NAT'L INST OF STAND & TECH R.I.C.
A11104 190132

NIST
PUBLICATIONS

NISTIR 5322

Computer Programs for Simulation of Lighting/HVAC Interactions

George N. Walton

Building and Fire Research Laboratory
Gaithersburg, Maryland 20899

QC
100
.U56
#5322
1993

NIST

United States Department of Commerce
Technology Administration
National Institute of Standards and Technology

NISTIR 5322

Computer Programs for Simulation of Lighting/HVAC Interactions

George N. Walton

December 1993
Building and Fire Research Laboratory
National Institute of Standards and Technology
Gaithersburg, MD 20899

U.S. Department of Commerce
Ronald H. Brown, *Secretary*
Mary L. Good, *Under Secretary for Technology*
National Institute of Standards and Technology
Arati Prabhakar, *Director*

U.S. Department of Energy
Washington, D.C., 20858

ABSTRACT

This report describes two computer programs developed for the analysis of lighting/HVAC interactions: HLITE and VLITE. VLITE is used to compute the coefficients (view factors) describing radiation interchange between surfaces. These coefficients are used in a thermal network model which is solved by HLITE for transient temperatures and cooling loads. These programs are research tools. HLITE is based on a simple finite volume model for heat transfer combined with sufficiently short time steps to permit explicit time integration in most of the simulation. Controls are modeled in a manner emulating the operation of controls in real buildings. The accuracy of the mathematical solution is appropriate to the models used and data available resulting in a fast, flexible simulation tool. The solution time tends to be directly proportional to the number items in the thermal network and inversely proportional to the time step. The simple methods used in HLITE could provide a general approach to modeling building systems.

Key words:

Energy calculation, HVAC, lighting, modeling, transient simulation, view factor

ACKNOWLEDGEMENTS

This work was sponsored by the U.S. Department of Energy and the Electric Power Research Institute. The review comments of Ed Sowell are gratefully acknowledged. Dr. Sowell's program, LIGHTS, is an interesting alternative to HLITE using a significantly different numerical approach. Thanks to Gene David of Ross & Baruzzini, Inc. for using and testing the program and providing his suggestions for improvements. The help of Steve Treado and John Bean in transferring and interpreting the experimental data is appreciated.

DISCLAIMER

The programs described in this report are furnished by the government and are accepted and used by any recipient with the express understanding that the United States Government makes no warranty, expressed or implied, concerning the accuracy, completeness, reliability, usability, or suitability for any particular purpose of the information and data contained in this program or furnished in connection therewith, and the United States shall be under no liability whatsoever to any person by reason of any use made thereof. These programs belong to the government. Therefore, the recipient further agrees not to assert any proprietary rights therein or to represent these programs to anyone as other than government programs. Any commercial products or tradenames mentioned in this report are included for informational purposes only, and do not constitute an endorsement or recommendation by the National Institute of Standards and Technology, the Department of Energy, or the Electric Power Research Institute.

TABLE OF CONTENTS

APPENDIX B: HLITE Input and Output

APPENDIX C: HLITE Test Cases

1. INTRODUCTION

This report describes two computer programs developed in support of "research to determine energy transfer from lighting systems to building spaces and thermal equipment as influenced by typical operating conditions and equipment configurations, and to develop procedures to promote the design of efficient lighting and HVAC systems, leading to energy and cost savings" [Treado & Bean, 1988]. Figure 1 depicts the data and programs used to simulate the lighting/HVAC interaction.

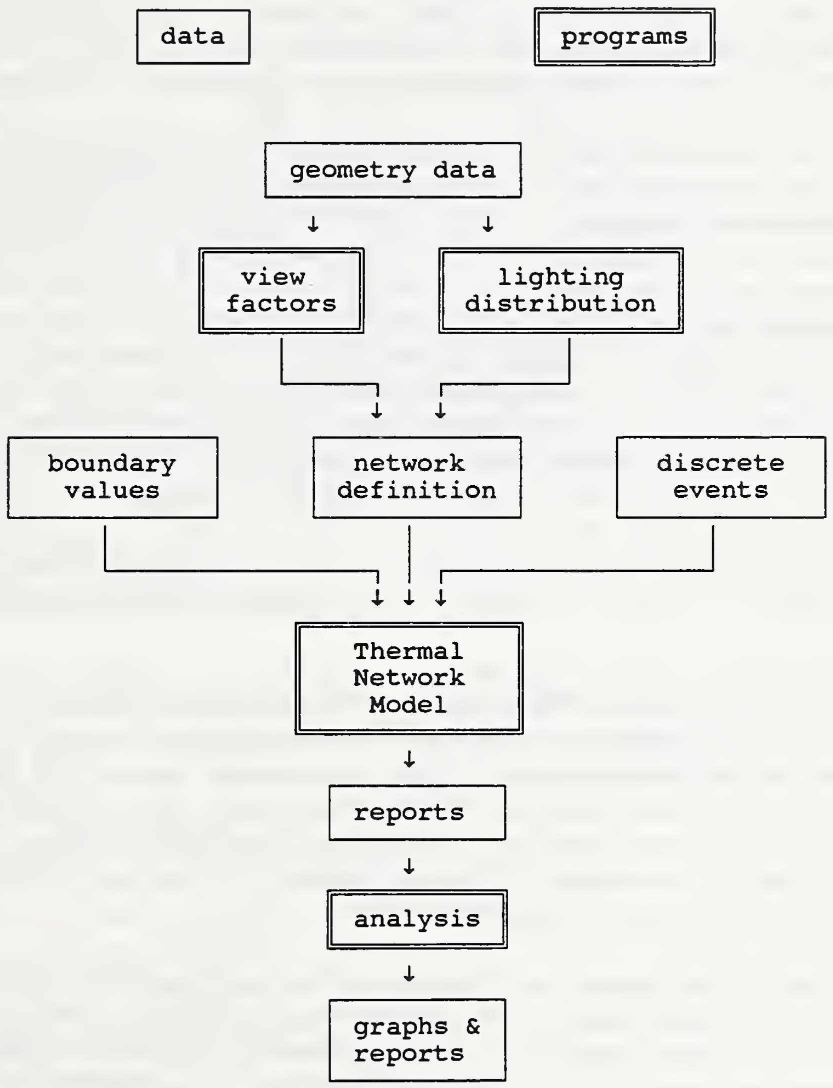


Figure 1. HVAC/Lighting Interaction Analysis

In general, this simulation involves determining the distribution of light onto the room surfaces, and then the transfer of heat by conduction, convection and radiation (using view factors) within the room and to the air conditioning system. The view factor calculation program is called VLITE; the thermal network model is called HLITE. The lighting distribution calculation and the analysis of the simulation must be done by commercially available programs.

These programs have been developed to extend the results of an experimental study in lighting/HVAC interaction being conducted at the National Institute of Standards and Technology (NIST). Reports on this project were presented by Treado and Bean, Sowell, and Rundquist at a 1990 ASHRAE symposium. They will be used in the development of design guidance and algorithms to be included in building energy simulation programs. These programs are research tools rather than engineering design tools. A primary goal of the computer model is simulation flexibility; that is, the ability to model a great variety of physical systems and operational strategies including many that were not anticipated during the development of the program. This includes the requirement of a simple fundamental program structure for the addition of new features. Another primary goal is that the simulation be accurate and sensitive to the parameters that the researcher wishes to study. These primary goals dictate that the program be based on fundamental physical principles rather than correlations to the experimental data. Good program performance in terms of execution time and program size as well as ease of use are secondary goals.

The simulation is centered around the HLITE program which uses a thermal network model to achieve the desired flexibility. In this model nodes representing various masses in the structure can be linked to each other by the different heat transfer processes. Expanding the model requires only the addition of new nodes or links to the existing structure which performs the transient heat transfer analysis. HLITE uses short time steps to model the transient performance of the lights and to handle the nonlinearities in the radiation and convective heat transfer models. HLITE uses three input files: the Network Definition File (NDF) is the description of the thermal network; the Boundary Values File (BVF) specifies the temperatures of certain nodes in the thermal network and sets the simulation start and stop times; the Discrete Events File (DEF) defines events such as switching lights on or off and changing setpoint temperatures which provides for flexible scheduling. These are all ASCII files which the user creates with a text editor. The contents of the output reports is defined by the user.

The capabilities of HLITE are described in section 2. The transient heat transfer theory used by HLITE is described in Appendix A, its input in Appendix B, and a number of test cases to validate the program in Appendix C.

The initial release of HLITE [Walton, 1990] called for view factors in the NDF but did not provide a way to calculate them. It was decided to require that these values be computed external to HLITE rather than provided by some over-simplified approximation in the program. The VLITE program has been developed to operate on the same computer as HLITE to provide accurately computed view factors in the form required by HLITE. The capabilities of VLITE are described in section 3. The theory for computing view factors is given in Appendix D; VLITE input is described in Appendix E; and a number of test cases to validate the program are in Appendix F.

HLITE also requires values for the fraction of light energy absorbed at each room surface. Again, rather than making some over-simplified approximation, HLITE requires that these values be determined externally. The calculation of these values in rooms with obstructing surfaces is non-trivial. Commercial lighting programs, such as Lumen-Micro (by Lighting Technologies Inc.), might be used for this purpose.

Section 4 gives examples of the use of these programs in simulating the NIST test facility for various operating conditions.

HLITE and VLITE are distributed on a diskette for use on an IBM PC compatible computer using the MS-DOS operating system. The programs, as distributed, require an 80286 or higher CPU and a math coprocessor. Installation of the program is described in the README file on the distribution diskette. It is assumed that the user is familiar with this operating environment. Execution of both programs is controlled interactively and some help in running the programs is available on the screen.

The programs are written in the C language. This language has excellent facilities for handling simulation problems of different sizes through the used of linked structures and memory allocation. Source code and compilation instructions and files are included on the diskette. The programs were compiled with Borland compilers; a different compiler would require some modifications to the source code. Recompilation can be used to run the programs on less capable PC's.

2. The HLITE Program

The simulation of the HVAC/lighting interaction is a two-step process involving first the creation of an abstract mathematical model of reality and second the solution of that mathematical model. The simulation is an approximation of reality limited by our knowledge of the system and the degree of complexity we are willing to incorporate in our model. HLITE has attempted to maintain an appropriate balance in these two factors. It has also attempted to create a model whose elements correspond directly to easily recognized elements in the real structure.

To achieve the desired flexibility, HVAC/lighting interactions are described in terms of a thermal network. The primary components of the thermal network are NODEs which correspond to some volume of material which can be characterized by a single temperature, and LINKs which describe the heat transfer paths connecting various NODEs. Since many links have identical thermal characteristics, these characteristics are described once as ELEMENTs, which are referenced by the LINKs. CONTROLs represent a different set of features. They convert NODE and LINK data such as temperatures and flows to SIGNALs which are processed by other CONTROLs to set values at other NODEs or LINKs. SIGNALS and CONTROLS operate in a network similar to the NODE/LINK network; the two networks are solved separately at each time step.

HLITE uses a simple finite volume approach to the simulation of transient heat transfer in a room (see Appendix A). The choice of the numerical methods used in the HLITE program is based on several factors. The handling of nonlinearities is considered critical. It is believed that control actions will produce the most important and difficult nonlinearities. One method to handle control actions within a long time step model is to note when an action occurs, or should have occurred, and adjust time steps and recompute values accordingly. An alternate method is to use a time step that is so short that the error in the timing of the control action is negligible. A sufficiently short time step combined with a finite volume method can be solved by an explicit time integration scheme. Explicit methods are much simpler and faster for a single time step than implicit methods. The finite volume method provides a simple theoretical basis for defining thermal network components. Explicit methods are especially good at handling nonlinearities. Implicit methods require some form of iteration in solving the simultaneous equations. Iterative methods can encounter difficulties in modeling control actions. The danger of implicit methods is that too long a time step may be used.

A forward Euler solution, the simplest of all explicit methods, is probably of sufficient accuracy at small time steps, especially when the uncertainty in knowledge of material properties and convection coefficients is considered. The cyclic nature of building operations tends to cancel the round-off errors over a long time. It is necessary to include the option of implicit integration for some nodes, e.g. massless nodes are unstable at all time steps. This leads to mixed explicit/implicit modeling. Actually, the method is only partially implicit; it uses coefficients computed at the start of the time step, and is therefore subject to error due to nonlinearities. This technique does need further testing, but good results have been obtained in the cases examined. The choice of explicit or implicit time integration for each node is based on a simple (approximate) stability test. Different choices for a minimum time step leads to different proportions of implicit and explicit nodes. This makes it possible to trade off the length of the time step with the number of nodes that are solved implicitly or explicitly. HLITE executes quite rapidly on a PC class computer for the typical research problem.

Some tests have indicated a relationship between the stability limit and the level of accuracy of the model for simple thermal conduction systems (see Appendix A, section 3.2). Consider the existence of implicit nodes as a warning signal. If there is direct heat input into such a node, the time step is probably too large to accurately model the node dynamics. If there is no direct heat input, the node is only influenced by heat transfer from other nodes, and an implicit solution is accurate. Similarly, check nodes that have a large stability limit compared to all the others; an accurate solution may require subdividing into several nodes.

The HLITE computation sequence for one time step consists of the following actions (described in more detail in Appendix A, section 2.2):

(1) Set boundary value node temperatures; new values read from the Boundary Values File.

(2) Process links to set temperature formula coefficients which will be used in steps (3) and (4). Check that the net air flow into air nodes sums to zero.

(3) Process nodes to compute explicit temperatures, simultaneously create list of temperatures to be solved implicitly.

(4) Compute implicit node temperatures; this is greatly speeded up by the use of a sparse matrix method.

(5) Compute heating and cooling loads at controlled temperature nodes; prepare all nodes for next time step by resetting heat gain values.

(6) Set control signals; new signal values are read from the Discrete Events File.

(7) Process controls to set remaining control signals; the control/signal network is processed in order producing an instantaneous transfer of signals through that network.

(8) Adjust loads of nodes where control values have changes.

(9) Write reports: write selected values to the screen to indicate the status of the simulation and accumulate values to be integrated and write to report files at specified time intervals.

Simulation models are provided for the following building/lighting/HVAC features:

• Transient one-dimensional conduction in multi-layered walls - each layer being homogeneous with constant thermal properties. (The model could be extended to three-dimensional conduction with non-constant properties, but those additional complications are not needed for the present study.)

• Air in a homogeneous zone; movement of air between zones. (This is probably the most serious simplification of the physics. A full computational fluid dynamics (CFD) model would be extremely time consuming. The air in a single room may be divided into zones if it is known how air is transferred among the zones.)

• Convection between surfaces and air according to simple or very complex and non-linear formulae based on the latest experimental research (Spittler, 1990).

• Radiant interchange between diffuse gray surfaces using accurately computed view factors. Constant surface emissivities are assumed for two wavelength bands: thermal radiation and visible light. (More bands could be incorporated, but there is insufficient data to use such an extension in this project.)

- Fluorescent luminaires including temperature dependent lighting and power qualities.

- "Generic" equipment to simulate miscellaneous heat gains and "Generic" mass elements and conductive links to represent additional features. Generic elements give a great deal of flexibility to the program (at a cost of understandability).

- Dead band control of air temperature using a special node. This special node, "czn", allowed version 1 of HLITE to simulate many of the control strategies being considered and to compute the response to a change in lighting levels for the computation of weighting factors. (See Sowell [1989] for a discussion and program for this computation.)

- Proportional-integral controls. Controls and feed-back loops are used to model complex control actions instead of rearranging the equations that are being solved. (See Sowell [1991] for another method of integrating controls using the HVACSIM$^+$ program.)

- "Ideal" heating and sensible cooling coils. These are used together with the controls to compute loads, i.e. cooling requirements, in the more complex cases.

HLITE does not include many of the features necessary for a general building energy analysis program such as solar gains, latent loads, ground heat transfer, and detailed HVAC equipment models.

The input models for HLITE are described in Appendix B. These simple features can be combined to describe very complex situations. Examples of the input are given in the test cases provided in Appendix C. These test cases are used to verify the correct operation of the program.

The descriptions of the physical system being simulated and computed results are stored in ASCII files. Unfortunately, the input for HLITE is more complex than it would be in a more tightly constrained environment. It is not designed for the casual user. Interactive creation of the data files could be helpful. A graphic interface for the connection of nodes and links may be best, but such programming is beyond the scope of the current project.

There are three modes of simulation: steady-state, transient, cyclic. A steady-state simulation is performed by setting all node masses to zero and iterating the node temperatures to convergence. Unfortunately, this has not worked for simulations which include PI (proportinal-integral) control. A simple transient simulation begins at the first time listed on the BVF and continues to the last time with boundary values read from the BVF and signal values set from the DEF. A cyclic simulation reads the first 24 hours of the BVF and DEF and iterates that period until a steady-periodic condition is achieved. From this point the simulation proceeds in the same way as the simple transient simulation.

The execution of HLITE is controlled interactively. This control involves the selection of input files and selection of the mode of simulation to be performed. An alternate version of the program called HLITEA is provided which can use a redirected input file in place of the interactive inputs. An advantage of HLITE is access to interactive help messages which may explain problems in the input files. While the simulation progresses, the current time and some user selected values are displayed on the screen to show the progress of the simulation.

3. The VLITE Program

Although the calculation of view factors occurs only once at the start of a simulation, the calculation can be very time consuming. The problem with view factors is not that they are very difficult to compute, but that the calculation time increases exponentially with the number of surfaces involved. Algorithms which are effective for a small number of surfaces may require a hopelessly long computation time for many surfaces. Consider a situation involving N surfaces. Since each surface may potentially interact with every other surface, there are N^2 interactions, or view factors. Even simplifications such as the reciprocity relation and the fact that a flat surface cannot view itself, reduce the number of view factors only to N(N-1)/2, which is still of order $O(N^2)$. If it is known that some, but not which, surfaces can obstruct (or 'shade' or 'occlude') the views between surface pairs, it is necessary to check N-2 surfaces as possible obstructing surfaces for each view factor. This gives N(N-1)(N-2)/2 obstruction checks, that is, $O(N^3)$. In addition, the procedures for computing obstructed view factors tend to be much less efficient than those for unobstructed view factors.

View factors may be computed for either two-dimensional or three-dimensional geometries. See Appendix D for the theory of the calculations used in VLITE. These algorithms are appropriate for use on a PC class computer, although the performance of a 486 class machine is useful for rooms involving many view obstructing surfaces. A simple algorithm based on surface areas, which may or may not lead to significant errors, is also provided. This could be useful for older less powerful PC's.

The geometry is described in an ASCII file using a Cartesian coordinate system (see Appendix E, section 3). Again, graphic input would be very helpful, but such programming is beyond the scope of the current project. However, a graphic display of the geometry is possible for computers equipped with VGA graphics. This display is very helpful for finding errors in the description of the problem geometry. Simple typing and syntax errors are noted as the input file is read, and the interactive help messages may be useful in understanding and correcting such errors.

The execution of VLITE is controlled interactively (see Appendix E, section 2). The user controls factors which influence accuracy and execution time and also the output of the calculations. One of these outputs is some ASCII text or a separate file which can be edited directly into the HLITE NDF.

4. Sample HVAC/Lighting Calculations

A series of simple analytical tests have been developed to insure that HLITE and VLITE are performing as expected. They are described in detail in Appendices C and F. The tests are sequenced so that the most basic algorithms are tested first. Later tests of other algorithms often rely on the basic algorithms. The input files for these tests are included on the distribution diskette to provide the user simple cases that can run directly to gain familiarity with the program, and to maintain the set of test cases which must be run when the program is modified to insure that its current capabilities have not been altered.

The experimental results used to test the HLITE program are generated by the NIST test facility as described by Treado and Bean [1988]. This test facility is constructed on a large concrete slab within the NIST environmental chamber. The facility is divided into two sections, a large insulated shell enclosing the test room area, and a smaller attached control room for housing instrumentation as shown in figure 2. The overall height of the facility is 6.36m (20'10½"). The interior dimensions of the test room are 4.23m (13'10½") long by 3.73m (12'3") wide with a suspended ceiling 2.44m (8') high. Above the ceiling is a 0.76m (2'6") high plenum. There are four luminaires mounted in the ceiling.

The test room floor slab is elevated to accommodate a lower plenum beneath the floor, and all other room surfaces are adjacent to temperature-controlled guard air spaces. Duplicate lighting and HVAC systems are installed in both the test room plenum and the lower plenum. The test room floor and ceiling slabs are 6.4 cm (2½ inch) thick concrete built on steel decks supported by a structural steel framework. The walls are constructed of 1.6 cm (5/8 inch) thick gypsum board fastened to steel studs.

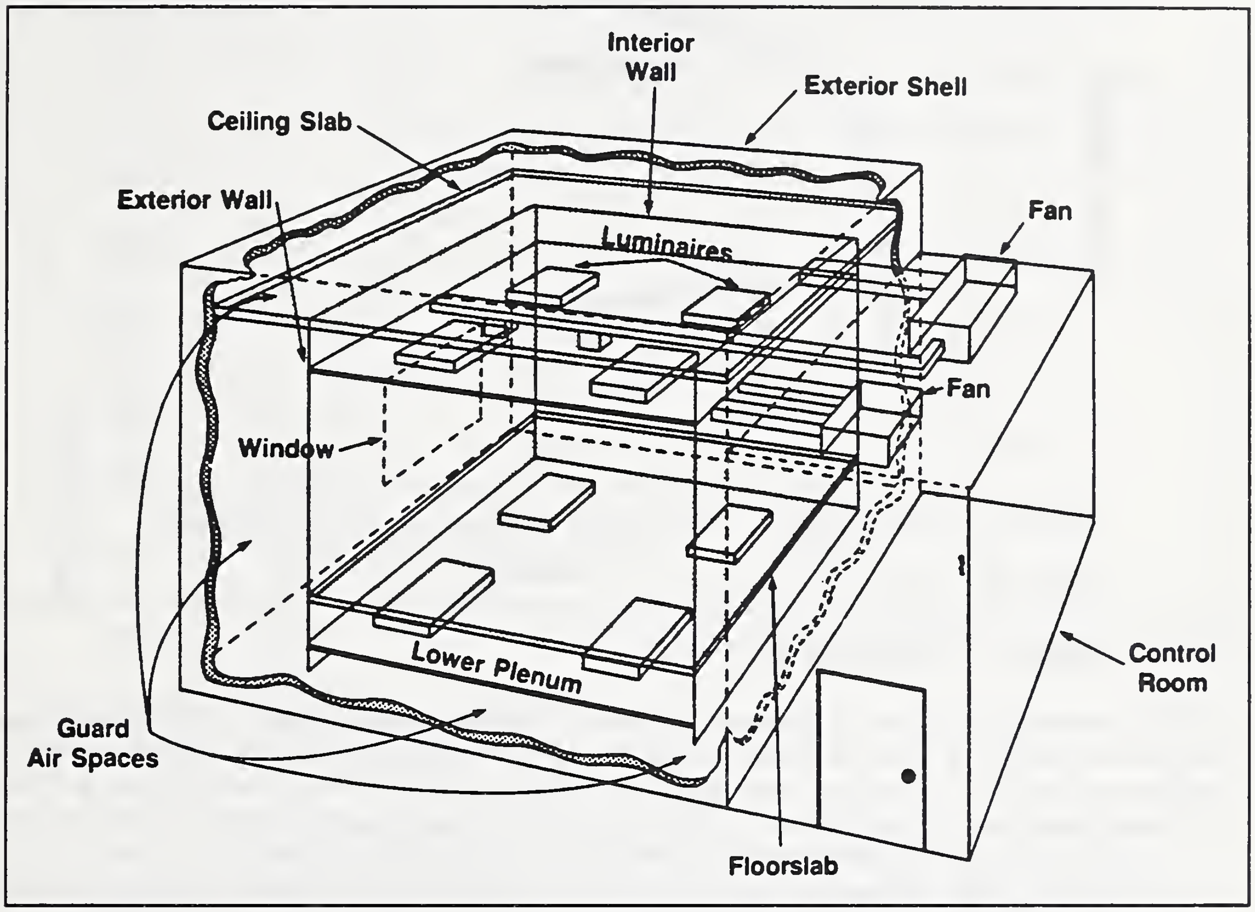


Figure 2. Cut-away Schematic View of Test Facility

A comparison of the experimentally measured loads with the results of an HLITE simulation for the case of ceiling grill air return follows. The NDF describing the test room is included in Appendix C. Results of the modeling are shown in figure 3 which compares the measured and computed transient cooling loads and shows reasonably good agreement between the two. Adjustments were made in two areas to get this agreement. First all the measured data was adjusted so that the average load and temperature values in the hour before lights on are at the correct steady state values. Second the parameters influencing the luminaire performance were adjusted to give correct steady state power consumption. These adjustments were in a range of reasonable values, but the values selected are not necessarily obvious. All other values were taken from the best available published data.

Considerably more data is available than was used in generating this initial comparison. These data, generated over a broad range of operating conditions and lighting configurations, will be used to define a set of modeling parameters which simulate the test room under all those conditions. The results of these comparisons will be reported at a later date.

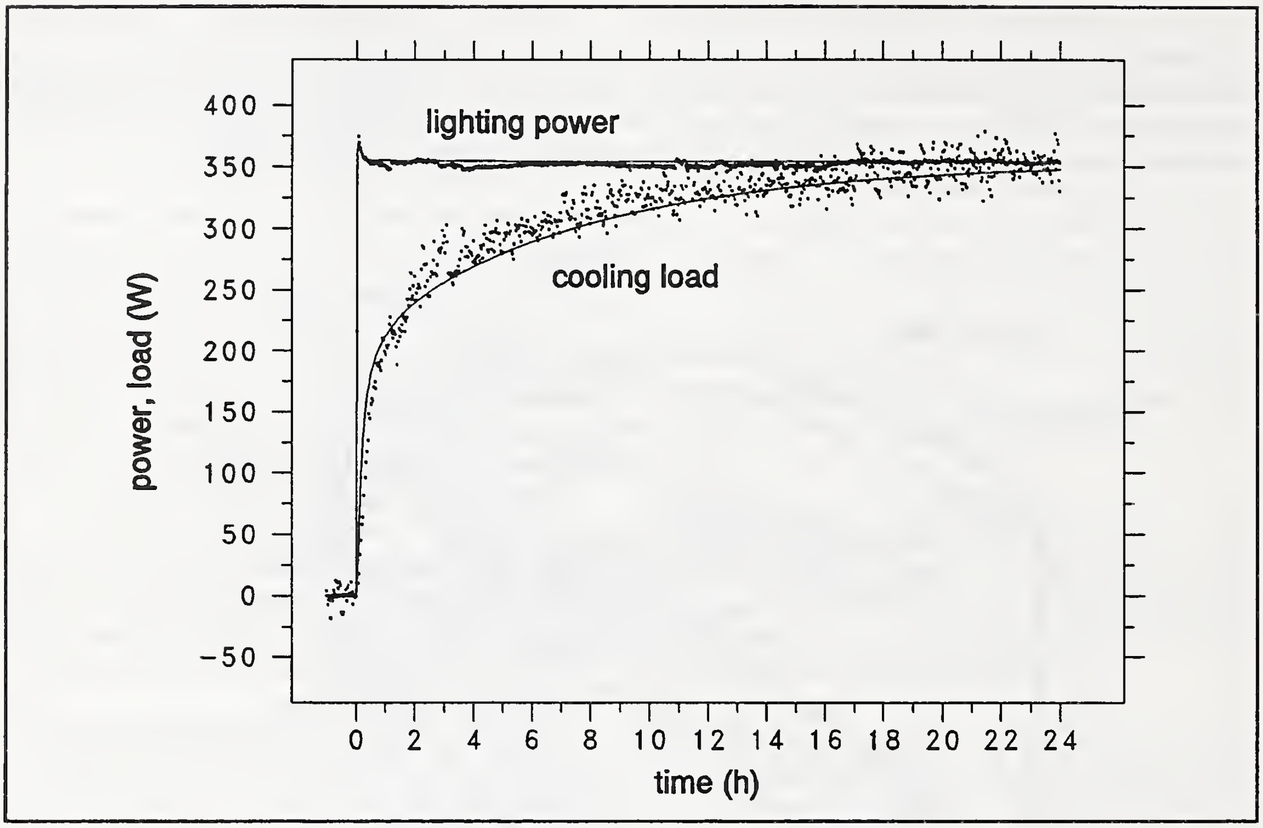


Figure 3. Comparison of Test ITd to Simulation

HLITE can perform two types of transient simulation. The first type begins at the NDF initial values and proceeds according to the contents of the BVF and DEF. Sample results are plotted in Figure 4 (Similar to Figure 3). The second type begins by iterating through the first 24 hours of the BVF and DEF until steady periodic performance is achieved. Then transient simulation proceeds from the new node temperatures in the normal manner. The results of such a periodic simulation are plotted in Figure 5. Note the small spike in lighting power curve which is caused by the temperature dependence of the fluorescent lamp power consumption.

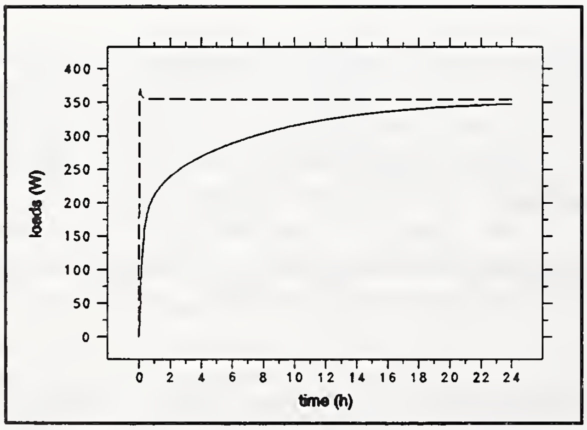


Figure 4. Transient Simulation

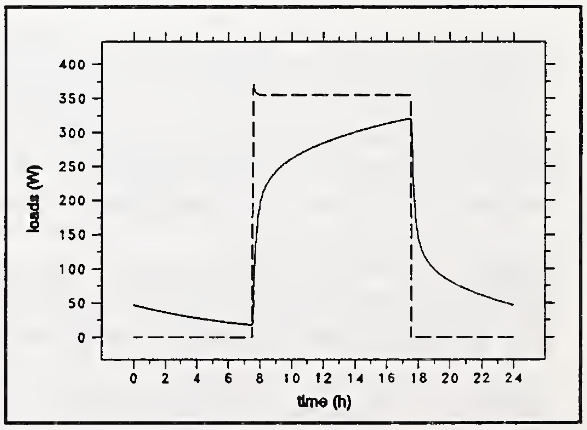


Figure 5. Cyclic Simulation

The simulations in figures 4 and 5 used the simple constant temperature air node available in version 1 of HLITE. This and the special "czn" node allowed the simulation of several control strategies, but in order to allow the simulation of many more control strategies a control model was implemented in version 2 of HLITE. Rather than considering the controls to be constraints on the thermal simulation equations, controls are made to emulate the performance of digital controls. This method has both the advantages and disadvantages of real controls. The biggest disadvantage is that the controls must tuned to provide optimum performance.

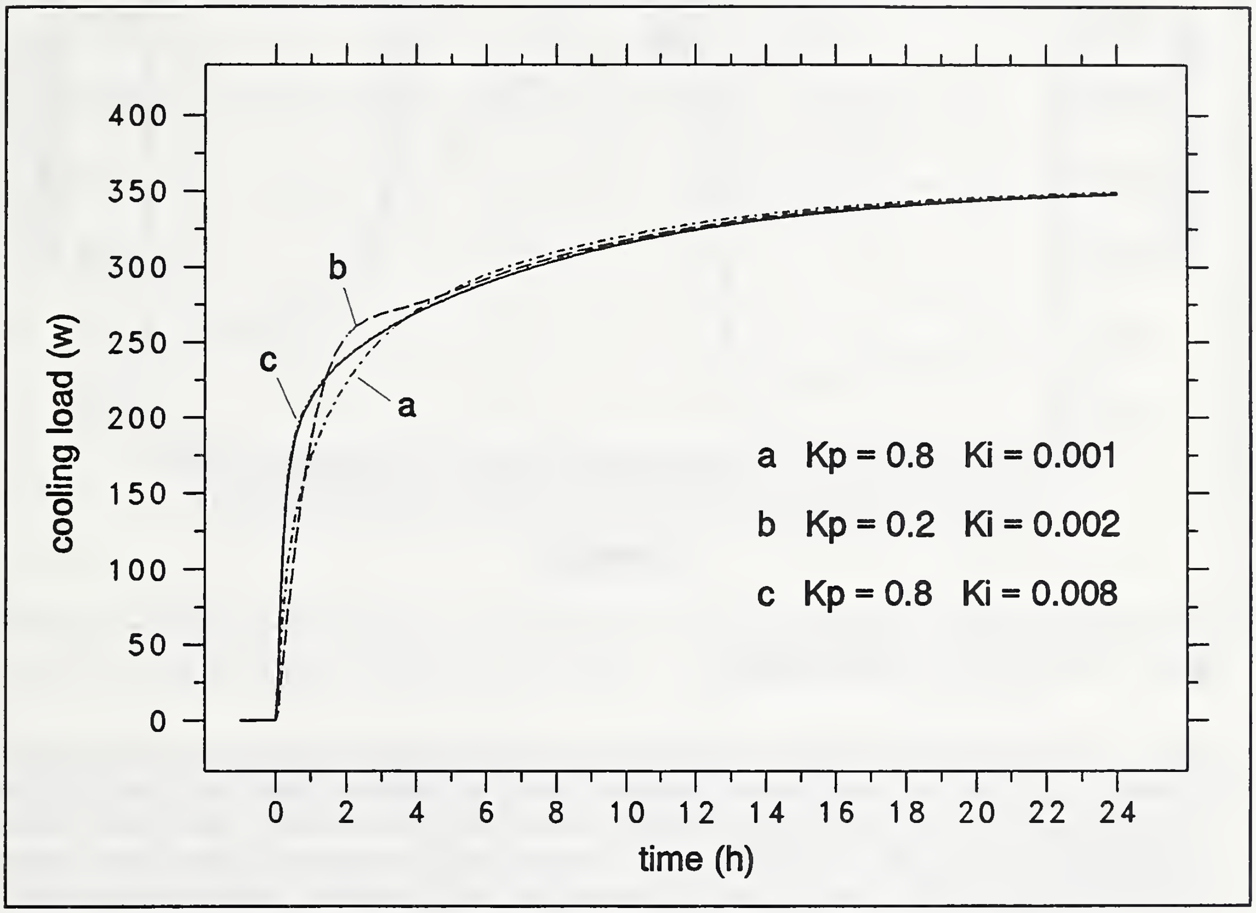


Figure 6. Effect of PI Control Parameters

Figure 6 compares the ideal constant air temperature transient simulation with several simulations using PI control with different coefficients. All cases were run with a minimum time step of 15 seconds. In this figure three combinations of the proportional factor Kp and the integral factor Ki are plotted. Curve c is very close to the ideal case. The controls model can be considered a source of error when compared to the theoretical or ideal case, or a means of more accurately simulating the performance of real buildings. How we feel about this question depends on the philosophy with which we approach the modeling process.

In any event, care must be taken in selecting the control coefficients. A tuning process is required. The values obtained for one time step are not optimal for another. Therefore, in order to compare the results of different runs, the time step and control parameters should be the same. Figures on the following pages will show simulations for cases involving relatively complex control actions.

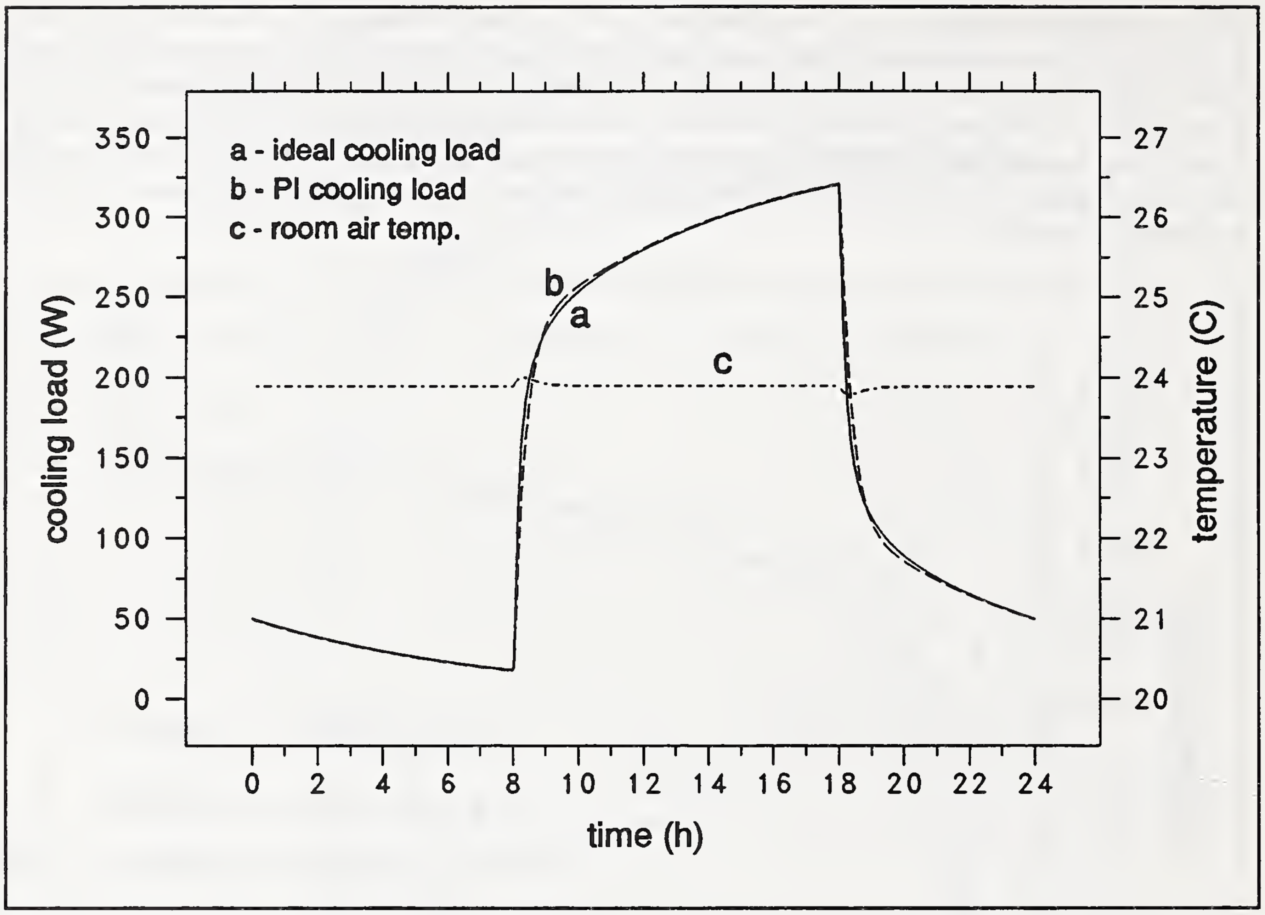


Figure 7. PI Control vs. Analytic Solution

Figure 7 compares the ideal constant air temperature model with the PI controlled model for a cyclic simulation. The lights are turned on for ten hours starting at 08:00 and turned off at 18:00 for the remaining fourteen hours of the day. The room air temperature is controlled to 23.9C (75.0F). Under PI control this varies by ±0.1°C (0.2°F). These variations are a necessary part of the control actions. The very minor differences in cooling loads are inversely related to variation in room air temperature.

