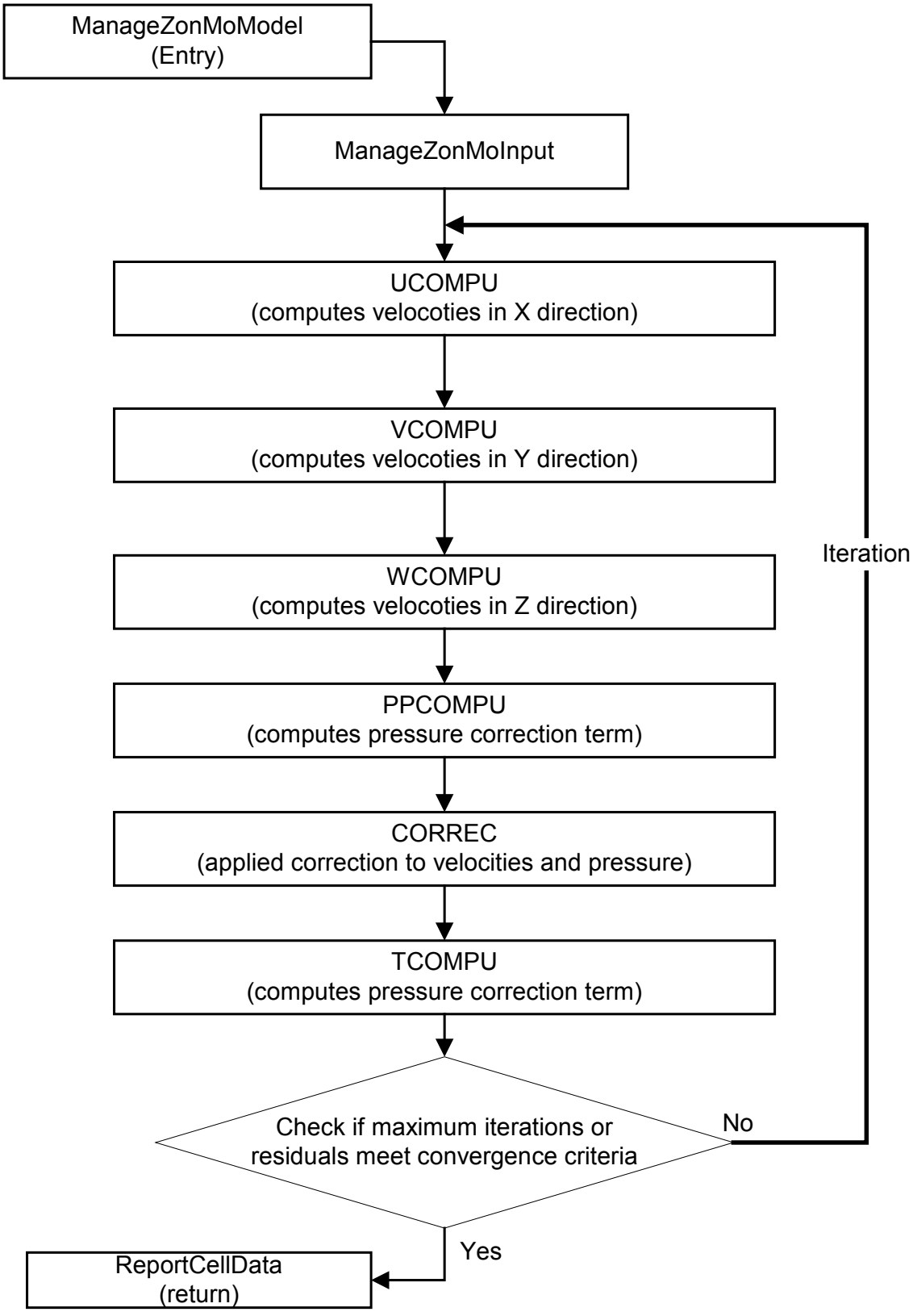


Figure A.15 Momentum-zonal program calling sequence

Variables in Momentum-Zonal

This section provides more detailed information on variables used inside the momentum-zonal program. Table A.20 list the dimensions of variables that have been grouped into different categories.

Table A.20 Variable names and array dimensions .

Variable name	Dimensions
Geometry and grid definitions	
XX, YY, ZZ	1
NNX, NNY, NNZ	1
DELX	NNX
DELY	NNY
DELZ	NNZ
X	NNX
Y	NNY
Z	NNZ
AREX	NNY, NNZ
AREY	NNX, NNZ
AREZ	NNX, NNY
VOL	NNX, NNY, NNZ
VolumeU, VolumeV , VolumeW	NNX, NNY, NNZ
Boundary conditions	
NBIN	1
LBIN	NBIN, 6
UBIN	NBIN
VBIN	NBIN
WBIN	NBIN
AMASSIN	NBIN
TBIN	NBIN
NBOUT	1
LBOUT	NBOUT, 6
PBOUT	NBOUT
TBOUT	NBOUT
NBL	1
IBL	NBL, 6
HSOU	NBL
NAWT	1
LNAW	NAWT, 6
TNAW	NAWT
NAWQ	1
LNAWQ	NAWQ, 6
QNAW	NAWQ

Control parameters for calculation and printing	
NITMAX	1
CRITE	1
TINIT	1
TRef	1
UMIN, UMAX	1
VMIN, VMAX	1
WMIN, WMAX	1
TMIN,TMAX	1
URFU,URFV,URFW,URFP,URFT	1
DTU, DTV, DTW, DTT	1
PINTV	1
IP1, IP2, IPSTP	1
JP1, JP2, JPSTP	1
Fluid properties	
DENS	1
ANU	1
ACP	1
PRL	1

Notes on variables.

XX, YY, ZZ - real variables for the flow domain in the x-, y-, and z-direction, respectively. Note that the gravitational force is always in the negative z-direction. For a two-dimensional case, use 0.0 m for the third dimension, but the program will adjust it to 1.0 m automatically.

NNX, NNY, NNZ - the number of control volumes (grids) in the x-, y-, and z-direction, respectively. Those numbers cannot be greater than parameters NI, NJ and NK, respectively, which are parameters set in *ZonalMomentumData.f90*.

DELX, DELY, DELZ - the arrays that store the dimension for each control volume (grid size) in the x-, y-, and z-direction, respectively. The sum of all the control volumes in the x-direction must be equal to XX. The same rule applies to YY and ZZ.

X, Y, Z - the arrays that contain the coordinates for the grid nodes (control volume centers).

AREX, AREY, AREZ - the arrays that store the surface area for each control volume normal to the x-, y-, and z-axis, respectively.

VOL - a three-dimensional array with volume size for each control volume.

NBIN - the total number of the inlet openings. This number should be equal or less than ten.

LBIN - an array to define the location of an inlet opening by the first and last control volume occupied by the inlet in the x-direction, the first and last control volume occupied by the inlet in the y-direction, and the first and last control volume occupied by the inlet in the z-direction, respectively. For a two-dimensional inlet, the third direction is represented by the two same numbers. If the inlet opening is

on the x=0 wall, use 1, 1 for instance. The inlet must be on a wall, not in the flow domain.

`UBIN, VBIN, WBIN` - the arrays that store inlet velocities in the x, y and z direction, respectively.

`AMASSIN` - an array for the mass flows through the inlet.

`TBIN` - an array for the inlet temperature.

`NBOUT` - the total number of the outlet openings. This number should be equal or less than ten.

`LBOUT` - an array that defines location of the outlet openings in the same way as that for `LBIN`. Outlet openings are treated as two-dimensional objects that can be placed on the enclosure walls. Hence, rules explained for the definition of location for inlet openings also apply for the outlet opening.

`PBOUT` - an array for the outlet pressure.

`TBOUT` - an array for the outlet temperature. These values should be set as close as possible to the real value. The outlet temperature and velocity have an impact on the convergence speed.

`NBL` - the total number of the blockages used in the flow field. The blockages are usually used for furniture, lights, occupants, partition walls, etc. This number should be equal or less than fifty.

`IBL` - an array that defines positions for each blockage in the same way as LBIN. The blockages are always three-dimensional and must be in the flow domain.

`HSOU` - an array that stores the heat generated by each blockage.

`NAWT` - the total number of the enclosure surfaces with a constant temperature. The maximum number of the non-adiabatic surfaces is fifty.

`LNAW` - an array that defines the positions for the surfaces in the same way as that for `LBIN`. The definition of the location has the same restrictions as for the inlets.

`TNAW` - an array for the temperature of the non-adiabatic walls.

`NAWQ` - the total number of enclosure surfaces with constant heat flux. The maximum number of the non-adiabatic surfaces is fifty.

`LNAWQ` - an array that defines the positions of the enclosure areas with heat flux. The definition of the location has the same restrictions as for inlet openings.

`QNAW` - an array for the heat from the non-adiabatic walls

`NITMAX` - the maximum iteration number for the run.

`CRITE` - a convergence criterion. The computation ends when the total residual is smaller than CRITE or when the maximum iteration number is reached.

`TINIT` - an initial value for the temperature field. The initial velocities are internally set to zero.

`TRef` - the reference temperature

`UMIN, UMAX, VMIN, VMAX, WMIN, WMAX, TMIN, TMAX` - the minimum and maximum values for U, V, W, and T, respectively, during the solving procedure. This may help the solution because the values may run out of control during the iteration. Those ranges, however, should be wide enough.

`URFP, URFU, URFV, URFW, URFT` - the linear under-relaxation factors for P, U, V, W, and T, respectively. The value of these factors should be between 0 and 1. A value of 1 implies no relaxation. A lower value will slow the convergence but will stabilize the calculation.

DTU, DTV, DTW, DTT - the false time steps for U, V, W, and T, respectively. The false time steps operate in a similar fashion as the linear under-relaxation factors. However, the range of the false time steps can be from 10^{-10} to 10^{10}. A large value means no relaxation and a small value implies heavy relaxation. As a rule of thumb, the following estimates the values for the false time steps: false time step relaxation factor = smallest grid size / largest velocity in the flow field. The false time steps can be increased one order higher or lower in the calculation to achieve a converged solution.

PINTV - a real variable setting the printing frequency for the residuals during the calculation

IMON, JMON, KMON - the grid number in the x-, y-, and z-direction, respectively, for the monitoring point. When the values of the variables at the monitoring grid do not change significantly, a converged solution is reached. Must be within the the air domain

IREF, JREF, KREF - the grid number for the reference point. It should be inside the air domain and not inside blockage.

IP1, IP2, IPSTP - The number of the first control volume, last control volume and interval for printing results in the x-direction.

JP1, JP2, JPSTP - The number of the first control volume, last control volume and interval for printing results in the y-direction

ICOMPU - Set to 0 if not starting from the previous results and set to 1 if starting from the previous results

DENS - Fluid density (kg/m^3)

ANU - Fluid kinematic viscosity (m^2/s)

ACP - Fluid specific heat at constant pressure (J/kg K)

PRL - Fluid Prandtl number

Additional output files

For the Air Model Toolkit the computed results of interest are passed to the AirDataManger and output to the file *ZonalData_.out*. The momentum-zonal program however generates additional results files that may be useful in stand-alone. These are fairly traditional output for CFD programs. All data files are formatted ASCII (text).

Balance.dat contains values of the residuals for mass, T, U, V and W equation for each INTVAL iterations during the calculation. The information is the same as that sent to the display device during the execution. Monitoring, and documenting, the residuals can help the user to monitor the convergence and tune relaxation parameters.

Result.dat contains (1) the basic thermal and fluid boundary information and (2) the results. The beginning of this file describes the basic information about geometry, boundary conditions and relaxation factors. Then the file gives the results of the velocity components U, V, W, the mean scalar velocity, and temperature in the z-direction, respectively. The printing on the x-direction (I) and the y-direction (J) can be controlled.

The user can set the control parameters in the input file – see the input reference for 'Momentum-Zonal Controls.' .

TECPxxxx.out are files formatted for Tecplot software. Tecplot is a commercial plotting program for visualizing and analyzing the numerical data. The data provided in this file include: the position of the center of the control volumes and the cell-center values of the velocity components, mean velocity, pressure, and temperature. In coupled mode many but not all of the TECPxxxx.out files are created. This data is organized as "IJK-Point."

A.9 Implementing Additional Air Models

This section provides information on using the Air Model Toolkit to couple different air models to the loads routines. Section A.2 introduced the overall coupling framework and the preceding sections discussed how the other air models are arranged. Recall that the overall intent is to use a model to provide an array of zone air temperatures to the load calculations. The model and its data are "connected" to the load calculation in two areas, (1) intialization/setup and (2) normal calculation. We first describe the overall relationships and then discuss where in the code to make alterations for these two areas. These are only suggestions and the user is obviously free to do what he or she likes with the code!

A.9.1 overview

Figure A.16 diagrams the interaction of the modules. The "package" would be the users desired new air model called "My Model." The main subroutines of interest in the loads calculation domain are `CalcHeatBalance` and `CallAirModel` which reside in the source code file *HeatBalanceSimMgr.f90.* This file evolved from *SuccessiveSubstitutionSolution.f90* in the Loads Toolkit. Most of the important loads domain data are stored at the module level here, while the detailed air domain data are stored in `AirDataManager` at the module level. The movement of user input data is through the `InputProcessor` utility module. The `AirDataManager` already loads and processes a number of different input objects useful for air models. Unique data for a new model can be input by developing new input object definitions for the dictionary file *ToolkitTest.idd* and then using the `InputProcessor` utility. This is fairly easy and the user is referred to EnergyPlus documentation (*InterfaceDeveloper.pdf* and *ModuleDeveloper.pdf*). Thus, one-time input data are brought in through both the *AirDataManager* and calls to the `InputProcessor`. In a loads calculation the air model will get called perhaps many thousands of times and these subsequent calls would likely only get updated boundary condition data from the `AirDataManager` rather than continuing to access data from the input file. Much of what the input routines have to do is keep track of the relationships between different surfaces and nodes or cells so that input and output are coordinated for the correct entities. Using the `AirDataManger` should facilitate this coordination. Once the air model has acquired all the input it needs for a particular run, it computes its results and passes them back to the `AirDataManager`. The load calculation routines then get the new data from the `AirDataManager`.

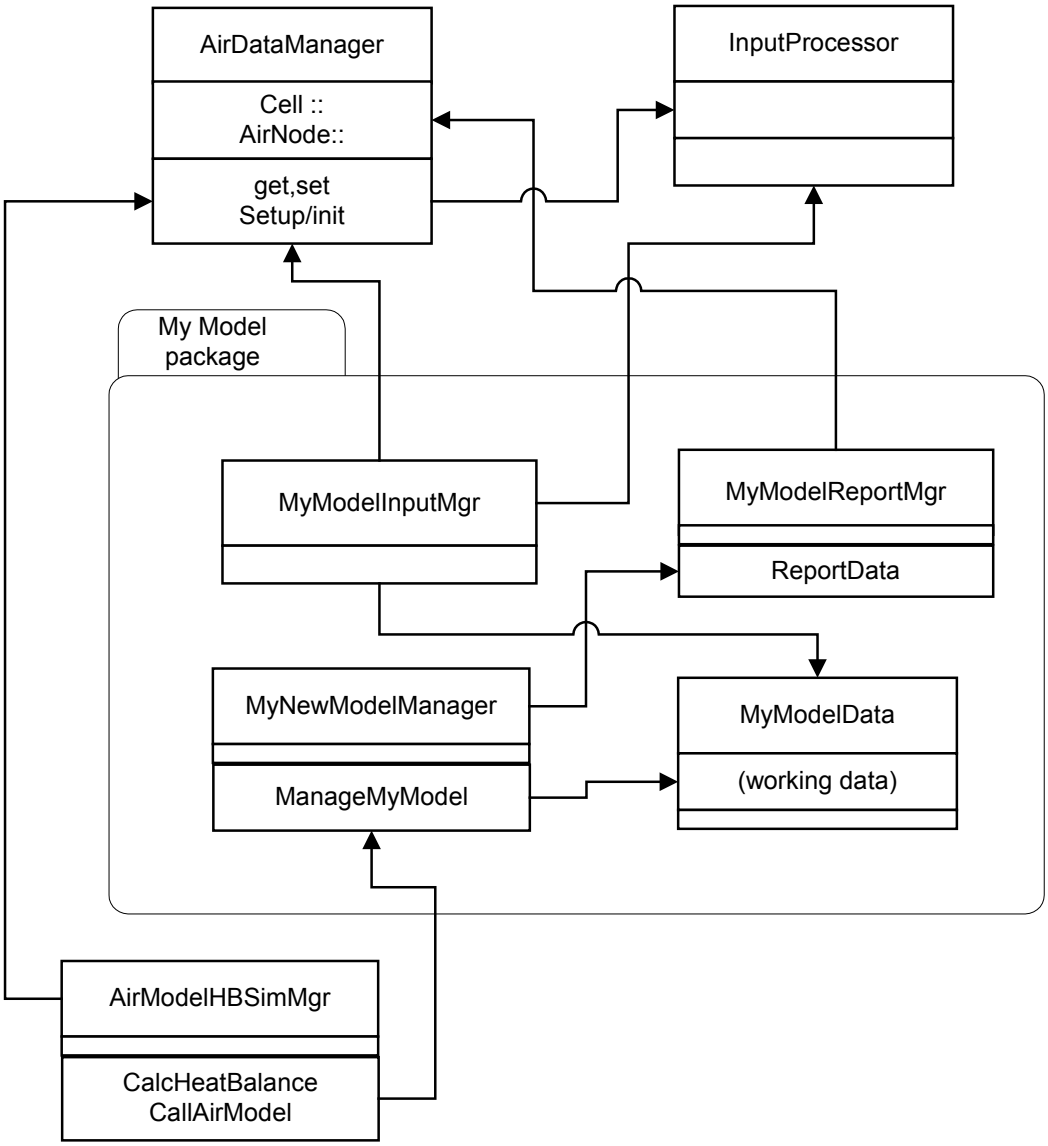


Figure A.16 Generic USE diagram for a user's new model.

The new air model should choose an integer type code to identify it in control statements though the variable `AirModelType`. For example the momentum-zonal model uses "6" and the next available type code is "7". This type code will be filled into `AirModelType` based on the users input in the object 'Select Air Model' and the code can vary from day to day. The type code is used in program control statements.

A.9.2 Setup and Initialization.

The routine `CalcHeatBalance` is the main controlling routine for a loads calculation. The first part of this routine calls set up and initialization routines to allocate, initialize and otherwise prepare variables for the computations. The second part implements the

main iteration loops. The following excerpt shows how two of the routines implemented in the toolkit are initialized (one nodal and one zonal) and shows how the user might edit the file to initialize their new model assuming that it is a nodal model.

```
Select Case (AirModelType(day))

Case (2) ! Mundt Nodal Model
         CALL ConstructNodes   ! in AirDataManager
         CALL ResolveSurfIDs   ! in this module
         CALL SetupMundtModel(ZoneHeight, FloorArea)

Case (6) ! momentum zonal model
         CALL ReadIDFZoneGrid  ! in AirDataManager
         CALL InitAirDataCells ! in AirDataManager
         CALL ResolveSurfIDs   ! in this module
         CALL setBarametricPressure(BarometricPressure)

Case (7) ! my new model
         CALL ConstructNodes   ! in AirDataManager
         CALL ResolveSurfIDs   ! in this module
         CALL SetupMyNewModel

END SELECT
```

The example above shows two types of init routines in the `AirDataManager`, one for nodal models and one for zonal models. The `AirDataManager` has a "split personality" in that it has data structures that differ between nodal models and zonal models. This is because nodal models have a somewhat free geometry whereas the zonal models are considered to use a structured grid (or 3-D orthogonal mesh). The `AirNode` structure is 1-D; the `cell` structure is 3-D. The `ResolveSurfIDs` routine handles coordinating the array locations of surface data in the loads domain to the array locations of surface data in the air domain. These can be different because of the policy that order not matter in the input file. The new air model can also be setup to obtain startup data and initialize data the first time it is called to do a computation as is done with the momentum-zonal model. However the setup/initialization routines for using the `AirDataManager` must be called in order to pass the data out of the loads domain into the air domain.

Including a `USE AirDataManager` statement and using public access is the simplest method of obtaining input data from the `AirDataManager`. There are numerous structures, or derived types, that can hold useful data. Inspect the code to see what is useful. The main structures are listed in Table A.20.

Table A.20 Primary derived types in AirDataManager

Structure name	dimension	notes
`AirNode`	`(TotalAirNodes)`	nodal models, model results each timestep
`Cell`	`(NX, NY, NZ)`	zonal models, model results each timestep
`SurfSet`	`(numSurfSettings)`	store boundary conditions and results for near-wall air temps
`Inlet`	`(numInlets)`	boundary conditions
`Outlet`	`(numOutlets)`	boundary conditions
`Content`	`(numContents)`	boundary conditions

A.9.3 Calling Air Model

The actual call to the air model takes place in the routine `CallAirModel`. This routine is called from the `CalcHeatBalance` routine if the `doAirModelSim` has been set to `.true.` A `Select Case` construct chooses which air model to handle based on the value in `AirModelType` for that `day`. In order to add an additional model, add an new `Case ( )` with the appropriate calls. The process is also diagrammed in Figures 3.3, A.3 and A.4. The following is an example of how one might add a call for a new zonal model with type code "8."

```
Case (8)                                  ! select "my new model"

  DO SurfNum = 1, NumOfSurfs
    CALL setSurfTemps(surface(SurfNum)%AirModelID, &
           TsIn(SurfNum, TimeStep)) ! pass T_s_i 's to air domain
  ENDDO

  CALL setInletZonal(AirModelInletID, MdotIn, SupplyAirTemp, &
           HUMRATsupply)                  ! pass inlet flow rate to air domain

  Call setZonalContent(timestep, PeopleQc,&
           ElecEquipQc, LightingQc) ! pass internal loads to air domain

  Call ManageMyModel(timestep)            ! call to compute air model

  CALL getEffBulkAirNodal(timestep,& ! call to retrieve T_a_i 's from air domain
    Surface([1:NumOfSurfs])%AirModelID, TeffBulkAir(:,timestep))

  CALL GetZonalLeavingConditions(timestep,Tleaving(timestep),&
           Mdotleaving, Wleaving)    ! call to retrieve T_leaving

  Call GetZonalControlTemp(timestep,TcurControl(timestep)) !  T_statDB
```

```
    DO SurfNum = 1, NumOfSurfs
      CALL getSurfHc(timestep, surface(SurfNum)%AirModelID,&
               HcIn(SurfNum, TimeStep)) ! retrieve h_ci 's from air domain
    ENDDO

  END SELECT
```

Not shown is the DT-coupling option that shifts results based on any deviation between the current reading of the thermostat, `TcurControl` , and the desired room air set point, `TroomSetpoint`.

APPENDIX B: PROGRAM USER GUIDE

The Air Model Toolkit contains programs that perform load calculations with a variety of air models. These "programs" are implemented as test/demonstration/research code and would need additional development and validation before serving as robust commercial software.

This Appendix compiles information on how to prepare input and run the programs. The programs were compiled for 32-bit Windows. The user can recompile source code for other operating systems. The overall process of running simulations and creating input files is comparable to that required to use the original ASHRAE Loads Toolkit (Pederson 2001) and EnergyPlus. This guide augments the Loads Toolkit and refers the user to that documentation for important information on the input for models in the loads domain of the computations.

The following typographical conventions are used in this report.

text	general document text (Times New Roman 12pt)
filename.idf	the name of a computer file (Italic Times New Roman 12pt)
`CodeFont`	actual fortran90 code or program input (Courier New 10pt)

All of the toolkit programs use an input data dictionary called, *ToolkitTest.idd*, along with an input data file *in.idf* to provide user input data to routines. Considerable effort was made to consolidate the entire input required to run air models *and* load calculations into this one input file. Many new input objects were developed in order to describe air models. In general, when trying to understand the input parameters for an object, the first place to look is the entry for that object in the data dictionary contained in the file, *ToolkitTest.idd.*

Most of this Appendix B discusses input for running the models. The first section gives an overview of the programs and provides instructions for a quick start. The second section provides a discussion of general modeling considerations when describing a thermal zone in the context of coupled air models. The next five sections provide details on running both stand-alone and coupled versions of the individual air models. The last section presents and encyclopedic reference for the new input objects used to input data for the air models.