The simulated peak cooling load at 18:00 using PI control is 320.9W versus 320.4W for the ideal case. Greater differences appear in the hourly cooling energy for the first few hours, but daily total cooling energy requirements are nearly identical:

time	ideal	PI control
08:00 - 09:00	173.6 Wh	159.2 Wh
09:00 - 10:00	241.9 Wh	264.2 Wh
10:00 - 11:00	261.1 Wh	263.6 Wh
...	...	...
24 hours	3548.1 Wh	3547.8 Wh

Because of the transient heat transfer within the room, the peak cooling load is less than the coincident lighting power, 354.4W. This difference, here 33.5W, will be called the "peak cooling load reduction".

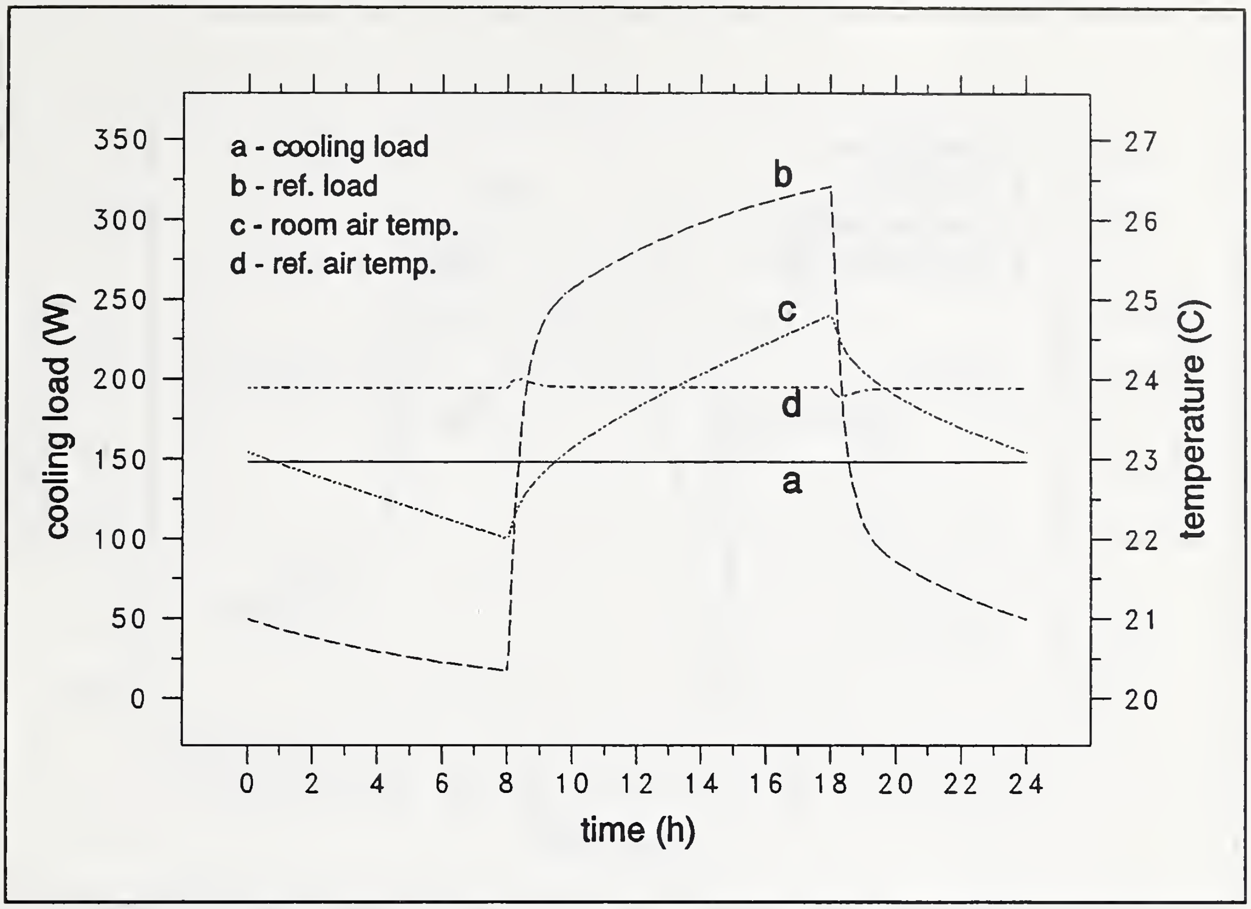
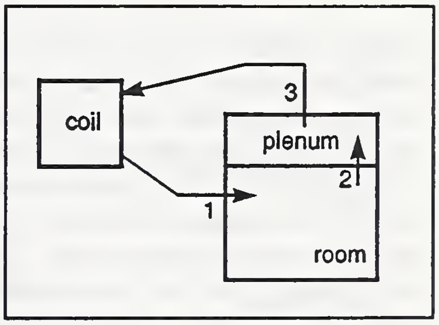


Figure 8. Constant Load vs. Constant Temperature



The test facility represents a thermally closed system in that all energy into the lights must eventually be removed from the room as a cooling load. Therefore, HVAC control actions cannot actually save energy, but they can affect the peak cooling loads which may significantly affect the cost of energy used.

The ideal peak load reduction would occur when the cooling power is kept constant for 24 hours and the room temperature is allowed to drift. Using the standard pattern of air flows (see sketch) and manually iterating with various loads, a constant hourly cooling load of 148.2W was found to provide steady cyclic performance. This corresponds to a peak cooling load reduction of 206.2W. This reduction in the peal cooling load is achieved at cost of temperature drift in the room. The case of nearly constant room air temperature and varying load (Figure 7) has been replaced by constant load and varying temperature (Figure 8). The minimum room air temperature 1s 22.0C (71.6F) and the maximum room air temperature is 24.8C (76.6F). Including other room heat gains, such as equipment and occupants, would lead to even greater temperature swings. Consider this and the following runs as a demonstration of simulation capabilities rather than a search for optimum peak cooling load reduction.

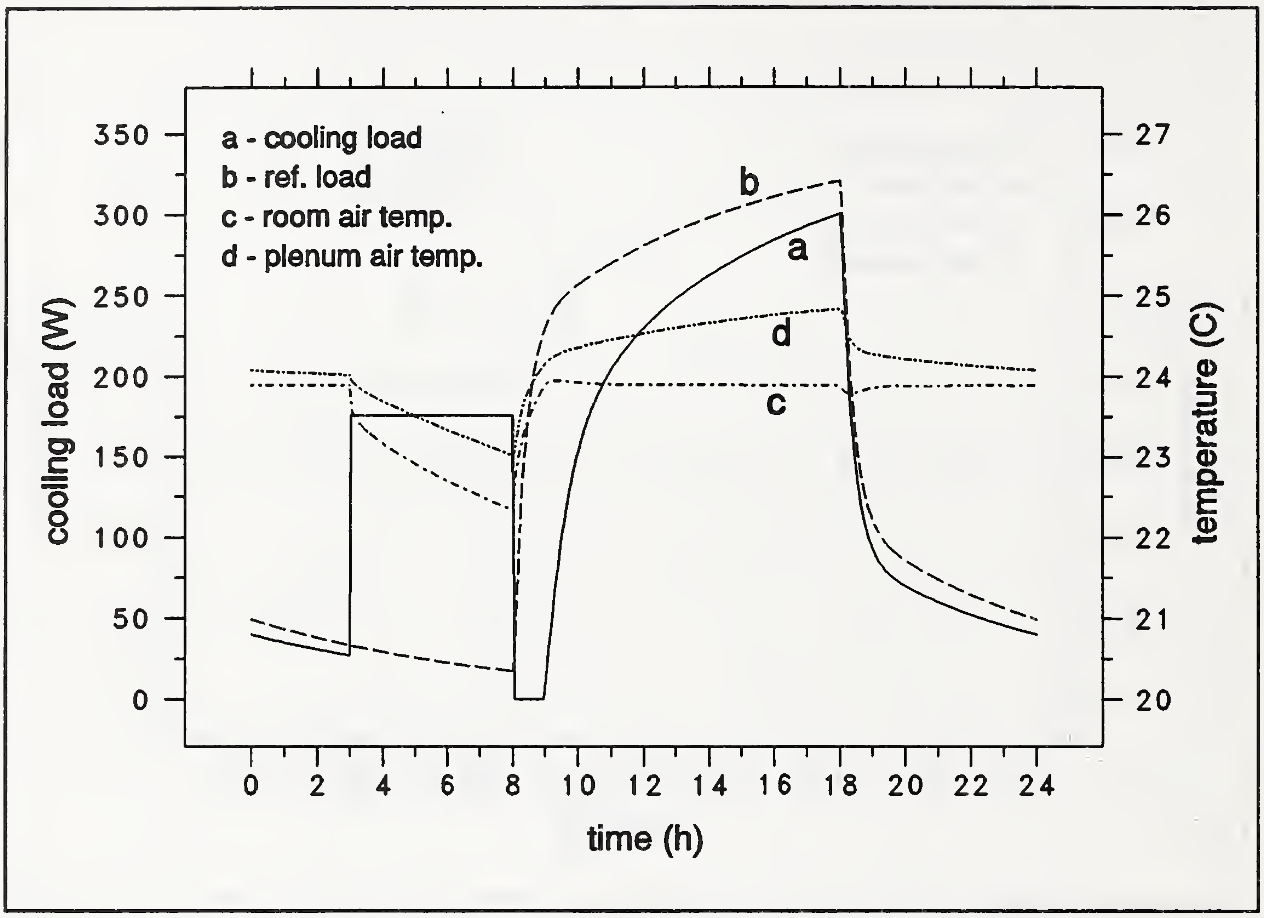


Figure 9. Peak Load Reduction Using Precooling

Perfect leveling of the cooling load is not possible. Loads vary from day to day and controls that anticipate the next day's load would be required. A control strategy which uses some precooling of the room (during the early morning hours when the cost of energy is presumably cheaper) will provide some peak cooling load reduction. Figure 9 shows the results of cooling the room at a rate of 175.8W from 03:00 to 08:00. This figure includes the cooling load for this control strategy (solid line), the cooling load for the constant temperature control strategy (dashed line) for comparison, the room air temperature, and the plenum air temperature. The plenum air temperature is included as an indicator of the use of the heat storage capacity of the materials enclosing the plenum, especially the underside of the concrete floor.

The maximum cooling load is 300.5W for a peak load reduction of 53.9W. The minimum room air temperature is 22.3C (72.2F) and the maximum remains near the desired value.

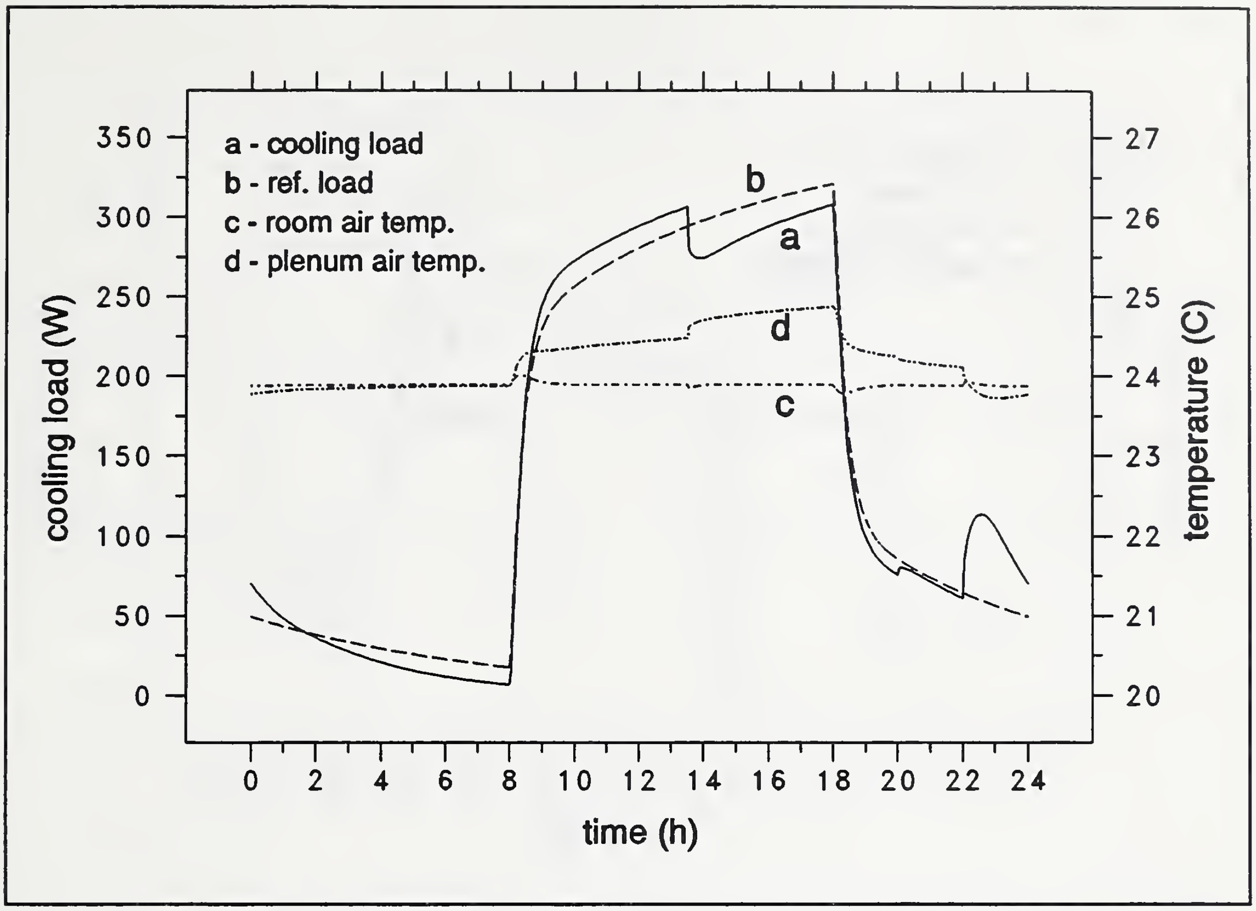
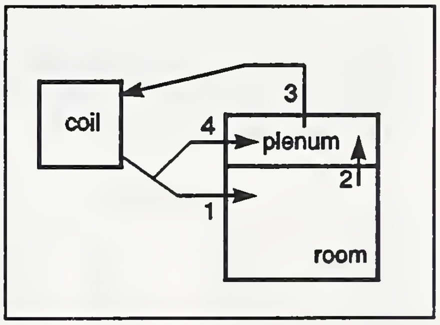


Figure 10. Peak Load Reduction Using Bleed Air



Large changes in room air temperature are not desirable for the comfort of the occupants. Therefore, another strategy is employed where some cooling air is bled from the supply duct directly into the plenum (path 4 in the sketch) at certain times as summarized below.

time		fraction of total flow in path			
from	to	1	2	3	4
08:00	13:30	0.75	0.75	1.00	0.25
13:30	20:00	1.00	1.00	1.00	0.00
20:00	22:00	0.75	0.75	1.00	0.25
22:00	08:00	0.15	0.15	1.00	0.85

This strategy produces two peaks in the cooling load curve: 306.4W at 13:30 and 308.0W at 18:00 with negligible variation in room air temperature. The peak cooling load reduction is 46.4W. The time for shutting off the bleed air was determined from several runs so as to give a near maximum peak reduction for this particular set of air flows.

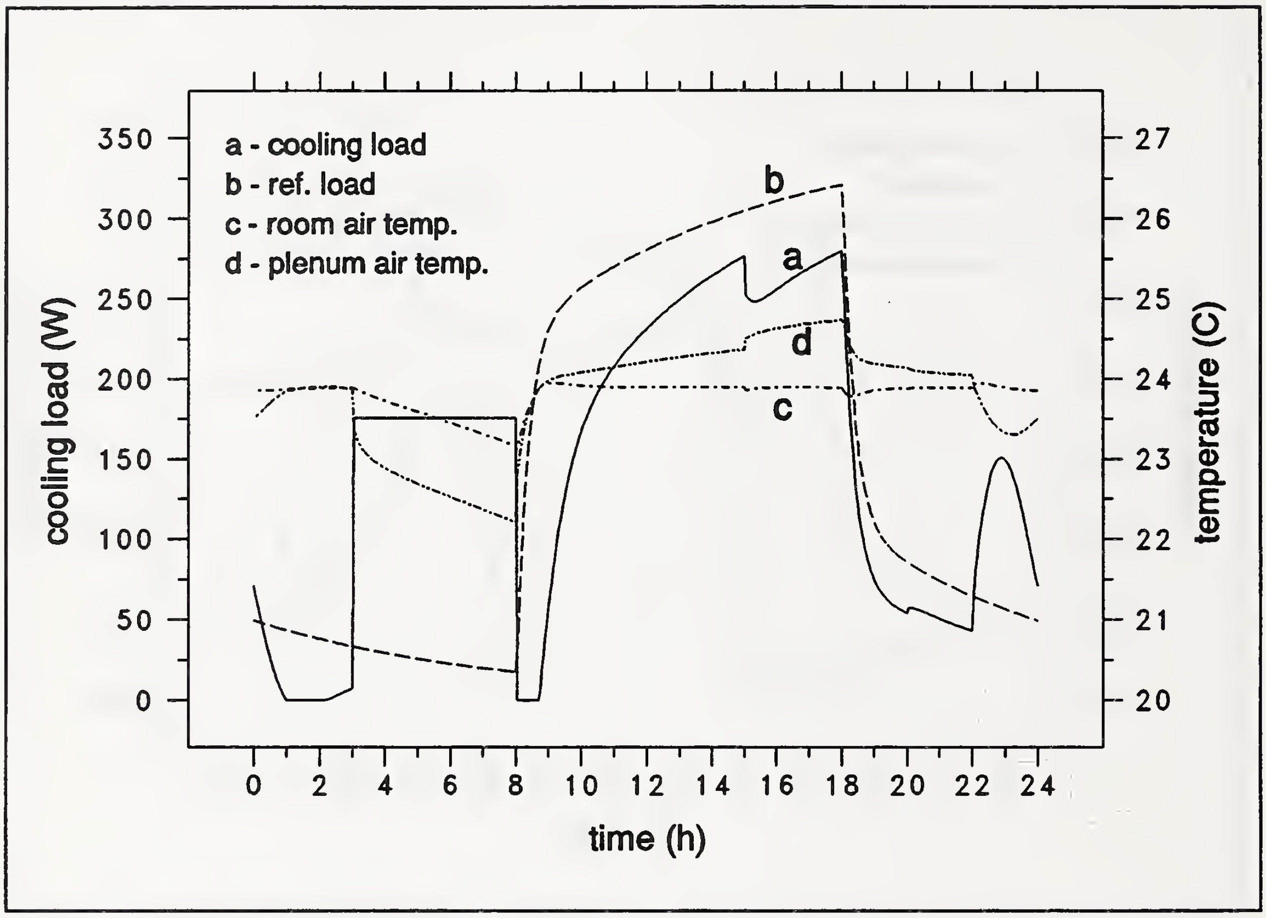


Figure 11. Peak Load Reduction Using Bleed Air & Precooling

Some combination of pre-cooling and bleed air may provide an increase in the peak load reduction while not varying the room temperature as much as in the precooling case. Adding pre-cooling (175.8W) between 03:00 and 08:00 while moving more air through the plenum than in the previous case produces the results shown in Figure 11.

```
    time     fraction of total flow in path
 from    to       1     2     3     4
08:00 - 13:30   0.75  0.75  1.00  0.25
13:30 - 20:00   1.00  1.00  1.00  0.00
20:00 - 22:00   0.75  0.75  1.00  0.25
22:00 - 08:00   0.01  0.01  1.00  0.99
```

The peak cooling loads are 276.7W at 15:00 and 279.7W at 18:00 for a peak load reduction of 74.7W. The minimum room air temperature is 23.1C (73.6F) and the maximum remains near the set point. The plenum air temperature falls significantly below the room air temperature at 08:00. This allows the storage of much more heat in the plenum later in the day than in the previous case.

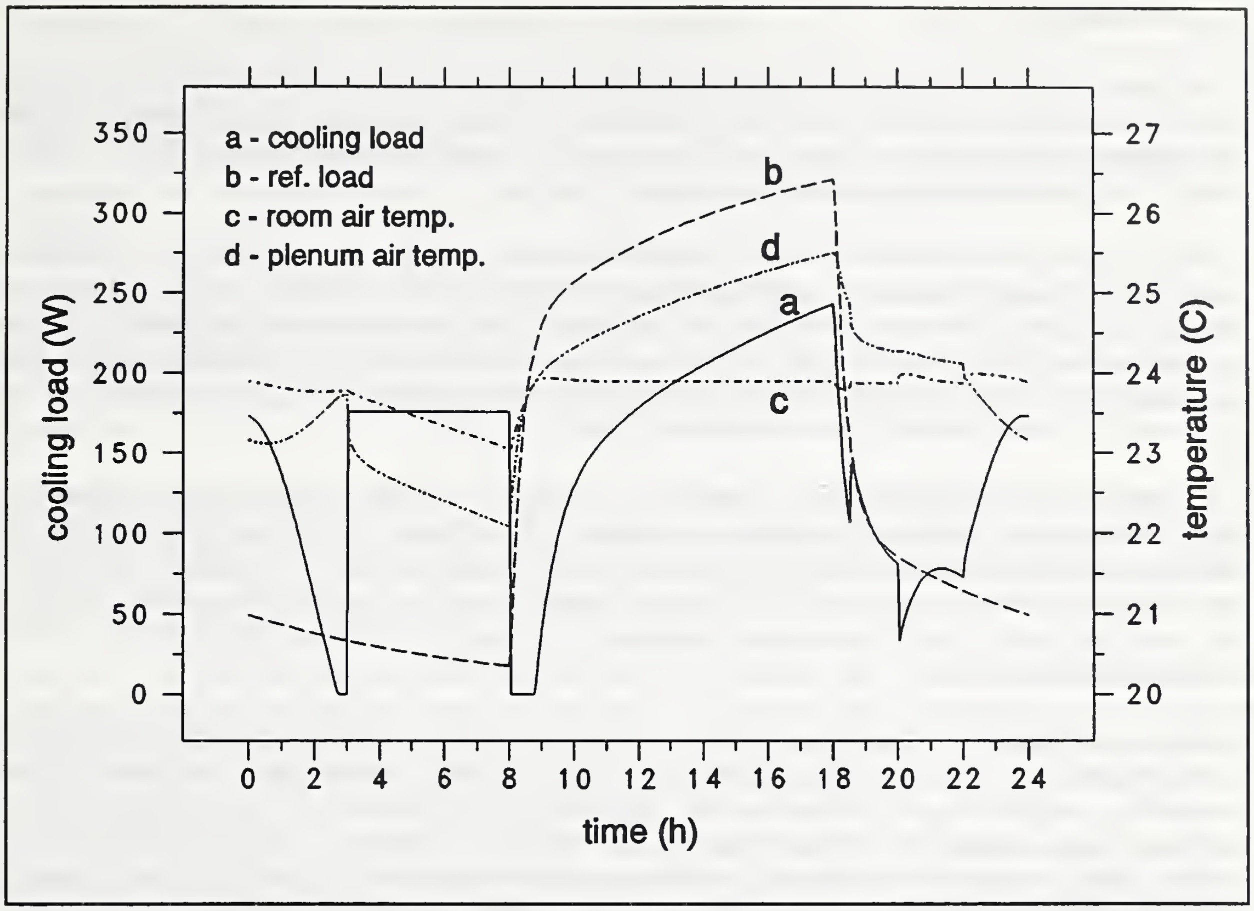
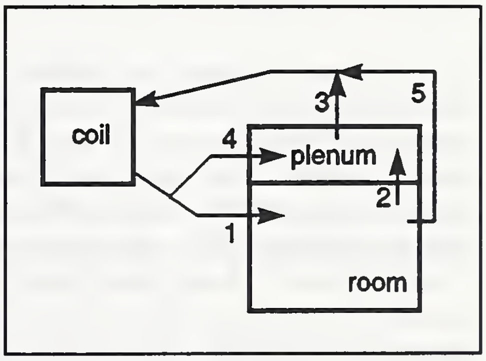


Figure 12. Peak Load Reduction Using Ducted Return

A final alternative will attempt to reduce the peak cooling load by returning the room air directly to the return air duct during occupied hours instead of passing air through the plenum. The accompanying sketch shows an additional air flow path labeled 5. The following air flow schedule was used:



time		fraction of total flow in path				
from	to	1	2	3	4	5
08:00	18:30	1.00	0.00	0.00	0.00	1.00
18:30	20:00	1.00	1.00	1.00	0.00	0.00
20:00	22:00	0.75	0.75	1.00	0.25	0.00
22:00	08:00	0.01	0.01	1.00	0.99	0.00

The maximum cooling load is 243.0W for a peak load reduction of 111.9W. The minimum room air temperature is 23.0C (73.4F). This strategy makes good use of the floor mass as indicated by plenum air temperatures ranging from 22.1C (71.8F) to 25.5C (77.9F). In effect, the stored "cool" in the plenum picks up almost the entire heat gain from the lights which goes upward into the plenum. This strategy does, however, produce larger temperature swings (and inefficiency) in the luminaires as indicated by a lighting power consumption of 350.4W instead of 354.9W in the original constant temperature case.

5. SUMMARY

This report has described two programs developed for the analysis of lighting/HVAC interactions: HLITE and VLITE. VLITE is used to compute the coefficients (view factors) describing radiation interchange between surfaces. These coefficients are used in a thermal network model which is solved by HLITE for transient temperatures and cooling loads.

These programs are research tools rather than engineering design tools. The primary goal is simulation flexibility (which led directly to a thermal network approach in HLITE). Another important goal is that the simulation be accurate and sensitive to the parameters that the researcher wishes to study. These primary goals dictate that the program be based on fundamental physical principles rather than correlations to the experimental data. Good program performance in terms of execution time and program size as well as ease of use are secondary goals.

The simulation of the lighting/HVAC interaction is a two-step process. First is the development of a model of the system; second is the mathematical solution of the model. This concept applies to the user who must develop the thermal network description data files which are solved by HLITE. The concept was also applied in development of HLITE where "models" refer to the physical processes chosen and the "solutions" refer to the computational techniques. The results of the computation can be no more accurate than the model.

Models were chosen according to their importance to lighting/HVAC interaction, the programming and computational costs of implementing solutions, and the availability of physical data. Such models include air in fully mixed zones, one-dimensional heat conduction, radiative transfer between constant property diffuse surfaces, and the two band (thermal and visible) model of radiation. All of the strictly thermal process could be modeled in terms of conventional differential equations. Controls, which are critical to the overall project, are fundamentally simple but may present a problem in the solution. One approach is to treat the controls as additional equations which are enforced as the thermal network is solved. This produces a differential algebraic system of equations which require iteration for a solution. It was decided to develop a solution which uses no iteration.

This solution is based on an imitation of the processes present in the test facility which is controlled by a microcomputer. Every 12 seconds this computer reads all the sensors in the test facility and generates signals to control the operation of the HVAC system. HLITE emulates this process with a thermal network of differential equations to model the heat transfer and an algebraic controls network to process signals and set control actions. This method of modeling controls requires a short (no more than one or two minutes) time step. A short time step is useful for simulating the other nonlinear effects in the thermal network. A sufficiently short time step combined with a finite volume method can be solved by an explicit time integration scheme. Explicit methods are much simpler and faster for a single time step than implicit methods. The finite volume method provides a simple theoretical basis for defining thermal network components. A forward Euler solution, the simplest of all explicit methods, is probably of sufficient accuracy at small time steps, especially when the simplicity of the models and uncertainty in knowledge of material properties and convection coefficients is considered. The cyclic nature of building operations tends to cancel the round-off errors over a long time. It has been necessary to include the option of implicit integration for some nodes, e.g. massless nodes are unstable at all time steps. This leads to mixed explicit/implicit modeling where the few implicit equation are handled by a fast sparse matrix solver.

The net result of these decisions in modeling and computation is a fast, flexible simulation tool. The accuracy of the mathematical solution seems appropriate to the models used. The solution time is directly proportional the number items in the thermal network and inversely proportional to the time step. The simple explicit methods used in HLITE could provide a general approach to modeling building systems. The use of simple explicit methods may match very well with the coming generation of multi-processor computers and an apparent trend in using very simple computational methods adaptable to multi-processors.

HLITE is supported by the VLITE program. Radiation interchange between diffusely reflecting and absorbing surfaces is important for problems involving either radiant heat transfer or distribution of light. Although the calculation of view factors occurs only once at the start of a simulation, the calculation can be very time consuming. View factors may be computed for either two-dimensional or three-dimensional geometries.

HLITE also requires values for the fraction of light energy absorbed at each room surface. The calculation of these values in rooms with obstructing surfaces is non-trivial. Commercial lighting programs might be used for this purpose. If it can be assumed that light is distributed diffusely from the luminaires, VLITE can be used to calculate that distribution.

The use of HLITE and the test facility measurements to improve the models employed in common building energy analysis programs remains to be done. A few points have been made obvious by the use of HLITE. The performance of the luminaire as a function of temperature is critical. The convection coefficients around and within the luminaire as a function of various air flows are also critical. The tests have indicated typical ranges of luminaire performance for a series of operating conditions. There is a fundamental mismatch between the time scales used in energy analysis programs and the time scale necessary to model the dynamics of lights. Fortunately, the transient behavior of the luminaires has been shown to have only a minor impact on total energy requirements. Programs using weighting factors cannot include the effects of nonlinear and variable heat transfer coefficients. HLITE can show how important these details really are.

6. REFERENCES

These include all references to the report and Appendices A through F.

Abramowitz, M. and I.A. Stegun. 1964. *Handbook of Mathematical Functions with Formulas, Graphs, and Mathermatical Tables*, National Bureau of Stantards Applied Mathematics Series #55, Washington DC, (also Dover Publications, New York NY, 1965) p 916.

ASHRAE. 1987. *ASHRAE Handbook - 1987 HVAC Systems and Applications*, ASHRAE, Atlanta GA.

ASHRAE. 1989. *ASHRAE Handbook - 1989 Fundamentals*, ASHRAE, Atlanta GA.

Ball, H.D. & D. Green. 1983. "The Impact of LIghting Fixtures on Heating and Cooling Loads - Mathematical Model", *ASHRAE Transactions*, Vol 89, Pt 2.

Belytschko, T. 1983. "An Overview of Semidiscretization and Time Integration Procedures", Chapter 1 in *Computational Methods for Transient Analysis*, edited by T. Belytschko and T.J.R. Hughes, North-Holland.

Burns, P.J. 1983. "MONTE -- A Two Dimensional Radiative Exchange Factor Code," Colorado State University, Ft. Collins CO.

Clausing, A.M. 1969. "Numerical Methods in Heat Transfer" in *Advanced Heat Transfer*, Ed. B.T. Chao, University of Illinois Press, Urbana IL.

Cohen, M.F. and D.P. Greenberg. 1985. "The Hemi-Cube -- A Radiosity Solution for Complex Environments," *ACM SIGGRAPH*, Vol 19, No 3.

Cohen, M.F., D.P. Greenberg, D.S. Immel and P.J. Brock. 1986. "An Efficient Radiosity Approach for Realistic Image Synthesis," *IEEE Computer Graphics and Applications*, Vol 6, No 2.

Conte, S.D. and C. de Boor. 1972. *Elementary Numerical Analysis*, McGraw-Hill, New York NY, pp 299-306.

Emery, A.F. 1986. "VIEW -- A Radiation View Factor Program with Interactive Graphics for Geometry Definition (Version 5.5.3)," NASA's Computer Software Management and Information Center, Athens GA.

Hottel, H.C. and A.F. Sarofim. 1967. *Radiative Transfer*, McGraw-Hill, New York NY.

IES. 1984. *IES Lighting Handbook, 1984 Reference Volume*, Illumination Enginering Society of North America, New York, 1984.

Judkoff, R., D. Wortman, R. O'Doherty, and J. Burch. 1983. "A Methodology for Validating Building Energy Analysis Simulations", Solar Energy Research Institute report SERI/TR-254-1508, Golden, CO.

Kreith, F. 1973. *Principles of Heat Transfer*, 3rd ed., Intext Educational Publishers.

Lewis, P.T., & D.K. Alexander. 1990. "HTB2: A Flexible Model for Dynamic Building Simulation", *Building and Environment*, Pergamon Press, Vol 25, No 1, pp 7-16.

Lipps, F.W. 1983. "Geometric Configuration Factors for Polygonal Zones Using Nusselt's Unit Sphere", *Solar Energy*, Vol. 30, No. 5.

Liu, W.K. & J.I. Lin. 1983. "Mixed Time Integration Schemes for Transient Conduction Forced-Convection Analysis" in *Numerical Properties and Methodologies in Heat transfer: Proceedings of the Second National Symposium*, Edited by T.M. Shih, Hemisphere Publishing Corp.

Mitalas, G.P. and D.G. Stephenson. 1966. "FORTRAN IV Programs to Calculate Radiant Interchange Factors," National Research Council of Canada, Division of Building Research, Ottawa, Canada, BDR-25.

Newman, W.M., and R.F. Sproull. 1973. *Principles of Interactive Computer Graphics*, McGraw-Hill, Appendix II.

Pavlidis, T. 1982. *Algorithms for Graphics and Image Processing*, Computer Science Press, Rockville MD.

Press, W.H., B.P. Flannery, S.A. Teukolsky, & W.T. Vetterling. 1988. *Numerical Recipes in C: the Art of Scientific Computing*, Cambridge University Press.

Puccinelli, E.F. 1973. "View Factor Computer Program (Program VIEW) User's Manual," Goddard Space Flight Center, Greenbelt MD, X-324-73-272

Putnam, K.L. and P.A. Subrahmanyam. 1986. "Boolean Operations on n-Dimensional Objects," *IEEE Computer Graphics & Applications*, Vol 6, No 6.

Rundquist, R.A. 1990. "Considerations in Developing a Transient Geat-from Lights Design", *ASHRAE Transactions*, Vol. 96, Part 2.

Selby, S.M., ed. 1974. *Standard Mathematical Tables*, 22nd Edition, CRC Press, p 369.

Sebald, A.V., 1979. "Efficient Solution of Large, Controlled Passive Solar Energy Systems: Forward Differencing in Thermal Networks", UCSD Energy Center Report, University of California, San Diego, La Jolla CA.

Shapiro, A.B. 1983. "FACET -- A Radiation View Factor Computer Code for Axisymmetric, 2D Planar, and 3D Geometries with Shadowing," University of California, Lawrence Livermore National Laboratory, UCID-19887.

Shapiro, A.B. 1985. "Computer Implementation, Accuracy, and Timing of Radiation View Factor Algorithms," *ASME Journal of Heat Transfer*, Vol 107.

Siegel, R. and J.R. Howell. 1978. *Thermal Radiation Heat Transfer*, Hemisphere Publishing Corp., Washington DC.

Sowell, E.F. 1972. "Environmental Radiation for Fluorescent Ceiling Systems", PhD Thesis, U. California, Los Angeles.

Sowell, E.F. 1989. "LIGHTS User's Guide", California State University, Fullerton.

Sowell, E.F. 1990. "A Numerical Lighting/HVAC Test Cell", *ASHRAE Transactions*, Vol. 96, Part 2.

Sowell, E.F. 1991. "A General Zone Model for HVACSIM$^+$ -- Users' Manual", Report No. OUEL 1889/91, University of Oxford, England.

Sparrow, E.M. and R.D. Cess. 1976 *Radiation Heat Transfer*, Hemisphere Publishing Corp., Washington DC.

Spittler, J.D. 1990. "An Experimental Investigation of Air Flow and Convective Heat Transfer in Enclosures Having Large Ventilative Flow Rates", PhD Thesis, University of Illinois, Urbana-Champaign, IL.

Stoecker, W.F, & P.A. Stoecker. 1989. *Microcomputer Control of Thermal and Mechanical Systems*, Van Nostran Reinhold, New York.

Treado, S.J., & J.W. Bean. 1988. "The Interaction of Lighting, Heating and Cooling Systems in Buildings - Interim Report", National Institute of Standards and Technology, NISTIR 88-3860.

Treado, S.J., & J.W. Bean. 1990. "Experimental Evaluation of Lighting/HVAC Interaction", *ASHRAE Transactions*, Vol. 96, Part 2.

Walton, G.N. 1986,87. "Algorithms for Calculating Radiation View Factors Between Plane Convex Polygons With Obstructions", National Bureau of Standards, NBSIR 86-3463 (1987 - shortened report in *Fundamentals and Applications of Radiation Heat Transfer*, HTD-Vol.72, American Society of Mechanical Engineers).

Walton, G.N. 1989. "AIRNET - A Computer Program for Building Airflow Network Modeling" National Institute of Standards and Technology, NISTIR 89-4072.

Walton, G.N. 1990. "A Computer Program for Simulation of HVAC / Lighting Interactions: Initial Report" National Institute of Standards and Technology, NISTIR 90-4472.