B.1 Quick Start for Running pre-compiled Programs

The Air Model Toolkit contains ready-to-run demonstration programs and sample test cases. The executables were compiled on a Windows 2000 platform using Compaq Visual Fortran version 6.6 (workspace files for this development environment are also

included). The quickest way run a program is to navigate to one of the TestCases directories and choose a model subdirectory such as,

```
\AirModelToolkit\TestCases\DisplaceVent_B3\ReesHavesModel\coupled
```

Then execute a batch file such as *RunAirModelTest.bat* , perhaps by double-clicking on it. Directories with test case input files have DOS batch files for convenience in handling the file moving and renaming involved in running a program. Input file names can be edited in the batch file to select a different input file. In some cases batch files are set up to run multiple cases in sequence. The batch files do useful things like copy over the results files, log what would be sent to the screen during command line operation, and record the starting and ending system clock times. The basic loads output is in the file *toolkit.out* but the batch files will prepend the input file name to the beginning of the file name for each of many output files.

The primary program distributed with the toolkit is called *AirModelLoads.exe* and it is located in,

```
\AirModelToolkit\SamplePrograms\AirModelLoads\debug\AirModelLoads.exe
```

Other programs are provided that have been compiled to run in slightly different ways (such as Rees and Haves model "A" vs. model "B"). There are also programs associated with air model components that offer stand-alone air modeling without the load calculations; these are located in the component sub-directories.

If you do not want to use the batch file method, you are free to rearrange files at will. In order to run any of the programs, all that is needed is to put two input files, *ToolkitTest.idd* and *in.idf*, in the same directory as the program executable.

B.2 Thermal Zone Modeling Considerations

It is assumed that the user is familiar with modeling building thermal zones for load and/or energy calculations. The first section here provides a short review of the input needed to describe a thermal zone to the sample programs, but detailed explanations should be sought elsewhere. The second section discusses the issue of sub-dividing surfaces beyond normal building energy simulation practice. The third section discusses partitioning a thermal zone into a structure grid for zonal models.

B.2.1 Modeling a thermal zone with the Heat Balance Model

This section provides a short review of the input needed to describe a thermal zone to run the sample programs. The ASHRAE Loads Toolkit (Pederson 2001) is the starting point for the Air Model Toolkit and roughly all the input associated with describing a zone for the Heat Balance Model apply to the coupled programs. Table B.1 lists input objects an input file should probably contain in order to define a single zone. The air models will

require additional input objects. See the Loads Toolkit documentation for more information on these input objects (or as always the input data dictionary *ToolkitTest.idd*)

Table B.1 Summary of Loads Toolkit input objects for a load calculation

Input objects	notes
Zone	overall zone, height and area
SURFACE	section of building fabric, orientation, area, radiation properties, reference to construction
Location	lat, lon for solar calcs
Date	day of year for solar calcs
Environment *	pressure in Pa , wind, elevation
SuccessiveSubstitutionData	number of days in steady-periodic calc, iterations on each time step
TempInside	drybulb Temperature of room set point
TempOutside	Outdoor air dry bulb
TempWetOutside	Outdoor air wet bulb
TempDeck	Supply air temperature
SystemMaximumAirFlow	limit on supply air, 0.0 triggers VAV
Lighting	total [W/m2] internal loads from lights
LightingSplits	splits for radiation/convection/return air
Equipment	total [W/m2] internal loads, equipment
EquipmentSplits	splits for radiation/convection
People	number of people in zone
PeopleSplits	activity level and splits for radiaton/convection/
Construction	assembly of 'MaterialLayer'
MaterialLayer	define construction material properties
Window	define window material properties

***** Changes have been made to the handling of atmospheric pressure and so the 'Environment' input object should use Pascal units (N/m^2). Before, it appeared to be in bars.

B.2.2 Sub-Dividing Surfaces

Surfaces might be subdivided to take advantage of modeling the distribution of room air temperature during a building load calculation. One of the non-uniform conditions being modeled is a vertical temperature gradient and sub-dividing vertical surfaces is done with the intent that applying better values for near-wall air temperatures might improve the modeling of surface heat transfer. To accomplish this, the wall that might formerly have been modeled as one entity is sub-divided in the vertical direction. The most useful resolutions are still an open question but four roughly equal parts seems reasonable for typical floor to ceiling heights. For example, the Rees and Haves model uses four nodes. For this type of nodal model we consider one node to represent the adjacent air control volume for an entire band (or "doughnut") of wall surfaces that share a common height.

Figure B.1 depicts sub-surfacing of vertical walls for the Rees and Haves model. In this toolkit, zonal models are considered to use coarse, 3-D Cartesian grids and to provide additional resolution in the horizontal direction. But additional sub-dividing of surfaces in the horizontal is not necessarily required since many researchers have concluded that horizontal variations in air temperature are minimal outside of thermal plumes.

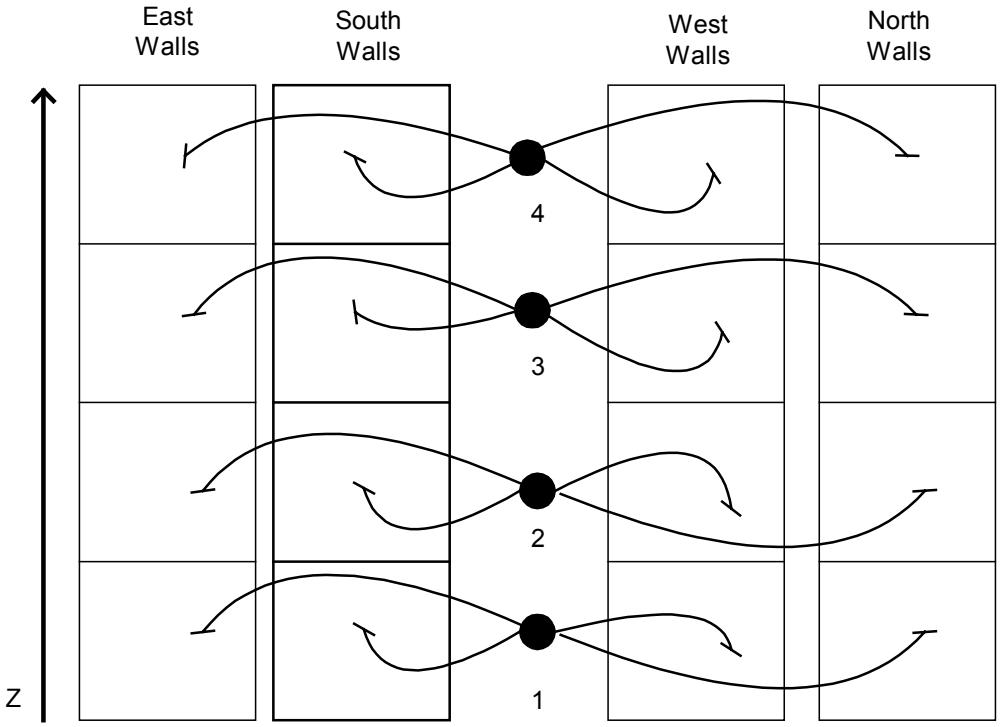


Figure B.1 Sub-surfaces for Rees and Haves model, four nodes in the vertical direction, each node is associated with walls sub-surfaces near its height

B.2.3 Zonal Model Gridding

This section discusses how to choose a grid with which to model a zone, or room, when using one of the zonal room air models. The analyst needs to decide how to grid the zone so that the input objects can be defined. Zonal models are based on a partitioning of the zone into an assembly of control volumes. A structured grid is used in this toolkit meaning that the grid arranges the control volumes, or cells, into an orthogonal, three-dimensional assembly where the distribution of cells in a particular direction can vary. A distribution is "extruded" into the volume as an unchanging array of cell displacements. This section develops an overall discussion of coarse grids for a thermal zone, additional model-specific details are discussed in the Input Object Reference in Appendix C.

While it is possible to represent partition walls within a zone, for simplicity, the term *room* is used to mean a building thermal zone comprised of a single room with its associated volume of room air and enclosure surfaces. The overall dimensions are

considered the inside dimensions for modeling room air. Currently, the programs require the room to

Ultimately of course some type of helpful graphical user interface would be very useful, but for now the analyst is likely to need to generate a grid "by hand." This is not overly difficult. The following is a list showing the tasks useful for collecting the data needed for the input object definitions. Figure B.2 diagrams a section of a 6x6x6 grid for reference.

Zone Gridding Outline

1. collect overall room dimensions
2. select grid resolution, eg. 6 x 6 x 6
3. choose 3 arrays of cell sizes by setting first and last at 0.2 m and distributing the remaining grid number evenly across each of the room dimension
4. determine actual or model positions of inlets, outlets, and contents
5. realign grid cells to align with objects identified in Step 4 (or alter model positioning of those objects)
6. determine actual or model positions of changes in surface construction
7. realign grid so that room surface construction transitions lie on grid lines
8. If Step 7 conflicts with Step 5 increase grid resolution or resize Step 4objects
9. sketch room views with grid lines, surfaces, inlets, outlets and contents and annotate grid index numbers
10. Finalize selection of sub-surfaces and calculate surface areas from grid
11. write out input objects

Just just four input objects specify the basic description of the grid, for example:

```
Zonal Cell Config, main,4.2, 3.6, 2.75, 9, 8, 8, DXs,DYs,DZs;
DISPLACEMENT LIST, DXs, 0.200000 , 0.542857 , 0.542857, 0.542857,
      0.542857, 0.542857 , 0.542857 , 0.542857, 0.200000;
DISPLACEMENT LIST, DYs, 0.200000 , 0.533333 , 0.533333,
      0.533333,0.533333, 0.533333 , 0.533333 , 0.200000;
DISPLACEMENT LIST, DZs, 0.200000,  0.391667 , 0.391667,
      0.391667,0.391667, 0.391667 , 0.391667 , 0.200000;
```

This describes a small zone that has floor dimensions of 4.2 by 3.6 m. and 2.75 m ceiling height. The grid is 9 by 8 by 8 (or 576 total grid number) and is a typical grid for the momentum-zonal model but is likely too fine of a grid for the pressure-zonal models where a 5 x 5 x 6 grid is more typical. The `'DISPLACEMENT LIST'` objects define arrays of cell displacements as lists of the sizes of each cell in one of the three directions. The remaining input objects describe where to locate surfaces, inlets, outlets, equipment, lights, and people in terms of integer grid locations.

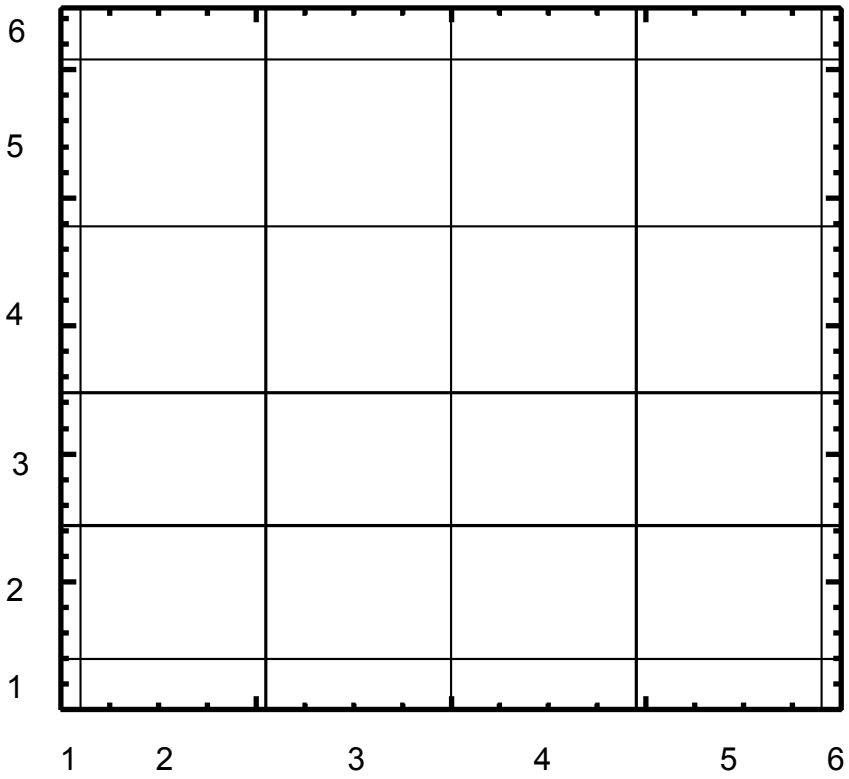


Figure B.2 Example of grid showing grid index numbering

Guidelines for selecting grid include:

- Cell size will depend on the choice of zonal model, allowable computation time, and the size of the zone, but a target size for coarse gridding is in the range from 0.5 x 0.5 x 0.25 m. to 1.0 x 1.0 x 0.5 m.
- Cells along the outer boundary of the air domain are those that interact directly with the inside face of the surface model. The size of these cells has direct implications for the effective air temperature used to compute surface convection heat transfer. This toolkit recommends (and suggests as a requirement) that the size of each boundary cell be 0.2 m normal to the surface.
- Transitions in surface constructions should occur at grid lines.
- Do not overlap the locations of objects for surfaces and inlets and outlets and contents.