Wong, R.L. 1976. "User's Manual for CNVUFAC -- The General Dynamics Heat Transfer Radiation View Factor Program," University of California, Lawrence Livermore National Laboratory, UCID-17275.

APPENDIX A

HLITE Theory

A.1 Introductory Example

The development of HLITE centers around the capability to perform transient simulation. A one-dimensional heat conduction problem provides the simplest example to illustrate transient simulation methods. Figure A.1 shows a portion of a one-dimensional conduction problem consisting of two material layers. Temperatures are computed at the boundaries of the layers because of (1) the importance of wall surface temperatures in a room and (2) the possibility of using a more exact finite element method for this analysis. This example will be described by physical instead of mathematical arguments following the description given by Clausing [1969, pp 157-213].

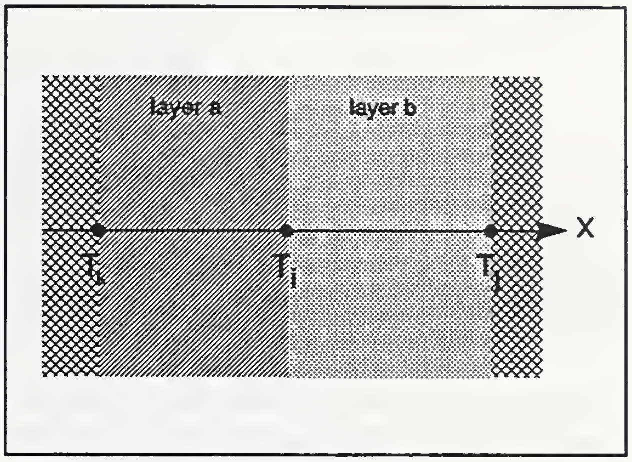


Figure A.1 1-D Conduction

The thermal properties of layers a and b are: the thicknesses, L_a and L_b, the thermal conductivities, κ_a and κ_b, the densities, ρ_a and ρ_b, and the specific heats, c_a and c_b. Each layer has identical surface areas, A. Subscripts i, j, and k refer to positions in space and n refers to time. In a lumped parameter representation, the temperature T_i can be considered to represent the temperature of a pseudo-layer extending from the midpoint of layer a to the midpoint of layer b. The internal energy of this pseudo-layer is given by:

$$U_i \equiv C_i T_i \equiv \rho V c T_i = \frac{\rho_a c_a L_a + \rho_b c_b L_b}{2} A T_i \tag{A.1}$$

which defines the heat capacity, C, assigned to node i. Heat is transferred to and from point i by three paths: conduction from point k:

$$q_a \equiv K_a(T_k - T_i) = (T_k - T_i)\kappa_a A / L_a \tag{A.2}$$

conduction from point j:

$$q_b \equiv K_b(T_j - T_i) = (T_j - T_i)\kappa_b A / L_b \tag{A.3}$$

and internal heat generation or radiation absorbed within the layer: q_i. Equations (A.2) and (A.3) define the overall conductances, K, to adjacent nodes.

The change in internal energy of the pseudo-layer between time t_n and t_{n+1} is given by

$$\Delta U_{n\rightarrow n+1} = (q_a + q_b + q_i)\Delta t \tag{A.4}$$

The calculation of $T_{i,n+1}$ is explicit or implicit depending on when the heat gains are evaluated. Evaluating at time n gives:

$$C_i(T_{i,n+1} - T_{i,n}) = [K_a(T_{k,n} - T_{k,i}) + K_b(T_{j,n} - T_{i,n}) + q_{i,n}]\Delta t \tag{A.5}$$

which is the standard Euler explicit time integration formula. Explicit means that $T_{i,n+1}$ can be directly computed from values known at time n. On the other hand, evaluating at time n+1 gives:

$$C_i(T_{i,n+1} - T_{i,n}) = [K_a(T_{k,n+1} - T_{k,i+1}) + K_b(T_{j,n+1} - T_{i,n+1}) + q_{i,n+1}]\Delta t \tag{A.6}$$

which is Euler's standard implicit time integration formula. Implicit means that $T_{i,n+1}$ is computed from other values at time n+1. These values depend implicitly on each other and must be computed by a solution of simultaneous equations.

Clausing [1969, p 190] also gives a discussion of stability in terms of thermodynamic laws.. Rearranging equation (A.5) to solve for $T_{i,n+1}$ gives

$$T_{i,n+1} = \frac{K_a T_{k,n} + K_b T_{j,n} + q_{i,n}}{C_i}\Delta t + \left(1 - \frac{K_a + K_b}{C_i}\Delta t\right) T_{i,n} \tag{A.7}$$

There is no solution if $C_i = 0$. If C_i is sufficiently small or Δt sufficiently large, then $(K_a + K_b) \Delta t / C_i > 1$, and as $T_{i,n}$ increases $T_{i,n+1}$ must decrease, and vice versa. This is thermodynamically impossible. It shows up in a numerical solution as oscillations, i.e. "instability", in the node temperatures at each time step. These oscillations tend to quickly increase to totally meaningless values. This suggests a simple technique to determine the minimum stable time step for any element in the system. In general, the smaller the thermal mass of the element, the smaller the time step for a stable explicit solution.

To get an idea of the magnitude of the time scale and size of the mass element required for stability, consider the case of a homogeneous material with uniform layers. Then $L_a = L_b = \Delta x$, $\kappa_a = \kappa_b = \kappa$, etc., and $\alpha = \kappa/\rho c$. The solution is stable if

$$\Delta x \geq \sqrt{2\alpha\Delta t} \tag{A.8}$$

The following values (English units) apply to typical building materials.

material	typical α ft^2/hr	Δt = 3600s	900s	300s	180s	60s
wood	0.0048	Δx > 1.17"	0.59"	0.34"	0.26"	0.15"
glass wool	0.011	1.78	0.89	0.51	0.40	0.23
brick	0.011-0.013	1.86	0.93	0.53	0.41	0.24
glass	0.013	1.93	0.96	0.56	0.43	0.25
concrete	0.019-0.027	2.79	1.39	0.80	0.62	0.36
marble	0.054	3.94	1.97	1.14	0.88	0.51

Rearranging the implicit equation (A.6) to solve for $T_{i,n+1}$ gives

$$(C_i + K_a\Delta t + K_b\Delta t)T_{i,n+1} = C_i T_{i,n} + (K_a T_{k,n+1} + K_b T_{j,n+1} + q_{i,n+1})\Delta t \tag{A.9}$$

This equation shows none of the computational or thermodynamic problems of equation (A.7) indicating that the standard implicit method is stable for all time steps.

The spatial discretization error for the standard explicit and standard implicit methods is proportional to $(\Delta x)^2$. The time discretization error is proportional to Δt.

The standard explicit and standard implicit methods err in opposite directions. Therefore, a more accurate solution can be obtained by combining the two methods. Expressing this combination generally in terms of β gives:

$$\Delta U_{n \to n+1} = [(1-\beta)(q_{a,n}+q_{b,n}+q_{i,n}) + \beta(q_{a,n+1}+q_{b,n+1}+q_{i,n+1})]\Delta t \qquad \text{(A.10)}$$

where: $0 \leq \beta \leq 1$,

$\beta = 0$ corresponds to the standard explicit method,
$\beta = 1/2$ corresponds to the Crank-Nicolson method,
$\beta = 2/3$ corresponds to the Galerkin method, and
$\beta = 1$ corresponds to the standard implicit method.

Equation (A.10) can be rearranged to

$$T_{i,n+1} = \frac{[C_i+\beta'(K_a+K_b)\Delta t]T_{i,n}+[\beta'(K_aT_{k,n}+K_bT_{j,n}+q_{i,n})+\beta(K_aT_{k,n+1}+K_bT_{j,n+1}+q_{i,n+1})]\Delta t}{C_i+\beta(K_a+K_b)\Delta t}$$

where (A.11)

$\beta' = 1 - \beta$.

Note that there is no problem for $C_i = 0$ as long as $\beta > 0$. For $\beta \geq 1/2$ this method is unconditionally stable, although the solution may be oscillatory. For $\beta > 3/4$ (approximately) the solution is stable and non-oscillatory. For $\beta = 1/2$, the time discretization error is proportional to $(\Delta t)^2$.

A.2 HLITE Implementation

A.2.1 Time Integration

We may now consider the choice of time integration method. Following the discussion of Belytschko [1983, pp 55, 419, 445], the advantages of explicit time integration are:
(1) Fewer calculations per time step.
(2) Algorithm logic and structure are simple; this implies that it is good for testing new ideas.
(3) Complex nonlinearities are easily handled.
(4) It requires little core storage compared to implicit methods using direct elimination procedures.
(5) It is very reliable in terms of accuracy and completing the computation.

The only notable disadvantage is that explicit time integration is only conditionally stable so that a very large number of time steps may be required.

With regard to accuracy, since implicit methods are unconditionally stable, they can easily be used with too large a time step leading to significant time integration errors. The stability requirements for explicit time integration force the time step to be so small that the time integration error is almost always smaller than the spatial discretization error. Of course, it is also possible to use a spatial discretization that is much too large.

Certain classes of problems require small time steps to achieve suitable accuracy. Long time steps are generally suitable for inertial problems, in which low frequencies dominate the response. Short time steps are required for problems with high frequency transients such as wave propagation problems, for example, simulation of airflow in ducts. Control actions also include high frequency transients.

Press, et.al. [1985] discuss several techniques for the solution of parabolic differential equations. Simple Euler methods are generally considered to be too inaccurate. More advanced methods include Runge-Kutta methods, the Bulirsch-Stoer method, and predictor-corrector methods. They report that the predictor-corrector methods are best for smooth functions. The prediction is explicit while the correction is implicit giving a combination of the characteristics of implicit and explicit methods. The Bulirsch-Stoer method is fully explicit and tends to be more accurate, but is only appropriate for smooth functions. The Runge-Kutta methods are best for non-smooth functions, especially when adaptive step sizes are used. HLITE uses a forward Euler solution, the simplest of all explicit methods, on the grounds that it is probably of sufficient accuracy at small time steps, especially when the uncertainty in knowledge of material properties and convection coefficients is considered. In addition, HLITE is not just solving differential equations; it is creating a finite difference model of a complex reality. It does not seem necessary to solve the finite difference equations to a greater accuracy than they are modeling the physical problem.

Another idea for simulating complex systems comes from work in coupled systems using mixed time integration schemes. The most promising of the mixed time integration schemes that have been described is an implicit-explicit partitioning of the system of equations [Liu & Lin, 1983]. That is, the equations for quickly responding elements are integrated using short explicit time steps, and the entire system is integrated implicitly at a long time step. This is in agreement with the prior observation that explicit integration is most applicable to problems with rapidly changing conditions, while implicit integration is best for inertial problems. Some work has been done with combined explicit and implicit modeling for building thermal performance. Sebald [1979] used explicit calculations of the massive elements in passive solar buildings together with implicit calculations for the massless elements. This method was used to reduce the number of simultaneous equations to speed the solution. Relatively long 20-minute time steps were used. Note that this is exactly the opposite of the method used by Liu and Lin, which indicates more study on explicit/implicit methods could be appropriate.

Two previous computer programs for evaluating the lighting/HVAC interaction [Ball & Green, 1983, and Treado & Bean, 1988] used primarily explicit time integration. This choice is consistent with Belytschko's observations, in that the proposed model needs to deal with control actions (short time steps) and nonlinearities (radiation) while execution time is a secondary concern for a research program. The general philosophy of the HLITE program is to use sufficiently small time steps that nonlinearities can be reasonably approximated in an explicit manner -- the problem is linearized at each time step, and the time steps are assumed short enough that iteration is not needed.

It is likely that the simulation will include a few nodes with very small stability limits, e.g. massless nodes are never stable during explicit time integration. Ball & Green report that one node in their model (plenum air mass) had so little mass that it had to be solved implicitly. HLITE therefore uses a combination of implicit and explicit methods to perform the transient simulation. This allows a variation of the length of the time step and the number of implicit nodes to give maximum program performance. The program determines which nodes should be treated implicitly and which explicitly according to stability criteria.

This numerical approach varies significantly from the LIGHTS program [Sowell, 1989]. This is probably due to HLITE starting from a transient analysis and LIGHTS starting from a steady state analysis [Sowell, 1972, the earliest detailed analysis of the lighting - thermal problem].

A.2.2 Computation Sequence

HLITE Computation sequence for one time step:
(1) Set boundary values (from Boundary Values File).
(-) Compute air flows and solar gains. (will not be implemented)
(2) Process links to set temperature formula coefficients.
(3) Process nodes to compute explicit temperatures,
simultaneously create list of temperatures to be solved implicitly.
(4) Compute implicit temperatures.
(5) Prepare all nodes for next time step,
simultaneously compute loads at controlled temperature nodes.
(6) Set control signals (from Discrete Events File).
(7) Process controls to set remaining control signals.
(8) Adjust loads where control values have changes.
(9) Write reports.

All controlled nodes operate at the minimum time step regardless of stability (to process control signals and compute loads).

(1) Set boundary value temperatures:

The BVF gives the temperatures of certain nodes as a function of time. Consider a single node which has temperatures $T_{i,n}$ and $T_{i,n+N}$, where difference in time is $N\Delta t$ and the difference in temperature is ΔT. Then

$$T_{i,n+1} = T_{i,n} + \Delta T/N; \quad T_{i,n+2} = T_{i,n+1} + \Delta T/N; \quad \text{etc.} \tag{A.12}$$

Algorithm: At time n, $T_{i,n}$ is set and $\Delta T/N$ is stored in the node data structure. At each time step T_i is incremented by $\Delta T/N$ until time n+N. Then $T_{i,n+N}$ is set (to eliminate accumulated round-off errors), the value for $T_{i,n+N+M}$ is read, and the process repeats. Note that the BVF need not use constant time increments, and that the BVF times need not exactly match the simulation time steps. Any mismatch is handled by linear interpolation.

(-) Compute air flows and solar gains: (will not be implemented)

An airflow network could be solved for the airflows in each link as is done in AIRNET [Walton, 1989]. The present method requires that the user specify the air flows. Solar gains could also be computed and added to internal heat gains of the nodes. These features would help to make HLITE a more general building thermal simulation program, but they are not necessary for the current project.

(2) Calculate heat fluxes and coefficients:

The heat flux from node j to node i due to conduction, convection, radiation, or air flow is given by

$$q_j = K_j(T_j - T_i) \tag{A.13}$$

where K_j is the appropriate linear coefficient. Heat gains internal to the node are called q_i. The q_j and q_i values are stored in the appropriate node data structures. The K_j are also stored if the node requires implicit computation.

(3) Calculate explicit temperatures:

$$T_{i,n+1} = T_{i,n} + \frac{dT}{dt}\Delta t \tag{A.14}$$

where

$$\frac{dT}{dt} = \frac{\sum q_{j,n} + q_{i,n}}{C_i} \tag{A.15}$$

The heat gains to node i were computed and summed during link processing. For nodes in the multiple time step mode, it is not necessary to recompute dT/dt at every time step.

(4) Calculate implicit temperatures:

$$T_{i,n+1} = \frac{C_i T_{i,n} + [(1-\beta)(\Sigma q_{j,n} + q_{i,n}) + \beta(\Sigma K_j T_{j,n+1} + q_{i,n+1})]\,\Delta t}{C_i + \beta\,\Sigma K_j\,\Delta t} \tag{A.16}$$

Note that this step uses heat gains and K_j coefficients which were computed and summed during link processing as for the explicit method, step (3). Therefore, the method is not truly implicit, but relies on the relatively small changes of node temperatures at each time step to give reasonably accurate results. The solution for all temperatures at time n+1 can be expressed in matrix form as

$$[A]\,\{T\} = \{B\} \tag{A.17}$$

where $[A]$ is a square matrix, and $\{T\}$ and $\{B\}$ are vectors. The solution of the simultaneous equations is done by a sparse matrix technique which is optimal for a matrix structure in which most of the temperatures are already known from the explicit solutions.

Figure A.2 shows the structure of $[A]$ for a case involving 42 equations for 23 explicit nodes and 19 implicit nodes. In this figure
blanks represent coefficients which must be zero,
★'s represent coefficients on the diagonal which must be 1,
•'s represent other coefficients which must be non-zero, and
+'s represent coefficients which are initially zero, but which can become non-zero during solution of the equations.

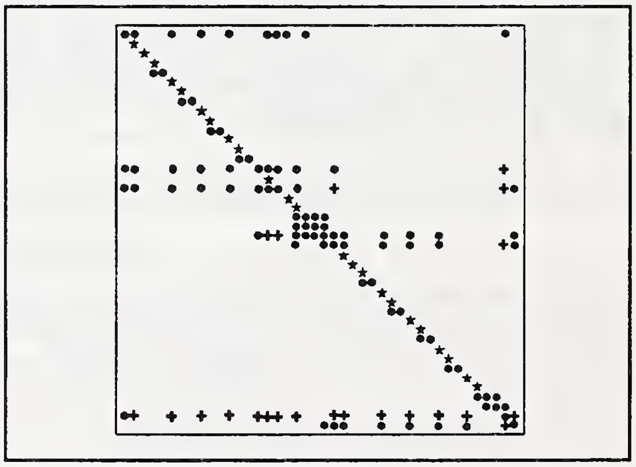


Figure A.2 Structure of [A] matrix

Although this matirx is non-symmetric, it is diagonally dominant which implies that pivoting is not necessary in the solution. This allowed the development of a simple sparse matrix solution using a row-wise Gaussian elimination algorithm. The sparse solution involves only the • and + elements of $[A]$ for a considerable savings of memory and execution time compared to a full matrix solution. As the time step is shortened, formerly implicit nodes become explicit causing the matrix to become even more sparse and quicker to solve. This is discussed further in Section A.3.3. The ordering of the equations becomes relatively unimportant when the matrix is very sparse.

(5) Compute heating and cooling loads:

A heating (cooling) load at a controlled node is defined as the rate at which heat must be added to (removed from) a node to cause the temperature to go from $T_{i,n}$ to $T_{i,n+1}$. This heat, Q, is in addition to the naturally occurring heat flows.

$$C_1(T_{i,n+1} - T_{i,n}) = (\Sigma q_j + q_i + Q)\Delta t \tag{A.18}$$

If $T_{i,n+1} = T_{i,n}$, then $Q = -(\Sigma q_j + q_i)$. Since positive q_j are heat flows into the node, a positive Q is a heating load and negative Q is a cooling load.

Loads may also be computed using the **qnd** link (B.3.4.10) which adds energy to a node based on a capacity and a signal value. The signal will probably have been generated through the controls.

(6) Set control signals:

The DEF gives the values of certain signals. Each line consists of a time, a signal name, and a value. At this point in the computation sequence, the time is incremented by Δt and the DEF is checked. If the current time is greater or equal to the time given on the DEF, the value of the signal is updated.

(7) Process controls:

The remaining signal values are processed as indicated by the control links. In contrast to the heat transfer links which connect nodes allowing heat flow in either direction, controls take an input signal (or signals) and produce an output signal. Control processing is considered to be instantaneous. Each control link is processed in the order in which it appears in the NDF. A different order of evaluation will probably give different results because of the direcional transfer of signal data.

(8) Adjust Loads:

It is possible that steps (6) and (7) will have reset a signal used in setting the temperature of a controlled node. If this has occurred, the load is recomputed to account for the new signal value.

Deadband operation of the **czn** node (B.3.3.6) requires special processing. Passing from the deadband into a controlled temperature causes the temperature to be reset to the control point and a load computed appropriate to that reset. If the time step begins with the temperature set at a control point and a "negative" load is then computed, the node temperature is revised into the deadband by the amount of that load.

(9) Write reports:

At each time step the variables which are to be integrated, as defined by the user, are processed. If the time step matches the reporting time step, the designated values are written to files or the terminal. The values displayed on the terminal allow the user to monitor the progress of the simulation.

A.3 HLITE Performance

A.3.1 Flexibility

Since HLITE was intended as a research tool, a primary goal in its development was flexibility. That is, the program should be easily modifiable when it is found that some unanticipated phenomenon must be added to the model. This was done with the NODE / LINK organization which had given such flexibility to the AIRNET [Walton, 1989] program. This process has served well in extending HLITE from its initial version to include the modeling of controls by an analogous SIGNAL/CONTROL organization. Further additions should also be relatively easy.

Unfortunately, the input for HLITE is more complex than it would be in a more tightly constrained environment. It is not designed for the casual user. A graphic interface for the connection of nodes and links could be helpful, but such programming is beyond the scope of the current project.

A.3.2 Numerical Accuracy

It is reasonable to expect that a research tool provide greater numerical accuracy than a design tool. However, the use of time consuming techniques which are significantly more accurate than the inherent uncertainties in describing the problem is debatable. The following considerations were made in developing HLITE.

The handling of nonlinearities is considered critical. Control actions produce the most important and difficult nonlinearities. One method to handle control actions within a long time step model is to note when an action occurs, or should have occurred, and adjust time steps and recompute values accordingly. An alternate method is to always use a time step that is so short that the error in the timing of the control action is negligible. This latter method corresponds to how most digital controls operate. The controlled value is sampled at a regular interval, it is compared to a setpoint, and a signal is generated which controls some piece of equipment. The decision to emulate this process led directly to the computational scheme in HLITE. A sufficiently short time step combined with a finite volume method can be solved by an explicit time integration scheme. Explicit methods are much simpler and faster for a single time step than implicit methods and are especially good at handling nonlinearities.

A forward Euler solution, the simplest of all explicit methods, may be of sufficient accuracy at small time steps, especially when the uncertainty in knowledge of material properties and convection coefficients is considered. One problem with such a simple method is that there is no control over the truncation error during the time integration. One is forced to do special test runs to insure that the time step is small enough. Examples of two such cases are presented.

Figure A.3 shows a worst case: a model of an incandescent light bulb. A high power is applied to a very small mass resulting in a rapid temperature change when the bulb is switched on. See TEST6D (C.2.6d) for details. Simulations were run using three time steps: 1, 20, and 60 seconds. The stability limit for the node representing the bulb is about 37 seconds, and the bulb reaches steady state in about three minutes. The errors in the 20 second (curve a) simulation are not large, and even the 60 second (curve b) simulation may be acceptable if the transient performance of the bulb is not of primary concern. These errors might be contrasted to the error made by assuming the bulb has no mass, which would be line c. The error in this model is of a similar magnitude to the error made by using the 60 second time step.

Most of the error in this test is due to the combination of explicit integration and linearization of the radiant interchange. A 1°C error in the average surface temperature leads to about a 0.25% error in the radiant flux. The radiant heat transfer coefficient is re-linearized at every time step, so very accurate results are obtained as the simulation approaches steady state. Errors in the transient case depend on how rapidly the temperatures are changing. In this case the initial temperature gradient is about 4.4°C (8°F) per second. Tests (TEST6B) of the fluorescent luminaire model indicate an initial temperature gradient of about 4.4°C (5°F) per minute. This indicates that time steps of up to a few minutes will not introduce a significant error in simulations using fluorescent lumianires.

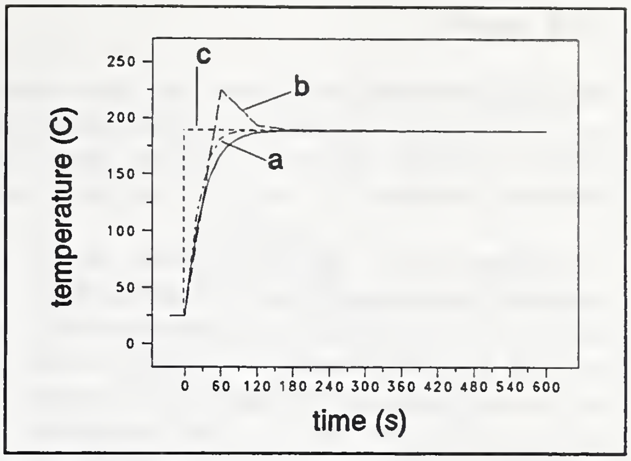


Figure A.3 A Worst Case Test

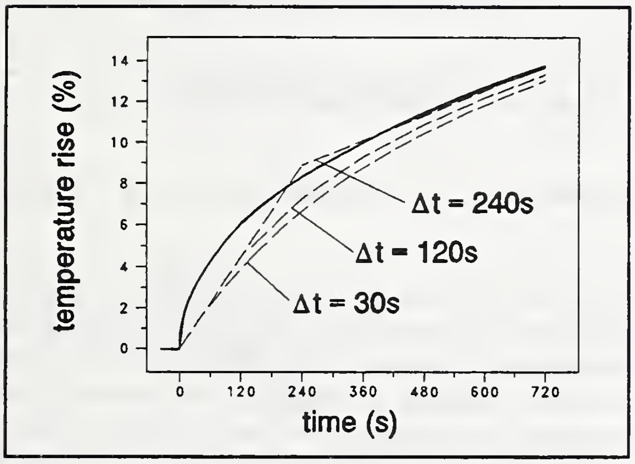


Figure A.4 Slab Surface Temperature

Figure A.4 shows the results of another test case (TEST2A). Here the surface temperature of a homogeneous slab of material which has been exposed to a sudden change in air temperature is plotted for a few minutes. Comparisons can be made to the analytic solution of this linear problem. The slab has been divided into layers having stability limits slightly greater than 300 seconds. Simulations were run using 240, 120, and 30 second time steps. Note that decreasing the time step does not improve the accuracy of the model! Creation of the difference equations from the differential equations introduces both temporal and spatial discretization errors. In these cases, where $\Delta t < 120s$, error is dominated by the spatial discretization.

In both of these examples, there appears to be a relationship between the stability limit of a given node and the level of accuracy of the model of that node. This gives an indication of the spatial and temporal discretizations necessary. Certainly, the time step must be less than the characteristic time of the phenomena studied. The time step for the controls should be sufficiently small to cause insignificant errors. Once an appropriate time step has been selected, the spatial discretizations can be selected accordingly. Such factors must be kept in mind while developing the HLITE input model.

A.3.3 Speed

Although quick execution time was a secondary goal in the development of HLITE, some excellent results have been achieved in that area. Several factors contribute to this performance: mixed implicit and explicit simulation, the use of a sparse matrix technique to solve implicit node temperatures, and simulation at different time steps for the explicit nodes.

A significant factor in building heat transfer simulation has been the solution of simultaneous equations to determine implicit node temperatures. Early in the development of HLITE it was decided to emphasize short time step simulation for the modeling of controls. This opened the possibility of doing most of the calculations explicitly, with only a few nodes requiring an implicit solution. The rules for stability limits indicate that as the simulation time step is shortened, more of the building nodes can be handled explicitly. Figure A.5 shows the stability limits of the 42 nodes used to simulated the test room. There are 2 nodes with stability limits between 0 and 60 seconds, 3 nodes between 60 and 120 seconds, 11 nodes between 120 and 180 seconds, etc. This means, for example, that using a time step of 120 seconds will result in 5 implicit nodes.

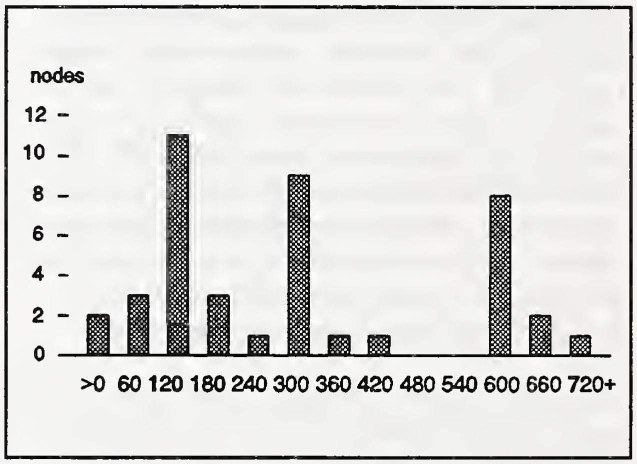


Figure A.5 Minimum Δt for Stability

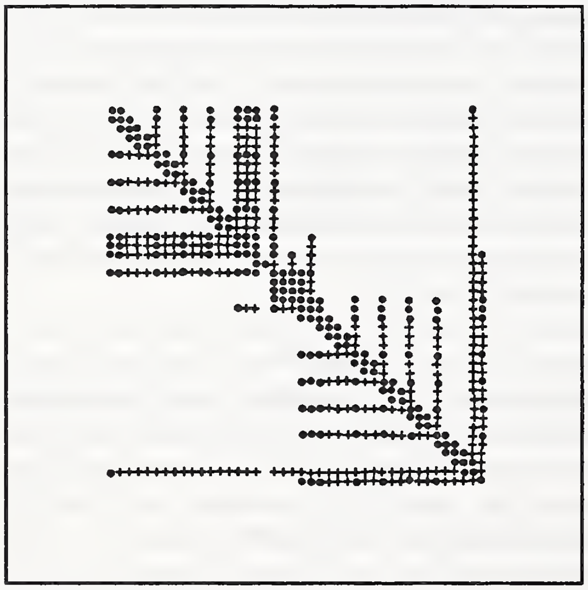


Figure A.6 Δt = 720s; time = 46.7s

It is necessary to include the option of implicit integration for some nodes, e.g. massless nodes are unstable at all time steps. This leads to mixed explicit / implicit modeling. In order to allow for this, it was necessary to develop a sparse matrix method to solve the simultaneous equations (A.17). The sparse solution involves only the non-zero elements of $[A]$ for a considerable savings of memory and execution time compared to a full matrix solution. The accompanying sketches (A.6 through A.10) show how the matrix become sparser and quicker to solve as the time step is shortened. These are from the test room model described in section C.3.

It is worth noting that the presence of low mass nodes tends to increase execution time. Either they must be treated implicitly which is slower than an explicit solution, or the time step must be reduced until they can be treated explicitly.

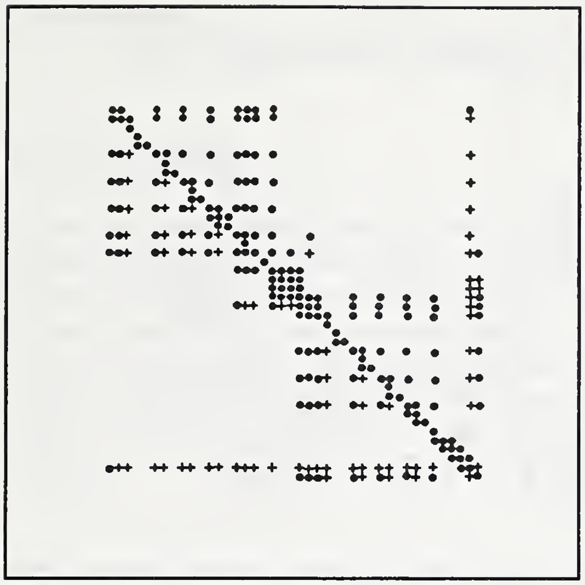


Figure A.7 Δt = 360s; Time = 51.4

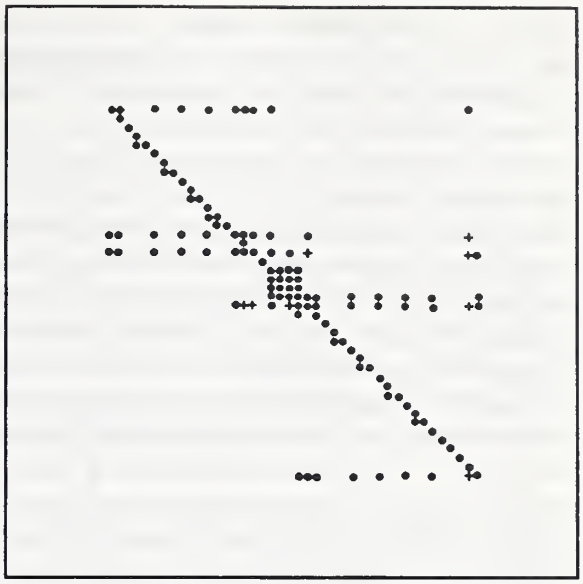


Figure A.8 Δt = 240s; time = 36.8s

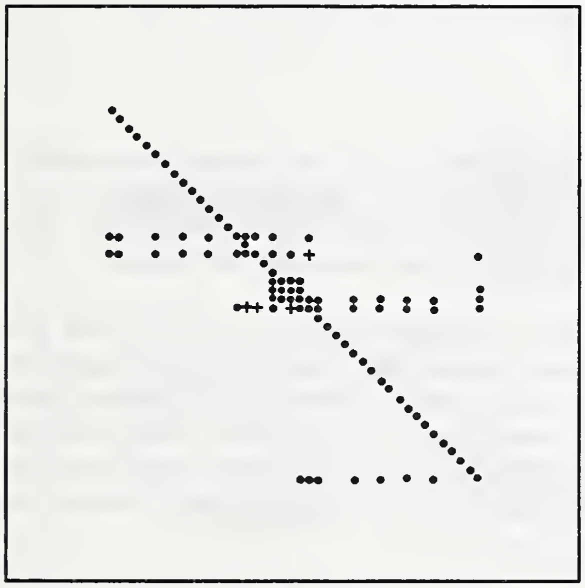


Figure A.9 Δt = 120s; time = 59.4s

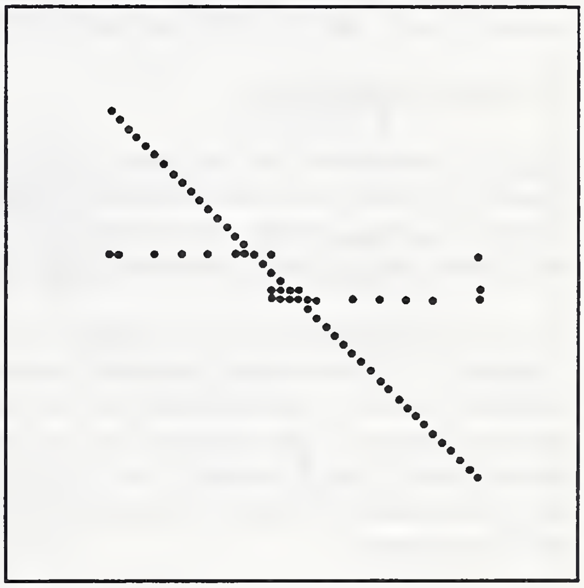


Figure A.10 Δt = 60s; time = 91.3s

Another special feature to improve program execution time is the use of multiple time steps. These operate in a manner very similar to that used for boundary value temperatures (A.2.2(1)). If a node is stable up to $N \cdot \Delta t$, where Δt is the shortest time step used in the simulation, and dT/dt has been computed at time t, the temperature of the node at time $t+M \cdot \Delta t$ is approximately $T_t + M \cdot dT/dt$. The key to fast calculation using multiple time steps is to avoid unnecessary computations. At certain time steps it is not necessary to compute dT/dt at some nodes or the heat transferred along some links. This technique trades some accuracy for some gain in execution time.

Figure A.11 summarizes the effects on execution time of the implicit / explicit trade-off and the use of multiple time steps for some nodes. These tests were run on a PC AT class computer using the test room NDF and single step 25 hour BVF described in section C.3. It is the relative rather than absolute execution times that are important. These results show that the execution time increases linearly for time steps of 1800 and 720 seconds (2 and 5 steps/hour) when there is a single Euler backward solution for almost all the nodes at each times step. Execution time remains nearly constant for time steps between 720 and 120 seconds. The cost of the increasing number of time steps is compensated by the simplicity of explicit simulation for a greater number of nodes. At shorter time steps (60 and 30 seconds) fewer implicit nodes become explicit so execution times again increase almost linearly. Below about 10 steps/hour each time step requires about 10 seconds execution time; above about 100 steps/hour each time step requires about 1 second execution time. The use of multiple time steps is helpful in this range. These results depend on the distribution of critical time steps for the nodes. Once most nodes can be treated explicitly, execution time is roughly proportional to the number of nodes and links times the number of time steps.

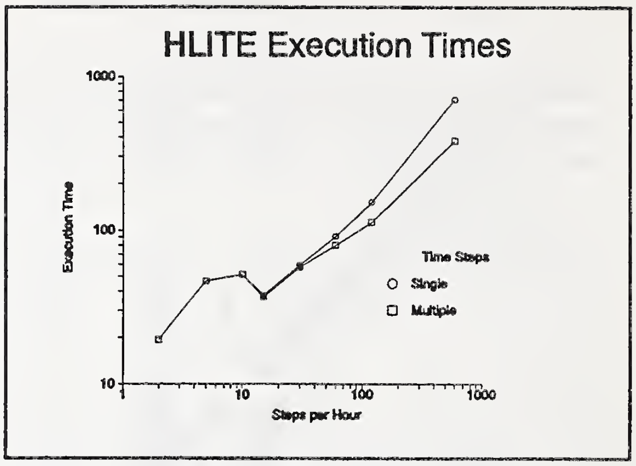


Figure A.11 Effect of Time Step

A.4 Possible Extensions

HLITE must retain a constant time step even when the system is in a near steady-state condition. However, increasing the time step may not significantly reduce execution time as shown above. A good variable time step algorithm would require an explicit calculation for nodes beyond their stability limits and controls with a variable sampling period. These extensions would require further research.

The need to develop a more flexible method for mathematical modeling of physical systems has long been recognized. It is presently the subject of considerable research using new languages and modeling tools. Lewis and Alexander [1990] believe that many of the features of such a system could be achieved using current methods. Their approach is to extend the assumptions of the explicit finite difference approximation to the full range of thermal processes in a building. They have implemented their ideas in a program called HTB2 which models heat transfer through the building structure, control actions, and HVAC equipment.

HLITE and HTB2 were developed independently, but they share the fundamental idea of explicit time integration. Both programs take advantage of the simplification that can occur using explicit methods with only the penalty of short time steps for stability. Lewis and Alexander claim that "information on the behavior of the building at high temporal resolution can be obtained without any increase in computation time". This claim is at least partially confirmed by HLITE. Many of the techniques used in HLITE could also be used in a general building simulation program.