The orientation of objects with respect to the grid needs to be declared. Table B.2 lists the integer type codes used to declare directions in the input (and inside the code) and Figure B.3 diagrams the coordinate system. These directions are from the perspective of inside the air or cell(s). To figure out the direction of something tion of an inlet, imagine sitting in the block of air cells adjacent to the inlet and determine which direction to travel to reach that inlet starting from the center of the air cell(s). So an inlet in a West wall will

have direction 1 and its flow direction will be in the + X direction. A north wall will have direction 4.

Table B.2 Zonal model direction definitions

Direction Identifier 'type code'	*Coordinate System*	Cardinal
1	- X	West
2	+ X	East
3	- Y	South
4	+ Y	North
5	- Z	Bottom
6	+ Z	Top

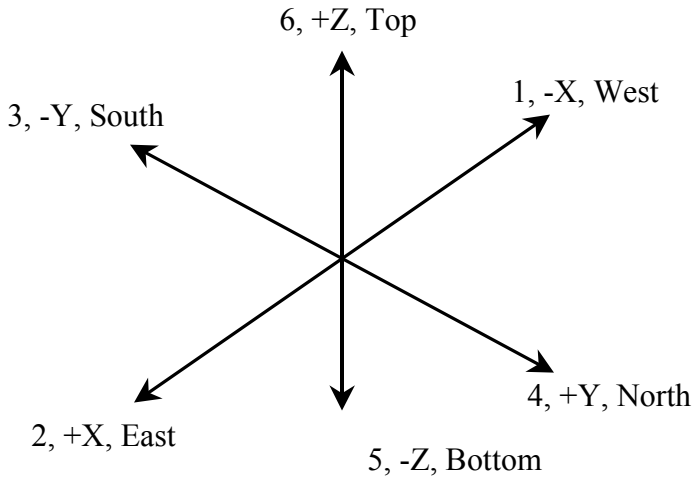


Figure B.3 Zonal model directions and coordinate system

The contents of a zone such as equipment, furniture, or people, are represented in the zonal models using a "ZONE CONTENT BLOCK" input object. Zone contents can be blocking or non-blocking. Non-blocking zone contents are used to add convection heat gains at locations interspersed within the room air. When coupled to the loads calculations, these convection heat gains are from the convective portion of internal loads from people, lights and equipment. The momentum-zonal model is constructed to model blockages; use locally finer grid scales (~0.1 m) to enable defining walls, tables tops and the like. For the coarsest grids, using blocking contents is not recommended. While the

momentum model is setup for zone contents to span blocks of multiple cells (hence the term “Block” meaning block-of-cells), the pressure zonal models are setup so that only a single cell can be a zone content.

B.3 Mundt model

Using the Mundt model provides a simple method of predicting air temperatures in a room with displacement ventilation. The model is for cooling applications where it is reasonable to model the air distribution as ideal displacement ventilation meaning the distribution system succeeds in spreading all new air out right at the floor. and with no heating load (reverts to mixing if encountered). The model is mainly interesting when coupled to the loads calculations, although stand-alone operation is possible by editing the driver test program *MundtDriver.f90*.

B.3.1 Input

To run a load calculation with zone air temperature distribution predicted using the Mundt model add a ries of input object
Table B.3 lists the input objects required for using the Mundt model in addition to the objects required for a mixing model load calculation (Table B.1)

Table B.3 Input Objects for Mundt Model

Input objects	notes
`SELECT AIR MODEL`	Use air model type = `2`
`SURFACE SETTINGS`	Use one for each `'SURFACE'`
`AIR NODE`	See table 4.3
`SURFACE LIST`	
`OUTSIDE SURFACE BC:NOD`	If desired
`Mundt Model Controls`	

Table B.4 Air Node Input for using Mundt Model

Typical name (user selectable)	Parameter 2 Node Type	notes
inlet	0	
Floor air	1	
Control point	2	
Ceiling air	3	
Room nodes	4	1 for each group of walls in vertical direction
Return	10	

The exact number of nodes in the model can be varied based on the desired amount of resolution in the vertical walls. The linear vertical temperature gradient can produce a result for any graduation but the implementation in the toolkit is limited to ten groups in

the vertical. All such room nodes have type 4 but are given different coordinates in the vertical direction and a different list of surfaces with which to interact.

B.3.2 Running Mundt Simulation

The model is run by executing the program AirModelLoads.exe located in the directory,

`\AirModelToolkit\SamplePrograms\AirModelLoads\debug\AirModelLoads.exe`

The data dictionary file, *toolkittest.idd*, and the user input file, *in.idf*, also need to be located in the directory.

B.3.3 Mundt Output

The output are sent to *toolkit.out* and *nodeData_.out*.

B.4 Rees and Haves Nodal Model

The Rees and Haves nodal model has been implemented in both stand-alone and coupled versions. The model is applicable to cooling load conditions using low-velocity displacement ventilation using diffusers of the side-wall type. The model is not used during heating load conditions and may not be applicable to other types of displacement ventilation diffusers, such as under-floor air.

B.4.1 Input

The additional input object definitions required for running the Rees and Haves air model in the Loads toolkit are listed in Table B.5.

Table B.5 Additional Input Objects Required for Rees and Haves Model: Coupled Air and Surface

Input objects	notes
`SELECT AIR MODEL`	Use air model type = 3
`SURFACE SETTINGS`	Use one for each 'SURFACE'
`AIR NODE`	See table 4.5
`NODAL FLOW PATH`	See table 4.6
`SURFACE LIST`	See table 4.5
`OUTSIDE SURFACE BC:NOD`	1 each 'SURFACE' w/ NOD
`THERMOSTAT LOCATION`	Only needs z direction in [m]
`ZONE CONTENT BLOCK`	Transfers changing internal loads

In the Rees and Haves Model there are four air nodes that are linked to the inside face surfaces of the room's vertical surfaces or walls. The walls are therefore subdivided into four groups at different distances above the floor. Each group is composed of all wall surfaces that share (roughly) the same elevation off the floor. Groups of surfaces that all interact with the same node are specified by listing the surface names in a '`Surface`

`list`' input object. The temperature setting in Surface Setting objects is used when running a stand-alone model; for a coupled model the surface temperatures are recalculated by the loads routines.

Table B.6 Air Node Input for Rees and Haves Model

Typical name (user selectable)	Parameter 2 Node Type	Typical Z Height above floor	Notes
Inlet	0	0.4	No surface
Floor	1	0.1	Floor surface(s)
Plume load 1	2	¼ x height	internal heat, no surface
Plume load 2	3	½ x height	internal heat, no surface
Plume3	4	¾ x heigth	No surface,
Plume4	5	Height – 0.1	Ceiling surface(s)
Room1	6	1/8 x height	Lower walls
Room2	7	3/8 x height	Middle lower walls
Room3	8	5/8 x height	Middle upper walls
Room4	9	7/8 x height	Upper
Outlet	10	7/8 x height	No surface

The nodal model is based on having as input the movement of air amongst the different nodes. The movement of air between the various nodes is prescribed using the input object '`Air Flow Path`' which includes a capacity rate factor for each branch of the nodal network. The flow paths required are listed in table B.7 with nominal capacity factors derived from the "rules" suggested by Rees and Haves. Although the implementation only allows for the one network configuration documented here, the performance of the model can be altered by changing the values for the flow rate factors. It is critical that a self-consistent set of mass flow factors be given where the sum of mass flow into a node equals the sum of mass flows leaving the node. The flow rate fractions are values used to scale the total system air flow rates since during a simulation the actual air system flow rates are varied to find the cooling load for a VAV system. Values greater than 1.0 are possible because of entrainment and circulation caused by the plumes. For more information, refer to Figure 3.2 and Section 3.8 as well as Rees and Haves (2001).

Table B.7 Air Flow Path Input for Rees and Haves Model

Typical name (user selectable)	Parameter 2 Path Type	Upwind Node name's type	Downwind Node name's type	Capacity "Rules" Flow rate factors
Cs	0	0	1	1.0
Cfp	1	1	2	0.15
Cp1	2	2	3	0.65
Cp2	3	3	4	1.15
Cp3	4	4	5	1.3
Cp4	5	5	9	1.3
CR4	6	9	8	0.3
Ce3	7	8	4	0.15
CR3	8	8	7	0.15
Ce2	9	7	3	0.5
CR2	10	7	6	0.35
Ce1	11	6	2	0.5
CR1	12	1	6	0.85
Cse	13	9	10	1.0

As an example, the complete set of "nodal flow path" input objects for using the "rules given by Rees and Haves (2001) for displacement ventilation are,

```
NODAL FLOW PATH, Cs,  0,  Inlet, floor,     1.0;
NODAL FLOW PATH, Cfp, 1,  Floor, Plume1,   0.15;
NODAL FLOW PATH, Cp1, 2, Plume1, Plume2,   0.65;
NODAL FLOW PATH, Cp2, 3, Plume2, Plume3,   1.15;
NODAL FLOW PATH, Cp3, 4, Plume3, Plume4,    1.3;
NODAL FLOW PATH, Cp4, 5, Plume4, Room4,     1.3;
NODAL FLOW PATH, CR4, 6,  Room4, Room3,     0.3;
NODAL FLOW PATH, Ce3, 7,  Room3, Plume3,   0.15;
NODAL FLOW PATH, CR3, 8,  Room3, Room2,    0.15;
NODAL FLOW PATH, Ce2, 9,  Room2, Plume2,    0.5;
NODAL FLOW PATH, CR2,10,  Room2, Room1,    0.35;
NODAL FLOW PATH, Ce1,11,  Room1, Plume1,    0.5;
NODAL FLOW PATH, CR1,12,  Floor, Room1,    0.85;
NODAL FLOW PATH, CsE,13,  Room4, Outlet,    1.0;
```

The internal loads to account for equipment, people, and lights can be specified separately but since they are all recombined for use in the model only one distribution object, `Equipment Distribution:heatgains,` should be specified. The internal loads are all lumped together and split between nodes 2 and 3 per the fractions in the distribution.

B.4.2 Execution

The model is run by executing the program *AirModelLoads.exe* located in the directory,

```
\AirModelToolkit\SamplePrograms\AirModelLoads\debug\AirModelLoads.exe
```

The data dictionary file, *toolkittest.idd*, and the user input file, *in.idf*, also need to be located in the directory.

Alternatively, stand-alone modeling can be accomplished by running the program, *ReesHaves_IMSL.exe* located in the directory,

```
\AirModelToolkit\components\ReesHavesNodal\ReesHaves_IMSL\debug\
```

B.4.3 Output

Running the Rees and Haves model will also produce an output file. For coupled air and surface models this file will get overwritten and so it is just the a report form the last time the model was called. The file is called *ReesHavesModel_.out* and its organization is explained in Table B.8. While most of the air data will be reported to the air data manager and also available in the file *NodeData_.out* this file directly from the nodal model has useful information about the surfaces in stand-alone operation and shows how nodes are connected to each other and how surfaces are connected to nodes.

Table B.8 File data in ReesHavesModel_.out file reported from ReesHaves Model

Record	1st value in record	2nd value in record	3rd value in record	4th value in record
1	*N*, Num of unknown Nodes	*M*, Number of paths (edges) in network	*L*, Number of surfaces	
2	T_{supply}	$\dot{m}$	*W*	
3	TstatDB	Height of Thermostat		
4	Total Wall convection	Qsys Air system mcDT		
5 to 13 (N)	Node height	Node Temp. [K]	$\dot{m}$, magnitude [kg/s]	
5+N to 5+N +M	Upstream Node ID	Downstream Node ID	Massflow factor,	Enthalpy flux [W]
6+N+M to 6+N+M+L	Surface ID	Adjacent Air Node ID	Surface Temp. [K] (input echo)	Surface Convection0 [W]

B.5 Inard Pressure-Zonal Model

The toolkit implementation of the Inard model does not work properly and should not be used.