APPENDIX B

HLITE Input and Output

B.1 Introduction

There are three primary input files for HLITE: the Network Definition File (NDF) which defines the model, the Boundary Values File (BVF) which sets the boundary value temperatures, and the Discrete Events File (DEF) which sets control signals. There are two optional files: the configuration file (CFG) which defines the interactive screen appearance, and the help (HLP) file which provides additional information to the user. Several aspects of the simulation are controlled interactively from the keyboard, and information is displayed on the screen including during the simulation. One or more files are created which can be used to generate reports and graphs. *Run HLITE with all files in the current working directory.*

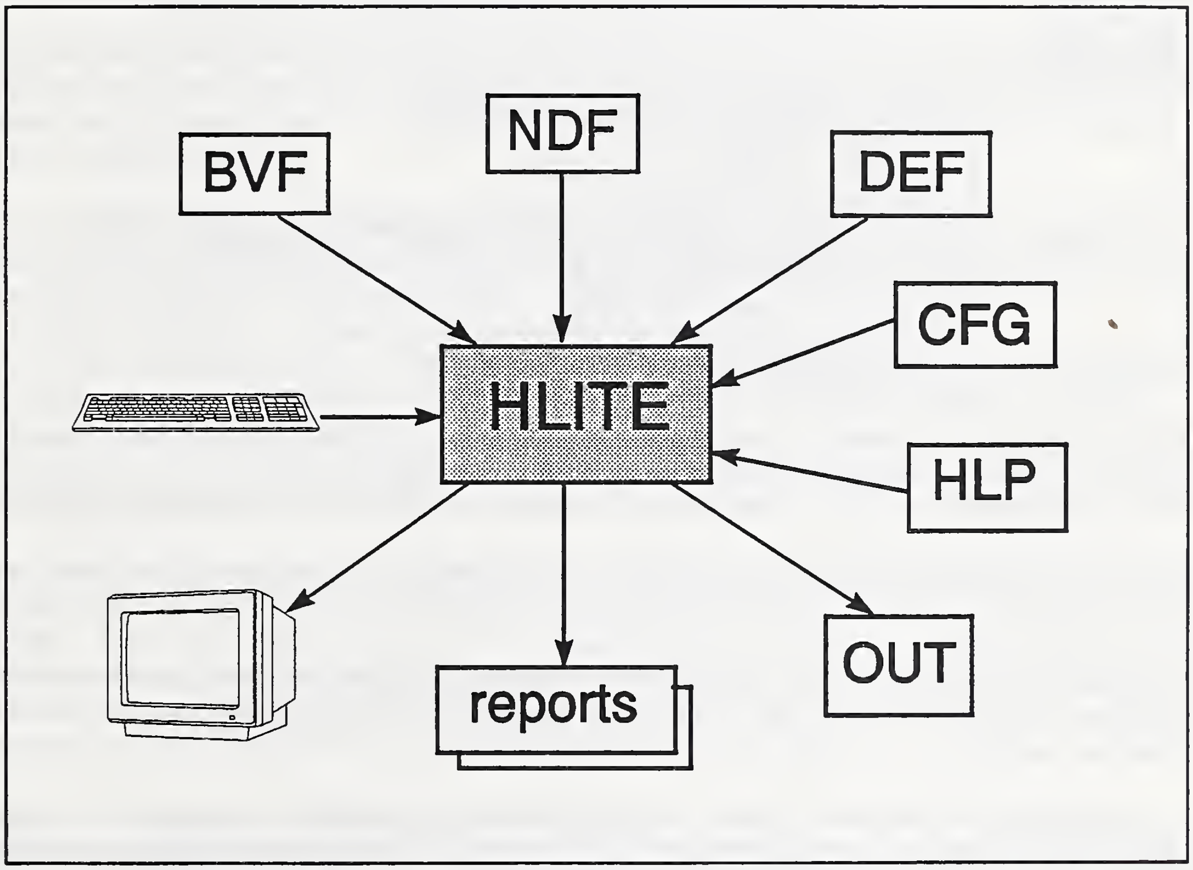


Figure B.1 HLITE Input and Output

There is a second version of HLITE called HLITEA which does not use the configuration or help files. It can use either the keyboard or redirected input for execution control.

B.2 Execution Control

The execution of HLITE is controlled interactively by various keyboard entries (no mouse). You may have to press a key to end a pause, enter a file name or a number, or enter a 'y' or 'n' in response to a question. When HLITE requires an entry, help can usually be obtained by pressing the 'F1' function key. Default values are often provided; if still in doubt, use the default. When a default value is indicated, pressing the ENTER key is equivalent to entering the value.

The execution of HLITEA may be controlled interactively, but it is intended that HLITEA be controlled by the use of a redirected input file: HLITEA <INPUT.

Since the input for both programs is essentially the same, a redirectable input file for HLITEA (file TEST1A) will now be described in detail.

```
test1a.ndf     ! (1) enter file names
test1a.bvf
test1a.def
test1a.out
y              ! (2) echo input file to output file
1              ! (3) 1 = English units, 0 = metric
1.0            ! (4) beta value
1              ! (5) test temperature limits
n              ! (6) commas in reports
n              ! (7) y = steady state simulation; remove / on next line
/  y  0.0001   ! (8) y = list stability data; 0.0001 = convergence test
y              ! (9) transient simulation
y              ! (10) list stability data
n              ! (11) y = transient initialization; remove / on next line
/  0.0001      ! (12) 0.0001 = convergence test
y              ! (13) continue transient simulation
```

The input file can be commented. Any information after a / or a ! to the end of the line is ignored by the input processor. Responses to consecutive questions may be placed on one line. They are spread out in the example above for clarity. The following numbered comments refer to the same numbers above.

(1) Enter, in order, the names of the NDF, BVF, and DEF data files and then the name of the list output file.

(2) Echoing the input file to the output file is useful for associating any error messages with the appropriate NDF input file lines. It is better to use HLITE for checking the NDF because of the interactive help messages.

(3) The units of values in all output files are determined by this parameter (0 = metric, 1 = English). Note that the output units are independent of the input units, and HLITE uses SI units in its internal calculations.

(4) The implicit integration factor refers to the value of β in equation (A.11).

(5) It is important to insure that the temperatures achieved have not exceeded the temperatures specified in the NDF for the stability calculations. Once this is known, it may be desirable to eliminate this test (set the parameter to 0) to save execution time. Enter 2 to determine the maximum and minimum node temperatures.

(6) Adding commas to reports (and quote marks around times) may make the HLITE reports easier to import into a spreadsheet program for analysis.

(7) A steady state simulation may be performed. It is done by setting all node masses to zero and iterating the node temperatures to convergence. It uses the initial values of boundary node temperatures and signal settings. It does not work for simulations which include PI control.

(8) If line (7) indicates a steady state simulation, remove the / to activate line (8). You may list the stability tests and linkages for all nodes. This list may help to identify missing or misplaced links. Convergence is controlled by entering the maximum permitted change in node temperatures in successive iterations.

(9) A transient simulation may be performed. A "no" here will terminate program execution.

(10) You may list the stability tests and linkages for all nodes.

(11) You may find the solution to a schedule that is repeated every 24 hours by performing a transient initialization. This means that the first day on the BVF and DEF is repeated until temperatures at the ends of two successive days have converged. Use this feature to set the transient heat storage in the model to realistic values.

(12) If line (11) indicates transient initialization, remove the / to activate line (12). Convergence is controlled by entering the maximum permitted change in node temperatures at the end of successive days. Reports are not generated during initialization.

(13) You will probably want to continue the transient simulation from the node temperature values determined by either steady state or transient initialization or from the initial settings in the NDF.

B.3 Network Definition File (NDF)

The primary components of the thermal network are NODEs which correspond to some volume of material with a certain temperature, and LINKs which describe the heat transfer paths connecting various NODEs. Since many links or nodes have identical thermal characteristics, these characteristics are described once as DATA, which are referenced by the LINKs or NODEs. CONTROLs are handled separately; they convert NODE and LINK data such as temperatures and flows to SIGNALs which are processed by other CONTROLs to set values at other NODEs or LINKs.

Almost every item in the NDF is given a name by the user. Names may contain no blanks (use -, _, or selective capitalization instead), and the name "report" is reserved. Names are used instead of numbers to simplify modifying the network and add an element of self-documentation. Choose names to help document the network. When an input item refers to another named input item, that item must have previously been defined. Because of the possible cross references, it may be safest to define all DATA, all SIGNALs, all NODEs, all LINKS, and all CONTROLs, in that order. On the other hand, the NDF may be more understandable if physically related items are grouped together. There is some redundancy in the input data which is used to help check for errors.

bold indicates required spelling.

[ ] indicates an optional parameter.

{ metric units | English units } gives the input units (see also section B.8).

Individual data elements, or words, in the input files are separated by blanks and the end-of-line. Lines of data should not be more than 80 characters long. Data may be continued onto the next line, but a single word may not be continued to the next line.

A ! or a / indicates that the rest of the line is a comment. Specifically, the comment indicator must appear as the first character on a line or be separated from the previous word by a blank. A ! or / embedded within a word is treated as part of that word.

Similarly, a * indicates the end of data on the file. This can be useful for placing additional information or modelling options at the end of the file after the * where they will not be read.

The first non-comment line in the NDF should be the TITLE line. It has the form:

title Anything you want to say

The data on this line is echoed as a title in the output report. It helps to relate the input file to the output file and to document the output file.

The second non-comment line in the file should be the UNITS line which has the form:

units type

where type is either **English** or **metric**. This line must precede any numeric data in the NDF.

B.3.1 Signals

signal name type value

name	name of signal point.
type	**d** = dimensionless, **t** = temperature { C | F }, **q** = (heat) power { W | Btu/h }, **e** = (electric) power { W }, or **f** = mass flow { standard L/s | standard ft^3/min (cfm)}.
value	initial value.

Signal values can be set by data on the Discrete Events File (DEF)(Section B.5) or by the operation of controls (Section B.3.5). Signal values remain at their initial value until reset by the DEF or a control. Values which are not reset remain at their initial value for the entire simulation.

B.3.2 Elements

Elements are used to consolidate often used data. It is then used by nodes or links which refer to the element name instead of repeating the data.

B.3.2.1 mat: material

element name **mat** L κ ρ c [R]

L	thickness of layer { m | ft }.
κ	thermal conductivity { W/(m·K) | Btu/(h·ft·F) }.
ρ	density { kg/m^3 | lb/ft^3 }.
c	specific heat { kJ/(kg·K) | Btu/(lb·F) }.
R	if L, κ, ρ, or c = 0, thermal resistance { $m^2 \cdot K/W$ | $h \cdot ft^2 \cdot F/Btu$ }.

The material data is used by the surface and internal nodes. This data defines the thermal properties of a conductive material. Note that every material has mass.

These data are used in TEST1A, TEST1B, TEST2A, and TEST2B.

B.3.2.2 cfr: controlled flow rate

element name **cfr** flow

flow	maximum mass flow rate { standard L/s | standard ft^3/min (cfm) }.

All air flows in the present version of HLITE are specified using **cfr** elements and **afp** links to specify flows between air nodes.

Possible extension: Use of additional elements as described in the AIRNET program to do relatively detailed combined air flow and heat transfer simulations.

B.3.2.3 hcv: variable convection coefficient

element name **hcv** δ ϵ α_+ β_+ γ_+ α_- β_- γ_-

δ ϵ	coefficients for correlation: $h = \delta + \epsilon\sqrt{J}$.
α_+ β_+ γ_+	coefficients for correlation: $h = \alpha + \beta\lvert\Delta T\rvert^{\gamma}$.
α_- β_- γ_-	+ for $\Delta T >= 0$, - for $\Delta T < 0$ ($\Delta T = T_{srf} - T_{air}$).

The convection coefficients in real buildings varies with conditions and cannot be accurately represented by a single constant value. The variable convection coefficient data, **hcv**, is designed to allow consideration of the primary factors according to the most current correlations.

Simple natural convection model:

Most energy analysis programs base their simple natural convection models on data from ASHRAE [1989]. This data includes the effect of surface slope. That is, when warm air is below a cooler horizontal surface (or cool air above a warm surface), there is more convective mixing than for air next to a vertical surface. Conversely, when cool air is below a warmer surface (or warm air above a cool surface) there is reduced convection. The values reported in Table 1, p 22.2 of [ASHRAE, 1989] are for combined convection and radiation, so a radiative component of 1.02 times emissivity of 0.9 (English units) is subtracted from the tabulated values to give the following convection coefficients:

surface tilt	heat flow	$W/m^2 \cdot K$	h: $Btu/h \cdot ft^2 \cdot {}^\circ F$
horizontal	enhanced	4.04	0.712
45°	enhanced	3.87	0.682
vertical	-	3.08	0.542
45°	reduced	2.28	0.402
horizontal	reduced	0.92	0.162

The α-β-γ values of **hcv** data for a ceiling and a floor using the above data (English units) would be:
0.162 0.0 0.0 0.712 0.0 0.0, and
0.712 0.0 0.0 0.162 0.0 0.0, respectively, since the temperature difference is always based on surface temperature minus air temperature.

Detailed natural convection model:

A more detailed study of natural convection indicates that the convection coefficient is also a function of the magnitude of the temperature difference. ASHRAE presents two sets of values which could apply to detailed convection model. From page 3.12 of [ASHRAE, 1989]:

surface tilt	heat flow	SI	English
horizontal	enhanced	$h = 1.52(\Delta T)^{.33}$	$h = 0.22(\Delta T)^{.33}$
vertical	-	$h = 1.31(\Delta T)^{.33}$	$h = 0.19(\Delta T)^{.33}$
horizontal	reduced	$h = 0.70(\Delta T)^{.33}$	$h = 0.10(\Delta T)^{.33}$

Convection coefficients for radiant heating panels is given on page 7.2 of [ASHRAE, 1987]:

surface tilt	heat flow	SI	English
horizontal	enhanced	$h = 2.18(\Delta T)^{.31}$	$h = 0.32(\Delta T)^{.31}$
vertical	-	$h = 1.78(\Delta T)^{.32}$	$h = 0.26(\Delta T)^{.32}$
horizontal	reduced	$h = 0.14(\Delta T)^{.25}$	$h = 0.021(\Delta T)^{.25}$

The α-β-γ values of **hcv** data for a ceiling and a floor using the first set of values (English units) would be:

0.0 0.10 0.33 0.0 0.22 0.33, and
0.0 0.22 0.33 0.0 0.10 0.33, respectively.

Simple forced convection model:

The other primary factor affecting convection coefficients is the operation of the HVAC system forcing air through the room. One correlation for this case uses a combined convection and radiation coefficient of 2.0 $Btu/h \cdot ft^2 \cdot {}^\circ F$. This corresponds to a convection coefficient of 1.08 $Btu/h \cdot ft^2 \cdot {}^\circ F$ or 6.14 $W/m^2 \cdot K$, which gives **hcv** δ and ϵ values (English units) of 1.08 and 0.0 respectively.

Detailed forced convection model:

A recent study by Spittler [1990] on room convection coefficients at high flow rates found that the coefficients are dependent on the room/vent geometry and the momentum of the ventilation air. A correlation of the form

$$h = \delta + \epsilon \sqrt{J}$$

where J is the dimensionless jet momentum number is proposed.

$$J = FU / \rho g V$$

where
- F = mass flow rate,
- U = velocity of the air as it enters the room,
- ρ = density of the ventilation air,
- g = acceleration of gravity, and
- V = room volume.

Letting A_e be the effective opening area of the air inlet, gives $\sqrt{J} = \dfrac{F}{\rho\sqrt{gVA_e}}$.

Spittler recommends the following correlations:

Ceiling inlet configuration (for $0.001 < J < 0.03$):

	$W/m^2 \cdot K$	$Btu/h \cdot ft^2 \cdot °F$
ceiling	$h = 11.4 + 209.7 \sqrt{J}$	$h = 2.01 + 36.9 \sqrt{J}$
walls	$h = 4.2 + 81.3 \sqrt{J}$	$h = 0.74 + 14.3 \sqrt{J}$
floor	$h = 3.5 + 46.8 \sqrt{J}$	$h = 0.62 + 8.24 \sqrt{J}$

Side wall inlet configuration (for $0.001 < J < 0.03$, $Ar < 0.3$ (Archimedes Number))

	$W/m^2 \cdot K$	$Btu/h \cdot ft^2 \cdot °F$
ceiling	$h = 0.6 + 59.4 \sqrt{J}$	$h = 0.11 + 10.5 \sqrt{J}$
walls	$h = 1.6 + 92.7 \sqrt{J}$	$h = 0.28 + 16.3 \sqrt{J}$
floor	$h = 3.2 + 44.0 \sqrt{J}$	$h = 0.56 + 7.75 \sqrt{J}$

The calculation of J requires the mass flow rate, room volume, and effective inlet area. These values are determined by the **cnv** link description.

notes:

If δ and ϵ are both set to 0, the natural convection correlation is used even when there is a system air flow.

See TEST5A for the simple convection models and TEST5B for the detailed convection models.

The algorithm implemented in HLITE does not check the limits for the forced convection correlation, since no other correlation outside those limits is available.

B.3.2.4 lmp: lamp

element name lmp power light leff area emit mass c N
T_1 Q_1 L_1
...
T_N Q_N L_N

power	rated input power { W }.
light	rated light output { lumen }.
leff	conversion factor: Watts • leff = lumens.
area	surface area { m^2 | ft^2 }.
emit	emittance.
mass	mass { kg | lb }.
c	average specific heat { kJ/(kg·K) | Btu/(lb·F) }.
N	number of points used to define power and light curves.
Ti	minimum lamp wall temperature { C | F }.
Qi	fraction of rated power at given temperature.
Li	fraction of rated light output at given temperature.

This is a model of a fluorescent lamp. The rated input power is for a single lamp and includes the power dissipated in the ballast.

Area, mass, and specific heat are used for radiant and convective interchange and transient heat transfer analysis. Additional data must be included in the links and nodes that make up the complete description of the luminaire.

The characteristics of fluorescent lamps are strongly dependent on the mercury vapor pressure within the lamp which in turn depends on the minimum lamp temperature. The following performance data was obtained from figure 8-34 of the Lighting Handbook [IES, 1984]:

Ti	Qi	Li
0.0	0.661	0.022
20.0	0.672	0.068
40.0	0.704	0.165
60.0	0.818	0.476
70.0	0.894	0.701
80.0	0.971	0.865
90.0	0.999	0.960
100.0	0.965	0.995
120.0	0.858	0.850
140.0	0.721	0.697
215.0	0.000	0.000

This may be used as a reasonable approximation of fluorescent lighting unless more specific data are available. These data are converted to a spline fit.

B.3.2.5 eqp: equipment

element name **epq** power area emit mass c

power	rated input power { W }.
area	surface area { m^2 | ft^2 }.
emit	emittance.
mass	mass { kg | lb }.
c	average specific heat { kJ/(kg·K) | Btu/(lb·F) }.

This is the generic equipment element. It describes a single unit of equipment; multiple units may be considered together under the equipment link command.

Area, emittance, mass, and specific heat are used for radiant and convective interchange and transient heat transfer analysis. Additional data must be included in the equipment node and equipment link to complete the description of the equipment.

B.3.3 Nodes

Each node description contains eight identical types of data plus other data specific to the individual type of node. The first word on a node data line is **node**. The second word is a user specified identifier which is unique for each node. The third word is one of the following node types:
air lyr srf mas eqp czn

The fourth word is a single character which tells how the node temperature is determined:
- **b** indicates a temperature from the boundary values file,
- **c** indicates a controlled temperature, and
- **v** indicates a variable temperature.

The fifth word is the fraction of the stability limit, fsl. If the actual time step exceeds fsl times the maximum stable time step, then the implicit solution method is used.

The sixth and seventh words are the expected minimum and maximum node temperatures. These values are used to determine the maximum stable time step for the node.

The eighth word is the initial temperature of the node.

B.3.3.1 air: air

node name **air** type fsl Tmin Tmax Ti vol [sig]

type	temperature type: **b**, **c**, or **v**.
fsl	fraction of stability limit
Tmin	minimum expected temperature { C | F }.
Tmax	maximum expected temperature { C | F }.
Ti	initial temperature { C | F }.
vol	volume { m^3 | ft^3 }.
sig	if type=**c**, name of signal setting the temperature, type **t**.

The air density and specific heat used to determine the heat capacity of the air node are set at standard default values. Together with the volume, they determine the heat capacity of the air.

The following values can be reported for air nodes:
T temperature - instantaneous or averaged.
Q load (= power required to maintain the set temperature, if it is set)
- instantaneous, integrated, or averaged.

B.3.3.2 srf: surface

node name **srf** type fsl Tmin Tmax Ti area emit [sig]

type	temperature type: **b**, **c**, or **v**.
fsl	fraction of stability limit
Tmin	minimum expected temperature { C | F }.
Tmax	maximum expected temperature { C | F }.
Ti	initial temperature { C | F }.
area	surface area { m^2 | ft^2 }.
emit	emittance.
sig	if type=**c**, name of signal setting the temperature, type **t**.

The heat capacity of a surface node is determined by the material element which is linked to the node.

The surface area is used in determining the conductive and convective heat transfer to adjoining nodes.

The emittance is used in determining radiant heat transfer to other surface nodes.

The following values can be reported for surface nodes:
T temperature - instantaneous or averaged.
Q load (= power required to maintain the set temperature, if it is set)
- instantaneous, integrated, or averaged.

B.3.3.3 lyr: layer

node name **lyr** type fsl Tmin Tmax Ti area [sig]

type	temperature type: **b**, **c**, or **v**.
fsl	fraction of stability limit
Tmin	minimum expected temperature { C | F }.
Tmax	maximum expected temperature { C | F }.
Ti	initial temperature { C | F }.
area	surface area { m^2 | ft^2 }.
sig	if type=**c**, name of signal setting the temperature, type **t**.

The following values can be reported for layer nodes:

T temperature - instantaneous or averaged.

Q load (= power required to maintain the set temperature, if it is set) - instantaneous, integrated, or averaged.

B.3.3.4 mass: mass

node name **mas** type fsl Tmin Tmax Ti area mass c emit [sig]

type	temperature type: **b**, **c**, or **v**.
fsl	fraction of stability limit
Tmin	minimum expected temperature { C | F }.
Tmax	maximum expected temperature { C | F }.
Ti	initial temperature { C | F }.
area	surface area { m^2 | ft^2 }.
mass	mass { kg | lb }.
c	specific heat { kJ/(kg·K) | Btu/(lb·F) }.
emit	emittance.
sig	if type=**c**, name of signal setting the temperature, type **t**.

This is a generic thermal mass. It may be used in place of surface and layer nodes. It improves the modeling of thin sheets of material which are exposed to air at both surfaces. Thin, high conductivity masses can usually be adequately modeled by a single node, which is effectively assuming that the entire mass, including both surfaces, is at a uniform temperature. This is expressed mathematically in terms of the Biot number: $Bi = hL/k$. In a thin plate mass the error in assuming that the internal temperature is uniform is less than 5% when $Bi < 10\%$ [Kreith, 1973, p 140]. The surface area may represent either one side or both sides of the mass, depending on how it is convectively and radiatively linked to other nodes.

If the node is being used to represent equipment, the type should be **v**. Control of an equipment node is determined by the appropriate **link** command.

The following values can be reported for mass nodes:

T temperature - instantaneous or averaged.

Q load (= power required to maintain the set temperature, if it is set) - instantaneous, integrated, or averaged.

B.3.3.5 eqp: equipment node

node name **eqp v** fsl Tmin Tmax Ti element N

fsl	fraction of stability limit.
Tmin	minimum expected temperature { C | F }.
Tmax	maximum expected temperature { C | F }.
Ti	initial temperature { C | F }.
element	material element, type **eqp** or **lmp**.
N	equipment multiplier.

The surface properties and heat capacity of an equipment node are determined by data in the element and link commands relating to this node.

An equipment node is always subject to a control (defined in the link command) and therefore is simulated at the shortest possible time step in order to catch any changes in the control point status.

The following values can be reported for equipment nodes:
- T temperature - instantaneous or averaged.
- Q power - electric power into the equipment.

B.3.3.6 czn: controlled zone node

node name **czn** type fsl Tmin Tmax Ti vol shtmp shcap sctmp sccap

type	temperature type: **c**.
fsl	fraction of stability limit
Tmin	minimum expected temperature { C | F }.
Tmax	maximum expected temperature { C | F }.
Ti	initial temperature { C | F }.
vol	volume { m^3 | ft^3 }.
shtmp	name of signal for the heating set point temperature, type **t**.
shcap	name of signal for the heating capacity, type **q**.
sctmp	name of signal for the cooling set point temperature, type **t**.
sccap	name of signal for the cooling capacity, type **q**.

The air density and specific heat used to determine the heat capacity of the air node are set at standard default values. Together with the volume, they determine the heat capacity of the air.

The shtmp and sctmp signals allow the definition of a deadband with floating temperature between set points, or a set temperature if the set point values are equal. The set point values are guaranteed to be equal if smin and smax refer to the same signal point.

The following values can be reported for czn nodes:
- T temperature - instantaneous or averaged.
- Q load (= power required to maintain the set temperature, if it is set)
  - instantaneous, integrated, or averaged.

B.3.4 Links

Links describe the heat transfer paths connecting various nodes.

B.3.4.1 knd: constant heat transfer coefficient

Link name **knd** K node1 node2

K	overall conductance value { W/K | Btu/(h·°F) }.
node1	name of node 1.
node2	name of node 2.

This is a generic simple link. It can be used to connect any two nodes. Note that this link makes no reference to the surface areas of the nodes; its effect is included in the K value. The heat transfer from node1 to node2 is given by

$$q = K (T_1 - T_2).$$

This link is used in TEST2D.

B.3.4.2 cnd: conduction

link name **cnd** element node1 node2

element	name of element, type **mat**.
node1	name of node 1, type **lyr** or **srf**.
node2	name of node 2, type **lyr** or **srf**.

The overall conductance is determined by the node areas and the **mat** element thermal properties: $K = A \kappa / L$.

The element properties are also used in determining the node heat capacities: $C = A L \rho c$. Half of C is assigned to each node.

The surface areas of the two nodes must be identical.

This link is used in TEST1A, TEST1B, TEST2A, and TEST2B.

B.3.4.3 hcc: constant convection

link name **hcc** h node1 node2

h	convection coefficient { $W/(m^2 \cdot K)$ | $Btu/(h \cdot ft^2 \cdot F)$ }.
node1	name of node 1, type **srf** or **mas** or **eqp**.
node2	name of node 2, type **air** or **czn**.

The constant convection coefficient is a simple model which is primarily useful in developing HLITE test cases which can be solved analytically. The heat transfer from the surface to the air is given by: $q = h\ A\ (T_1 - T_2)$, where A is the area of the surface or mass.

This link is used in TEST2A and TEST2B.

B.3.4.4 cnv: variable convection

link name **cnv** data node1 node2 sig [VAe Fmax]

data	name of data, type **hcv**.
node1	name of node 1, type **srf** or **mas** or **eqp**.
node2	name of node 2, type **air** or **czn**.
sig	name of signal giving a flow value, type **f**.
VAe	room volume times effective area of flow opening { m^5 | ft^5 }, used to compute velocity from flow rate.
Fmax	maximum flow rate { standard L/s | standard ft^3/min (cfm) }.

This links all the pieces necessary to describe a variable convection heat transfer. The appropriate **hcv** data must be used; there must be a node with a surface area and an air node; and the signal must be describing an air flow unless the special signal name **null** is used. A **null** signal means that the flow rate portion of the **hcv** data is never used.

The VAe value is necessary to define the jet momentum number (section D.2) and Fmax is used in determining the maximum possible convection coefficient to evaluate stability. These values are included in the link data because they are room specific and some care must be used in defining values when there are multiple air inlets in the room or the room is described by multiple air nodes.

The heat transfer from the surface to the air is given by: $q = h\ A\ (T_1 - T_2)$, where A is the area of the surface or mass.

The following value can be reported for variable convection links:

h convection coefficient - instantaneous or averaged.

This link is used in TEST5A and TEST5B.

B.3.4.5 afp: air flow path

link name **afp** element node1 node2 sig

node1	name of node air flows from, type **air** or **czn**.
node2	name of node air flows to, type **air** or **czn**.
sig	name of signal giving a flow value, type **f**.

The mass flow rate from node 1 to node 2 is set to the current value of the signal.

The user is responsible for determining that a mass balance is maintained at each air node. Air flow path links should be described before any other links because they are referenced by the cnv links.

Possible extension: use additional links as described in the AIRNET program to do relatively detailed combined air flow and heat transfer simulations.

The following value can be reported for air flow path links:

F	mass flow rate - instantaneous or averaged.

B.3.4.6 rad: radiant exchange

link name **rad** $\mathcal{F}_{12}$ node1 node2

$\mathcal{F}_{12}$	view factor - includes surface emissivity effect.
node1	name of node 1, type **srf** or **mas** or **eqp**.
node2	name of node 2, type **srf** or **mas** or **eqp**.

This is a simple radiation exchange model in which the effect of the emittances of both surfaces have been combined into the $\mathcal{F}$ view factor as defined by Kreith [1973, p 260]. Therefore, the emittance values given with the node data are not used. The user must insure that the data is consistent.

Two useful formulae [Kreith, 1973, pp 260,261] for the calculation of $\mathcal{F}$ are:

(1) two infinitely large parallel plates:

$$\mathcal{F}_{12} = 1 / (1/\epsilon_1 + 1/\epsilon_2 - 1)$$

(2) two concentric spheres or cylinders:

$$\mathcal{F}_{12} = 1 / ((1-\epsilon_j)/\epsilon_i + 1 + A_1(1-\epsilon_2)/A_2\epsilon_2)$$

The radiation heat transfer coefficient, K_j in eqn (A.13), is computed from $K_j = \sigma A_i \mathcal{F}_{ij}(T_i^2+T_j^2)(T_i+T_j)$. For surface temperatures near 300K, a change of 1°C in one of the surface temperatures during a single time step corresponds to an error in the average value of K during that step of 0.25%; a change of 10°C corrsponds to an error of 2.55%. Temperatures typically change by less than these amounts.

This link is used in TEST4A.

B.3.4.7 vfm: view factor matrix

link name **vfm** N
node1
 F_{11} F_{12} ... F_{1N}
...
nodeN
 F_{N1} F_{N2} ... F_{NN}

N	number of nodes (surfaces) in matrix.
nodei	name of node i, type **srf** or **mas** or **eqp**.
F_{ij}	view factor from surface i to surface j.

This link uses the view between a set of N surfaces, and the emittances given with the node data to compute the radiation interchange between the set of surfaces. The algorithm uses the view factors and emittances to compute script F view factors (described in section B.3.4.6). These are computationally more efficient than using an iteration to compute radiosities.

The sum of the view factors in each row should equal 1 (tested to $\pm1\%$) and reciprocity should apply: $A_i \cdot F_{ij} = A_j \cdot F_{ji}$ (tested to $\pm1\%$).

This link is used in TEST4B.

B.3.4.8 lum: luminaire

link name **lum** nodeL offset sig balf nodeB N
node1 fr1
...
nodeN frN

nodeL	name of lamp node, type **eqp**.
offset	average lamp temperature - minimum lamp temperature { C | F }.
sig	name of signal point: 0 = off; 1 = on, type **d**.
balf	fraction of input energy to the ballast.
nodeB	if balf > 0, name of node containing the ballast, type **mas**; otherwise use **null** as a place holder.
N	number of nodes (surfaces) in light distribution matrix.
nodej	name of node j, type **srf** or **mas** or **eqp**.
frj	fraction of light from luminaire absorbed at surface.

The luminaire link is used to describe one or more luminaires.

A reasonable approximation for the offset value is that the minimum bulb temperature is 5°C (9°F) less than the average temperature. This is based on some measurements in the test facility and may not be applicable to all conditions.

The following value can be reported for luminaire links:

Q power used - instantaneous, integrated, or averaged.

This link is used in TEST6A and TEST6B.

B.3.4.9 eqp: equipment link

link name **eqp** node sig

node	name of equipment node, type **eqp**.
sig	name of signal point: fraction of rated wattage, type **d**.

The equipment link is used to describe one or more pieces of equipment which release heat into the room. The actual power into the node is the rated power times the signal value.

The equipment node can be linked to the room air by convection and to other surfaces by radiant interchange. Equipment with cooling fans may require very high convection coefficients for accurate modeling.

This link is used in TEST6C.

B.3.4.10 qnd: heat flux link

link name **qnd** [cap | qsig] node fsig

cap	cooling capacity [Btu/h | W] -or-
qsig	name of a signal: capacity, type **q**.
node	name of node, type **air**, **lyr**, **srf**, or **mas**.
fsig	name of a signal: fraction of rated capacity, type **d**.

This link adds energy to the node at the rate cap • sig. The capacity may be constant or it can be made to vary by using the capacity signal. An ideal heating coil is modeled by adding heat to an air node. A radiant panel could be modeled by adding heat to a surface node.

This link is used in TEST8B and TEST8C where it is used to represent an ideal sensible cooling coil by using a negative capacity to indicate the removal of heat. TEST8B uses the capacity signal; TEST8C uses the capacity value input.

note: HLITE determines whether the capacity is a value or a signal by first attempting to process the word as a number and, if that fails, attempting to find a matching signal name. Therefore, the signal cannot have a name that could be interpreted as a number.

B.3.5 Controls

Controls link signals to each other and to various nodes and links in the network. They may be used to control the system or to process values for reports. Some signals have units which determines conversion during reporting. The dimensionless signal corresponds to the void class in C. There are controls which act as sensors converting some node of link value to a dimensionless signal. Corresponding to each such sensor is a control which has an inverse operation. These are primarily useful for generating report values. Another group of controls do simple mathematical operations on the signal values. The final group of controls perform the classical control operations, such as proportional/integral control. The sequence in which controls are listed on the NDF is important. When the input signal for "control A" is an output from "control B", list "control B" first.

B.3.5.1 ts, ps, eps, fs: sensors

All sensors have the following form:

control name type node out offset gain tau

node	name of node for sensed value, type according to type of control.
out	name of output signal, type **d**.
offset	adjustment factor
gain	adjustment factor
tau	sensor time constant

If tau = 0.0, output = (Tnode - offset) / gain.

If tau > 0.0, output = Tnode - (Tnode - out*) $\cdot$ exp(Δt $\cdot$ tau)
where out* is the value of output at the prior timestep.

If a value between 0 and 1 is required as a controller input, the offset is the minimum allowable temperature and the gain is the maximum allowable temperature minus the minimum temperature.

The following sensor types are available:

ts	temperature sensor.
ps	power sensor. The node should have control = **c**.
eps	electric power sensor. The node should be type **eqp**.
fs	flow rate sensor. Instead of a node, this sensor must refer to a type **afp** link.

B.3.5.2 st, sp, sep, sf: set value

These controls are the opposite of sensors. They set a signal of the appropriate type and units. (Other controls operate to set dimensionless signals.) They all have the following form:

control name type in out offset scale

in	name of input signal, type **d**.
out	name of output signal, type according to type of control.
offset	adjustment factor
scale	adjustment factor

out = in · scale + offset.

The following set types are available:

st	set temperature. Signal type **t**.
sp	set power. Signal type **q**.
sep	set electric power. Signal type **e**.
sf	set flow rate. Signal type **f**.

If the input signal value was created by a sensor with a given offset and gain, the original value can be computed by a set using the same offset and scale equal to the gain.

B.3.5.3 mod: modify signal

control name **mod** in out offset scale

in	name of input signal, any type.
out	name of output signal, any type.

out = in · scale + offset

This allows special processing of signals. If you are using this function to do a units conversion, check the units in section B.8.

B.3.5.4 bin, inv, pos: unary controls

These controls take a single input signal and create an output signal without any other parameters. All unary controls are of the form:

control name type in out

in	name of input signal, type **d**.
out	name of output signal, type **d**.

The following unary control types are available.

bin	binary signal:	out = 1 if in > 0.0,
		out = 0 if in <= 0.0

inv	invert signal:	out = 1.0 - in

pos	positive signal:	out = in if in > 0.0,
		out = 0 if in <= 0.0

B.3.5.5 and, or, add, sub, mul, div: binary controls

These controls take two input signal and create an output signal without any other parameters. All unary controls are of the form:

control name type in1 in2 out

in1	name of input signal #1.
in2	name of input signal #2.
out	name of output signal.

The following unary control types are available.

and	and signals:	out = 1 if in1 > 0 and in2 > 0,
		out = 0 otherwise.

or	or signals:	out = 1 if in1 > 0 or in2 > 0,
		out = 0 otherwise.

add	add signals:	out = in1 + in2.
sub	subtract signals:	out = in1 - in2.
mul	multiply signals:	out = in1 · in2.
div	divide signals:	out = in1 / in2.

B.3.5.6 sum: sum signals

control name **sum** N in1 ... inN out

N	number of input signals
in1	name of input signal1.
...	
inN	name of input signalN.
out	name of output signal.

$$out = \sum_{i=1}^{N} in_i$$

B.3.5.7 ulc, llc, uls, lls: limits

These controls process a single input signal and a parameter or a second input signal to create an output signal. All limit controls are of the form:

control name type in out [limit | limsig]

in	name of input signal, type **d**.
out	name of output signal, type **d**.
limit	limit value -or-
limsig	name of limit signal, type **d**.

The following types of limit controls are available.

ulc	upper limit control:	out = in - limit.
llc	lower limit control:	out = limit - in.
uls	upwer limit set point:	out = 1 if (in - limit) > 0, out = 0 otherwise.
lls	lower limit set point:	out = 1 if (limit - in) > 0, out = 0 otherwise.

Use the upper and lower limit controls to prepare an error signal for the proportional controls. If the input signal to a **ulc** is greater than the limit value, a positive output results; e.g., if the air temperature is too high, the positive signal could activate some cooling. If the input signal to a **llc** is lower than the limit value, a positive output results; e.g., if the air temperature is too low, the positive signal could activate some heating.

Use the upper and lower limit setpoint controls to prepare a signal to turn something on or off.

B.3.5.8 pc: proportional control

control name **pc** in out Kp

in	name of input signal, type **d**.
out	name of output signal, type **d**.
Kp	proportional factor

$out = Kp \cdot in; \; 0 <= out <= 1.$

This is a model of a simple proportional controller where the input signal is the error signal. This error signal is some sensed value minus a setpoint value which is obtained by using other control functions. The output signal is limited to values between 0 and 1.

B.3.5.9 pic: proportional-integral control

control name **pic** in out Kp Ki

in	name of input signal, type **d**.
out	name of output signal, type **d**.
Kp	proportional factor
Ki	integral factor

$out = out^* + Kp \cdot (in - in^*) + Ki \cdot (in + in^*); \; 0 <= out <= 1.$
where out* and in* are the values of out and in at the previous time step.