B.6 POMA Pressure-Zonal Model

The POMA model is available but probably only useful for stand-alone air modeling. The model works but has problems fully converging.

B.6.1 Input

For stand-alone model POMA needs the input objects listed in Table B.6.1

```
SURFACE SETTINGS, (12)
ZONAL CELL CONFIG ,  (10)
DISPLACEMENT LIST, (2..60)
ZONAL INLET,   (16)
ZONAL OUTLET,  (12)
ZONE CONTENT BLOCK, (13)
REFERENCE PRESSURE,   (4)
POMA INLET,     (12)
POMA OUTLET,   (4)
```

Here is an example input file for a stand-alone POMA run,

```
ZONAL CELL CONFIG, MiniBat chamber, 3.1, 3.1, 2.5, !
               5, 5, 6, ! number of cells in X Y Z [integer]
               DXs, DYs, DZs; ! names of Displacement List objects
DISPLACEMENT LIST, DXs, 0.200000 ,    0.900000 ,    0.900000  ,
               0.900000 ,    0.200000;
DISPLACEMENT LIST, DYs,    0.200000,     0.900000,     0.900000 ,
               0.900000 ,    0.200000;
DISPLACEMENT LIST, DZs  ,   0.200000,     0.525000   ,  0.525000  ,
               0.525000    , 0.525000  ,   0.200000;
SURFACE SETTINGS, Southwall, 1, 3, 1, 5, 1, 1, 1, 6  290.05, 4.0,;

SURFACE SETTINGS, Northwall, 1 , 4, 1, 5, 5, 5, 1, 6, 306.15, 4.0,;

SURFACE SETTINGS, eastwall,  1, 2,  5, 5, 1, 5, 1, 6, 300.05, 4.0,;

SURFACE SETTINGS, westwall,  1, 1,  1, 1, 1, 5, 1, 6, 300.48, 4.0,;

SURFACE SETTINGS, ceiling,   1, 6,  1, 5, 1, 5, 6, 6, 301.65, 4.0,;

SURFACE SETTINGS, floor,     1, 5,  1, 5, 1, 5, 1,  1, 299.05, 4.0,;

Reference Pressure,101230.0, 1, 1, 1;
THermostat Location, 3,3,3,;
```

B.6.2 Execution

Stand-alone modeling uses a different program called *PomaPressureZonal.exe* located in the directory,

```
\AirModelToolkit\components\POMApressurezonal\debug\
```

The data dictionary file, *toolkittest.idd*, and the user input file, *in.idf*, also need to be located in the directory.

The coupled model performs poorly but is functioning in the program *AirModelLoads.exe*.

B.6.3 Output

The data are located in output files *ZonalData_.out* and *output_room.txt*.

B.7 Momentum-Zonal Model

The momentum-zonal model is provided as a working coarse-grid 3-D flow program. This program simulates room airflow with inlets, outlets, blockages, heat sources, walls, etc. It is often helpful to test zonal model input in this stand-alone program before running as a coupled case in order to explore convergence and relaxation parameters. The model is not intended for forced air systems.

B.7.1 Input

Table B.9 lists the additional input objects needed to run a model using the momentum-zonal program.

Table B.9 Input objects for momentum-zonal model component

Input objects	notes
ZONAL CELL CONFIG	domain size and grid resolution
DISPLACEMENT LIST	array of cell sizes
SURFACE SETTINGS	Use one for each 'SURFACE'
MOMENTUM-ZONAL CONTROLS	program controls and paramters
THERMOSTAT LOCATION	about 1.1 m. above floor
REFERENCE PRESSURE	location in grid
ZONE CONTNET BLOCK	optional, heat sources, blockages
ZONAL INLET	optional, use with air system
ZONAL OUTLET	optional, use with air system

Dealing with the input of values for "under-relaxation" and "false-time step" factors is going to remain a significant aspect of working with the zonal momentum-model. This is a consequence of the numerical method used to iteratively solve what are non-linear relations. These parameters are always going to be grid and problem dependent and some amount of testing and trial-and-error is to be expected. The following table lists some guidelines to use to get started. This is an area where it simply helps to have experience. It is recommended that a new airflow problem be run in stand-alone mode to manually test different relaxation factors before running coupled to the loads calculations. Running a mixing model beforehand can provide temperatures and conventional film coefficients to use in a stand-alone model while tuning relaxation parameters. The user may have to go through a process of trial and error to find combinations that produce rapid convergence. This is one of the drawbacks of the numerical method and is a long-standing issue that is not likely to be resolved. Record factors and iterate to find combinations that produce convergence quickly (<1000 iterations). While there are a total of nine factors typically there would be only three unique values. All the false time steps usually have the same value. The velocity and temperature under relaxation factors also usually have the same value. The pressure under relaxation factor has its own value but it does not tend to need much variation. So really the process involves find a combination of two values that work well, one for relaxation and another for false time step. The term "false time step" comes from the fact that the factor must have units of time for terms to agree and that the terms look like a transient contribution to the balance equations. The convergence criteria is adjustable (e.g. 0.01) and represents a normalized total mass residual.

Table B.10 Under Relaxation and False-Time Step Factors

Factor	Variable in code	absolute range	practical range
U dir. Relaxation	URFU	[0, 1]	(0.02 – 0.5)
V dir. Relaxation	URFV	[0, 1]	(0.02 – 0.5)
W dir. Relaxation	URFW	[0, 1]	(0.02 – 0.5)
Pressure Relaxation	URFP	[0,1]	(0.4-0.8)
Temperature	URFT	[0, 1]	(0.02 – 0.5)
U dir. False Time Step	DTU	[1e-10, 1e10]	(1e-1, 1e10)
V dir. False Time Step	DTV	[1e-10, 1e10]	(1e-1, 1e10)
W dir. False Time Step	DTW	[1e-10, 1e10]	(1e-1, 1e10)
Temperature False Time Step	DTT	[1e-10, 1e10]	(1e-1, 1e10)

B.7.2 Execution

The model is run by executing the program *AirModelLoads.exe* located in the directory,

```
\AirModelToolkit\SamplePrograms\AirModelLoads\debug\AirModelLoads.exe
```

The data dictionary file, *toolkittest.idd*, and the user input file, *in.idf*, also need to be located in the directory.

Stand-alone modeling uses a different program called *ZonalMomentum.exe* located in the directory,

```
\AirModelToolkit\components\momentumZonal\debug\momentumzonal.exe
```

B.7.3 Output

The data are reported in the files *toolkit.out* (see Table A.3) and *ZonalData_.out* (see Table A.5). There are other files created by the model itself rather than the framework for coupling (see section A.7), *result.dat*, *balance.dat*, *Tecp___x.dat*.

B.8 Input Object Reference

This section is an encyclopedic reference on the input objects developed for the Air Model Toolkit. The same methods as the Loads Toolkit and the program EnergyPlus are used. New input objects have been added to the data dictionary, *ToolkitTest.idd*. For more information on the idd/idf input process see Section A.1.2 in Appendix A of this report as well as the Loads Toolkit documentation, and EnergyPlus documentation (InputOutputReference.pdf). In order to run models of complete zones it is necessary to include input objects from the Loads Toolkit as well as those documented here. These new objects are listed here with the number of comma-separated data fields that each uses.

```
SELECT AIR MODEL, (7*#days)
TOOLKIT OUTPUT, (1)
SURFACE SETTINGS, (12)
AIR NODE,        (9)
SURFACE LIST, (8+)
NODAL FLOW PATH, (5)
OUTSIDE SURFACE BC:NOD, (2)
ZONAL CELL CONFIG ,  (10)
DISPLACEMENT LIST, (2..60)
ZONAL INLET,    (16)
ZONAL OUTLET,  (12)
ZONE CONTENT BLOCK, (13)
MOMENTUM-ZONAL CONTROLS, (42)
REFERENCE PRESSURE,    (4)
THERMOSTAT LOCATION, (4)
POMA TEMP GUESS,  (5)
POMA INLET,      (12)
POMA OUTLET,    (4)
ZONAL WALL PLUME BLOCK, (12)
LIGHTING DISTRIBUTION:HEATGAINS, (3..21+)
EQUIPMENT DISTRIBUTION:HEATGAINS, (3..21+)
PEOPLE DISTRIBUTION:HEATGAINS, (3..21+)
```

The remainder of this section provides details on these input object. The input object descriptions are arranged alphabetically. The text file *ToolkitTest.idd* is an important aid to generating input and should also be examined by anyone developing input files "by hand." The input object reference descriptions in this section count parameters by comma locations whereas the .idd file distinguishes between character (e.g. A1, A2, A3,..) and numeric data (e.g. N1, N2, ..).

'AIR NODE'

This input object defines nodes in a nodal model. Zonal models do not use these objects. The presence of these objects in the input file determines the number of elements in the `AirNode` structure in the `AirDataManager`. The nodal model selected will dictate the number of air node objects to include in the input file. For the Mundt model, use 6 to 10 or more nodes depending on the desired sub-surfacing of vertical walls. Here is an

example of the 'Air Node' object definitions for using the Mundt linear model with vertical walls divided into four vertical bands.

```
Air Node, Inlet,         0, 0, 0, 0.4 , , 0, none,         290.37;
Air Node, Floorair,      1, 0, 0, 0.1 , , 3, Floorlist,    297.15;
AIR NODE, Thermostat,    2, 0, 0, 1.1 , , 0, none,         297.15;
AIR NODE, Return,       10, 0, 0, 2.5 , , 0, none,         297.15;
AIR NODE, ceilingair,    3, 0, 0, 2.64, , 3, CeilingList, 297.15;
AIR NODE, Room1,         4, 0, 0, 0.34, , 4, wallsOne,     297.15;
AIR NODE, Room2,         4, 0, 0, 1.03, , 4, wallsTwo,     297.15;
AIR NODE, Room3,         4, 0, 0, 1.72, , 4, wallsThree,   297.15;
AIR NODE, Room4,         4, 0, 0, 2.40, , 4, wallsFour,    297.15;
```

For the Rees and Haves model, use 11 nodes (see section B.4.1). Here is an example of the Air Node object definitions for using the Rees and Haves model.

```
AIR NODE, Inlet, 0, 0, 0,      0.4,       , 0,    none,   291.15;
AIR NODE, Floor, 1, 0, 0,      0.1,       , 1,   Floor,   297.15;
AIR NODE, Plume1,2, 0, 0,   0.6875, load 1, 0,    none,   297.15;
AIR NODE, Plume2,3, 0, 0,    1.375, load 2, 0,    none,   297.15;
AIR NODE, Plume3,4, 0, 0,   2.0625,       , 0,    none,   297.15;
AIR NODE, Plume4,5, 0, 0,     2.65,       , 1,    Roof,   297.15;
AIR NODE, Room1, 6, 0, 0,  0.34375,       , 4,wallsOne,   297.15;
AIR NODE, Room2, 7, 0, 0,  1.03125,       , 4,wallsTwo,   297.15;
AIR NODE, Room3, 8, 0, 0,  1.71875,       , 4,wallsThree, 297.15;
AIR NODE, Room4, 9, 0, 0,  2.40625,       , 4,wallsFour,  297.15;
AIR NODE, Outlet,10,0, 0,  2.40625,       , 0,   none,    297.15;
```

Parameters 1 is a unique name for the air node.

Parameter 2 sets integer type codes, that are discussed in sections B.3 and B.4. Generally the codes used in the above examples should be used.

Parameters 3, 4, and 5 can be used to "position" the node inside the zone in meters. The origin is the lower, south-west corner. However, for nodal models there is not really a fixed geometry to the control volumes with a clearly defined control volume center point. The Z coordinate, in meters, is used in the displacement ventilation models for interpolating between nodes in order to produce a result for the drybulb temperature at the thermostat height. The X and Y coordinates are not (yet) used.

Parameter 6 identifies a zone content block that will be associated with the node. The internal loads passed to the air model will be added to the node. The Rees and Haves model is set up for this and only for nodes with type code 2 or 3. No such modeling is done for the Mundt model.

Parameter 7 is the integer number of surfaces associated with the node.

Parameter 8 is the name of an input object. If parameter 7 is "1" then this should be the name of a single surface. If more than one '`surface`' is involved then this parameter should be the name of a '`Surface List Object`'.

Parameter 9 is a temperature in Kelvin used to initialize the node. This is not critical but a reasonable value for indoor air should be used.

'DISPLACEMENT LIST'

This input object defines an array of cell sizes for describing a structured grid. This object is used with zonal models and not with nodal models. The object simply consists of a name followed by a list of the sizes of the cells in a given direction. Section B.2 discusses zone gridding in more detail. The sum of the cell sizes should equal the overall zone dimension declared in the 'Zonal Cell Config' object. The current limit is 60 cells in any direction. The beginning and ending entries will be along the air domain boundaries and it is suggested that the size of these cell be "standardized" at 0.2 m. The following example is for a grid number of 10 with an overall length of 8.0 m.

```
DISPLACEMENT LIST, DXs,0.20 , 0.95, 0.95, 0.95, 0.95, 0.95, 0.95,
      0.95, 0.95, 0.20;
```

'EQUIPMENT DISTRIBUTION:HEATGAINS'

This input object is used in the momentum-zonal model to redistribute loads from equipment inside a thermal zone. The overall loads from equipment are set using the Loads Toolkit input objects 'Equipment' and 'EquipmentSplits'. In the event that a zonal model represents internal contents as separate distributed objects, this object can be used to distribute the loads within the space. The program uses the distributions defined in this object to redistribute the current total equipment convection loads amongst 'Zone Content Block' input objects. A distribution is defined in this object and the sum of the real numbers should be 1.0. The first parameter is the name of the distribution input object. After the first parameter, list pairs of parameters one for each zone content block. If you run out of room you can edit the ToolkitTest.idd file and add more pairs of block names and distribution real numbers. The following example spreads equipment convection evenly across 7 content blocks named PC1 through PC7.

```
Equipment Distribution:heatgains,  !
    Computers all on,
    PC1, 0.142857,
    PC2, 0.142857,
    PC3, 0.142857,
    PC4, 0.142857,
    PC5, 0.142857,
    PC6, 0.142857,
    PC7, 0.142857;
```

'LIGHTING DISTRIBUTION:HEATGAINS'

This input object is used in the momentum-zonal model to redistribute loads from lighting inside a thermal zone. The overall loads from lighting are set using the Loads Toolkit input objects 'Lighting' and 'LightingSplits'. In the event that a zonal model represents internal contents as separate distributed objects, this object can be used

to distribute the loads within the space. The program uses the distributions defined in this object to redistribute the current total lighting convection loads amongst 'Zone Content Block' input objects. A distribution is defined in this object and the sum of the real numbers should be 1.0. The first parameter is the name of the distribution input object. After the first parameter, list pairs of parameters one for each zone content block. If you run out of room you can edit the *ToolkitTest.idd* file and add more pairs of block names and distribution real numbers. The following example spreads equipment convection evenly across 4 content blocks named Lamp 1, Lamp 2, Lamp 3, and Lamp 4.

```
Lighting distribution:heatgains, !
    Lights uniform on,
    Lamp 1,  0.33333,
    Lamp 2,  0.33333,
    Lamp 3,  0.33334;
```

'MOMENTUM-ZONAL CONTROLS'

This object is used to pass input data to the momentum-zonal model. This is a fairly large input object with many parameters that control the momentum-zonal model. Many can be left alone, but some will need to be adjusted in order to adjust the relaxation behavior of the program. The input controls will be discussed by the line numbers indicated in this example,

```
MOMENTUM-ZONAL CONTROLS, ,
      0.0, 0.0, -9.81,! (1) gravity force vector
      1.189, 1005.6,  ! (2) density and specific heat
      295.15, 295.15, ! (3) air temp (K) initial , reference(Boussinesq)
      5, 5, 5,        ! (4) reference location grid index
      1, 1000, 0.01,  ! (5) restart , number of iter , relaxation criteria
     -4.0, 4.0, -4.0, 4.0, -4.0, 4.0, 273.15, 353.15,! (6) limits
      0.02,0.02, 0.02, 0.5, 0.02,  ! (7) linear under-relaxation factors
      5.0E0, 5.0E0, 5.0E0, 5.0E0, ! (8) false time step relaxation factors
      1, 10, 2,       ! (9) rinting controls for result
      1, 10, 5,       ! (10) printing controls
      50, 5, 5, 5, 1.0;! (11) residual report freq., monitor location, "Cd"
```

(1) These three parameters define the gravity force vector and should not need to be changed.

(2) These two parameter define the air density (kg/m3) and specific heat (J/kg·K).

(3) The two initialization values here should be reasonable indoor air temperatures in Kelvin. The first is used to initialize the cell air temperatures. The second is used for reference in the buoyancy calculation.

(4) The reference cell location in integer grid numbers for the temperature reference discussed in (3)

(5) The first parameter is the restart control. If set to 0 then no-restart; if set to 1 then restart. In the *AirModelLoads.exe* program, this will determine whether or not the program discards whatever current solution it might have and always goes back to an initial restart. Normal operation is that the current data from the `AirDataManager are` updated into the momentum-zonal working variables. These values may be just the last run from the model or they may be new if the time step has changed and the last result from the prior day is loaded. There is little reason other than testing to set the restart = 0. . (variable: `ICOMPU)` .

(5) The second parameter is the maximum number of iterations. Hopefully the simulation is exiting early, but if not, then this limits the number of iterations. Typical useful range for coarse grids is probably 200 to 2000, but it could go quite high if necessary. If your model is not converging after many thousands of iterations, try adjusting the relaxation parameters. But keep in mind there is no guarantee that the numerical method will always converge.

(5) The third parameter is the convergence criteria. This value will be compared to and error level computed in the program. A smaller criteria will cause the program to iterate more and run with more converged solutions. A larger criteria will cause the program to exit sooner and run with less well converged solutions. The value being compared is a normalized residual for mass. As the program cycles through its mass balance check, it sums all of the mass inconsistencies and then normalizes them based on the inlet mass flow. The criterion used most often is 0.01. The useful range of this value is probably 0.1 to 0.001. This value can be left unchanged or increased to shorten computing time or decreased to improve "accuracy."

(6) Limits on variables are used to constrain flow field values that may (temporarily) try to diverge. The example shows restricting interim computed velocities to within ± 4 m/s. The first four parameters are x-direction minimum, x-direction maximum, y-direction minimum, y-direction maximum, z-direction minimum, z-direction maximum. The last two parameters are bounds on air temperatures in Kelvin. The example values should suffice. The model is not really intended for higher velocity flows.

(7) These five parameters are important parameters controlling the numerical iterations. The first three parameters are the linear (under-) relaxation factors for the u, v, and w velocity components. These parameters are always going to be grid and problem dependent and some amount of testing and trial-and-error is to be expected in developing a set of values that exhibit rapid convergence. Table B.10 lists some guidelines to use to get started. This is an area where it simply helps to have experience. It is recommended that a new airflow problem be run in stand-alone mode to manually test different relaxation factors before running coupled to the loads calculations. Trial and error may be necessary to find combinations that produce convergence quickly (<1000 iterations). While there are a total of five factors typically there would be only two unique values. The velocity and temperature under relaxation factors can usually have the same value. The pressure under relaxation factor has its own value but it does not tend to need much

variation. So mostly the process involves finding a combination of two values that work well, one for relaxation and another for false time step. The linear relaxation factors are between 0.0 and 1.0.

Table B.10 Linear under relaxation factors

Factor	variable in code	recommend	practical range
U dir. Relaxation	URFU	0.03	(0.02 – 0.5)
V dir. Relaxation	URFV	0.03	(0.02 – 0.5)
W dir. Relaxation	URFW	0.03	(0.02 – 0.5)
Pressure Relaxation	URFP	0.5	(0.4 – 0.8)
Temperature	URFT	0.03	(0.02 – 0.5)

(8) These four parameters set important controls in the model that are another type of numerical weighting factor called "false-timestep." This is something of an inertia term that demonstrates good behavior with buoyancy driven flows. The use of these factors does not imply transient analysis or actual time steps. Trial and error should be used to adjust these factors (and the linear factors) to obtain a quickly converging model before running coupled models. The allowable range is from 10^{-10} to 10^{10}. When adjusting values make changes of about one-half an order of magnitude.

Table B.11 false-timestep factors

Factor	Variable in code	recommend	practical range
U dir. False Time Step	DTU	5.0	(0.01, 50)
V dir. False Time Step	DTV	5.0	(0.01, 50)
W dir. False Time Step	DTW	5.0	(0.01, 50)
Temperature False Time Step	DTT	5.0	(0.01, 10)

(9) and (10) These six parameters control printing to the results.dat file. This file is created but frequently overwritten for loads calculation. The data are also reported to the central AirDataManager and written to the *ZonalData_.out*.

(11) Monitoring controls determine what is reported to the monitor (or display device) during program execution. The first parameter is the frequency of reporting in terms of iterations. The next three parameters select a grid location for reporting computed variables. The last parameter here is an experimental "discharge coefficient" that should be set to 1.0 until its behavior is determined. The factor may be used in the future to modify pressure terms inspired by the empirical value (Cd = 0.83) used in some pressure-zonal model.

‘MUNDT MODEL CONTROLS’

This object is used to pass input data for the Mundt model. Currently the only extra parameters needed are for indicating “floor splits”. When using the Mundt model, one of these objects should be present and would typically be,

```
Mundt Model Controls, 0.1, 0.0;
```

This allows the user to model a portion of the internal loads as being near (≈0.1 m) the floor. The parameters can be a real numbers on [0..1]. The first parameter indicates what portion of the internal source convection split is actually released into the air near the floor. Similarly the second parameter can indicate what portion of the infiltration air might enter near the floor. These floor splits are only implemented for the Mundt model.

‘NODAL FLOW PATH’

This object is used to specify the relative air flow rates along the paths connecting nodes in a nodal model. The current implementations require that nodes be configured in just one specific way corresponding to the network of air nodes in the Rees and Haves nodal “model B.” If a toolkit user wishes to alter the network configuration (by altering code in the balance function) these input field should still be useful. But for now the list objects should be similar to the following example,

```
NODAL FLOW PATH, Cs,  0,  Inlet, floor,     1.0;
NODAL FLOW PATH, Cfp, 1,  Floor, Plume1,   0.15;
NODAL FLOW PATH, Cp1, 2, Plume1, Plume2,   0.65;
NODAL FLOW PATH, Cp2, 3, Plume2, Plume3,   1.15;
NODAL FLOW PATH, Cp3, 4, Plume3, Plume4,    1.3;
NODAL FLOW PATH, Cp4, 5, Plume4, Room4,     1.3;
NODAL FLOW PATH, CR4, 6,  Room4, Room3,     0.3;
NODAL FLOW PATH, Ce3, 7,  Room3, Plume3,   0.15;
NODAL FLOW PATH, CR3, 8,  Room3, Room2,    0.15;
NODAL FLOW PATH, Ce2, 9,  Room2, Plume2,    0.5;
NODAL FLOW PATH, CR2,10,  Room2, Room1,    0.35;
NODAL FLOW PATH, Ce1,11,  Room1, Plume1,    0.5;
NODAL FLOW PATH, CR1,12,  Floor, Room1,    0.85;
NODAL FLOW PATH, CsE,13,  Room4, Outlet,    1.0;
```

The first parameter is a user selected name and is not yet used inside the code. The second parameter identifies the path and is very important that all 14 entries appear with the integer codes shown in the example above. The third parameter is the name of an Air Node object that is upwind. The fourth parameter is the name of an Air Node object that is downwind. The program is not currently setup to work out an arbitrary network; the network defined has to follow that that of the example above. The fifth parameter is very important as it describes the movement of air between the individual node heat balances. The user can vary these flow rate factors and alter the behavior of the nodal model. However, the set of values *must* be consistent in that flows in and out of a node must agree with conservation of mass. The user should verify that the sum of flows into a node equals the sum of all flows leaving that node. The values are non-dimensional

factors that are eventually applied to the air system flow rate (1.0). A value more than one is allowed and indicates entrainment or other additional overall mixing is being modeled. In this model the there are three plume-only nodes that are not used in (or passed to) the loads calculation. The four room air nodes provide parts of the set of T_{ai}'s.