This is a model of a simple proportional integral controller where the input signal is the error signal. This error signal is some sensed value minus a setpoint value which is obtained by using other control functions. The output signal is limited to values between 0 and 1. The Kp and Ki factors must be tuned for the specific problem and time step.

This is similar to an "incremental" PI algorithm described by Stoecker [1989, eqns (17-37) and (18-3)].

B.3.6 Time Steps

times nr t1 t2 .. tn

nr	number of different time steps
t1 ... tn	shortest to longest step (in seconds)

Each successively longer time step must be a simple multiple of the previous time step. One hour (3600 sec) must be a simple multiple of the longest time step.

examples:

```
times 5 12  60 300 1200 3600
times 3 60 180 900
times 1 60
```

When multiple time steps are used, each node uses the longest time step that it can while maintaining stability for explicit simulation. See also Section A.3.2.

B.3.7 Reports

B.3.7.1 List Reports

report file N incr
type1 name value action format
...
typeN name value action format

file	name defining this report, an MS-DOS file name (no \'s).
N	number of items reported.
incr	time increment between reports { sec }.
typej	**n** for node, **l** for link, or s for signal.
name	identifies which one.
value	selects the value to be reported: **T** = temperature { C | F], **Q** = power { W | Btu/h }, **f** = mass flow rate { kg/s | lbm/s }, **h** = convection coefficient { $W/(m^2 \cdot K)$ | $Btu/(h \cdot ft^2 \cdot °F)$ }, **0** = any signal value {units from signal type}.
action	**i** = instantaneous value, s = integrated (trapezoidal) value, **a** = averaged value.
format	C language format specifier, e.g. %8.3f or %12.5e .

The time step must be defined before reports are defined.

The value reported must be one permitted for the particular node or link. Power is defined for controlled nodes only. It is the rate a which heat must be added to the node to maintain the specified temperature. This may also be called a load. A cooling load would be a negative number. Equipment nodes (B.3.3.5) are an exception: power represents the electric power into the equipment node whose temperature is allowed to float.

Each report is written to a different file. The maximum number of reports is 5, which is set by the parameter MAXRPTFL (in the source code).

This output can be modified for direct import to spreadsheet programs by one of the interactive execution control parameters. In that case it may also be desirable to eliminate blanks from the report file, which can be done by changing the format specifiers, e.g. %.3f or %.5e .

(Possible extension: reports that can also be activated from the DEF. This would allow printing of values at irregular intervals or for just a few time steps during the simulation.)

B.3.7.2 Screen Report

display N incr
type1 name value action format
...
typeN name value action format

This defines the values displayed at the bottom of the screen during simulation. The input is similar to that for list reports except no file name is needed. Only a limited number of values can be displayed because of the width of the screen. The first value displayed is the simulation time in day/hr:mn:sc format (see B.4). The second varaible is the real clock time since the start of simulation. The third value displayed is the first value under the **display** command, etc.

B.4 Boundary Values File (BVF)

The BVF sets node temperature values. The first record of the boundary values file tells the number of values in each record. The second record gives the names of the nodes corresponding to the respective values. The remaining records give a time and the values for that time. During simulation, values at intermediate times are obtained by linear interpolation. The BVF determines simulation start and stop times.

Consider the following BVF (file TEST1B.BVF):

```
1
node-5
001/00:00:00  50.0
001/01:00:00  50.0
001/02:00:00  32.0
001/03:00:00  50.0
001/06:00:00  50.0
*
```

The 1 on the second line indicates that each line of data will report one value, the temperature of node-5 as indicated on the third line, which was also identified as a boundary value node in the NDF. The BVF indicates that this temperature is constant for the first hour, decreases to 32°F at the end of the second hour, returns to 50°F during the next hour, and remains at that value until the end of simulation. Note that the time between data points need not be constant. The times set the total period of simulation to six hours. The values, in this case temperature, should be in the same units as used in the NDF.

The * on the last line indicates the end of BVF information. This is the same as for the NDF. Also similar is the use of / or ! to indicate a comment.

Times are indicated by:

day number /	3 digits, 000 to 999
hour :	2 digits, 00 to 23
minute :	2 digits, 00 to 59
second	2 digits, 00 to 59

(Possible extension: the addition of (or a separate file of) heat gains for selected nodes at selected times with interpolation. This might be used to model solar gains.)

B.5 Discrete Events File (DEF)

The DEF sets signal values. This allows the changing of set points, etc., according to a very general user-defined schedule.

The following contents of TEST1B.DEF indicate that the control signal named "inside" is set to 77°F for one hour and then returned to 68°F, which is its initial value as specified in the NDF. The values should also be in the same units as used in the NDF.

```
001/00:00:00
001/04:00:00  inside  77.0
001/05:00:00  inside  68.0
002/00:00:00
*
```

A single time equal to or greater than the final BVF time is required. Other times outside the simulation period specified by the BVF are ignored.

B.6 Configuration File (CFG)

HLITE.CFG is an optional configuration file. It determines how the screen appears during processing. This is normally handled by default, but the user can create a custom configuration file by using the MAKECFGT program. The configuration file controls the size of the cursor and the foreground and background colors for various types of messages.

B.7 Help File (HLP)

HLITE.HLP contains the text that is displayed when the F1 key is pressed. Therefore, this file must be present for interactive help to work, but the rest of HLITE will run without it.

B.8 Units

"External" units are those used in the input and output files.

symbol	description	internal	external metric	external English
t Δt	time or time difference	s	s	s
	simulation time	s	day/hh:mm:ss 000/00:00:00 through 999/23:59:59	day/hh:mm:ss
T	temperature	°C	°C	°F
T	absolute temp.	°K (= °C + 273.15)	(not used externally)	
L	length or thickness	m	m	ft
A	area	m^2	m^2	ft^2
V	volume	m^3	m^3	ft^3
m	mass	kg	kg	lb (mass)
ρ	density	kg/m^3	kg/m^3	lb/ft^3
F	air flow rate (volumes referenced to standard air)	kg/s	L/s $\rho = 1.2$ kg/m^3	ft^3/min (cfm) $\rho = 0.075$ lb/ft^3
Q	power electric power	W W	W W	Btu/h W
E	energy electric energy	J J	kJ Wh	Btu Wh
κ	thermal conductivity	W/(m·K)	W/(m·K)	Btu/(h·ft·°F) [note that the English values are 1/12 of those for Btu·in/(h·ft^2·°F)]
h	convection coefficient	W/(m^2·K)	W/(m^2·K)	Btu/(h·ft^2·°F)
c	specific heat	J/(kg·K)	kJ/(kg·K)	Btu/(lb·°F)
K	overall conductance	W/K	W/K	Btu/(h·°F)
C	overall heat capacity	J/K	kJ/K	Btu/°F

APPENDIX C

HLITE Test Cases

C.1 Program Testing

Testing is an essential part of the development of a computer simulation. Inaccurate results are not always due to program errors as shown in the SERI report on validation of building energy analysis programs [Judkoff, 1983] which identified seven error sources classified into two groups. External sources are those which are not under the control of the developer of the computer code. These errors include:
(1) differences between the actual weather around the building and the weather used in the simulation;
(2) differences between the actual effect of occupant behavior and those effects assumed by the user;
(3) user error, including inappropriate simplifying assumptions, in deriving the input files; and
(4) differences between the actual thermal and physical properties of the building and those input by the user.

Internal error sources are those contained within the coding of the program. They include:
(1) differences between the actual heat/mass transfer mechanisms and the algorithmic representations of those mechanisms;
(2) differences between the actual interactions of heat/mass transfer mechanisms and those interactions between the algorithms; and
(3) coding errors.

Three types of tests have been used to validate building energy analysis programs. These tests involve comparison to other simulation programs, comparison to analytically calculated results, and comparison to experimental data. The following table from the SERI validation report summarizes the advantages and disadvantages of each method.

VALIDATION TECHNIQUES

Technique	Advantages	Disadvantages
Comparative Relative test of different programs	No input uncertainty Any level of complexity Inexpensive Many comparisons possible	No truth standard
Analytical Test of numerical solution	No input uncertainty Exact truth standard given the simplicity of the model Inexpensive	Does not test the model Limited to cases for which analytical solutions can be derived
Empirical Comparison to measured building performance	Approximate truth standard within accuracy of data acquisition Any level of complexity	Measurements have some input uncertainty High quality, detailed measurements are time consuming & expensive

The analytic validation tests are best for detecting coding errors. The empirical validation tests are necessary to insure that the actual physical processes are modeled correctly and that all significant phenomena are being considered.

C.2 Analytic Tests

A series of simple analytical tests have been developed to insure that HLITE is performing as expected. The tests are sequenced so that the most basic algorithms are tested first. Later tests of other algorithms often rely on the basic algorithms. The input and report files for these tests are included on the HLITE distribution diskette to provide the user simple cases that can run directly to gain familiarity with the program, and to maintain the set of test cases which must be run when the program is modified to insure that its current capabilities have not been altered. All tests are written in English units, as the test cell was designed in English units, which forces HLITE to convert both input and output values for the calculations which are performed in SI units within HLITE.

TEST1A through TEST1D check the explicit and implicit calculation of heat conduction as well as the processing of the BVF and DEF and the calculation of loads for controlled temperature nodes.

TEST2A through TEST2D check the calculation of transient heat conduction including the use of multiple time steps.

TEST3A through TEST3C check the calculation of loads for air nodes for fixed temperature and dead band control.

TEST4A and TEST4B check the simulation of radiant heat transfer between surfaces.

TEST5A through TEST5C check the various convection algorithms.

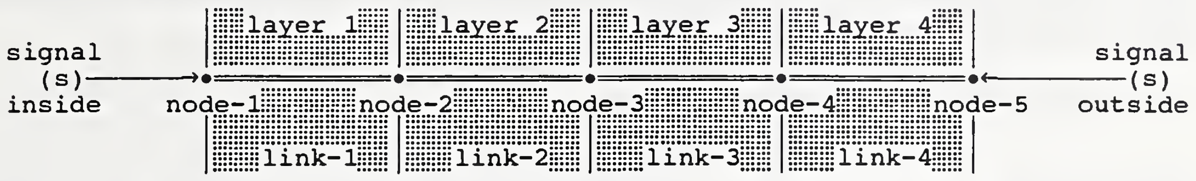
TEST6A through TEST6D check the equipment algorithms: the modeling of fluorescent luminaires and the impact of time step on the modeling of small mass, high energy components.

TEST7A and TEST7B check the air flow link calculations.

TEST8A through TEST8D check the operation of controls.

C.2.1a TEST1A: lyr, cnd, explicit

TEST1A involves the simplest heat transfer simulation, conduction within a homogeneous wall divided into 4 layers as shown.



There are 2 temperature signals which specify the temperatures at nodes 1 and 5. The nodes are all of the same type, **lyr**, with initial temperature of 20 C (68 F) or 10 C (50 F), and areas of 0.929 m^2 (10 ft^2). Nodes 1 and 5 are controlled, **c**, by the specified signal points, inside and outside. The material layers represent 1 inch of a masonry material (κ = 0.778 W/mK (0.45 Btu/h•ft•F), ρ = 1922. kg/m^3 (120. lb/ft^3), and c = 0.837 kJ/kgK (0.20 Btu/lb•F). The five nodes are linked by the four material layers.

This is described in the following Network Data File:

```
title TEST 1A: steady conduction, explicit.
units English

signal inside t 68.
signal outside t 50.

node node-1 lyr c 1.0 68. 68. 68. 10. inside
node node-2 lyr v 1.0 50. 68. 68. 10.
node node-3 lyr v 1.0 50. 68. 68. 10.
node node-4 lyr v 1.0 50. 68. 68. 10.
node node-5 lyr c 1.0 50. 50. 50. 10. outside

element masonry-1" mat 0.083333 0.45 120. 0.2

link link-1 cnd masonry-1" node-1 node-2
link link-2 cnd masonry-1" node-2 node-3
link link-3 cnd masonry-1" node-3 node-4
link link-4 cnd masonry-1" node-4 node-5

times 1 300

report test1a.rpt  7  600
 n  node-1  T  i  %8.4f
 n  node-2  T  i  %8.4f
 n  node-3  T  i  %8.4f
 n  node-4  T  i  %8.4f
 n  node-5  T  i  %8.4f
 n  node-1  Q  a  %12.4e
 n  node-5  Q  a  %12.4e

display  2  300
 n  node-2  T  i  %8.4f
 n  node-4  T  i  %8.4f

* end of data
```

The NDF indicates that transient simulation will be done with a 300 second (5 minute) time step. The temperatures for all the nodes and the loads at the controlled nodes are written to a file named "test1a.rpt" at every other time step. During the simulation, a line at the bottom of the screen is updated every (in this case) time step. This line gives the simulated time, the actual time since the simulation began, and the temperatures of nodes 2 and 4.

The contents of the BVF for this test are:

```
0
001/00:00:00
001/08:00:00
```

and the contents of the DEF are:

```
001/00:00:00
001/24:00:00
```

The BVF causes simulation for 8 hours; the DEF indicates no changes in the control signals.

The results of the simulation are given in file test1a.rpt:

```
001/00:00:00  68.0000 68.0000 68.0000 68.0000 50.0000  0.0000e+00  0.0000e+00
001/00:10:00  68.0000 68.0000 67.0887 61.7225 50.0000  0.0000e+00 -6.7433e+02
001/00:20:00  68.0000 67.4567 65.1671 59.2802 50.0000  2.7680e+00 -6.4378e+02
001/00:30:00  68.0000 66.6655 63.6523 57.9147 50.0000  3.0078e+01 -5.0449e+02
001/00:40:00  68.0000 65.9421 62.5070 57.0175 50.0000  7.1784e+01 -4.2891e+02
001/00:50:00  68.0000 65.3578 61.6434 56.3806 50.0000  1.1064e+02 -3.7980e+02
001/01:00:00  68.0000 64.9055 60.9925 55.9124 50.0000  1.4223e+02 -3.4512e+02
001/01:10:00  68.0000 64.5610 60.5019 55.5631 50.0000  1.6674e+02 -3.1967e+02
001/01:20:00  68.0000 64.3002 60.1321 55.3008 50.0000  1.8543e+02 -3.0070e+02
001/01:30:00  68.0000 64.1033 59.8533 55.1035 50.0000  1.9958e+02 -2.8646e+02
001/01:40:00  68.0000 63.9548 59.6432 54.9548 50.0000  2.1026e+02 -2.7575e+02
001/01:50:00  68.0000 63.8428 59.4848 54.8428 50.0000  2.1832e+02 -2.6769e+02
001/02:00:00  68.0000 63.7584 59.3654 54.7584 50.0000  2.2440e+02 -2.6161e+02
...
...
...
001/07:40:00  68.0000 63.5000 59.0000 54.5000 50.0000  2.4300e+02 -2.4300e+02
001/07:50:00  68.0000 63.5000 59.0000 54.5000 50.0000  2.4300e+02 -2.4300e+02
001/08:00:00  68.0000 63.5000 59.0000 54.5000 50.0000  2.4300e+02 -2.4300e+02
```

This output shows gradual change in the temperature distribution to the expected values. The computed loads, 71.1 W (243 Btu/h), are also correct.

$$Q = \kappa\, A\, \Delta T / L = 0.778 \cdot 0.929 \cdot 10 / .1016 = 71.1 \text{ W}$$
$$= 0.45 \cdot 10. \cdot 18. / 0.333 = 243. \text{ Btu/h}$$

C.2.1b TEST1B: 1A+implicit, BVF, DEF

TEST1B is a quick test of the implicit method of calculation and the operation of the BVF and DEF. It uses the same thermal network as TEST1A, but the layers have no mass. Therefore, there are no transients in the solution.

```
title TEST 1B: steady conduction, implicit.
units English

signal inside t 68.

node node-1 lyr c 1. 68. 68. 68. 10. inside
node node-2 lyr v 1. 50. 68. 68. 10.
node node-3 lyr v 1. 50. 68. 68. 10.
node node-4 lyr v 1. 50. 68. 68. 10.
node node-5 lyr b 1. 50. 50. 50. 10.

element masonry-1" mat 0.083333 0.45 120. 0.0 0.0  no thermal mass

link link-1 cnd masonry-1" node-1 node-2
link link-2 cnd masonry-1" node-2 node-3
link link-3 cnd masonry-1" node-3 node-4
link link-4 cnd masonry-1" node-4 node-5

times 1 900

report test1b.rpt  7  900
 n  node-1  T  i  %8.4f
 n  node-2  T  i  %8.4f
 n  node-3  T  i  %8.4f
 n  node-4  T  i  %8.4f
 n  node-5  T  i  %8.4f
 n  node-1  Q  i  %12.5e
 n  node-1  Q  s  %12.5e

display  3  900
 n  node-2  T  i  %8.4f
 n  node-4  T  i  %8.4f
 n  node-5  T  i  %8.4f

* end of data
```

The following data is in TEST1B.BVF:

```
1
node-5
001/00:00:00  50.0
001/01:00:00  50.0
001/02:00:00  32.0
001/03:00:00  50.0
001/06:00:00  50.0
*
```

This indicates that each line of data will report one value, the temperature of node-5, which was also identified as a boundary value node in the NDF. The BVF indicates that this temperature is constant for the first hour, decreases to 0 C (32 F) at the end of the second hour, returns to 10 C (50 F) during the next hour, and remains at that value until the end of simulation. Note that the time between data points is arbitrary.

The contents of TEST1B.DEF indicate that the control signal "inside" is set to 25 C (77 F) for one hour and then returned to 20 C (68 F), which is its initial value as specified in the NDF.

```
001/00:00:00
001/04:00:00  inside  77.0
001/05:00:00  inside  68.0
002/00:00:00
```

The following report file was generated:

```
001/00:00:00  68.0000 68.0000 68.0000 68.0000 50.0000  0.0000e+00  0.0000e+00
001/00:15:00  68.0000 63.5000 59.0000 54.5000 50.0000  2.4300e+02  3.0375e+01
001/00:30:00  68.0000 63.5000 59.0000 54.5000 50.0000  2.4300e+02  6.0750e+01
001/00:45:00  68.0000 63.5000 59.0000 54.5000 50.0000  2.4300e+02  6.0750e+01
001/01:00:00  68.0000 63.5000 59.0000 54.5000 50.0000  2.4300e+02  6.0750e+01
001/01:15:00  68.0000 62.3750 56.7500 51.1250 45.5000  3.0375e+02  6.8344e+01
001/01:30:00  68.0000 61.2500 54.5000 47.7500 41.0000  3.6450e+02  8.3532e+01
001/01:45:00  68.0000 60.1250 52.2500 44.3750 36.5000  4.2525e+02  9.8719e+01
001/02:00:00  68.0000 59.0000 50.0000 41.0000 32.0000  4.8600e+02  1.1391e+02
001/02:15:00  68.0000 60.1250 52.2500 44.3750 36.5000  4.2525e+02  1.1391e+02
001/02:30:00  68.0000 61.2500 54.5000 47.7500 41.0000  3.6450e+02  9.8719e+01
001/02:45:00  68.0000 62.3750 56.7500 51.1250 45.5000  3.0375e+02  8.3532e+01
001/03:00:00  68.0000 63.5000 59.0000 54.5000 50.0000  2.4300e+02  6.8344e+01
001/03:15:00  68.0000 63.5000 59.0000 54.5000 50.0000  2.4300e+02  6.0750e+01
001/03:30:00  68.0000 63.5000 59.0000 54.5000 50.0000  2.4300e+02  6.0750e+01
001/03:45:00  68.0000 63.5000 59.0000 54.5000 50.0000  2.4300e+02  6.0750e+01
001/04:00:00  77.0000 63.5000 59.0000 54.5000 50.0000  7.2900e+02  1.2150e+02
001/04:15:00  77.0000 70.2500 63.5000 56.7500 50.0000  3.6450e+02  1.3669e+02
001/04:30:00  77.0000 70.2500 63.5000 56.7500 50.0000  3.6450e+02  9.1125e+01
001/04:45:00  77.0000 70.2500 63.5000 56.7500 50.0000  3.6450e+02  9.1125e+01
001/05:00:00  68.0000 70.2500 63.5000 56.7500 50.0000 -1.2150e+02  3.0375e+01
001/05:15:00  68.0000 63.5000 59.0000 54.5000 50.0000  2.4300e+02  1.5188e+01
001/05:30:00  68.0000 63.5000 59.0000 54.5000 50.0000  2.4300e+02  6.0750e+01
001/05:45:00  68.0000 63.5000 59.0000 54.5000 50.0000  2.4300e+02  6.0750e+01
001/06:00:00  68.0000 63.5000 59.0000 54.5000 50.0000  2.4300e+02  6.0750e+01
    time         T1      T2      T3      T4      T5        Q           Q
```

Note the loads at time 4:00:00 and 5:00:00 which include the power required to change the temperatures. This is caused by the time stepping scheme. The temperature of the node has been set to the control values but the load is based on the temperature at the start of the time step.

Note instantaneous loads and total energy for each quarter hour time step.

C.2.1c TEST1C: mas, knd

TEST1C tests the **mas** nodes and **knd** links. The network is equivalent to TEST1A, and it should give identical results.

```
title TEST 1C:  MAS nodes and KND links; identical to TEST1A.
units English

signal inside t 68.
signal outside t 50.

node node-1 mas c 1.0 68. 68. 68. 10.0  50.0 0.2 0.9 inside
node node-2 mas v 1.0 50. 68. 68. 10.0 100.0 0.2 0.9
node node-3 mas v 1.0 50. 68. 68. 10.0 100.0 0.2 0.9
node node-4 mas v 1.0 50. 68. 68. 10.0 100.0 0.2 0.9
node node-5 mas c 1.0 50. 50. 50. 10.0  50.0 0.2 0.9 outside

link link-1 knd 54.0 node-1 node-2
link link-2 knd 54.0 node-2 node-3
link link-3 knd 54.0 node-3 node-4
link link-4 knd 54.0 node-4 node-5

times 1 300

report test1c.rpt  7  600
 n  node-1  T  i  %8.4f
 n  node-2  T  i  %8.4f
 n  node-3  T  i  %8.4f
 n  node-4  T  i  %8.4f
 n  node-5  T  i  %8.4f
 n  node-1  Q  a  %12.4e
 n  node-5  Q  a  %12.4e

display  2  300
 n  node-2  T  i  %8.4f
 n  node-4  T  i  %8.4f

* end of data
```

C.2.1d TEST1D: hcc, thin layer

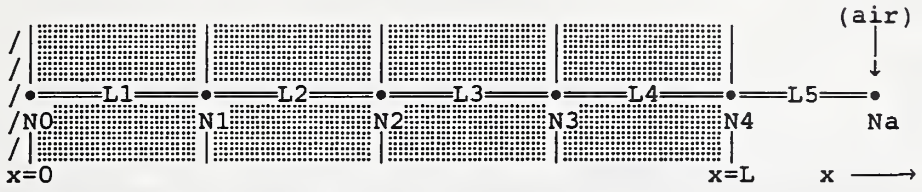
TEST1D tests the simulation of a thin layer. Earlier tests indicated that this requires a direct solution of implicit (simultaneous) equations. Convergence is very slow for an iterative solution.

```
title TEST 1D: effect of thin layer:  requires direct solution of implicit eqns
units English

signal inside t 68.
signal outside t 50.

node node-1 air c 1. 68. 68. 68. 100. inside
node node-2 srf v 1. 50. 68. 68. 10.  0.90
node node-3 srf v 1. 50. 68. 68. 10.  0.90
node node-4 air c 1. 50. 50. 50. 100. outside

element metal mat 0.0052 118. 169. 0.21

link link-1 hcc  1.0  node-2 node-1
link link-2 cnd metal node-2 node-3
link link-3 hcc  1.0  node-3 node-4

times 1 300

report test1c.rpt  6  600
 n  node-1  T  i  %8.3f
 n  node-2  T  i  %8.3f
 n  node-3  T  i  %8.3f
 n  node-4  T  i  %8.3f
 n  node-1  Q  i  %9.3f
 n  node-4  Q  i  %9.3f

display  3  300
 n  node-1  T  i  %8.3f
 n  node-2  T  i  %8.3f
 n  node-4  T  i  %8.3f

* end of data
```

```
001/00:00:00   68.000  68.000  68.000  50.000    0.000    0.000
001/00:10:00   68.000  61.485  61.485  50.000   42.707 -137.289
001/00:20:00   68.000  59.686  59.686  50.000   76.941 -103.055
001/00:30:00   68.000  59.190  59.189  50.000   86.393  -93.603
001/00:40:00   68.000  59.053  59.052  50.000   89.003  -90.993
001/00:50:00   68.000  59.015  59.014  50.000   89.723  -90.273
001/01:00:00   68.000  59.004  59.004  50.000   89.922  -90.074
001/01:10:00   68.000  59.001  59.001  50.000   89.977  -90.019
001/01:20:00   68.000  59.001  59.000  50.000   89.992  -90.004
001/01:30:00   68.000  59.000  59.000  50.000   89.996  -90.000
001/01:40:00   68.000  59.000  59.000  50.000   89.998  -89.998
001/01:50:00   68.000  59.000  59.000  50.000   89.998  -89.998
001/02:00:00   68.000  59.000  59.000  50.000   89.998  -89.998
```

C.2.2a TEST2A: transient conduction, explicit

TEST2A checks the simulation of transient conduction within a homogeneous wall. The physical problem consists of an infinite flat plate starting at temperature T_i which is suddenly exposed to fluid at temperature T_∞. The HLITE model of the wall consists of 4 identical layers:



By linking node-0 only to node-1, there is no heat transfer across the x=0 plane. This is equivalent to a plate of thickness 2L exposed to the fluid at both faces. This is done in the following NDF:

```
title TEST 2A: transient conduction.
units English

signal air-temp t 0.0

node node-0 lyr v 1. 0. 100. 0.0 10.
node node-1 lyr v 1. 0. 100. 0.0 10.
node node-2 lyr v 1. 0. 100. 0.0 10.
node node-3 lyr v 1. 0. 100. 0.0 10.
node node-4 srf v 1. 0. 100. 0.0 10.   .90
node node-a air c 1. 0. 100. 0.0 100.  air-temp

element masonry-0.75" mat 0.0625 0.45 120. 0.2

link link-1 cnd masonry-0.75" node-0 node-1
link link-2 cnd masonry-0.75" node-1 node-2
link link-3 cnd masonry-0.75" node-2 node-3
link link-4 cnd masonry-0.75" node-3 node-4
link link-5 hcc  1.0          node-4 node-a

times 1 120

report test2a.rpt  4  120
 n  node-a  T  i  %8.4f
 n  node-4  T  i  %8.4f
 n  node-0  T  i  %8.4f
 n  node-a  Q  i  %12.5e

report test2a.sum  1  3600
 n  node-a  Q  s  %12.5e

display  3  120
 n  node-a  T  i  %8.3f
 n  node-4  T  i  %8.4f
 n  node-0  T  i  %8.4f

* end of data
```

TEST2A uses the following BVF:

```
0
000/23:00:00
001/06:00:00
```

and the following DEF:

```
001/00:00:00
001/00:00:00  air-temp  100.0
002/00:00:00
```

to compute the following reported values:

```
000/23:00:00  -0.0000 -0.0000 -0.0000  0.0000e+00
 ...
000/23:58:00  -0.0000 -0.0000 -0.0000  0.0000e+00
001/00:00:00 100.0000 -0.0000 -0.0000  5.4127e+03
001/00:02:00 100.0000  4.4444 -0.0000  1.0000e+03
001/00:04:00 100.0000  7.2691 -0.0000  9.5556e+02
001/00:06:00 100.0000  9.2919 -0.0000  9.2731e+02
001/00:08:00 100.0000 10.8769 -0.0000  9.0708e+02
001/00:10:00 100.0000 12.1972  0.0058  8.9123e+02
001/00:12:00 100.0000 13.3419  0.0254  8.7803e+02
001/00:14:00 100.0000 14.3609  0.0656  8.6658e+02
001/00:16:00 100.0000 15.2845  0.1315  8.5639e+02
001/00:18:00 100.0000 16.1324  0.2257  8.4716e+02
001/00:20:00 100.0000 16.9186  0.3491  8.3868e+02
001/00:22:00 100.0000 17.6529  0.5015  8.3081e+02
001/00:24:00 100.0000 18.3432  0.6816  8.2347e+02
001/00:26:00 100.0000 18.9953  0.8877  8.1657e+02
001/00:28:00 100.0000 19.6140  1.1179  8.1005e+02
001/00:30:00 100.0000 20.2035  1.3699  8.0386e+02
001/00:32:00 100.0000 20.7670  1.6419  7.9797e+02
 ...
001/01:00:00 100.0000 27.0200  6.6457  7.3363e+02
 ...
001/02:00:00 100.0000 36.8839 18.5749  6.3414e+02
 ...
001/03:00:00 100.0000 45.1663 29.2346  5.5091e+02
 ...
001/04:00:00 100.0000 52.3527 38.5081  4.7871e+02
 ...
001/05:00:00 100.0000 58.5970 46.5667  4.1597e+02
 ...
001/06:00:00 100.0000 64.0229 53.5692  3.6146e+02
    time         Ta       T4       T0        Qa
```

The step change in temperature for -17.78 C (0 F) to 37.78 C (100 F) is chosen for easy comparison to the normalized temperatures of the analytic solution for this case. The stability analysis done by HLITE indicates stability limits of about 330 seconds for the surface node and 375 seconds for the interior nodes. The 120 second time step causes HLITE to use the explicit time integration method.

The computed values can be compared to the classical analytic solution:

$$\frac{T(x,t) - T_\infty}{T_i - T_\infty} = 2 \sum_{n=1}^{\infty} e^{-\delta_n^2 t\alpha/L^2} \frac{\sin\delta_n \cos(\delta_n x/L)}{\delta_n + \sin\delta_n \cos\delta_n}$$

with

$\delta_n \tan\delta_n = hL/\kappa$

The following results were computed for the test case using normalized temperatures of $T_i = 0$ and $T_\infty = 1$.

time	T at L	T at 0
00:00:00	0.000000	0.000000
00:00:10	0.017842	0.000000
00:00:20	0.025086	0.000000
00:00:30	0.030588	0.000000
00:00:40	0.035188	0.000000
00:00:50	0.039213	0.000000
00:01:00	0.042828	0.000000
00:01:10	0.046134	0.000000
00:01:20	0.049195	0.000000
00:01:30	0.052056	0.000000
00:01:40	0.054749	0.000000
00:01:50	0.057299	0.000000
00:02:00	0.059726	0.000000
00:02:30	0.066402	0.000000
00:03:00	0.072374	0.000000
00:03:30	0.077811	0.000000
00:04:00	0.082828	0.000000
00:04:30	0.087500	0.000000
00:05:00	0.091884	0.000000
00:06:00	0.099949	0.000003
00:07:00	0.107265	0.000011
00:08:00	0.113990	0.000031
00:09:00	0.120233	0.000072
00:10:00	0.126073	0.000143
00:12:00	0.136772	0.000415
00:14:00	0.146426	0.000909
00:16:00	0.155257	0.001670
00:18:00	0.163418	0.002718
00:20:00	0.171021	0.004057
00:22:00	0.178151	0.005680
00:24:00	0.184872	0.007574
00:26:00	0.191239	0.009718
00:28:00	0.197293	0.012093
00:30:00	0.203072	0.014679
00:32:00	0.208604	0.017453
00:34:00	0.213918	0.020398
00:36:00	0.219034	0.023495
00:38:00	0.223973	0.026727
00:40:00	0.228753	0.030078
00:42:00	0.233388	0.033535
00:44:00	0.237893	0.037084
00:46:00	0.242279	0.040715
00:48:00	0.246558	0.044417
00:50:00	0.250738	0.048180
00:52:00	0.254829	0.051997
00:54:00	0.258838	0.055859
00:56:00	0.262772	0.059760
00:58:00	0.266637	0.063694
01:00:00	0.270440	0.067655
01:10:00	0.288662	0.087718
01:20:00	0.305872	0.107921
01:30:00	0.322342	0.128005
01:40:00	0.338234	0.147827
01:50:00	0.353644	0.167309
02:00:00	0.368634	0.186409
02:10:00	0.383239	0.205109
02:20:00	0.397485	0.223399
02:30:00	0.411390	0.241281
02:40:00	0.424968	0.258759
02:50:00	0.438227	0.275838
03:00:00	0.451179	0.292526
03:10:00	0.463831	0.308830
03:20:00	0.476190	0.324760
03:30:00	0.488265	0.340323
03:40:00	0.500060	0.355528
03:50:00	0.511583	0.370382
04:00:00	0.522841	0.384894
04:10:00	0.533839	0.399071
04:20:00	0.544584	0.412922
04:30:00	0.555081	0.426454
04:40:00	0.565336	0.439674
04:50:00	0.575355	0.452589
05:00:00	0.585142	0.465206
05:10:00	0.594704	0.477533
05:20:00	0.604046	0.489575
05:30:00	0.613173	0.501340
05:40:00	0.622089	0.512834
05:50:00	0.630799	0.524062
06:00:00	0.639309	0.535032
∞	1.000000	1.000000

C.2.2b TEST2B: transient conduction, implicit

TEST2B is a test of the implicit time integration method. It differs from the explicit test only in the size of the time step, which is set at 600 seconds.

```
000/23:00:00  -0.0000 -0.0000 -0.0000  0.0000e+00
 ...
000/23:50:00  -0.0000 -0.0000 -0.0000  0.0000e+00
001/00:00:00 100.0000 -0.0000 -0.0000  1.0825e+03
001/00:10:00 100.0000  9.7860  0.2744  1.0000e+03
001/00:20:00 100.0000 15.2648  0.9635  9.0214e+02
001/00:30:00 100.0000 18.9949  2.0556  8.4735e+02
001/00:40:00 100.0000 21.8838  3.4685  8.1005e+02
001/00:50:00 100.0000 24.3047  5.1083  7.8116e+02
001/01:00:00 100.0000 26.4364  6.8947  7.5695e+02
 ...
001/02:00:00 100.0000 36.5881 18.3097  6.4925e+02
 ...
001/03:00:00 100.0000 44.8416 28.8263  5.6449e+02
 ...
001/04:00:00 100.0000 51.9811 38.0295  4.9141e+02
 ...
001/05:00:00 100.0000 58.1934 46.0459  4.2783e+02
 ...
001/06:00:00 100.0000 63.6017 53.0256  3.7248e+02
```

General note: The finite difference method is actually solving a slightly different problem than the step change in air temperature. The change in temperature cannot be instantaneous; it occurs over a single time step. That is, during one time step the air temperature changes linearly from its initial to its final value. Shorter time steps are better at representing a step change, but they never really produce a step change.

C.2.2c TEST2C: transient conduction, mas, knd

TEST2C checks the mas elements and knd links in a transient simulation which is identical to TEST2A. The configuration of nodes and links remains the same. They must be computed to match the corresponding mat elements and lyr links.

```
                                                                      (air)
                                                                        ↓
•=====L0=====•=====L1=====•=====L2=====•=====L3=====•====L4====•
N0           N1           N2           N3           N4         Na
```

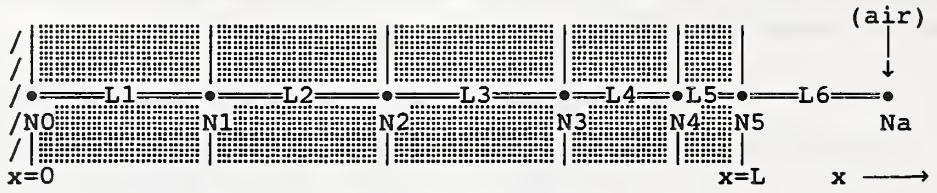
The masses of nodes 1, 2, and 3 are given by $m = \rho LA = 34.03$ kg (75.0 lb) Nodes 0 and 4 have half that mass. The heat transfer coefficient between nodes is given by $K = \kappa A/L = 94.96$ W/K (72.0 Btu/h • F). These values are used in the following NDF:

```
title TEST 2C: transient conduction - mass nodes and K links
units English

signal air-temp t 0.0

node node-0 mas v 1. 0. 100. 0.0 10. 37.5 0.2 .90
node node-1 mas v 1. 0. 100. 0.0 10. 75.0 0.2 .90
node node-2 mas v 1. 0. 100. 0.0 10. 75.0 0.2 .90
node node-3 mas v 1. 0. 100. 0.0 10. 75.0 0.2 .90
node node-4 mas v 1. 0. 100. 0.0 10. 37.5 0.2 .90
node node-a air c 1. 0. 100. 0.0 100.  air-temp

link link-1 knd  72.0  node-0 node-1
link link-2 knd  72.0  node-1 node-2
link link-3 knd  72.0  node-2 node-3
link link-4 knd  72.0  node-3 node-4
link link-5 hcc  1.0   node-4 node-a

times 1 120
 ...
```

This test gives results identical to TEST2A.