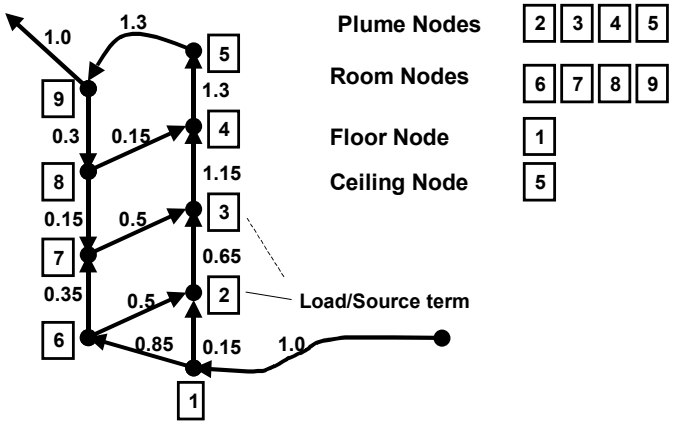


Figure B.4 Rees and Haves model "rules" for flow rates

'OUTSIDE SURFACE BC:NOD'

This object is used for each surface that is designated to have "NOD" outside boundary condition. The first parameter is the name of a 'surface' object with the "NOD" boundary condition and the second parameter is the name of a 'surface' object that will provide the boundary condition. It is provides "other side" outside air temperature boundary conditions. The surface can be self-referencing. This object allows adding modeling detail to interior partitions. This is used to set the other side of floor to the ceiling or give a lower wall surface the same lower air temperature on its other side. Such modeling allows taking advantage of symmetry when analyzing just one thermal zone in a building that is intended to be representative of all the zones with the same façade orientation and cooling system configuration.

'PEOPLE DISTRIBUTION:HEATGAINS'

This input object is used in the momentum-zonal model to redistribute loads from people inside a thermal zone. The overall loads from people are set using the Loads Toolkit input objects 'People' and 'PeopleSplits.' In the event that a zonal model represents internal contents as separate distributed objects, this object can be used to distribute the loads within the space Although people are usually integer values, the current code uses the distributions defined in this object to redistribute the current total convection loads amongst 'Zone Content Block' input objects. A distribution is defined in this object and the sum of the real numbers should be 1.0. The first parameter is the name of the distribution input object. After the first parameter, list pairs of parameters one for each zone content block. If you run out of room you can edit the ToolkitTest.idd file and add more pairs of block names and distribution real numbers. The following example spread people convection evenly across 7 content blocks named `occp1` through `occp7`.

```
People Distribution:Heatgains, !
    Uniform partial people,
    occp1,  0.142857,
    occp2,  0.142857,
    occp3,  0.142857,
    occp4,  0.142857,
    occp5,  0.142857,
    occp6,  0.142857,
    occp7,  0.142857;
```

'POMA INLET'

This input object allows specify additional input information for inlets required in the jet model in POMA. A '`ZONAL INLET`' object is also needed and must be consistent with this object. Here is an example of how to define objects for an inlet for use with POMA.

```
ZONAL INLET, maininlet,   3,   3, 3, 1, 1, 2,
      2,0.5,0.0,0.2053,0.0,0.041393,291.150,0.08,;
POMA INLET, maininlet, 3,1,2,y,20.0,0.5,0.9,0.587500
      ,1.26667 ,291.15,non;
```

The following discussion if for parameters in the POMA object.

The first parameter is a unique object name and should probably be the same as the 'Zonal Inlet' object.

Parameters 2, 3, and 4 declare the location as grid indices in the X-, Y-, and Z-directions. Inlets in POMA should span only one cell at a time.

Parameter 5 declares the direction. This should be set to either "x", "y", or "z". Use only one character. The program figures out which direction the jet must be direction along the axis specified.

Parameter 6 is the angle of the jet in degrees.

Parameter 7 is the velocity decay coefficient for the jet model.

Parameter 8 is the initial thickness of the jet in meters.

Parameter 9 is the initial width of the jet in meters.

Parameter 10 is the supply air temperature in Kelvin

Parameter 11 is a control for the whether or not the jet is isothermal. It should be set to either "iso" or "non".

‘POMA Outlet’

This input object allows specify the output location for POMA. A ‘ZONAL OUTLET’ object is also needed and must be consistent with this object. Outlets should be along domain boundary rather than in the interior of the air.

The first parameter is an object name that should probably be the same as the ‘Zonal outlet’ object.

Parameters 2, 3, and 4 declare the location as grid indices in the X-, Y-, and Z-directions. Outlets in POMA should span only one cell at a time.

‘POMA TEMP GUESS’

This object is used with the stand-alone Air Model Toolkit implementation of POMA, but has been phased out. The code can be changed to use this object, but we have moved in the direction of using routines that attempt to generate a reasonable guess from the surface temperatures. However, this object still exists and was used to set a temperature field to use as an initial guess. The object works by specifying the end points of an x-direction swath of cells. The temperature is then spread linearly across the cells. A separate input object is used for each (NY x NZ) possible X-direction swath through the cells. The first parameter is the object name and is not used in the code. The second parameter is an integer Y-direction grid number and the third parameter is the Z-direction grid number. These identify a set of X-direction cells that all have the same Y and Z cell index. The fourth and fifth parameters are the temperatures (in Kelvin) of the endpoints. Generally this temperature is somewhat close, but not too close to the wall temperature. It can be a difficult and important task when running POMA to find a set of temperature guesses that are successful.

‘REFERENCE PRESSURE’

This object is used to set the reference pressure for the zonal models. The object is not needed for nodal models nor the mixing models. It is necessary to describe where the reference pressure is to be located within the zone. For POMA the reference pressure must be in a specific place (grid location 1,1,1). For the momentum-zonal model this location can be the same as the thermostat location. For the Inard pressure-zonal model the location cannot be within a special cell block and should be one of the drift cells.

Parameter 1 is the absolute pressure in Pascals [N/m2] which should probably (but doesn’t need to) match the pressure value in ‘Environment’ object.

Parameters 2, 3, and 4 are the X , Y and Z grid index numbers identifying the cell location that is to be the reference. Pressures will be calculated relative to this reference.

‘SELECT AIR MODEL’

This object controls air model use when coupling them to the loads calculations in the program *AirModelLoads.exe*. If this object is not specified, the program runs the conventional complete mixing model. The object has seven parameters that are repeated the same number of times as the number of days. Table B.x provides a summary of the input parameters that are arranged as 7-tuples. Here is an example for running the Rees and Haves model.

```
select air model,    !
   1,  ,  ,  , , , , ! day 1 mixing warm up
   1,  ,  ,  , , , , ! day 2 mixing warm up
   3, 2, 2,1 , ,1,1, ! day 3 , Rees Haves DT-couple
   3, 2, 2,1 , ,1,1, ! day 4 , Rees Haves DT-couple
   3, 2, 2,1 , ,1,1, ! day 5 , Rees Haves DT-couple
   3, 2, 2,1 , ,1,1, ! day 6 , Rees Haves DT-couple
   3, 2, 2,1 , ,1,1, ! day 7 , Rees Haves DT-couple
   3, 2, 2,1 , ,1,1; ! day 7 , Rees Haves DT-couple
```

This input object allows fine control of when and how the coupled air and surface model are to work together. The seven input parameters from a set of 7 parameters, or a 7-tuple, that is to be repeated so that the number of 7-tuples in this input object equals the number of days in the simulation (declared in the object `SuccessiveSubstitutionData`). The individual parameters are discussed next, but keep in mind they are repeated.

Parameter 1 selects the air model to use for that day. For a building with conventional levels of thermal mass, a typical design day calculation repeats the same weather conditions for eight days in a row. This is used to find a steady-periodic solution. It is desirable to select simpler models and/or looser control and convergence criteria early in the design-day run for improved computational efficiency (and even success in some cases). Most test cases run for this research used two days of mixing model (parameter 1 = 1) at the beginning followed by 6 days using the desired air model.

Parameter 2 selects between two different coupling schemes T-couple and DT-couple that determine how the air temperature results from the air model are used by surface model. T-couple is selected by setting to “1” and DT-couple is selected by setting to “2”. T-couple uses the actual temperature values from the air model as the effective air temperatures regardless of their relation to the control temperature. DT-couple uses the relative *distribution* of air temperatures predicted by the air model and applies the relative values to the room control set point. The relative method amounts to a shift in the values by the current deviation in thermostat control. DT-coupling is more stable and efficient and has been most often. T-coupling may show control problems if the model does a poor job of predicting temperature at the thermostat.

Table B.12 Select Air Model – Parameters Description

Parameter	Value		Comments
1 Air Model	1	Complete Mixing	
	2	Mundt Nodal	
	3	Rees and Haves Nodal	
	4	POMA Pressure-Zonal	problems
	5	Inard Pressure-Zonal	Not implemented
	6	Momentum-Zonal	
2 coupling scheme	1	T-couple : direct air model	Instable some Air Models
	2	DT-couple: relative air model	More reliable
3 convection film coef. scheme	1	conventional	ASHRAE,TARP. Etc selected at compile time
	2	Air Model's Hc's	using to pass Hc from 'Surface Settings' to loads
4 air loop control	1	conventional	More stable, ignores comfort
	2	secondary air loop	Requires Air Model to predict Tstat well
	3	in-zone convective heater	no air system
5 control band	[ºC]	chooses tolerance on Tstat for secondary air loop	Testing 0.01ºC to 0.2ºC
6 control scheme	1	dry bulb at thermostat	
	2	comfort temperature	½(MRT + Tstat)
7 air data update scheme	1	Never use prior time step result	
	2	Use result from prior time step	

Parameter 3 selects between using conventional film coefficients (ASHRAE, BLAST, DOE2.0) for the inside face surface or values passed from AirDataManager to surface models. These values are currently input by the user, but future development may apply correlations designed to work with near-wall reference temperatures.

Parameter 4 chooses whether or not the secondary air-system flow rate control loop is used. A setting of "1" indicates no secondary air loop. A setting of "2" indicates to use a secondary air system iteration loop with thermostat control to find the air system flow rate. The control criteria is set in parameter 5. A setting of "3" indicates that this is a heating load calculation and that in-space convective equipment is used. This is not available with the displacement ventilation models. For the momentum model, use zone content blocks of equipment type to declare the heater position; in this case all equipment blocks are part of the heating system.

Parameter 5 chooses the control criterion for the air-system flow rate loop. This is ignored unless Parameter 4 is set to "2." This criterion affects the exit condition for a

secondary iteration loop running inside the usual day/time-step/iteration loops. A tight value (0.01ºC) will increase the number of calls to the air model and therefore make the overall run time increase. Too loose value (0.2ºC) could cause air system flow rates to oscillate between iterations (at the same time step) causing fluctuations in the load calculation (that could hurt convergence in the surface domain). The number of cycles executed is recorded and output and can be compared to choice of tolerance. For computation efficiency, on could use looser criteria for earlier days and tighten the criterion for the last day.

Parameter 6 chooses the control scheme for temperature referencing in adjusting the air system flow rate. The drybulb air temperature at the thermostat's location is the main mode and is selected by setting the parameter to "1". A "comfort temperature" is a combination of other comfort parameters rather than only the thermostat's drybulb. By setting this parameter to "2" the secondary air loop is controlled by a comfort temperature. Currently, the comfort temperature is implemented as a simple average of the mean radiant temperature and the thermostat drybulb.

Parameter 7 allows controlling how air domain results are made available to subsequent timesteps. A setting of "1" here sets the program to not do any shifting of results between time steps. A setting of "2" sets the program to shift data to the following timestep. These can be mixed so that say for the first day time step shifting occurs but not on the following days. The models generally attempt to use previous solutions to improve computational efficiency. One possibility is that at the end of an iteration loop for a given time step, the data for that time step are passed to the next. It is an open question for a load calculation whether it is better to use the data from the day before (but at the same time step) or to use the data from the current day but from the previous time step. Design-day loads calculation for could use either, while an annual energy calculation would use the later.

'SURFACE SETTINGS'

This input object provides additional information on the surfaces that is used by the air models. The presence of these input objects in the input file will determine the dimensions of the structure `SurfSet` when allocated by the `AirDataManager`. This provides a means of passing surface data between the air domain and the loads domain.

The first parameter is the name of the surface and should match the corresponding name of a 'SURFACE' object.

Parameter 2 selects the type of boundary condition that the surface provides to the air model. Currently only a value of 1 should be used for temperature boundary conditions since this is the coupling scheme that is primarily implemented in the sample program. The momentum-zonal model is setup to also allow heat flux boundary conditions internally and so a value of 2 can be used for stand-alone room air modeling using the momentum-zonal model.

Parameter 3 declares the wall orientation using an integer type code. Imagine sitting in the block of air cells adjacent to the wall and determine which direction to travel to reach the wall. So use orientation of 1 if wall is found by traveling in - X direction, 2 if in +X direction, 3 if in - Y direction, 4 if in +Y direction, 5 if in -Z direction, and 6 if in +Z direction.

Parameters 4 through 9 declare the portion of the grid of air cells that is to be associated with the surface. For nodal air models these fields are not used and can be left blank. The layer of air cells adjacent to the surface are the ones that interact with the surface producing an effective bulk air temperature for the surface model and experience surface convection heat transfer. The cells associated with a surface form a block that is identified by the beginning and ending grid numbers in each direction. Parameter 4 is the beginning grid index in the X direction and Parameter 5 is the ending grid index in the X direction. Similarly parameters 6 and 7 define the beginning and ending grid index in the Y direction and parameters 8 and 9 the Z direction. The block should be one-cell-thick so that for a surface of say orientation 1 would have its X indices begin and end at the same value (1 in this case) and would extend in the Y and Z directions. The first index in a pair should be less than or equal to the second.

Parameter 10 sets a surface temperature in Kelvin to use for that surface. In stand-alone air model this represents an important boundary conditions. For coupled air and loads modeling this value is only used for initialization and will be changed by the program as the simulation progress.

Parameter 11 sets a prescribed value for the surface convection heat transfer coefficient, h_c, in [W/m^2·K]. Future development may introduce an additional parameter for controlling choice of convection correlation so that temperature, length, and flow conditions be included in computing a continually updated value for hc.

Parameter 12 sets a prescribed value for a constant convective heat flux. This is not currently implemented for coupled air and surface models where temperature boundary condition are used, but it should work for stand-alone momentum-zonal model

‘SURFACE LIST’

This input object is used to specify a list of surfaces that interact with an air node via surface convective heat transfer. The first parameter is the list name and would also appear as parameter 8 in an Air Node object. The remaining parameters are the names of `‘surface’` and `‘surface settings’` objects to associate with the node.

‘THERMOSTAT LOCATION’

This input object defines where in the zone the thermostat is located. The first three parameters are integers that correspond to the X, Y, and Z direction grid index locations and are used with zonal models. The fourth parameter is the height in the zone at which

the thermostat is located (in [m]) for a nodal model. For nodal models the grid indexes are not used and for zonal models the height is not used.

'TOOLKIT OUTPUT'

Using this input object with its parameter value set at 2 creates a results file from the Heat Balance Simulation Manager. File has the name *toolkit.out*.

```
Toolkit OUTPUT, 2;
```

Other data files are created when running air models (that contain the air data) but the *toolkit.out* data can be compared to simulations that do not use any of the Air Models allowing comparisons to conventional simulations. A value 1 should produce original toolkit.out from the loads toolkit. The data format of in the file are described in Table A.3 and Section B.x.

'ZONAL CELL CONFIG'

This object provides the top-level of geometrical information for describing a grid for use with zonal models. Gridding a zone is also discussed in section 4.2.2. The first parameter is a name for the zone. Parameters 2, 3, and 4 describe the overall zone dimensions in the X, Y, and Z directions respective in units of meters. Parameters 5, 6, and 7 describe the number of cells in each direction. The last three Parameters 8, 9, and 10 refer to names of 'displacement list' objects that give the sizes of cells in the each of the three directions.

'ZONAL INLET'

This input object describes a single air flow inlet for a zonal model. Using this object with any parameters will allocate an additional element of the `inlet` array in *AirDataManager.f90*. The first parameter is a unique name (not really used), the second parameter is the direction integer code ([1..6]) representing the direction you would travel from the center of the adjacent air control volume(s) to find the associated inlet. The integer type codes are 1 = wall is found in -X dir, 2 = +X dir, 3 = -Y dir, 4 = +Y dir, 5 = -Z dir, and 6 = +Z dir. For the pressure zonal models the grid extension can be only 1 cell thick in all directions, but for finer grids a one-cell thick but 2-D block of cells can also be defined. Inlets in the Air Model Toolkit sample programs must be located along the outer domain of the air control volumes as along the surfaces of the boundary.

[[A bug warning on inlet directions: early versions of the momentum-zonal run poorly with negative direction inlets (directions 2, 4, and 6) different behavior may be observed for inlets located in surfaces that would be East, North, or the Ceiling. It is unfortunate but for now inlets should not be on these surfaces.]]

Parameters 3 through 8 declare the portion of the grid of air cells that is to be associated with the inlet. For nodal air models (types 1, 2, or 3) these fields are not used and can be

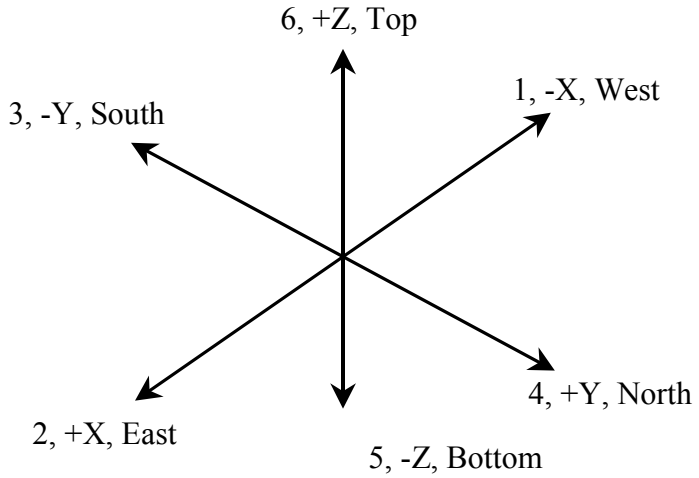
left blank. The layer of air cells in the block defined here should be one-cell thick. Each cell will get a uniform (by area) distribution of the outflow. The use of this object creates an important instance of the variables used to pass values for $T_{leaving}$ from the air models to the governing building simulation. Parameters 3 and 4 are the X-direction beginning and ending indices, 5 and 6 are the Y-direction begin and ending indices and 7 and 8 are Z-direction beginning and ending indices of the adjacent air control volumes associated with the inlet.

Parameter 9 is the effective area ratio for use in determining inlet flow velocity from total volume flow rate and actual area of the inlet cell block. The momentum technique allows accounting for increased velocity of a diffuser grill versus a completely clear opening. For example an effective area ration of 0.5 will set inlet velocity at twice that of 1.0 but still have the same mass flow rate.

Parameters 10, 11, and 12 indicate uniform velocity for the inlet for stand-alone operation. If used with coupled model, this is just an initialization value. A normal direction (as in perpendicular to inlet) is easily obtained and velocity computed from the inlet's volumetric flow and effective area. The value will be changed dynamically as for a VAV-style air sytem flow rate/load calculation.

Paramater 13 indicates the total mass flow for stand-alone operation. If used with coupled model, this is just an initialization value. The mass flow is generally adjusted to meet the load in typical design-day cooling load calculation.

Parameter 14 is the inlet drybulb temperature for stand-alone models. If used with coupled model, this is just an initialization value and the value will be set to the "Tdeck"



B.5 Coordinate system with integer direction codes

‘ZONAL OUTLET’

This input object describes a single air flow outlet. Using this object with any parameters will allocate an additional element of the `outlet` array *AirDataManager.f90*. The first parameter is a unique name (not really used), the second parameter is the direction integer code ([1..6]) representing the direction you would travel from the center of the adjacent air control volume(s) to find the associated with the outlet. The integer type codes are 1 = wall is found in -X dir, 2 = +X dir, 3 = -Y dir, 4 = +Y dir, 5 = -Z dir, and 6 = +Z dir. For the pressure zonal models the grid extension can be only 1 cell thick in all directions, but for finer grids a one-cell thick block of cells can also be defined. Outlets in the Air Model Toolkit sample programs must be located along the outer domain of the air control volumes as in a boundary.

Parameters 3 through 8 wall declare the portion of the grid of air cells that is to be associated with the outlet. For nodal air models (types 1, 2, or 3) these fields are not used and can be left blank. The layer of air cells in the block defined here should be one-cell thick. Each cell will get a uniform (by area) distribution of the outflow. The use of this object creates an important instance of the variables used to pass values for $T_{leaving}$ from the air models to the governing building simulation. Parameters 3 and 4 are the X-direction beginning and ending indices, 5 and 6 are the Y-direction begin and ending indices and 7 and 8 are Z-direction beginning and ending indicies of the adjacent air control volumes associated with the outlet.

Parameter 9 is a uniform “top-hat” velocity for the outlet. This is just an initialization feature, that is easily obtained from the inlet’s volumetric flow. The program will not maintain this flow rate but use of this object may help to get the numerics off to a good start. If used with coupled model the value here will get overwritten as determined from the flow at the start of the air model (typically) will be automatically updated to match the air system flow solution from “warmup days”

Parameter 10 is a return air temperature guess useful for stand-alone air modeling. Some models may, at times, run the outlet the wrong way during far from convergence iterations (or finite-difference Jacobian routines) and so the temperature is made available but this is generally used only for initialization and can usually be left at default. This object can be truncated (with a semi colon) after parameter 10 but parameters 11 and 12 are available for using with more detailed implementations. Some data structures have variables setup for working with humidity and pollution sources but these have not been implemented and there is little supporting code at this time.

‘ZONAL WALL PLUME BLOCK’

This input object is used in pressure-zonal model to declare a block of “special cells.” The flow in these special cells will be determined using a wall plume empirical law. It is used on “active” walls that would be expected to drive the flow. Blocks of cells should be one-cell-thick. Do not overlap blocks in corners. Special cell wall plume models are not implemented in POMA or Momentum-Zonal.

The first parameter is the name of this particular input object.

Parameters 2 and 3 are the X-direction beginning and ending indices, 4 and 5 are the Y-direction beginning and ending indices and 6 and 7 are Z-direction beginning and ending indices of the adjacent air control volumes associated with the wall.

Parameter 8 is the name of a 'SURFACE SETTING' object that provides association with a particular surface. The block of cells defined in this object should agree with the block of cells for the 'SURFACE SETTING' object. Future expansion may allow for multiple surface associations but for now a one-to-one association is needed.

'ZONE CONTENT BLOCK'

This input object is used to describe lights, equipment, and people that are modeled as the contents inside a thermal zone. This allows positioning internal sources of convection heat gain (or loss) and/or obstructions. Both nodal and zonal models use this input object. For nodal models the positioning information is not needed. The presence of these input objects in the input file will determine the dimensions of the structure `Content` when allocated by the `AirDataManager`. These structures are used to pass data for internal loads from the loads domain to the air model. Future expansion may allow associating these with internal mass wall constructions (loads domain), but for now no transient or thermal mass behavior can be modeled with these objects.

Parameter 1 is a unique name identifying this object. These are used in the code to cross reference with heat gain distributions and node associations.

Parameter 2 is an integer type code. A setting of "1" indicates the content is a blockage in the momentum-zonal program. A setting of "2" indicates the content is not a blockage. Non-blockage content block means that the airflow can pass right through it as the heat sources are distributed throughout the volume of the block. A blockage means that the flow cannot pass through and the load is "released" at the faces of the block. A setting of "2" is recommended for coarse grid models.

Parameter is 3 is not yet used.

Parameters 4 and 5 are the X-direction beginning and ending indices, 6 and 7 are the Y-direction beginning and ending indices and 8 and 9 are Z-direction beginning and ending indices of the air domain control volumes associated with the content block. These are integer grid index values. The block's geometry will "fill-out" the volume defined by the block of cells. For nodal models, these can be left empty. Values cannot exceed the maximum grid numbers defined in the `Zonal Cell Config` object.

Parameter 10 is the total convective heat load in Watts from internal loads associated with the block. The value here is only used for stand-alone testing of models, since these

load levels are determined by the loads calculations and continually updated and passed to the air model. The actual load during a coupled will be determined by the input ob

Parameter 11 defines the nature of the internal load where 1=lights, 2=equipment, 3= person. Using this parameter is optional since the information is also available from the various distributions (`Equipment Distribution:heatgains, Lighting distribution:heatgains, People Distribution:Heatgains`)