C.2.2d TEST2D: multiple time steps

TEST2D tests the use of multiple time steps. The surface layer of TEST2A is divided into two layers.



```
title TEST 2D: variable thickness layers for multiple time steps
title test2d
units English

signal air-temp t 0.0

node node-0 lyr v 1. 0. 100. 0.0 10.
node node-1 lyr v 1. 0. 100. 0.0 10.
node node-2 lyr v 1. 0. 100. 0.0 10.
node node-3 lyr v 1. 0. 100. 0.0 10.
node node-4 lyr v 1. 0. 100. 0.0 10.
node node-5 srf v 1. 0. 100. 0.0 10.  .90
node node-a air c 1. 0. 100. 0.0 100.  air-temp

element masonry-0.25" mat 0.02083 0.45 120. 0.2
element masonry-0.50" mat 0.04167 0.45 120. 0.2
element masonry-0.75" mat 0.0625 0.45 120. 0.2

link link-1 cnd masonry-0.75" node-0 node-1
link link-2 cnd masonry-0.75" node-1 node-2
link link-3 cnd masonry-0.75" node-2 node-3
link link-4 cnd masonry-0.50" node-3 node-4
link link-5 cnd masonry-0.25" node-4 node-5
link link-6 hcc  1.0          node-5 node-a

times 3  30 60 120
```

These times were chosen to allow comparison to the TEST2A results and to match the stability limits of the thinner layers. This is reported in TEST2D.OUT:

```
 ...
node:  node-3    C = 12.50 Btu/F    node # 4
   linked to node-4              by link-4              K = 107.99
   linked to node-2              by link-3              K = 72.00
   sum(K) = 179.99   stability limit = 250.0 sec   time step = 120

node:  node-4    C = 7.50 Btu/F    node # 5
   linked to node-5              by link-5              K = 216.03
   linked to node-3              by link-4              K = 107.99
   sum(K) = 324.03   stability limit = 83.3 sec   time step = 60

node:  node-5    C = 2.50 Btu/F    node # 6
   linked to node-a              by link-6              K = 10.0
   linked to node-4              by link-5              K = 216.03
   sum(K) = 226.03   stability limit = 39.8 sec   time step = 30
 ...
```

The results of simulation are reported at the same times used in TEST2A:

```
000/23:00:00  -0.0000 -0.0000 -0.0000  0.0000e+00
 ...
000/23:58:00  -0.0000 -0.0000 -0.0000  0.0000e+00
001/00:00:00 100.0000 -0.0000 -0.0000  2.1651e+04
001/00:02:00 100.0000  5.1262 -0.0000  9.5642e+02
001/00:04:00 100.0000  7.6559 -0.0000  9.2858e+02
001/00:06:00 100.0000  9.4215 -0.0000  9.0973e+02
001/00:08:00 100.0000 10.8447 -0.0000  8.9485e+02
001/00:10:00 100.0000 12.0689  0.0047  8.8220e+02
001/00:12:00 100.0000 13.1570  0.0217  8.7102e+02
001/00:14:00 100.0000 14.1425  0.0582  8.6094e+02
001/00:16:00 100.0000 15.0464  0.1193  8.5171e+02
001/00:18:00 100.0000 15.8833  0.2080  8.4319e+02
001/00:20:00 100.0000 16.6637  0.3254  8.3525e+02
001/00:22:00 100.0000 17.3958  0.4714  8.2782e+02
001/00:24:00 100.0000 18.0862  0.6450  8.2082e+02
001/00:26:00 100.0000 18.7400  0.8445  8.1420e+02
001/00:28:00 100.0000 19.3615  1.0680  8.0790e+02
001/00:30:00 100.0000 19.9543  1.3135  8.0191e+02
 ...
001/01:00:00 100.0000 26.8301  6.5212  7.3266e+02
 ...
001/02:00:00 100.0000 36.7421 18.4331  6.3332e+02
 ...
001/03:00:00 100.0000 45.0434 29.1072  5.5021e+02
 ...
001/04:00:00 100.0000 52.2447 38.3955  4.7811e+02
 ...
001/05:00:00 100.0000 58.5020 46.4673  4.1547e+02
 ...
001/06:00:00 100.0000 63.9393 53.4815  3.6103e+02
```

These results compare extremely well to the analytic solution. In addition this simulation is computing temperatures at the surface node every 30 seconds which would be useful for the analysis of dynamics at that time scale.

The following table summarizes the transient conduction tests.

Surface temperatures:

time	analytic	test2a	test2b	test2d
00:02:00	0.059726	0.044444	-	0.051262
00:04:00	0.082828	0.072691	-	0.076559
00:10:00	0.126073	0.121972	0.097860	0.120689
01:00:00	0.270440	0.270200	0.264364	0.268301
02:00:00	0.368634	0.368839	0.365881	0.367421
06:00:00	0.639309	0.640229	0.636017	0.639393
execution time (sec): (without reports)		0.93	0.82	2.53

C.2.3a TEST3A: air, controlled

TEST3A checks the operation of a controlled air node. The network consists of a single air node controlled by a single signal which causes a step change in temperature during the simulation. This also provides test of a degenerate network.

```
        (air) ————————→• a

title TEST 3A:  instantaneous and cumulative loads
units English

signal air-temp t 50.

node node-a air c 1. 50. 68. 50. 100.  air-temp

times 1 900

report test3a.rpt  4  900
 n  node-a  T  i  %8.4f
 n  node-a  Q  i  %12.4e      |
 n  node-a  Q  a  %12.4e      } Note 3 different reporting methods
 n  node-a  Q  s  %12.4e      |

report test3a.sum  1  3600
 n  node-a  Q  s  %12.4e

display  2  900
 n  node-a  T  i  %8.4f
 n  node-a  Q  i  %12.4f

* end of data
```

The BVF allows for two hours of simulation. The DEF causes the air temperature to increase from 10 C (50 F) to 20 C (68 F) after one hour:

```
001/01:00:00   air-temp   68.
```

The results from TEST3A.RPT are:

```
001/00:00:00  50.0000  0.0000e+00  0.0000e+00  0.0000e+00
001/00:15:00  50.0000  0.0000e+00  0.0000e+00  0.0000e+00
001/00:30:00  50.0000  0.0000e+00  0.0000e+00  0.0000e+00
001/00:45:00  50.0000  0.0000e+00  0.0000e+00  0.0000e+00
001/01:00:00  68.0000  1.2990e+02  6.4952e+01  1.6238e+01
001/01:15:00  68.0000  0.0000e+00  6.4952e+01  1.6238e+01
001/01:30:00  68.0000  0.0000e+00  0.0000e+00  0.0000e+00
001/01:45:00  68.0000  0.0000e+00  0.0000e+00  0.0000e+00
001/02:00:00  68.0000  0.0000e+00  0.0000e+00  0.0000e+00
```

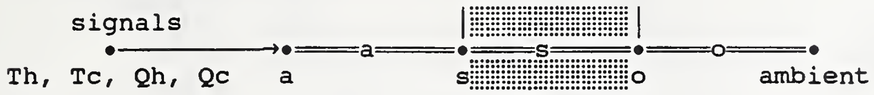
The results from TEST3A.SUM are:

```
001/00:00:00   0.0000e+00
001/01:00:00   1.6238e+01
001/02:00:00   1.6238e+01
```

It is expected that this change in air temperature will require ($Q=\rho Vc\Delta T$) 34.2 kJ (32.4 Btu). The temperature change occurs during a single time step. The expected rate at which heat is added to the air node is ($Q/\Delta t$) 38.0 W (129.6 Btu/h). The energy is split into adjacent time steps by the trapezoidal integration algorithm used to generate the report.

C.2.3b TEST3B: czn, deadband, explicit

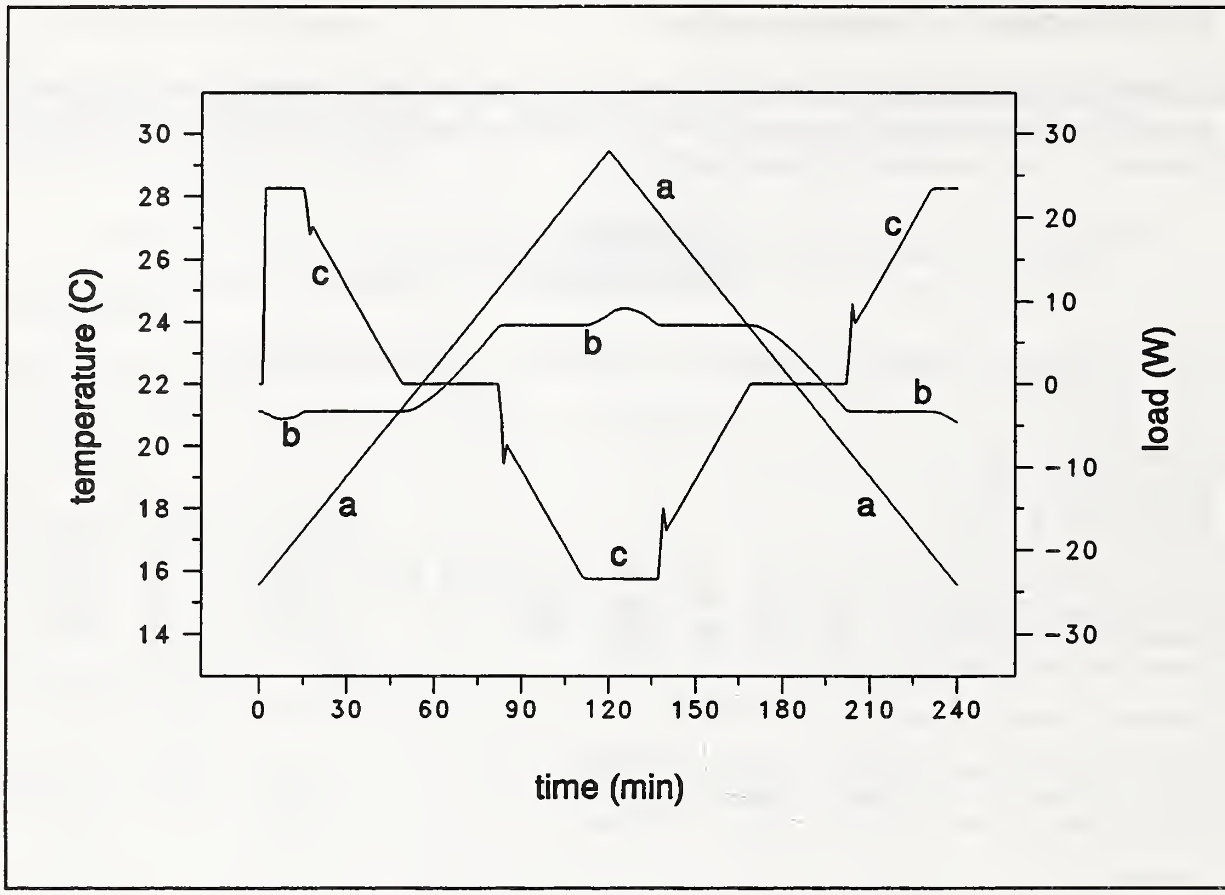
TEST3B checks the operation of **czn** node dead band operation. Simple convection and conduction links [overall U = 0.568 W/m^{2}K (0.10 Btu/h·ft^2·F)] connect the czn node to a boundary node representing ambient air. The czn node is simulated with mass.



```
title TEST 3B: room air temperature limits - explicit czn node
units English

signal Th t 70.
signal Tc t 75.
signal Qh q +80.
signal Qc q -80.

node node-a1 czn c 1. 70. 75. 70. 100.  Th Qh Tc Qc
node node-s1 srf v 1. 69. 76. 70. 100. .90
node node-o1 srf v 1. 60. 85. 60. 100. .90
node ambient air b 1. 60. 85. 60. 100.

element InsWall mat 0.0 0.0 0.0 0.0 8.0

link link-a1 hcc  1.0    node-s1 node-a1
link link-s1 cnd InsWall node-o1 node-s1
link link-o1 hcc  1.0    node-o1 ambient

times 1 60

report test3b.rpt  4  60
 n  ambient  T  i  %8.3f
 n  node-s1  T  i  %8.3f
 n  node-a1  T  i  %8.3f
 n  node-a1  Q  i  %9.3f

report test3b.sum  1  3600
 n  node-a1  Q  s  %12.4e

display  3  60
 n  ambient  T  i  %8.3f
 n  node-a1  T  i  %8.3f
 n  node-a1  Q  i  %9.3f

* end of data
```

The BVF is set up to cause the temperature to increase from 15.56 C (60 F) to 29.44 C (85 F) during the first two hours of simulation and return to 60 F during the last two hours.

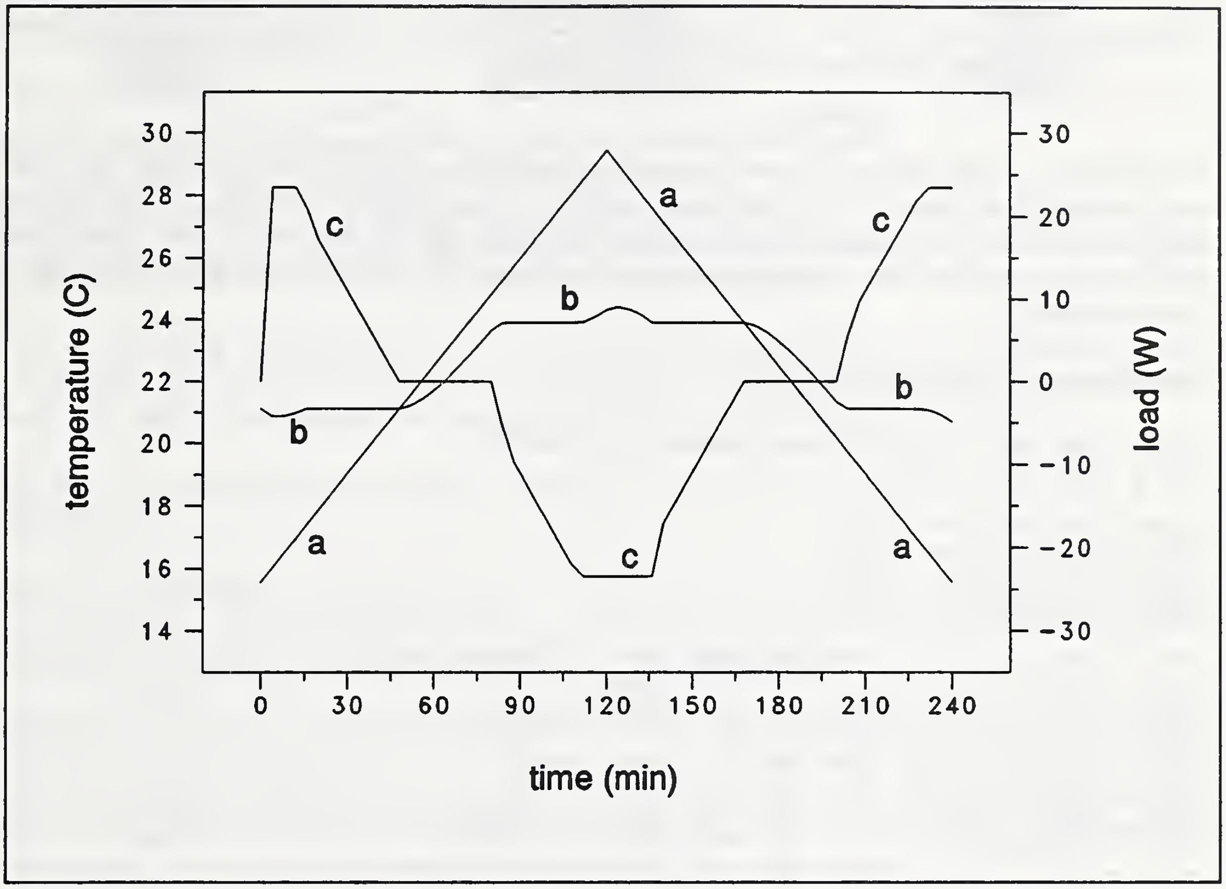
The results of this simulation are shown in the following figure.



Curve a is the ambient temperature, curve b is the room air temperature, and curve c is the load. At the start of the simulation it takes one time step for the load to change from its initial value of 0 to 23.4 W (80 Btu/h). Whenever the ambient temperature changes so that the room air temperature changes from a floating to a controlled mode, there is a small one time step spike in the load. Compare these results to test3c.

C.2.3c TEST3C: czn, implicit

This is the same as test3b except that the time step has been increased to 120 seconds forcing an implicit solution for the **czn** node temperature and load.



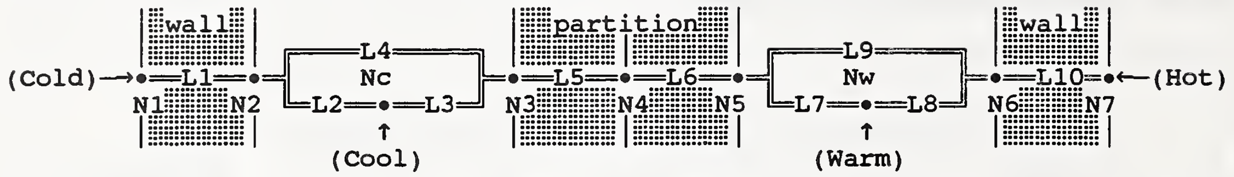
Curve a is the ambient temperature, curve b is the room air temperature, and curve c is the load. The computed load transitions smoothly as the room air temperature changes from floating to controlled mode.

The integrated loads (energies) using one hour time steps for tests 3b and 3c are:

```
      time           test3b        test3c      difference
001/00:00:00    0.0000e+00    0.0000e+00
001/01:00:00    3.8126e+01    3.7711e+01        -1.1%
001/02:00:00   -3.5520e+01   -3.6106e+01        +1.6%
001/03:00:00   -4.0390e+01   -3.9454e+01        -2.3%
001/04:00:00    3.5520e+01    3.6106e+01        +1.6%
```

C.2.4a TEST4A: rad

TEST4A checks the calculation of radiant heat transfer.



This test uses a mode complex thermal network representing two walls bounding two air spaces which are separated by an almost infinite resistance partition. The outside surface temperatures and the air temperatures are set constant. If there is no heat conduction through the partition, the temperatures of the four nodes bounding the two air spaces can be computed by solving two pairs of simultaneous non-linear equations.

The corresponding NDF description is:

```
title TEST 4A: radiant interchange calculation
units English

signal ColdSurface t 48.
signal HotSurface  t 98.
signal CoolRoom    t 68.
signal WarmRoom    t 78

node node-1 lyr c 1. 48. 48. 48. 2000.  ColdSurface
node node-2 srf v 1. 48. 68. 60. 2000.  .90
node node-c air c 1. 68. 68. 68. 10000. CoolRoom
node node-3 srf v 1. 68. 68. 68. 2000.  .90
node node-4 lyr v 1. 68. 78. 73. 2000.
node node-5 srf v 1. 78. 78. 78. 2000.  .90
node node-w air c 1. 78. 78. 78. 10000. WarmRoom
node node-6 srf v 1. 78. 98. 80. 2000.  .90
node node-7 lyr c 1. 98. 98. 98. 2000.  HotSurface

element ExtWall   mat  0.0 0.0 0.0 0.0 4.0
element Partition mat  0.0 0.0 0.0 0.0 1.0e8

link link-1 cnd ExtWall   node-1 node-2
link link-2 hcc 0.5424    node-2 node-c
link link-3 hcc 0.5424    node-3 node-c
link link-4 rad 0.818182  node-2 node-3
link link-5 cnd Partition node-3 node-4
link link-6 cnd Partition node-4 node-5
link link-7 rad 0.818182  node-5 node-6
link link-8 hcc 0.5424    node-5 node-w
link link-9 hcc 0.5424    node-6 node-w
link link-10 cnd ExtWall  node-6 node-7

times 1 900

report test4a.rpt   6   900
 n  node-2  T  i  %8.4f
 n  node-3  T  i  %8.4f
 n  node-5  T  i  %8.4f
 n  node-6  T  i  %8.4f
 n  node-c  Q  a  %12.4e
 n  node-w  Q  a  %12.4e
```
(continued on next page)

```
display  6  900
 n  node-2  T  i  %8.4f
 n  node-c  T  i  %8.4f
 n  node-3  T  i  %8.4f
 n  node-5  T  i  %8.4f
 n  node-w  T  i  %8.4f
 n  node-6  T  i  %8.4f

* end of data
```

For the "cool" side of the network, there are four active heat transfer paths:
(1) link-1 $q_1 = U\,(T_1 - T_2)$ conduction in the exterior wall,
(2) link-2 $q_2 = H\,(T_2 - T_c)$ convection from the wall to the air,
(3) link-3 $q_3 = H\,(T_3 - T_c)$ convection from the partition to the air, and
(4) link-4 $q_4 = \sigma\mathcal{F}\,(T_2^4 - T_3^4)$ radiation from the wall to the partition.

$U = 1.42$ W/m^{2}K (0.25 Btu/h·ft^2·F); $H = 3.08$ W/m^{2}K (0.5424 Btu/h·ft^2·F); $\sigma = 5.6697\text{E-8}$ W/m^{2}K^4 (1.7141E-9 Btu/h·ft^2·R^4) (Stefan-Boltzmann constant); $\mathcal{F} = 1\,/\,(1/\epsilon_1 + 1/\epsilon_2 - 1) = 0.818182$ for $\epsilon_1 = \epsilon_2 = 0.90$ (radiation interchange factor between two surfaces for which $F_{1,2} = F_{2,1} = 1$).

This leads to the following heat balances at surfaces 2 and 3: $q_1 - q_2 - q_4 = 0$, and $q_4 - q_3 = 0$, respectively.

Solving these equations simultaneously with $T_1 = 8.89$ C (48 F) and $T_c = 20$ C (68 F) gives $T_2 = 17.51$ C (63.523 F), $T_3 = 18.51$ C (65.322 F), and $q_1 = -12.24$ W/m^2 (-3.881 Btu/h·ft^2). For the entire 185.8 m^2 (2000 ft^2) wall the expected heating load is 2275. W (7762. Btu/h).

Similarly, for the "warm" side of the network setting $T_7 = 36.67$ C (98 F) and $T_w = 25.56$ C (78 F) gives $T_6 = 27.084$ C (80.752 F), $T_5 = 28.016$ C (82.428 F), and $q_1 = 12.28$ W/m^2 (3.893 Btu/h·ft^2). For the entire wall the expected cooling load is 2282. W (7786. Btu/h). Note that this load is not exactly equal to the heating load for the cool room even though there is a 11.1 C (20 F) temperature difference for both walls. This is caused by the nonlinear nature of radiant heat transfer.

The computed results from file test4a.rpt are:

```
001/00:00:00  60.0000 68.0000 78.0000 80.0000  0.0000e+00  0.0000e+00
001/00:15:00  63.5213 65.3246 80.7389 82.4352  3.8803e+03 -3.8912e+03
001/00:30:00  63.5226 65.3228 80.7495 82.4279  7.7610e+03 -7.7842e+03
001/00:45:00  63.5226 65.3228 80.7495 82.4279  7.7613e+03 -7.7860e+03
001/01:00:00  63.5226 65.3228 80.7495 82.4279  7.7613e+03 -7.7860e+03
     time        T2      T3      T6      T5       Qw          Qc
```

C.2.4b TEST4B: vfm

TEST4B is identical to test4a except that view factor, **vfm**, links are used in place of the radiant interchange, **rad**, links. The link portion of the test4b NDF is:

```
link link-1 cnd ExtWall   node-1 node-2
link link-2 hcc 0.5424    node-2 node-c
link link-3 hcc 0.5424    node-3 node-c
link link-4 vfm 2
  node-2
    0.0  1.0
  node-3
    1.0  0.0
link link-5 cnd Partition node-3 node-4
link link-6 cnd Partition node-4 node-5
link link-7 vfm 2
  node-5
    0.0  1.0
  node-6
    1.0  0.0
link link-8 hcc 0.5424    node-5 node-w
link link-9 hcc 0.5424    node-6 node-w
link link-10 cnd ExtWall  node-6 node-7
```

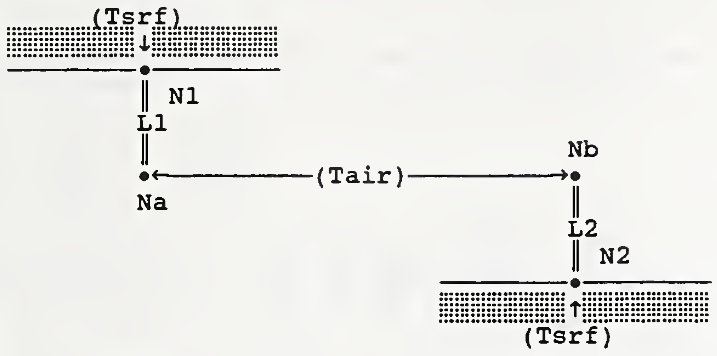
Note that node-2 has a view factor of 0.0 to itself and 1.0 to node-3; node-3 has a view factor of 1.0 to node-2 and 0.0 to itself. This is physically identical to the situation simulated in test4a. Therefore, the results of simulation should be identical, which is confirmed by the output file test2b.rpt:

```
001/00:00:00  60.0000 68.0000 78.0000 80.0000  0.0000e+00  0.0000e+00
001/00:15:00  63.5213 65.3246 80.7389 82.4352  3.8803e+03 -3.8912e+03
001/00:30:00  63.5226 65.3228 80.7495 82.4279  7.7610e+03 -7.7842e+03
001/00:45:00  63.5226 65.3228 80.7495 82.4279  7.7613e+03 -7.7860e+03
001/01:00:00  63.5226 65.3228 80.7495 82.4279  7.7613e+03 -7.7860e+03
```

This output shows how nonlinearities are handled over several timesteps converging to the exact answer in only a few steps.

C.2.5a TEST5A: cnv, simple model

TEST5A checks the simple models for the variable convection coefficient element (section B.3.2.2).



```
title TEST 5A:  Simple convection coefficients.
units English

signal Tair     t 70.
signal Tsrf     t 50.
signal flow     d 0.0

node node-a air c 1. 70. 70. 70. 1000.  Tair
node node-b air c 1. 70. 70. 70. 1000.  Tair
node node-1 srf c 1. 50. 90. 50. 10.  .90   Tsrf
node node-2 srf c 1. 50. 90. 50. 10.  .90   Tsrf

element hs_ceil  hcv 1.08 0.0 0.162 0.0 0.0 0.712 0.0 0.0
element hs_wall  hcv 1.08 0.0 0.542 0.0 0.0 0.542 0.0 0.0
element hs_floor hcv 1.08 0.0 0.712 0.0 0.0 0.162 0.0 0.0
element flow cfr 1.0

link link-0 afp flow node-a node-b flow
link link-1 cnv hs_ceil  node-1 node-a link-0
link link-2 cnv hs_floor node-2 node-b link-0

times 1 300

report test5a.rpt  7  300
 n  node-a  T  i  %8.4f
 n  node-1  T  i  %8.4f
 n  node-2  T  i  %8.4f
 n  node-a  Q  i  %12.4e
 n  node-b  Q  i  %12.4e
 l  link-1  h  i  %8.4f
 l  link-2  h  i  %8.4f

display  3  300
 n  node-a  T  i  %8.4f
 n  node-a  Q  i  %12.4e
 n  node-b  Q  i  %12.4e

* end of data
```

Since the ϵ coefficients in the **hcv** elements are 0, the VA_c and flow values in the **cnv** links are not needed.

Beginning at Tair = 21.1 C (70 F) and Tsrf = 10 C (50 F), Tsrf is incremented by 2.78°C (5°F) up to 32.2 C (90 F). The convection coefficient, h, is computed from the α_+ and α_- values. Then the flow signal is set to 1 changing h to the δ value. The following DEF was used:

```
00:00:00
00:10:00  Tsrf  55.
00:20:00  Tsrf  60.
00:30:00  Tsrf  65.
00:40:00  Tsrf  70.
00:50:00  Tsrf  75.
01:00:00  Tsrf  80.
01:10:00  Tsrf  85.
01:20:00  Tsrf  90.
01:30:00  flow  1.0
```

The expected loads, given by Q = -h·A·(Tsrf - Tair), are:

Tsrf	Qa	Qb
50.0	142.4	32.4
55.0	106.8	24.3
60.0	71.2	16.2
65.0	35.6	8.1
70.0	0.0	0.0
75.0	-8.1	-35.6
80.0	-16.2	-71.2
85.0	-24.3	-106.8
90.0	-32.4	-142.4

After the flow signal is set, the expected load is
Qa = Qb = -h·A·ΔT = -63.3 W (-216.0 Btu/h).

Five minute time steps reporting intervals were used. All the temperatures are fixed; none are computed. HLITE produced the following RPT file:

```
001/00:00:00  70.0000 50.0000 50.0000  0.0000e+00  0.0000e+00  0.0000  0.0000
001/00:05:00  70.0000 50.0000 50.0000  1.4240e+02  3.2400e+01  0.7120  0.1620
001/00:10:00  70.0000 55.0000 55.0000  1.4240e+02  3.2400e+01  0.7120  0.1620
001/00:15:00  70.0000 55.0000 55.0000  1.0680e+02  2.4300e+01  0.7120  0.1620
001/00:20:00  70.0000 60.0000 60.0000  1.0680e+02  2.4300e+01  0.7120  0.1620
001/00:25:00  70.0000 60.0000 60.0000  7.1200e+01  1.6200e+01  0.7120  0.1620
001/00:30:00  70.0000 65.0000 65.0000  7.1200e+01  1.6200e+01  0.7120  0.1620
001/00:35:00  70.0000 65.0000 65.0000  3.5600e+01  8.1000e+00  0.7120  0.1620
001/00:40:00  70.0000 70.0000 70.0000  3.5600e+01  8.1000e+00  0.7120  0.1620
001/00:45:00  70.0000 70.0000 70.0000 -0.0000e+00 -0.0000e+00  0.1620  0.7120
001/00:50:00  70.0000 75.0000 75.0000 -0.0000e+00 -0.0000e+00  0.1620  0.7120
001/00:55:00  70.0000 75.0000 75.0000 -8.1000e+00 -3.5600e+01  0.1620  0.7120
001/01:00:00  70.0000 80.0000 80.0000 -8.1000e+00 -3.5600e+01  0.1620  0.7120
001/01:05:00  70.0000 80.0000 80.0000 -1.6200e+01 -7.1200e+01  0.1620  0.7120
001/01:10:00  70.0000 85.0000 85.0000 -1.6200e+01 -7.1200e+01  0.1620  0.7120
001/01:15:00  70.0000 85.0000 85.0000 -2.4300e+01 -1.0680e+02  0.1620  0.7120
001/01:20:00  70.0000 90.0000 90.0000 -2.4300e+01 -1.0680e+02  0.1620  0.7120
001/01:25:00  70.0000 90.0000 90.0000 -3.2400e+01 -1.4240e+02  0.1620  0.7120
001/01:30:00  70.0000 90.0000 90.0000 -3.2400e+01 -1.4240e+02  0.1620  0.7120
001/01:35:00  70.0000 90.0000 90.0000 -2.1600e+02 -2.1600e+02  1.0800  1.0800
...
001/02:00:00  70.0000 90.0000 90.0000 -2.1600e+02 -2.1600e+02  1.0800  1.0800
                Tair     T1      T2        Q1          Q2         h1      h2
```

C.2.5b TEST5B: cnv, detailed model

TEST5B is similar to TEST5A except that the detailed convection models are used instead of the simple models.

```
title TEST 5B:  Detailed convection coefficients.
units English

signal Tair      t 70.
signal Tsrf      t 50.
signal flow      d 0.0

node node-a air c 1. 70. 70. 70. 1000.  Tair
node node-b air c 1. 70. 70. 70. 1000.  Tair
node node-1 srf c 1. 50. 90. 50. 10.  .90   Tsrf
node node-2 srf c 1. 50. 90. 50. 10.  .90   Tsrf

element hd_ceil  hcv 2.01 36.9 0.0 0.10 0.33 0.0 0.22 0.33
element hd_wall  hcv 0.74 14.3 0.0 0.19 0.33 0.0 0.19 0.33
element hd_floor hcv 0.62 8.24 0.0 0.22 0.33 0.0 0.10 0.33
element flow cfr 100.

link link-0  afp flow node-a node-b flow
link link-00 afp flow node-b node-a flow
link link-1 cnv hd_ceil  node-1 node-a link-0 1000. 100.
link link-2 cnv hd_floor node-2 node-b link-0 1000. 100.
```

The VA_e value of 1000 in the **cnv** link represents a room volume of 28.3 m^3 (1000 ft^3) (matching the volumes of node-a and node-b) and a flow area of .093 m^2 (1 ft^2). The flow rate of .047 m^3/s (100 cfm) gives a maximum air velocity of 30.5 m/s (100 ft/s) and 6 air changes per hour. This flow rate in the links matches the flow rate in the element.

Beginning at Tair = 21.1 C (70 F) and Tsrf = 10 C (50 F), Tsrf is incremented by 2.78°C (5°F) up to 32.2 C (90 F). The expected loads at the air nodes, given by $Q = -h \cdot A \cdot \Delta T = -h' \cdot A \cdot (Tsrf - Tair)^{1.33}$ are:

```
Tsrf        Qa        Qb
50.0      118.2      53.7
55.0       80.7      36.7
60.0       47.0      21.4
65.0       18.7       8.5
70.0        0.0       0.0
75.0       -8.5     -18.7
80.0      -21.4     -47.0
85.0      -36.7     -80.7
90.0      -53.7    -118.2
```

After the flow signal is set, the square root of the jet momentum number is

$$\sqrt{J} = F / [\rho(gVA_e)^{½}] = (100 \cdot .075/60)/[.075 \cdot (32.2 \cdot 1000 \cdot 1)^{½}] = 0.009288$$

and the expected loads are

$Q_a = -h \cdot A \cdot \Delta T = -137.9$ W (-470.5 Btu/h) and

$Q_b = -40.8$ W (-139.3 Btu/h).

C.2.5c TEST5C: **cnv** & **null**

TEST5C is a variation of TEST5A in which the **cnv** links refer to the **null** node, which causes the effect of air flow rate to be ignored in computing convection coefficients.

```
title TEST 5C:  Simple convection coefficients.
units English

signal Tair     t 70.
signal Tsrf     t 50.
signal flow     d 0.0

node node-a air c 1. 70. 70. 70. 1000.  Tair
node node-b air c 1. 70. 70. 70. 1000.  Tair
node node-1 srf c 1. 50. 90. 50. 10.  .90   Tsrf
node node-2 srf c 1. 50. 90. 50. 10.  .90   Tsrf

element hs_ceil  hcv 1.08 0.0 0.162 0.0 0.0 0.712 0.0 0.0
element hs_wall  hcv 1.08 0.0 0.542 0.0 0.0 0.542 0.0 0.0
element hs_floor hcv 1.08 0.0 0.712 0.0 0.0 0.162 0.0 0.0
element flow cfr 1.0

link link-0 afp flow node-a node-b flow
link link-1 cnv hs_ceil  node-1 node-a null    ←—— null links used
link link-2 cnv hs_floor node-2 node-b null

...
```

Using the same BVF and DEF as TEST5A, this NDF produces results identical to TEST5A except that the flow signal is now ignored.

C.2.6a TEST6A: lmp, steady performance

TEST6A checks the operation of the luminaire performance curves. By varying the air temperature, the luminaire temperature is changed. All light is absorbed by the "Walls" surface so that the cooling load for that surface equals the light emitted from the luminaire. Similarly, the load at the "Housing" surface equals the power into the ballast.

```
title TEST 6A:  luminaire calculations -- test temperature effect
units English

signal air-temp t 50.
signal switch d 1.0

element tube  lmp  48.0  3200.  372.  1.57  0.90  0.64  .2  11
      0.0         0.661         0.022
     20.0         0.672         0.068    (Ref: IES Lighting Handbook,
     40.0         0.704         0.165     Ref Volume, 1984, Fig 8-34)
     60.0         0.818         0.476
     70.0         0.894         0.701
     80.0         0.971         0.865
     90.0         0.999         0.960
    100.0         0.965         0.995
    120.0         0.858         0.850
    140.0         0.721         0.697
    215.0         0.000         0.000

node RoomAir air c 1. 50. 130. 50. 100.  air-temp
node Housing mas v 1. 50. 140. 50. 10.  0.1  0.2  .90
node Adb1    lyr c 1. 50. 130. 50. 10.  air-temp
node Walls   srf v 1. 50. 140. 50. 10.  .90
node Adb2    lyr c 1. 50. 130. 50. 10.  air-temp
node Luminaire eqp v 1. 50. 141. 50. tube  2

link light   lum  Luminaire  5. switch .15 Housing 1
    Walls   1.0
link ballast knd  5.0     Housing Adb1
link walls   knd  5.0     Walls   Adb2
link tubes   hcc  5.0     Luminaire RoomAir

times 1 120

report  test6a.rpt  6  360
 n  RoomAir    T  i  %8.3f
 n  Luminaire  T  i  %8.3f
 l  light      Q  i  %9.3f
 n  RoomAir    Q  i  %9.3f
 n  Adb1       Q  i  %9.3f
 n  Adb2       Q  i  %9.3f

display  3  360
 n  RoomAir    Q  i  %9.3f
 n  Luminaire  T  i  %8.3f
 n  Housing    T  i  %8.3f

* end of data
```

Contents of the DEF:

```
001/00:00:00
001/01:00:00  air-temp  55.
001/02:00:00  air-temp  60.
001/03:00:00  air-temp  65.
001/04:00:00  air-temp  70.
001/05:00:00  air-temp  75.
001/06:00:00  air-temp  80.
001/07:00:00  air-temp  85.
001/08:00:00  air-temp  90.
001/09:00:00  air-temp  95.
001/10:00:00  air-temp 100.
001/11:00:00  air-temp 105.
001/12:00:00  air-temp 110.
001/13:00:00  air-temp 115.
001/14:00:00  air-temp 120.
001/15:00:00  air-temp 125.
001/16:00:00  air-temp 130.
002/00:00:00
*
```

The following results were selected from the report file after disappearance of the transients due to temperature change.

```
001/00:00:00     10.000  10.000     0.000    0.000    0.000    0.000
001/00:30:00     10.000  17.010    76.919  -58.058  -11.538   -7.323
001/01:30:00     12.778  19.918    80.510  -59.136  -12.077   -9.297
001/02:30:00     15.556  22.841    84.365  -60.341  -12.655  -11.369
001/03:30:00     18.333  25.833    88.609  -62.112  -13.291  -13.206
001/04:30:00     21.111  28.843    92.560  -64.035  -13.884  -14.640
001/05:30:00     23.889  31.757    95.117  -65.166  -14.268  -15.684
001/06:30:00     26.667  34.531    95.938  -65.133  -14.391  -16.414
001/07:30:00     29.444  37.166    95.127  -63.954  -14.269  -16.904
001/08:30:00     32.222  39.732    93.318  -62.199  -13.998  -17.121
001/09:30:00     35.000  42.305    91.163  -60.505  -13.675  -16.984
001/10:30:00     37.778  44.906    88.888  -59.035  -13.333  -16.520
001/11:30:00     40.556  47.519    86.484  -57.670  -12.973  -15.841
001/12:30:00     43.333  50.128    83.941  -56.279  -12.591  -15.072
001/13:30:00     46.111  52.720    81.256  -54.734  -12.188  -14.334
001/14:30:00     48.889  55.286    78.427  -52.980  -11.764  -13.683
001/15:30:00     51.667  57.830    75.456  -51.046  -11.318  -13.091
001/16:30:00     54.444  60.356    72.346  -48.963  -10.852  -12.530
                   air    lamp      lamp      air    ballast    walls
                    C       C        C         W        W         W
```

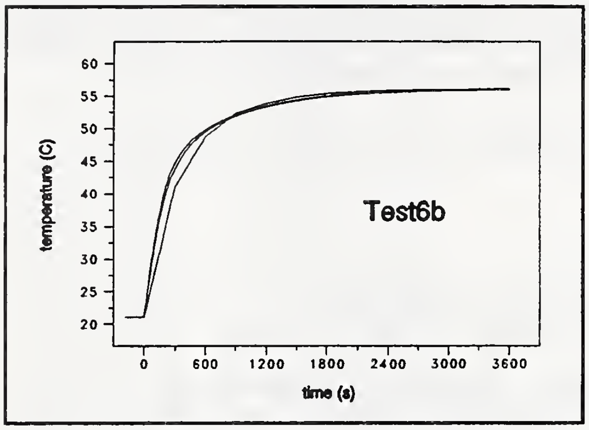
The sum of the cooling loads should be equal to (minus) the power into the light.

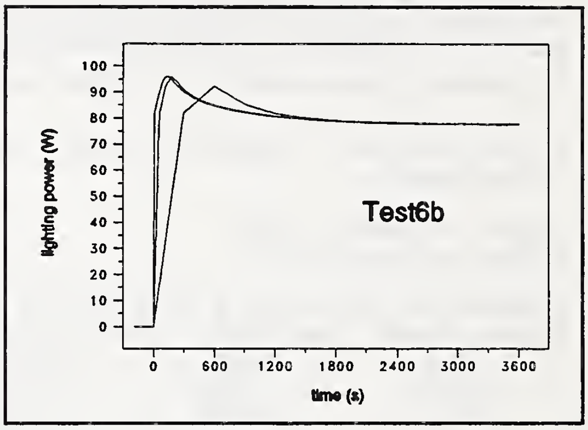
C.2.6b TEST6B: lmp, transient

TEST6B checks the transient operation of the luminaire model, in particular the effect of using different time steps. The thermal network consists of the lamps linked radiatively to wall and housing surfaces and room air linked convectively to all surfaces. Nonlinear convection and radiation models are used. The summary reports indicate the lighting energy and cooling requirements for each hour.

```
title TEST 6B:  luminaire calcs - test time step vs. transient response
units English

signal air-temp t 70.
signal switch d 0.0

element hd_tube  hcv 0.0  0.0  0.0 0.19 0.33 0.0 0.19 0.33
element tube  lmp  48.0  3200.  372.  1.57  0.90  0.64  .2  11
      0.0         0.661        0.022
     20.0         0.672        0.068  ! (Ref: IES Lighting Handbook,
     40.0         0.704        0.165  !  Ref Volume, 1984, Fig 8-34)
     60.0         0.818        0.476
     70.0         0.894        0.701
     80.0         0.971        0.865
     90.0         0.999        0.960
    100.0         0.965        0.995
    120.0         0.858        0.850
    140.0         0.721        0.697
    215.0         0.000        0.000

node Housing mas v 1. 65. 100. 70. 6.283  10.  0.2  .90   ! 6" dia, 10# mass
node Lum_Air air v 1. 65. 120. 70. 0.785                  ! 6" dia, 4' long
node Tubes   eqp v 1. 65. 140. 70. tube  2
node RoomAir air c 1. 65.  75. 70. 100.  air-temp
node Walls   mas v 1. 65.  75. 70. 400. 10.  0.2  .90

link light  lum  Tubes  5. switch .15 Housing 2
    Housing 0.1    Walls   0.9
link link-1 cnv  hd_tube  Housing   Lum_Air  null
link link-2 cnv  hd_tube  Tubes     Lum_Air  null
link link-3 hcc   1.50    Housing   RoomAir
link link-4 rad  0.8572   Tubes     Housing   ! A1=3.14; A2=6.283
! sF = 1 / ( (1-.9)/.9 + 1 + 3.14*(1-.9)/(6.28*.9) )
link link-5 hcc   1.00    Walls   RoomAir
! link Link-6 knd   4.0     Housing  Tubes

times 1 300

report  test6b.rpt  6  300
 n  Tubes      T  i  %8.3f
 l  light      Q  i  %9.3f
 n  RoomAir    Q  i  %9.3f
 n  Lum_Air    T  i  %8.3f
 n  Housing    T  i  %8.3f
 n  Walls      T  i  %8.3f

report  test6b.sum  2  3600
 l  light      Q  s  %9.3f
 n  RoomAir    Q  s  %9.3f

display  3  300
 n  RoomAir    Q  i  %12.4f
 n  Tubes      T  i  %8.3f
 n  Housing    T  i  %8.4f
* end of data
```

Results for tube temperature and power consumption for time steps of 10, 60, and 300 seconds are displayed in the following graphs.





The results for the 10 and 60 second time step runs are almost identical. The 300 second time step results are quite different. Most of the difference is because the transient response of the tube is much faster than 300 seconds. This shows up in the computed energies (trapezoidal integrations of the lighting power and cooling loads):

	10 seconds		60 seconds		300 seconds	
time	Qlight	Qroom	Qlight	Qroom	Qlight	Qroom
00:00:00	0.000	0.000	0.000	0.000	0.000	0.000
01:00:00	80.795	-62.121	80.127	-62.495	77.372	-61.172
02:00:00	77.532	-77.392	77.529	-77.414	77.517	-77.483

Most of the difference in the first hour is due to the inability of the trapezoidal integration to capture the rapidly changing loads. This should not lead to long term errors in energy predictions because a similar error of opposite sign occurs when the light is turned off.

In this case the choice of time step has very little difference on the modeling of energy requirements. This test should probably be performed on each piece of equipment to insure that an appropriate time step is being used.

C.2.6c TEST6C: eqp node

TEST6C is a test of the generic equipment model and the DEF signal controls. The computed loads relate directly to the equipment capacities.

```
title TEST 6C:  generic equipment model
units English

signal air-temp t 70.
signal switch d 0.0

element WithMass   eqp  100.  5.0  0.90  20.0  0.2
element Massless   eqp  100.  5.0  0.90  20.0  0.0

node RoomAir1 air c 1. 70. 70. 70. 1000.  air-temp
node Equip1   eqp v 1. 70. 77. 70. WithMass  2
node RoomAir2 air c 1. 70. 70. 70. 1000.  air-temp
node Equip2   eqp v 1. 70. 77. 70. Massless  1

link eqp1    eqp  Equip1  switch
link fan1    hcc 10.0  Equip1  RoomAir1
link eqp2    eqp  Equip2  switch
link fan2    hcc 10.0  Equip2  RoomAir2

times 1 120

report  test6c.rpt  4  120
 n  Equip1     T  i  %8.3f
 n  RoomAir1   Q  i  %9.3f
 n  Equip2     T  i  %8.3f
 n  RoomAir2   Q  i  %9.3f

display  4  120
 n  Equip1     T  i  %8.3f
 n  RoomAir1   Q  i  %9.3f
 n  Equip2     T  i  %8.3f
 n  RoomAir2   Q  i  %9.3f

* end of data
```

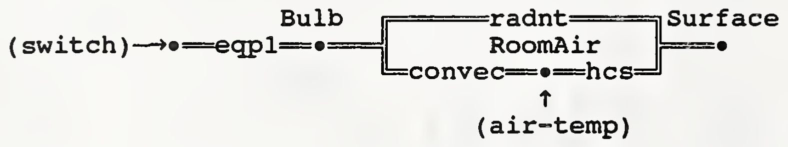
Contents of DEF:

```
001/00:00:00
001/00:30:00  switch   1.0
001/01:00:00  switch   0.5
001/01:30:00  switch   1.0
001/02:30:00  switch   0.0
002/00:00:00
```

Each equipment element has a nominal power consumption of 100W. Link eqp1 invokes two pieces of equipment which should create a cooling load on node RoomAir1 of 200W times the switch value. Link eqp2 represents on one piece of equipment, so the expected cooling load on node RoomAir2 is 100W times the switch value. The shape of the cooling power curves is different because one type of equipment has mass.

C.2.6d TEST6D: eqp, fast transient

TEST6D is a test of the modeling of nonlinear effects in a case involving very fast transients. Testing at different time steps indicates the largest time step necessary to get an accurate solution. This model of a 100 watt light bulb indicates the smallest time step necessary to model thermal equipment. The results of this simulation are described in Section A.3.2 and figure A.3.



```
title TEST 6D:  low mass, high power equipment
units English

signal air-temp t 77.
signal switch d 0.0

element 100Wbulb  eqp  100.  0.213  0.90  0.06  0.2
element hcsphere  hcv  0  0  0.118 0.785 0.25  0.118 0.786 0.25

node RoomAir  air c 1. 75.  79. 77. 10.    air-temp
node Bulb     eqp v 1. 75. 380. 77. 100Wbulb  1
node Surface  mas v 1. 75.  79. 77. 1000.  0.01  0.20  0.9

link eqp1    eqp  Bulb      switch
link convec  cnv  hcsphere  Bulb    RoomAir  null
link radnt   rad  0.8981    Bulb    Surface
link hcs     hcc  1.0       Surface RoomAir

times 1 60

report  test6d.rpt  3  60
 n  Bulb     T  i  %8.3f
 n  RoomAir  Q  i  %9.3f
 n  Surface  T  i  %8.3f

report  test6d.sum  1  60
 n  RoomAir  Q  s  %9.3f

display  2  60
 n  Bulb     T  i  %8.3f
 n  RoomAir  Q  i  %9.3f

* end of data
```

C.2.7a TEST7A: cfr, afp, DEF

TEST7A checks the simple air flow calculation procedure. This case consists of a specified air flow from a cold to a warm room and an equal return flow to maintain a mass balance at each node. These flows create easily computed heating and cooling loads. The interaction of DEF with the control signals is checked.

```
signal Cold t 50.
signal Room t 68.
signal Flow d 1.0

node node-1 air c 1. 32. 50. 50. 1000.  Cold
node node-2 air c 1. 68. 70. 68. 1000.  Room

element 100cfm cfr 100.

link link-1 afp 100cfm  node-1 node-2 Flow
link link-2 afp 100cfm  node-2 node-1 Flow

times 1 3600

report test7a.rpt  3  3600
 l  link-1  F  i  %12.5e
 n  node-1  Q  i  %12.5e
 n  node-2  Q  i  %12.5e

display  2  3600
 n  node-1  Q  i  %12.5e
 n  node-2  Q  i  %12.5e

* end of data
```

BVF:

```
0
001/00:00:00
001/06:00:00
```

DEF:

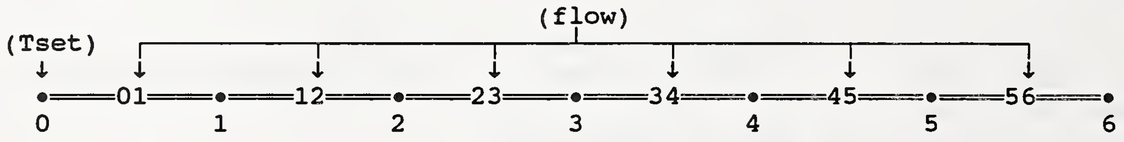
```
001/00:00:00
001/02:00:00 Flow 0.5
001/03:00:00 Cold 32.0
001/04:00:00 Flow 1.0
001/04:00:00 Cold 50.0
001/05:00:00 Room 70.0
002/00:00:00
```

Expected results:

```
     time period      flow (l/s)  ΔT (°C)  loads (W)
00:00:00 - 02:00:00     154.8      10.0      569.7
02:00:00 - 03:00:00      77.4      10.0      284.9
03:00:00 - 04:00:00      77.4      20.0      569.7
04:00:00 - 05:00:00     154.8      10.0      569.7
05:00:00 - 06:00:00     154.8      11.1      633.0
```

C.2.7b TEST7B: duct model

TEST7B shows what can happen if the simple air flow calculation is used to simulate transient flow in a duct. It consists of seven air nodes connected in series by six airflow paths. Each node contains 100 m^3 of air, and the air flow rate is 100 m^3/min. The air node is assumed to represent a fully mixed volume of air.



```
signal Tset t 0.0
signal Flow d 1.0

node node-0 air c 1. 0.0 1.0 0.0 100.  Tset
node node-1 air v 1. 0.0 1.0 0.0 100.
node node-2 air v 1. 0.0 1.0 0.0 100.
node node-3 air v 1. 0.0 1.0 0.0 100.
node node-4 air v 1. 0.0 1.0 0.0 100.
node node-5 air v 1. 0.0 1.0 0.0 100.
node node-6 air v 1. 0.0 1.0 0.0 100.

element flow cfr 100.

link link-01 afp  flow   node-0 node-1 Flow
link link-12 afp  flow   node-1 node-2 Flow
link link-23 afp  flow   node-2 node-3 Flow
link link-34 afp  flow   node-3 node-4 Flow
link link-45 afp  flow   node-4 node-5 Flow
link link-56 afp  flow   node-5 node-6 Flow
 ...
```

Consider the case of "plug flow" in a duct, i.e., there is no mixing in the direction of flow. In this case a temperature pulse entering at the upstream end of the duct will propagate along the duct without changing shape. This condition is achieved in the model presented above when the time step exactly matches the time for one air change in each air node, or one minute. The results for such a case are:

```
001/00:00:00  0.000 0.000 0.000 0.000 0.000 0.000 0.000
001/00:01:00  0.000 0.000 0.000 0.000 0.000 0.000 0.000
001/00:02:00  1.000 0.000 0.000 0.000 0.000 0.000 0.000
001/00:03:00  1.000 0.998 0.000 0.000 0.000 0.000 0.000
001/00:04:00  1.000 1.000 0.996 0.000 0.000 0.000 0.000
001/00:05:00  1.000 1.000 1.000 0.993 0.000 0.000 0.000
001/00:06:00  1.000 1.000 1.000 1.000 0.991 0.000 0.000
001/00:07:00  1.000 1.000 1.000 1.000 1.000 0.989 0.000
001/00:08:00  1.000 1.000 1.000 1.000 1.000 1.000 0.987
001/00:09:00  1.000 1.000 1.000 1.000 1.000 1.000 1.000
001/00:10:00  1.000 1.000 1.000 1.000 1.000 1.000 1.000
 ...
001/00:15:00  1.000 1.000 1.000 1.000 1.000 1.000 1.000
```

However, when the time step is reduced to 30 seconds or increased to two minutes, the step change in temperature is diffused as it propagates through the duct. This is an inherent property in this solution method and is called "numerical diffusion" as opposed to any real diffusion which might occur.

time step = 30 seconds:

```
001/00:00:00  0.000 0.000 0.000 0.000 0.000 0.000 0.000
 ...
001/00:02:00  1.000 0.000 0.000 0.000 0.000 0.000 0.000
001/00:02:30  1.000 0.499 0.000 0.000 0.000 0.000 0.000
001/00:03:00  1.000 0.749 0.249 0.000 0.000 0.000 0.000
001/00:03:30  1.000 0.874 0.498 0.124 0.000 0.000 0.000
001/00:04:00  1.000 0.937 0.686 0.311 0.062 0.000 0.000
001/00:04:30  1.000 0.968 0.811 0.498 0.186 0.031 0.000
001/00:05:00  1.000 0.984 0.890 0.654 0.342 0.108 0.015
001/00:05:30  1.000 0.992 0.937 0.772 0.498 0.225 0.062
001/00:06:00  1.000 0.996 0.964 0.854 0.634 0.361 0.143
001/00:06:30  1.000 0.998 0.980 0.909 0.744 0.497 0.252
001/00:07:00  1.000 0.999 0.989 0.945 0.826 0.620 0.374
001/00:07:30  1.000  1.00 0.994 0.967 0.885 0.723 0.497
001/00:08:00  1.000  1.00 0.997 0.980 0.926 0.804 0.610
001/00:08:30  1.000  1.00 0.998 0.989 0.953 0.865 0.707
001/00:09:00  1.000  1.00 0.999 0.993 0.971 0.909 0.786
001/00:09:30  1.000  1.00 0.999 0.996 0.982 0.940 0.847
001/00:10:00  1.000  1.00  1.00 0.998 0.989 0.961 0.893
001/00:10:30  1.000  1.00  1.00 0.999 0.993 0.975 0.927
001/00:11:00  1.000  1.00  1.00 0.999 0.996 0.984 0.951
001/00:11:30  1.000  1.00  1.00  1.00 0.998 0.990 0.968
001/00:12:00  1.000  1.00  1.00  1.00 0.999 0.994 0.979
001/00:12:30  1.000  1.00  1.00  1.00 0.999 0.996 0.986
001/00:13:00  1.000  1.00  1.00  1.00  1.00 0.998 0.991
001/00:13:30  1.000  1.00  1.00  1.00  1.00 0.999 0.995
001/00:14:00  1.000  1.00  1.00  1.00  1.00 0.999 0.997
001/00:14:30  1.000  1.00  1.00  1.00  1.00  1.00 0.998
001/00:15:00  1.000  1.00  1.00  1.00  1.00  1.00 0.999
```

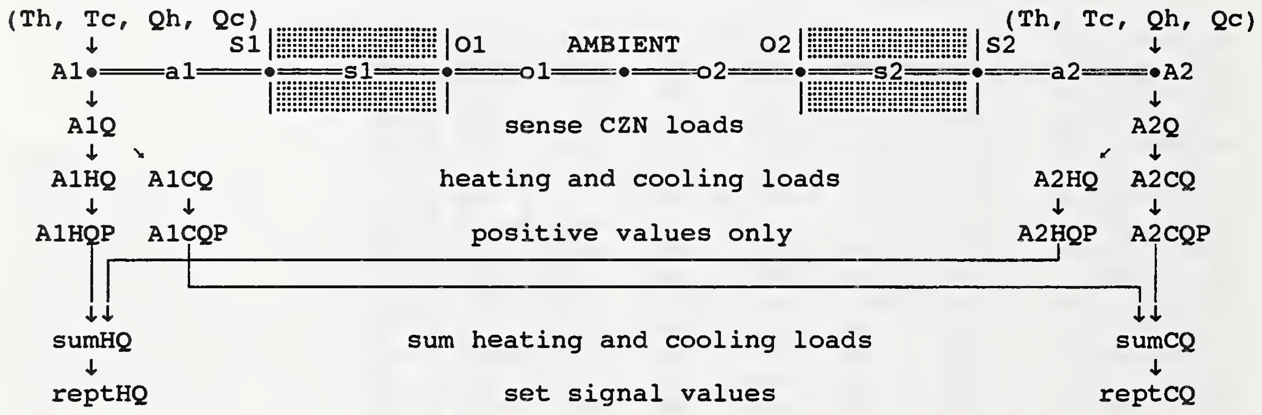
time step = 120 seconds:

```
001/00:00:00  0.000 0.000 0.000 0.000 0.000 0.000 0.000
001/00:02:00  1.000 0.000 0.000 0.000 0.000 0.000 0.000
001/00:04:00  1.000 0.666 0.444 0.296 0.197 0.131 0.087
001/00:06:00  1.000 0.889 0.740 0.592 0.460 0.350 0.262
001/00:08:00  1.000 0.963 0.888 0.789 0.679 0.570 0.467
001/00:10:00  1.000 0.988 0.954 0.899 0.826 0.740 0.649
001/00:12:00  1.000 0.996 0.982 0.954 0.912 0.854 0.786
001/00:14:00  1.000 0.999 0.993 0.980 0.957 0.923 0.877
```

These results indicate that a different solution procedure must be used to model transient flows in ducts. Fortunately, it should be relatively straight-forward to develop a method which employs a moving grid system, i.e. a Lagrangian solution instead of the more common Eulerain solution, taking advantage of features of the C language such as memory allocation.

C.2.8a TEST8A: controls for reports

TEST8A shows the use of controls and signals to process heating and cooling loads from multiple zones to create reportable values. The zones happen to be identical in this case and are taken from TEST3B. The following signal/control paths compute the total heating and cooling loads. Another set of paths not shown compute the average surface temperatures.

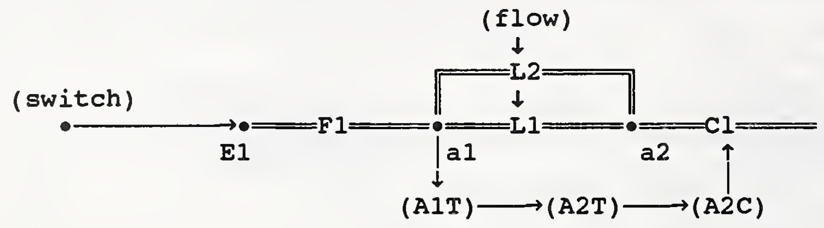


```
title TEST 8A:  control sensors and signal processors
units English

signal Th t 70.
signal Tc t 75.
signal Qh q +80.
signal Qc q -80.
signal zero       d 0.0
signal A1Q        d 0.0    /* a1 load */
signal A1CQ       d 0.0    /* a1 cooling load */
signal A1CQP      d 0.0    /* a1 cooling load (+) */
signal A1HQ       d 0.0    /* a1 heating load */
signal A1HQP      d 0.0    /* a1 heating load (+) */
signal A2Q        d 0.0    /* a2 load */
signal A2CQ       d 0.0    /* a2 cooling load */
signal A2CQP      d 0.0    /* a2 cooling load (+) */
signal A2HQ       d 0.0    /* a2 heating load */
signal A2HQP      d 0.0    /* a2 heating load (+) */
signal sumCQ      d 0.0    /* total cooling load */
signal sumHQ      d 0.0    /* total heating load */
signal reptCQ     q 0.0    /* cooling load as Btu/h */
signal reptHQ     q 0.0    /* heating load as Btu/h */
signal S1T        d 0.0    /* s1 temperature */
signal O1T        d 0.0    /* o1 temperature */
signal S2T        d 0.0    /* s2 temperature */
signal O2T        d 0.0    /* o2 temperature */
signal sumT       d 0.0    /* sum of temperatures */
signal avgT       d 0.0    /* average of temperatures */
signal reptT      t 0.0    /* average temperature [F] */

node node-A1 czn c 1. 70. 75. 70. 100.  Th Qh Tc Qc
node node-S1 srf v 1. 69. 76. 70. 100. .90
node node-O1 srf v 1. 60. 85. 60. 100. .90
node ambient air b 1. 60. 85. 60. 100.
node node-O2 srf v 1. 60. 85. 60. 100. .90
node node-S2 srf v 1. 69. 76. 70. 100. .90
node node-A2 czn c 1. 70. 75. 70.   0.  Th Qh Tc Qc
```

```
link link-a1 hcc  1.0  node-S1 node-A1
link link-s1 knd 12.5  node-O1 node-S1
link link-o1 hcc  1.0  node-O1 ambient
link link-a2 hcc  1.0  node-S2 node-A2
link link-s2 knd 12.5  node-O2 node-S2
link link-o2 hcc  1.0  node-O2 ambient

control A1Q   ps  node-A1  A1Q   0.0 1.0 0.0
control A2Q   ps  node-A2  A2Q   0.0 1.0 0.0
control A1CQ  llc A1Q  A1CQ  0.0
control A2CQ  llc A2Q  A2CQ  0.0
control A1CQP  pos A1CQ  A1CQP  0.0
control A2CQP  pos A2CQ  A2CQP  0.0
control sumCQ   sum  2 A1CQP A2CQP  sumCQ
control reptCQ  sp  sumCQ  reptCQ  0.0 1.0
control A1HQ  ulc A1Q  A1HQ  zero
control A2HQ  ulc A2Q  A2HQ  zero
control A1HQP  pos A1HQ  A1HQP  0.0
control A2HQP  pos A2HQ  A2HQP  0.0
control sumHQ   sum  2 A1HQP A2HQP  sumHQ
control reptHQ  sp  sumHQ  reptHQ  0.0 1.0
control S1T   ts  node-s1  S1T   0.0 1.0 0.0
control O1T   ts  node-o1  O1T   0.0 1.0 0.0
control S2T   ts  node-s2  S2T   0.0 1.0 0.0
control O2T   ts  node-o2  O2T   0.0 1.0 0.0
control sumT  sum  4 S1T O1T S2T O2T  sumT
control avgT  mod  sumT  avgT  0.0 0.25
control reptT st  avgT  reptT  0.0 1.0

times 1 60

report test8a.rpt  7  60
 n  ambient  T  i  %8.3f
 n  node-a1  T  i  %8.3f
 n  node-a1  Q  i  %9.3f
 n  node-a2  T  i  %8.3f
 n  node-a2  Q  i  %9.3f
 s  reptCQ   0  i  %9.3f
 s  reptHQ   0  i  %9.3f

report test8a.sum  5  60
 n  node-s1  T  i  %8.3f
 n  node-o1  T  i  %8.3f
 n  node-s2  T  i  %8.3f
 n  node-o2  T  i  %8.3f
 s  reptT    0  i  %9.3f

display  5  60
 n  ambient  T  i  %8.3f
 n  node-a1  T  i  %8.3f
 n  node-a1  Q  i  %9.3f
 n  node-a2  T  i  %8.3f
 n  node-a2  Q  i  %9.3f

* end of data
```

C.2.8b TEST8B: pc

This is a test of the proportional control. The network is similar to TEST6C with the addition of the controls and an air node to serve as a cooling coil. Results are summarized under TEST8D.



```
title TEST 8B:  proportional control & ideal cooling
units English

signal switch   d 1.0      ! equipment switch
signal flow     d 1.0      ! flow rate control
signal A1T      d 0.0      ! node-a1 temperature - cooling setpoint
signal A2T      d 0.0      ! positive A1T
signal A2C      d 0.0      ! signal to cooling coil


element WithMass   eqp  100.  5.0  0.90  20.0  0.2
element Massless   eqp  100.  5.0  0.90   0.0  0.2
element flow       cfr  100.0

node node-a1 air v 1. 70. 80. 70. 100.           ! controlled node
node node-a2 air v 1. 50. 85. 70.   0.           ! cooling coil
node Equip1  eqp v 1. 70. 90. 70. WithMass  2    ! imposed load

link link1   afp  flow node-a1 node-a2 flow
link link2   afp  flow node-a2 node-a1 flow
link eqp1    eqp  Equip1  switch
link fan1    hcc  10.0  Equip1 node-a1
link coil    qnd  -1200. node-a2 A2C

control A1T   ts  node-a1  A1T   75.0 1.0 0.0
control A2T  ulc  A1T  A2T  0.0
control cc1   pc  A2T  A2C  0.3

times 1 12

report test8b.rpt  7  60
 n  node-a2  T  i  %8.3f
 n  node-a1  T  i  %8.3f
 n  Equip1   Q  i  %8.3f
 s  A1T      0  i  %8.3f
 s  A2T      0  i  %8.3f
 s  A2C      0  i  %8.3f
 n  node-a2  Q  i  %9.3f

display  4  60
 n  node-a2  T  i  %8.3f
 n  node-a1  T  i  %8.3f
 n  Equip1   T  i  %8.3f
 n  node-a2  Q  i  %9.3f

* end of data
```

C.2.8c TEST8C: pic

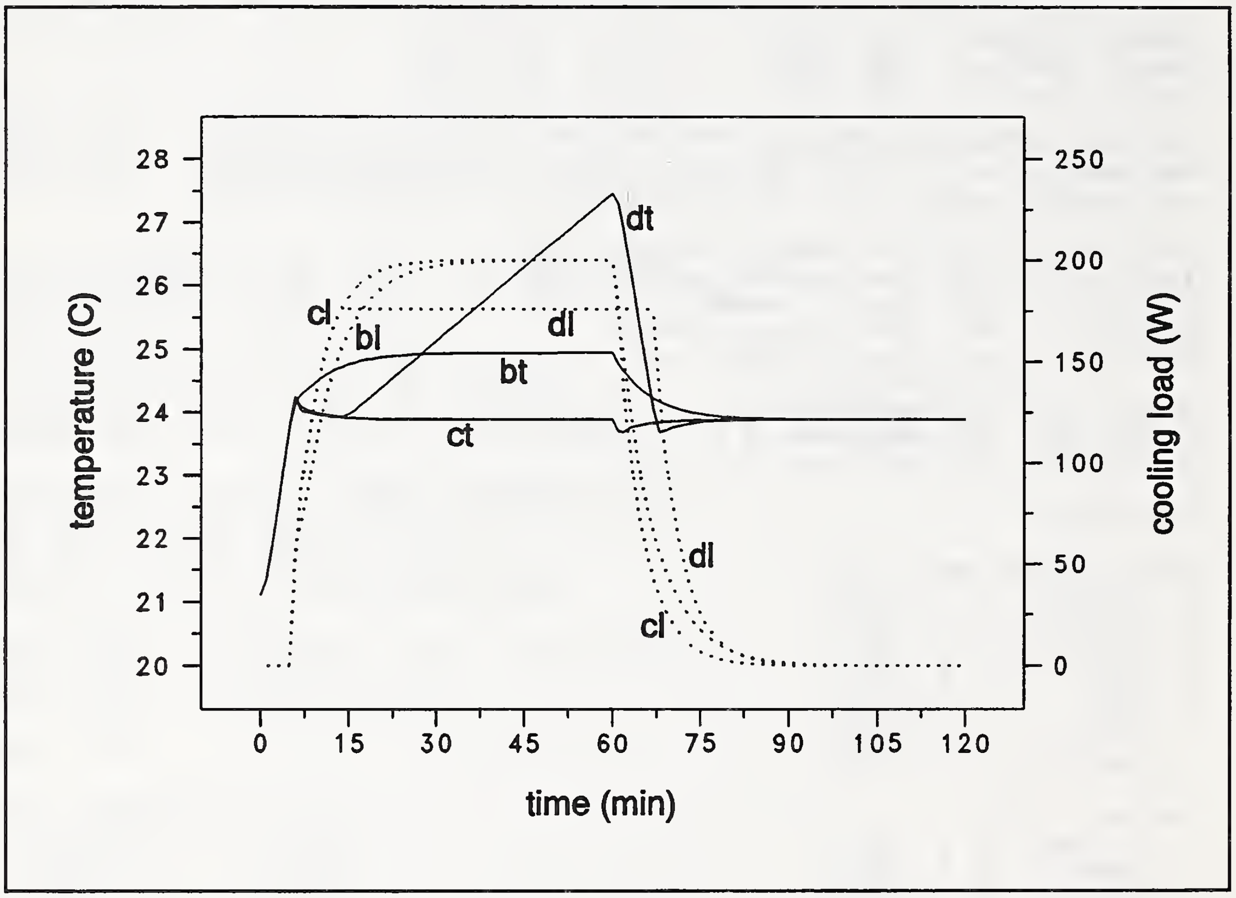
This tests is similar to TEST8B except that proportional/integral control is used instead of simple proportional. Results are summarized under TEST8D.

```
title TEST 8C:  proportional/integral control & ideal cooling
units English

signal switch   d 1.0      ! equipment switch
signal flow     d 1.0      ! flow rate control
signal A1T      d 0.0      ! node-a1 temperature - cooling setpoint
signal A2T      d 0.0      ! positive A1T
signal A2C      d 0.0      ! signal to cooling coil
signal ccap     q -1200.   ! cooling coil capacity

element WithMass   eqp  100.  5.0  0.90  20.0  0.2
element Massless   eqp  100.  5.0  0.90   0.0  0.2
element flow       cfr  100.0

node node-a1 air v 1. 70. 80. 70. 100.          ! controlled node
node node-a2 air v 1. 50. 90. 70.   0.          ! cooling coil
node Equip1  eqp v 1. 70. 90. 70. WithMass  2   ! imposed load

link link1   afp  flow node-a1 node-a2 flow
link link2   afp  flow node-a2 node-a1 flow
link eqp1    eqp  Equip1  switch
link fan1    hcc  10.0  Equip1 node-a1
link coil    qnd  ccap   node-a2 A2C

control A1T   ts  node-a1  A1T   75.0 1.0 0.0
control A2T  ulc  A1T  A2T  0.0
control cc1  pic  A2T  A2C  0.3 0.05

times 1 12

report test8c.rpt  7  60
 n  node-a2  T  i  %8.3f
 n  node-a1  T  i  %8.3f
 n  Equip1   Q  i  %8.3f
 s  A1T      0  i  %8.3f
 s  A2T      0  i  %8.3f
 s  A2C      0  i  %8.3f
 n  node-a2  Q  i  %9.3f

display  4  60
 n  node-a2  T  i  %8.3f
 n  node-a1  T  i  %8.3f
 n  Equip1   T  i  %8.3f
 n  node-a2  Q  i  %9.3f

* end of data
```

C.2.8d TEST8D: pic, capacity limited

TEST8D is identical to TEST8C except that the cooling capacity is limited to 175.8 W (600 Btu/h) which is inadequate to maintain the desired air temperature.

The following figure summarizes the results of tests 8B, 8C, and 8D. At time zero, the equipment heat gain is switched on; at one hour it is switched off. The air temperature responds by increasing from 21.1 C (70 F) into the region where the controls become active. For the proportional control test, the temperature increases to about 25.0 C (about 76 F) and the cooling load to 205.1 W (700 Btu/h). For the PI test the temperature properly seeks the 23.9 C (75 F) control point with a similar load. When the cooling capacity is limited, the air temperature continues increasing. When the equipment is switched off, the air temperature returns to 23.9 C (75 F) in all cases.

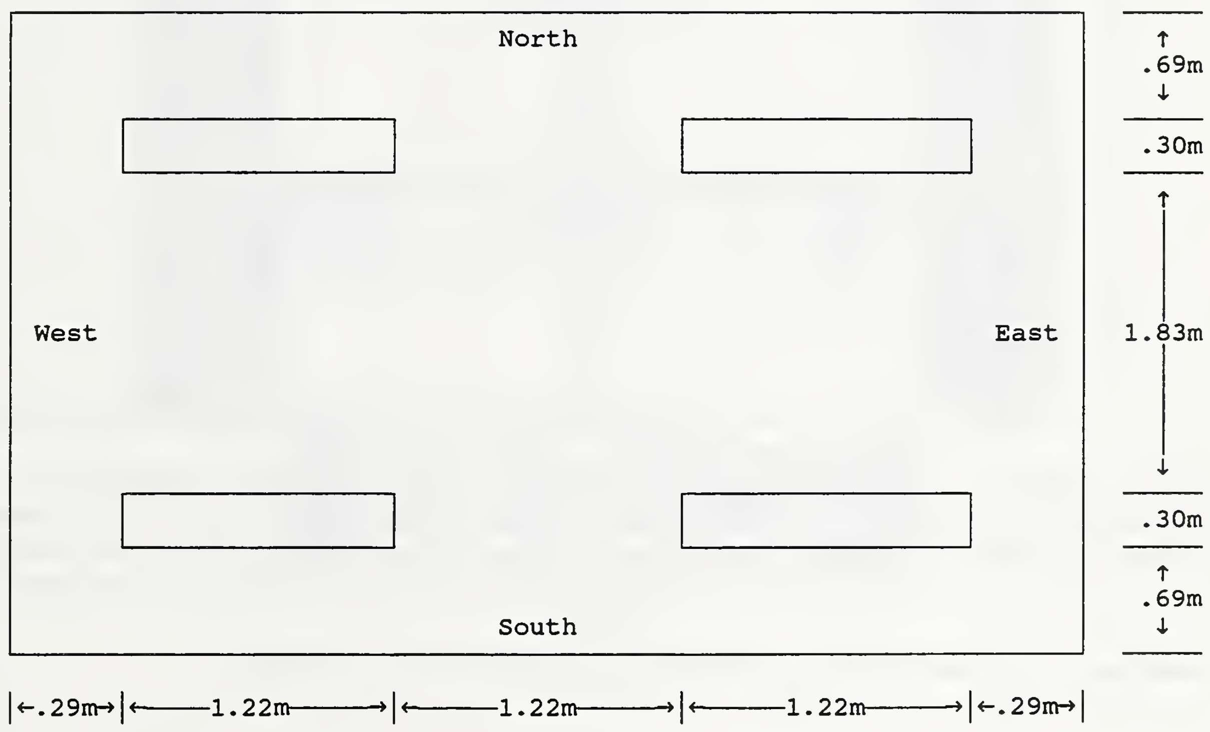


bt - temperature from test 8B
bl - load from test 8B
ct - temperature from test 8C
cl - load from test 8C
dt - temperature from test 8D
dl - load from test 8D

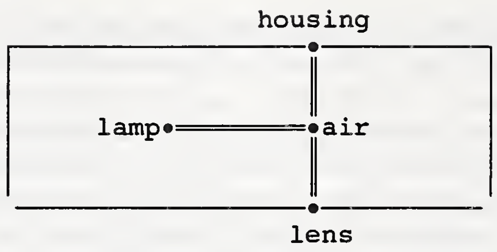
C.3 Experimental Test

The experimental results used to test the HLITE program are generated by the NIST test facility as described by Treado and Bean (1988). This test facility is constructed on a large concrete slab within the NIST environmental chamber. The facility is divided into two sections, a large insulated shell enclosing the test room area, and a smaller attached control room for housing instrumentation as shown in figure 4. The overall height is 6.36m (20'10½"). The test room floor slab is elevated to accommodate a lower plenum beneath the floor, and all other room surfaces are adjacent to temperature-controlled guard air spaces. Duplicate lighting and HVAC systems are installed in both the test room plenum and the lower plenum. The test room floor and ceiling slabs are 64 mm (2½") thick concrete built on steel decks supported by a structural steel framework. The walls are constructed of 16 mm (⅝") thick gypsum board fastened to steel studs.

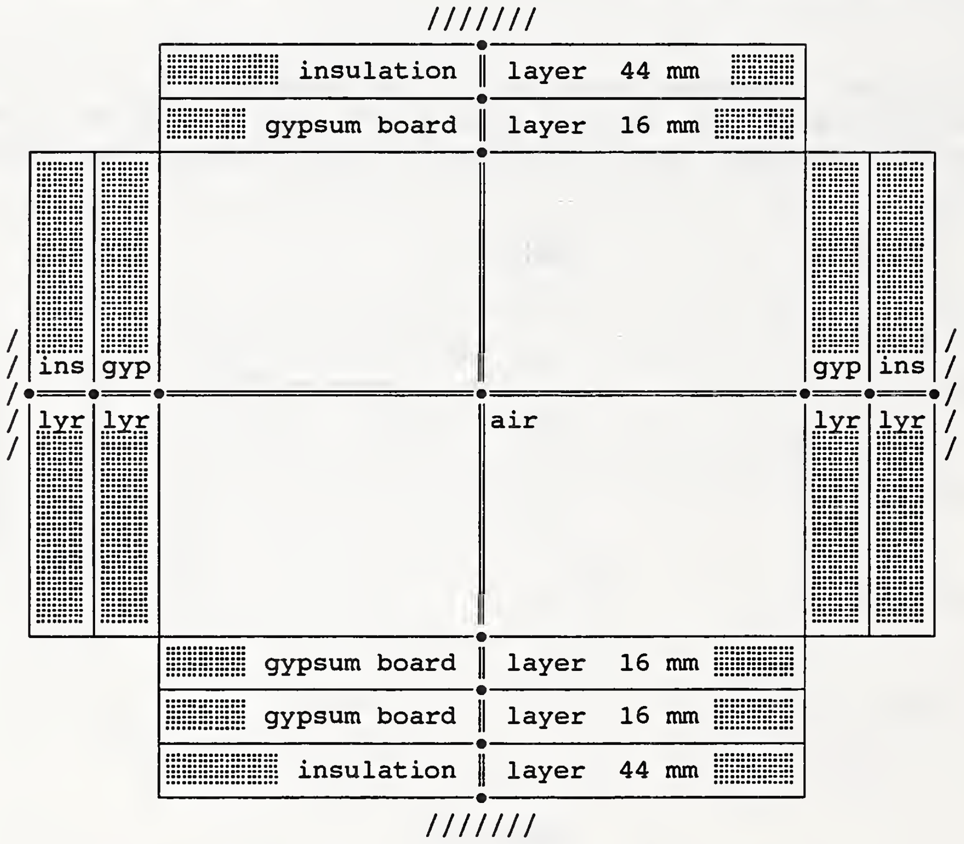
The NIST Test Room is rectangular room 4.23 m (13'10½") long by 3.73 m (12'3") wide with a suspended ceiling 2.44 m (8') high. Above the ceiling is a .76 m (2'6") high plenum. There are four luminaires mounted in the ceiling. The following sketch (not to scale) shows the layout of the luminaires.



Luminaire Model:

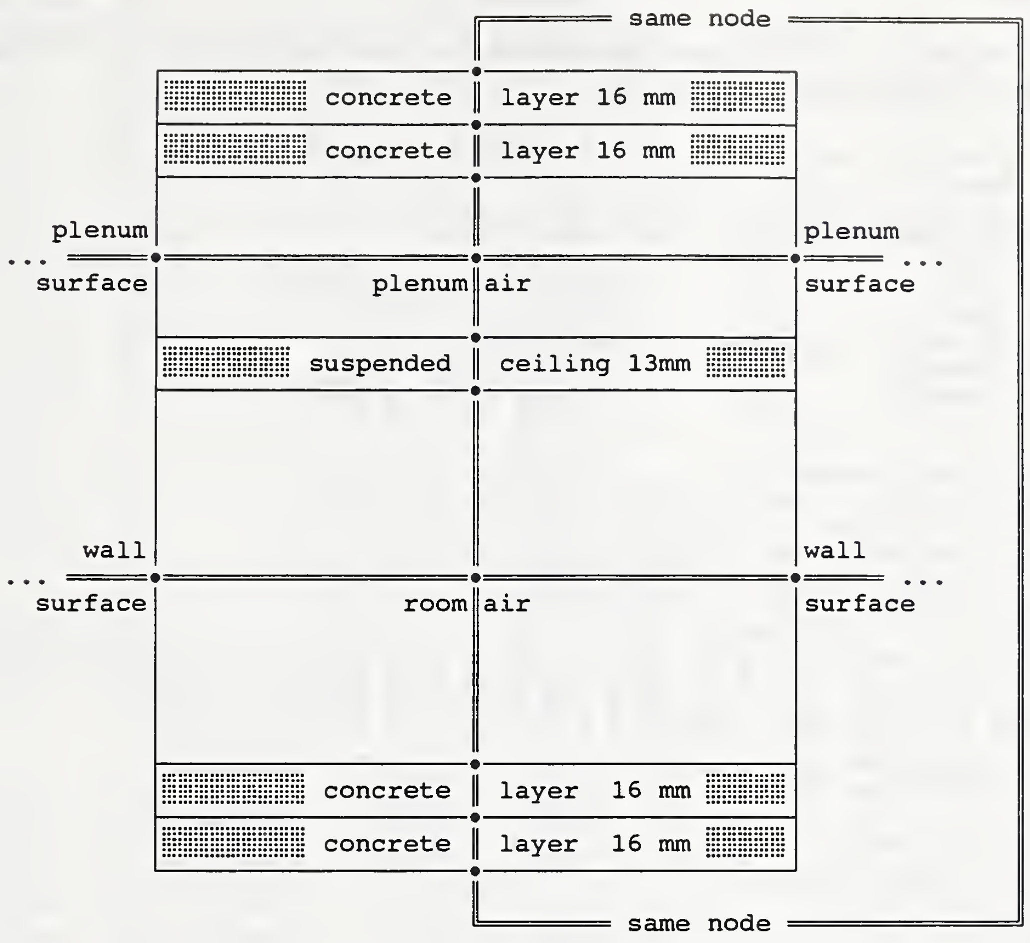


Walls model:



Radiant interchange links between all surfaces are not shown.

Floors and Ceilings:



Radiant interchange links between all surfaces are not shown.

One special feature shown here is that the concrete acts as both ceiling and floor to model a condition where identical rooms are above and below the room being modeled. The walls use the more common adiabatic surface at their centers. This cannot be done on the ceiling/floor because the boundary conditions are not symmetric.

Data Files:

These features of the NIST test room are described in the following NDF file. This file illustrates that complex simulations require long data files.

```
title Test ITd Trans-on CG 160 75 4x2 acryl carpet no furn
units English

signal RoomTemp    t 75.0   ! Room setpoint
signal off         d 0.0
signal flow        d 1.0    ! flow through main path
signal switch      d 0.0    ! switch for lights
signal RoomQ       d 0.0    ! room load
signal RoomCQ      d 0.0    ! room cooling load
signal reptCQ      q 0.0    ! cooling load as Btu/h
signal reptCE      e 0.0    ! cooling load as W

! conduction elements:
element DryWall    mat  0.0521 0.093 50.0 0.26 0.0
element FiberGlass mat  0.146  0.026 0.53 0.2  0.0
element Concrete   mat  0.0683 1.0   140. 0.22 0.0
element Carpet     mat  0.0728 0.035 40.  0.2  0.0
element AccTile    mat  0.0417 0.035 23.  0.2  0.0
! convection coefficients:
element hpl_ceil  hcv 0.21 21.2 0.0 0.13 0.33 0.0 0.29 0.33 !More surface area
element hpl_wall  hcv 0.54 16.0 0.0 0.19 0.33 0.0 0.19 0.33
element hpl_floor hcv 0.71 16.0 0.0 0.22 0.33 0.0 0.10 0.33
element hpl_hous  hcv 1.40 16.0 0.0 0.22 0.33 0.0 0.10 0.33
element hlm_hous  hcv 0.0 0.0 0.0 0.31 0.33 0.0 0.31 0.33
element hlm_tube  hcv 0.0 0.0 0.0 0.31 0.33 0.0 0.31 0.33
element hlm_lens  hcv 0.0 0.0 0.0 0.31 0.33 0.0 0.31 0.33
element hrm_ceil  hcv 2.01 36.9 0.0 0.10 0.33 0.0 0.22 0.33
element hrm_wall  hcv 0.74 14.3 0.0 0.19 0.33 0.0 0.19 0.33
element hrm_floor hcv 0.62 8.24 0.0 0.22 0.33 0.0 0.10 0.33
! standard fluorescent tube:
element tube  lmp  46.5  3200.  372.  1.57  0.90  0.64  .2  12
      0.0         0.661        0.022
     20.0         0.672        0.068
     40.0         0.704        0.165
     60.0         0.818        0.476
     70.0         0.894        0.701
     80.0         0.971        0.865
     90.0         0.994        0.960
    100.0         0.994        0.995
    110.0         0.965        0.930
    120.0         0.920        0.850
    140.0         0.750        0.697
    215.0         0.000        0.000
! flow element:
element maxflow  cfr 160.     ! max flow through any path

node UnderSide srf v 1. 73.0 80.0 75.0 169.97 0.90    ! underside of floor
node S-Plenum  srf v 1. 73.0 80.0 75.0  34.69 0.90    ! south wall in plenum
node S-Plen-2  lyr v 1. 73.0 80.0 75.0  34.69
node S-Plen-1  lyr v 1. 73.0 80.0 75.0  34.69
node S-Plen-0  lyr v 1. 73.0 80.0 75.0  34.69
node W-Plenum  srf v 1. 73.0 80.0 75.0  30.63 0.90    ! west wall in plenum
node W-Plen-1  lyr v 1. 73.0 80.0 75.0  30.63
node W-Plen-0  lyr v 1. 73.0 80.0 75.0  30.63
node N-Plenum  srf v 1. 73.0 80.0 75.0  34.69 0.90    ! north wall in plenum
node N-Plen-1  lyr v 1. 73.0 80.0 75.0  34.69
node N-Plen-0  lyr v 1. 73.0 80.0 75.0  34.69
node E-Plenum  srf v 1. 73.0 80.0 75.0  30.63 0.90    ! east wall in plenum
node E-Plen-1  lyr v 1. 73.0 80.0 75.0  30.63
node E-Plen-0  lyr v 1. 73.0 80.0 75.0  30.63
node PlenumSC  srf v 1. 73.0 80.0 75.0 153.97 0.90    ! plenum side of ceiling
node Steel     mas v 1. 73.0 80.0 75.0 200.0  625. 0.12 0.90
node PlAir     air v 1. 73.0 80.0 75.0 419.6
node housing   mas v 1. 73.0 90.0 75.0  29.33  64. 0.21 0.90  ! luminaire
node tubes     eqp v .8 73.0 130.0 75.0 tube  8
node LmAir     air v 1. 73.0 105.0 75.0  5.33
node AcrLens   mas v 1. 73.0 85.0 75.0  16.0   5.4 0.2  0.90
node SspCeil   srf v 1. 73.0 80.0 75.0 153.97 0.90    ! room side of ceiling
node S-Wall    srf v 1. 73.0 80.0 75.0 111.00 0.90    ! south wall in room
node S-Wall-2  lyr v 1. 73.0 80.0 75.0 111.00
node S-Wall-1  lyr v 1. 73.0 80.0 75.0 111.00
node S-Wall-0  lyr v 1. 73.0 80.0 75.0 111.00
node W-Wall    srf v 1. 73.0 80.0 75.0  98.00 0.90    ! west wall in room
node W-Wall-1  lyr v 1. 73.0 80.0 75.0  98.00
node W-Wall-0  lyr v 1. 73.0 80.0 75.0  98.00
node N-Wall    srf v 1. 73.0 80.0 75.0 111.00 0.90    ! north wall in room
node N-Wall-1  lyr v 1. 73.0 80.0 75.0 111.00
node N-Wall-0  lyr v 1. 73.0 80.0 75.0 111.00
node E-Wall    srf v 1. 73.0 80.0 75.0  98.00 0.90    ! east wall in room
node E-Wall-1  lyr v 1. 73.0 80.0 75.0  98.00
node E-Wall-0  lyr v 1. 73.0 80.0 75.0  98.00
node Floor     srf v 1. 73.0 80.0 75.0 169.97 0.90    ! floor
node Floor-4   lyr v 1. 73.0 80.0 75.0 169.97 0.90    ! floor layers
node Floor-3   lyr v 1. 73.0 80.0 75.0 169.97 0.90
node Floor-2   lyr v 1. 73.0 80.0 75.0 169.97 0.90
node Floor-1   lyr v 1. 73.0 80.0 75.0 169.97 0.90
node RmAir     air c 1. 73.0 80.0 75.0 1360.0 RoomTemp ! const T room air

link Flow-rp   afp  maxflow     RmAir    PlAir   flow   ! air flow paths
link Flow-rl   afp  maxflow     RmAir    LmAir   off
link Flow-lp   afp  maxflow     LmAir    PlAir   off
link Flow-pr   afp  maxflow     PlAir    RmAir   flow
link S-Plen-2  cnd  DryWall     S-Plenum S-Plen-2    ! plenum conduction paths
link S-Plen-1  cnd  DryWall     S-Plen-2 S-Plen-1
link S-Plen-0  cnd  FiberGlass  S-Plen-1 S-Plen-0
link W-Plen-1  cnd  DryWall     W-Plenum W-Plen-1
link W-Plen-0  cnd  FiberGlass  W-Plen-1 W-Plen-0
link N-Plen-1  cnd  DryWall     N-Plenum N-Plen-1
link N-Plen-0  cnd  FiberGlass  N-Plen-1 N-Plen-0
link E-Plen-1  cnd  DryWall     E-Plenum E-Plen-1
link E-Plen-0  cnd  FiberGlass  E-Plen-1 E-Plen-0
link Ceiling   cnd  AccTile     PlenumSC SspCeil
link FinEffct  knd  200.        UnderSide Steel
! plenum convection:
link Undsid-c  cnv  hpl_ceil    UnderSide PlAir Flow-pl  420.  200.
link S-Plen-c  cnv  hpl_wall    S-Plenum PlAir  Flow-pl  420.  200.
link W-Plen-c  cnv  hpl_wall    W-Plenum PlAir  Flow-pl  420.  200.
link N-Plen-c  cnv  hpl_wall    N-Plenum PlAir  Flow-pl  420.  200.
link E-Plen-c  cnv  hpl_wall    E-Plenum PlAir  Flow-pl  420.  200.
link C-Plen-c  cnv  hpl_floor   PlenumSC PlAir  Flow-pl  420.  200.
link H-Plen-c  cnv  hpl_hous    housing  PlAir  Flow-pl  420.  200.
link Steel-c   cnv  hpl_wall    Steel    PlAir  Flow-pl  420.  200.

link PlenRad vfm 8      ! data for plenum geometry
UnderSide               ! area: 169.9688
.000000 .066438 .055097 .068282 .054832 .367727 .046498 .341126
S-Plenum                ! area: 34.6875
.325547 .000000 .063638 .019546 .063338 .300087 .058326 .169518
W-Plenum                ! area: 30.6250
.305786 .072080 .000000 .073309 .008029 .303667 .051566 .185562
N-Plenum                ! area: 34.6875
.334582 .019546 .064723 .000000 .064525 .310398 .053571 .152655
```

```
E-Plenum                  ! area: 30.6250
.304319 .071740 .008029 .073084 .000000 .303338 .051649 .187842
PlenumSC                  ! area: 153.9688
.405940 .067606 .060401 .069929 .060335 .000000 .040425 .295364
housing                   ! area: 29.3200
.269552 .069003 .053862 .063378 .053947 .212284 .002752 .275222
Steel                     ! area: 200.0000
.289903 .029401 .028414 .026476 .028763 .227384 .040348 .329311

link Luminaire lum  tubes  8.0
    switch 0.1625 housing 8
    tubes   0.0296
    housing 0.0734
    Floor   0.493
    SspCeil 0.0269
    E-Wall  0.0925
    S-Wall  0.0965
    W-Wall  0.0959
    N-Wall  0.0922
link Lamphs-c  cnv  hlm_hous    housing  LmAir  null  ! luminaire heat flow paths
link Lamptb-c  cnv  hlm_tube    tubes    LmAir  null
link Lampln-c  cnv  hlm_lens    AcrLens  LmAir  null
link LumCond1  knd   4.0  housing PlenumSC
link LumCond2  knd   4.0  housing SspCeil
link LampRad vfm 3        ! data for 2 tube luminaire
AcrLens                   ! area: 1.0000 * 16
.000000 .709451 .290549
housing                   ! area: 1.8340 * 16
.386833 .362745 .250422
tubes                     ! area: 0.7806 * 16
.372213 .588362 .039426

link S-Wall-2  cnd  DryWall     S-Wall   S-Wall-2    ! room conduction paths
link S-Wall-1  cnd  DryWall     S-Wall-2 S-Wall-1
link S-Wall-0  cnd  FiberGlass  S-Wall-1 S-Wall-0
link W-Wall-1  cnd  DryWall     W-Wall   W-Wall-1
link W-Wall-0  cnd  FiberGlass  W-Wall-1 W-Wall-0
link N-Wall-1  cnd  DryWall     N-Wall   N-Wall-1
link N-Wall-0  cnd  FiberGlass  N-Wall-1 N-Wall-0
link E-Wall-1  cnd  DryWall     E-Wall   E-Wall-1
link E-Wall-0  cnd  FiberGlass  E-Wall-1 E-Wall-0
link Floor-4   cnd  Carpet      Floor   Floor-4
link Floor-3   cnd  Concrete    Floor-4 Floor-3
link Floor-2   cnd  Concrete    Floor-3 Floor-2
link Floor-1   cnd  Concrete    Floor-2 Floor-1
link Floor-0   cnd  Concrete    Floor-1 UnderSide
! room convection:
link SspCel-c  cnv  hrm_ceil    SspCeil  RmAir  Flow-pl  1360.  200.
link AcLens-c  cnv  hrm_ceil    AcrLens  RmAir  Flow-pl  1360.  200.
link S-Wall-c  cnv  hrm_wall    S-Wall   RmAir  Flow-pl  1360.  200.
link W-Wall-c  cnv  hrm_wall    W-Wall   RmAir  Flow-pl  1360.  200.
link N-Wall-c  cnv  hrm_wall    N-Wall   RmAir  Flow-pl  1360.  200.
link E-Wall-c  cnv  hrm_wall    E-Wall   RmAir  Flow-pl  1360.  200.
link Floor-c   cnv  hrm_floor   Floor    RmAir  Flow-pl  1360.  200.

link RoomRad vfm 7        ! data for room geometry
Floor                     ! area: 169.9688
.000000 .174201 .152542 .174201 .152542 .314427 .032086
S-Wall                    ! area: 111.0000
.266746 .000000 .154335 .157839 .154335 .241291 .025455
W-Wall                    ! area: 98.0000
.264566 .174808 .000000 .174808 .121254 .238852 .025713
N-Wall                    ! area: 111.0000
.266746 .157839 .154335 .000000 .154335 .242618 .024128
```

```
E-Wall                   ! area: 98.0000
.264566 .174808 .121254 .174808 .000000 .238823 .025742
SspCeil                  ! area: 153.9688
.347101 .173953 .152028 .174909 .152009 .000000 .000000
AcrLens                  ! area: 16.0000
.340854 .176594 .157495 .167387 .157671 .000000 .000000

control RoomQ   ps  RmAir  RoomQ   0.0 1.0 0.0  ! controls used to compute
control RoomCQ  llc RoomQ  RoomCQ  0.0          ! total cooling load
control reptCQ  sp  RoomCQ reptCQ  0.0 1.0      ! in Btu/h
control reptCE  sep RoomCQ reptCE  0.0 1.0      ! and W.

times  3  60  120  240

report stepitd.rpt  5  120
 n  RmAir    Q  i  %10.3f
 l  Luminaire  Q  i  %10.3f
 s  reptCQ  O  i  %10.3f
 s  reptCE  O  i  %10.3f
 n  PlAir   T  i  %8.3f

report stepitd.sum  4  3600
 n  RmAir    Q  s  %10.3f
 l  Luminaire  Q  s  %10.3f
 s  reptCQ  O  s  %10.3f
 s  reptCE  O  s  %10.3f

report stepitd.wls  6  900
 n  Floor-4  T  i  %8.3f
 n  UnderSide T i  %8.3f
 n  W-Wall    T  i  %8.3f
 n  W-Plenum T  i  %8.3f
 n  SspCeil   T  i  %8.3f
 n  PlenumSC T  i  %8.3f

display  6  3600
 n  RmAir    Q  i  %10.3f
 l  Luminaire  Q  i  %10.3f
 s  reptCE  O  i  %10.3f
 n  tubes    T  i  %8.3f
 n  LmAir    T  i  %8.3f
 n  PlAir    T  i  %8.3f

* end of data
```

Boundary values file:

```
0                       ! 0 indicates no boundary values
000/23:00:00            ! simulation starts one hour before start of day 1
002/00:00:00            ! simulation ends at start of day 2
*
```

Discrete events file:

```
000/23:00:00                   ! start matches BVF
001/00:00:00   flow 1.0        ! at midnight start the airflow
001/00:00:00   switch 1.0      ! and the luminaire
004/00:00:00                   ! end the DEF after the BVF
*
```

APPENDIX D

VLITE Theory

D.1 Introduction

Some of the following material is taken from prior reports by the author [Walton, 1986, 1987]. It is repeated here to provide a convenient reference on the methods used in VLITE.

Radiation interchange between diffusely reflecting and absorbing surfaces is important for problems involving either radiant heat transfer or distribution of light. Analysis of radiation interchange begins with the computation of view factors (also called 'shape factors', 'angle factors', and 'configuration factors') which are a function only of the geometry of the participating surfaces. The theory for computing view factors is covered in several textbooks [Hottel & Sarofim, 1967; Siegel & Howell, 1978; Sparrow & Cess, 1978]. Most of the theoretical methods have been implemented in numerical algorithms for computers. Several use the double area integration or related line integration techniques [Wong, 1976; Puccinelli, 1973; Shapiro, 1983; Emery, 1986]. Another method uses what is called the Nusselt sphere [Lipps, 1983]. A recent variation of this method [Cohen & Greenberg, 1985, 1986] is being used to determine the intensity of diffuse light at a point for accurately displaying scenes on a computer terminal. (The application of view factors to a computer graphics problem contrasts to the algorithms in this report which apply some computer graphics techniques to the calculation of view factors.) A Monte Carlo method [Burns, 1983] (essentially a form of ray tracing) may also be used to compute view factors, but such a method is very time consuming. It is useful for handling problems involving surfaces which are not perfectly diffuse.

Although the calculation of view factors occurs only once at the start of a simulation, the calculation can be very time consuming. The problem with view factors is not that they are very difficult to compute, but that the calculation time increases exponentially with the number of surfaces involved. Algorithms which are effective for a small number of surfaces may require a hopelessly long computation time for many surfaces. Consider a situation involving N surfaces. Since each surface may potentially interact with every other surface, there are N^2 interactions, or view factors. Even simplifications such as the reciprocity relation (section 2.1) and the fact that a flat surface cannot view itself, reduce the number of view factors only to N(N-1)/2, which is still of order $O(N^2)$. If it is known that some, but not which, surfaces can obstruct (or 'shade' or 'occlude') the views between surface pairs, it is necessary to check N-2 surfaces as possible obstructing surfaces for each view factor. This gives N(N-1)(N-2)/2 obstruction checks, that is, $O(N^3)$. In addition, the procedures for computing obstructed view factors tend to be much less efficient than those for unobstructed view factors.

This appendix will examine algorithms for computing view factors using improved line integral and computer graphics methods with special emphasis on the processing of view obstructions.

D.2 2-D Planar View Factors

The fundamental formula for a view factor between isothermal, black-body, diffusely emitting and reflecting surfaces, $F_{1,2}$, is

$$A_1 F_{1,2} = \int_{A_1}\int_{A_2} \frac{(dA_1 \cos\theta_1)(dA_2 \cos\theta_2)}{\pi r^4} \tag{D.1}$$

where A_1 and A_2 are the areas of surfaces 1 and 2, θ_1 and θ_2 are the angles between the normals to surface differential elements dA_1 and dA_2 and the vector between those elements, and r is the length of that vector. Hottel [1967] calls the term $A_1F_{1,2}$ the direct radiation interchange area. It is a convenient method of recording view factor information because of the reciprocity relationship, $A_1F_{1,2} = A_2F_{2,1}$, which can be used to reduce the number of view factors to be computed or stored.

Two-dimensional systems are those where all surfaces can be represented in a single plane, X-Y, with infinite extent in the Z direction. Such a geometry has practical use (e.g., to compute view factors in a fluorescent tube luminaire), a simple computational formula, and is easy to display.

D.2.1 Unobstructed Views; the Crossed-Strings Algorithm

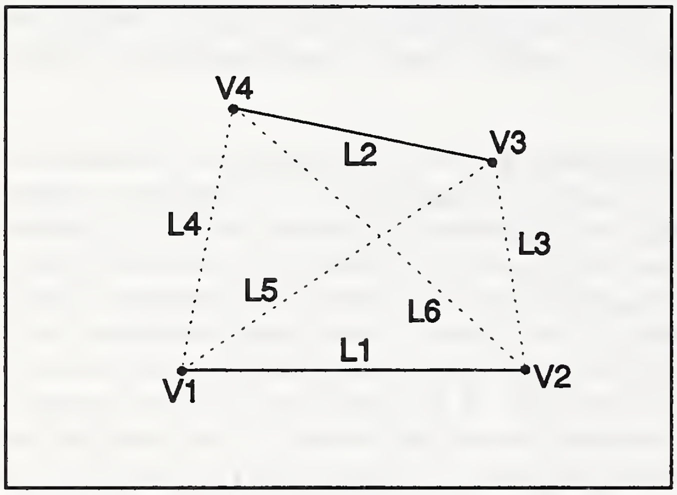


Figure D.1 2-D Surface Configuration

Consider the two surfaces labeled 1 and 2 in figure D.1 which extend infinitely in the direction normal to plane of the paper. In this 2-D representation these flat surfaces are represented by line segments. Their areas are proportional to the segment lengths.

The view factor is given by [Hottel, 1967, p 33]

$$F_{1,2} = \frac{(\text{sum of crossed strings}) - (\text{sum of uncrossed strings})}{2\,(\text{length of surface } 1)} \tag{D.2}$$

or

$$L1\, F_{1,2} = \frac{1}{2}\left[(L5 + L6) - (L3 + L4)\right] \tag{D.3}$$

where L1 = distance from V1 to V2, L3 = distance from V2 to V3,
L4 = distance from V1 to V4, L5 = distance from V1 to V3, and L6 = distance from V2 to V4.

Figure D.2 shows two boxes consisting of eight vertices and eight surfaces. If only the coordinates of the vertices were considered, a view factor between surfaces S1 and S3 could be computed with equation D.2 although they are on opposite sides of the same box and cannot view each other. On the other hand, surfaces S1 and S5 can view each other. Since it assumed that each line segment forms one surface of an opaque object, it is necessary to establish a convention relating the line to the object. It is assumed that the solid object lies to the "right" (as viewed from the first vertex) of the vector from the first to the second vertex. Only that portion of the surfaces which are in front of each other can see each other, i.e., have a non-zero view factor. For example, in figure D.2 surfaces S1 and S3 are behind each other, so $F_{1,3} = 0$. Surfaces S1 and S5 are totally in front of each other so equation D.2 can be used to compute the view factor. Surface S1 is partially behind surface S8. Therefore, only that portion of S1 in front of S8 (below the dashed line) is used in computing $F_{1,8}$. For this particular view factor it is as if surface S1 extended only from Vx to vertex V2. Methods for numerically determining the relationships between points and lines are given in section D.4.

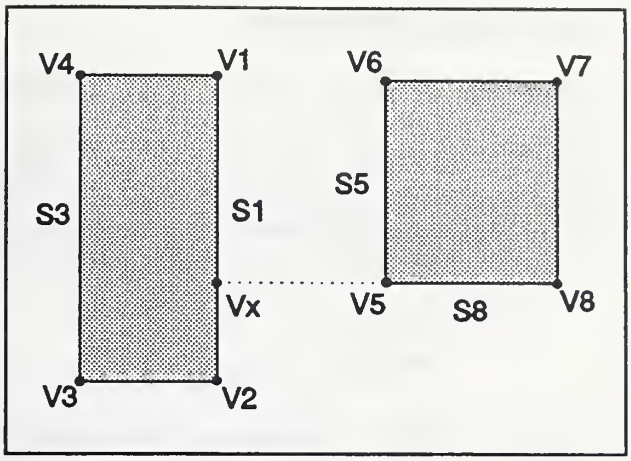


Figure D.2 2-D Surface Orientations

D.2.2 Obstructed Views

When more than two surfaces are involved, some surface(s) may fully or partially obstruct the view between some other pair of surfaces. Figure D.3 shows a simple case where portions of surfaces S3 and S4 partially obstruct the view between surfaces S1 and S2. As indicated in this figure, a surface will at least partially obstruct the view between two other surfaces if some of the surface is within the convex polygon defined by the two surfaces and the "uncrossed strings".

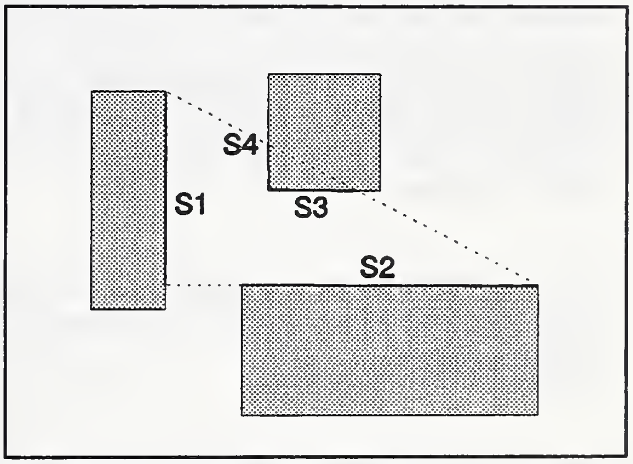


Figure D.3 2-D View Obstruction

The following algorithm is used to compute F_{12}:
(1) transform the coordinates so that S2 starts at the origin and follows the X-axis;
(2) divide S1 into many subsurfaces;
(3) from the center of each subsurface project a shadow of each obstructing surface onto the $Y=0$ line;
(4) merge the shadows to determine those portions of S2 which can be seen from each subsurface;
(5) compute the radiation interchange areas from the subsurface to the visible portions of S2;
(6) sum these interchange areas for all subsurfaces of S1.

D.2.3 Overall 2-D Algorithm

The following overall algorithm is used to compute 2-D view factors.

```
                    2-D View Factor Algorithm
    Compute homogeneous coordinates of all surfaces and vertices.
    Set up list of possible view obstructing surfaces.
    Process rows n from 1 through N.
        Process column elements m from 1 through N.
            If m < n, view already computed, so ─────────────────┐
            If m = n, view = 0 ──────────────────────┐           │
            Check for self-obstructed view; if so ───┤           │
            Eliminate possible obstructing surfaces. │           │
            Compute unobstructed view factor.        ↓           │
            If needed, compute obstructing effect.               │
            Set Am*Fmn and An*Fnm as computed.   Set to 0.       ↓
        Sum all An*Fnm in row n.
```

A list of potential obstructing surfaces is created so that surfaces which can never obstruct the view between any other pair of surfaces are not considered later in the algorithm when the obstructing surfaces between each pair of surfaces must be determined. Figure D.4 shows a 2-dimensional enclosure where surface 2, 3, 8, and 9 can obstruct views, but the other surfaces cannot. Surface n cannot be an obstructing surface if all other surfaces are on or in front of the plane of surface n.

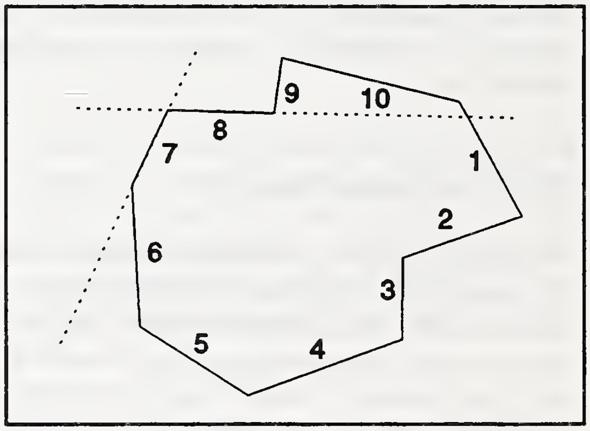


Figure D.4 Possible Obstructions

This is an $O(N^2)$ algorithm, although potential obstructing surfaces tend to be found quickly since no further tests need be made relative to remaining surfaces. This algorithm is very useful for surfaces forming an enclosure.

D.3 3-D Planar View Factors

D.3.1 Unobstructed Views

The development of algorithms for computing view factors in geometries with obstructing surfaces must begin with the implementation of fast and accurate calculation of unobstructed view factors. The description that follows is based on the review presented by Shapiro [1983, 1985] but the numerical implementations have been modified for greater speed and accuracy.

D.3.1.1 Double Area Integration

Figure D.5 shows the basic geometry and nomenclature involved in computing a radiation view factor. The fundamental formula for a view factor between isothermal, black-body, diffusely emitting and reflecting surfaces, F_{12}, is

$$A_1 F_{1,2} = \int_{A_1} \int_{A_2} \frac{(dA_1 \cos\theta_1)(dA_2 \cos\theta_2)}{\pi r^4} \tag{D.4}$$

where A_1 and A_2 are the areas of surfaces 1 and 2, θ_1 and θ_2 are the angles between the normals to surface differential elements dA_1 and dA_2 and the vector between those elements, and r is the length of that vector. Equation (D.4) can be integrated numerically by dividing both surfaces into finite subsurfaces as also shown in figure D.5. When each edge of a quadrilateral is divided into n elements, its surface is divided into n^2 subsurfaces. Therefore, equation (D.4) can be numerically approximated by

$$A_1 F_{1,2} = \frac{1}{\pi} \sum_{i=1}^{m^2} \sum_{j=1}^{n^2} \frac{-(r \cdot u_1)(r \cdot u_2) \Delta A_i \, \Delta A_j}{(r \cdot r)^2} \tag{D.5}$$

where m and n are the edge division factors for surfaces 1 and 2, $\mathbf{u}_1$ and $\mathbf{u}_2$ are the unit normal vectors of the surfaces, ΔA_i and ΔA_j are the areas of the subsurfaces, and $\mathbf{r}$ is the vector between the centroids of the subsurfaces.

D.3.1.2 Line Integral Method

The area integrals in equation (D.4) can be transformed into line integrals using Stoke's theorem and is expressed by

$$A_1 F_{1,2} = \oint_{C_1} \oint_{C_2} \ln r \, dl_1 \, dl_2 \tag{D.6}$$

where C_1 and C_2 are the boundary contours of the surfaces and r is distance between dl_1 and dl_2 which are vector elements on the two contours. Equation (D.6) can be evaluated numerically by dividing the edges of the quadrilaterals into elements v_i and v_j to give

$$A_1 F_{1,2} = \frac{1}{4\pi} \sum_{p=1}^{4} \sum_{q=1}^{4} \Phi_{pq} \sum_{i=1}^{m} \sum_{j=1}^{n} \ln r^2 v_i v_j \tag{D.7}$$

where v_i and v_j are the length of the elements, Φ_{pq} is the cosine of the angle between edges p and q (which is equal to the dot product of the unit vectors along p and q), and the term (ln r) is replaced by (ln r^2)/2 to avoid computing a square root since r^2 can be directly computed from the coordinates of the midpoints of v_i and v_j.

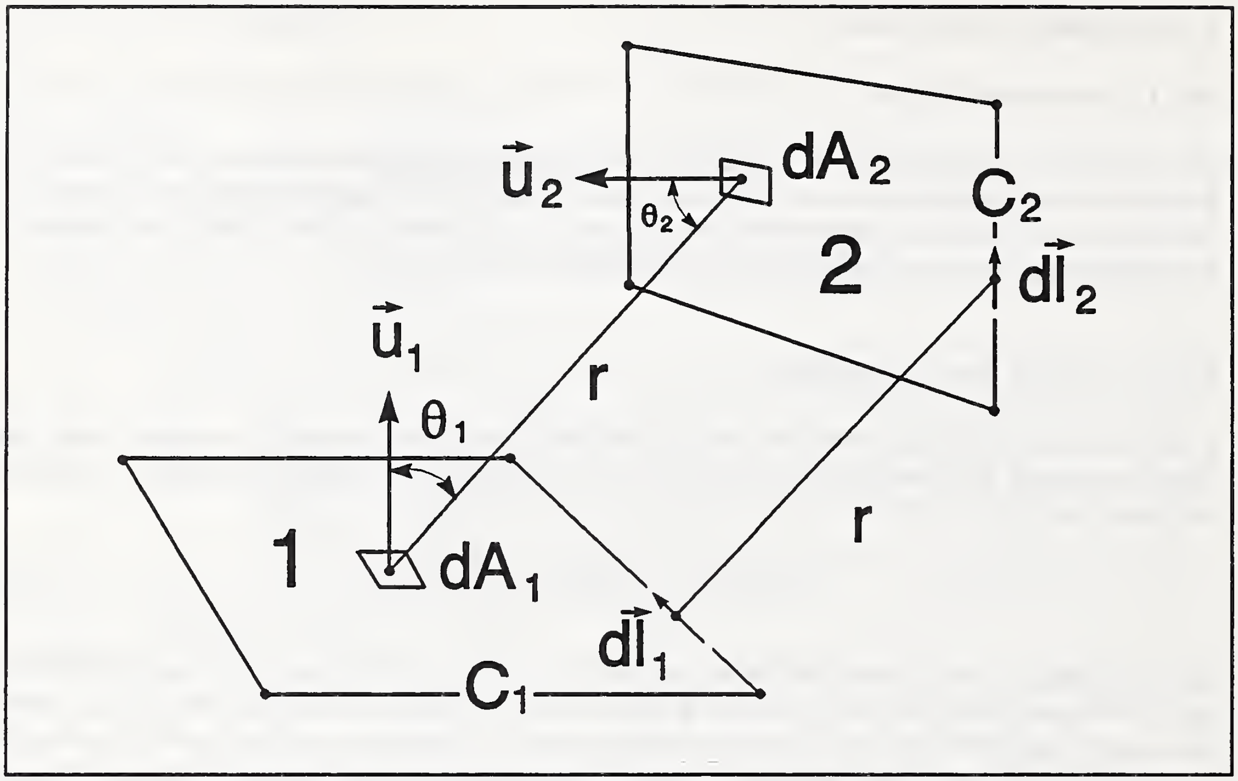


Figure D.5 Symbols for Area and Line Integral Methods

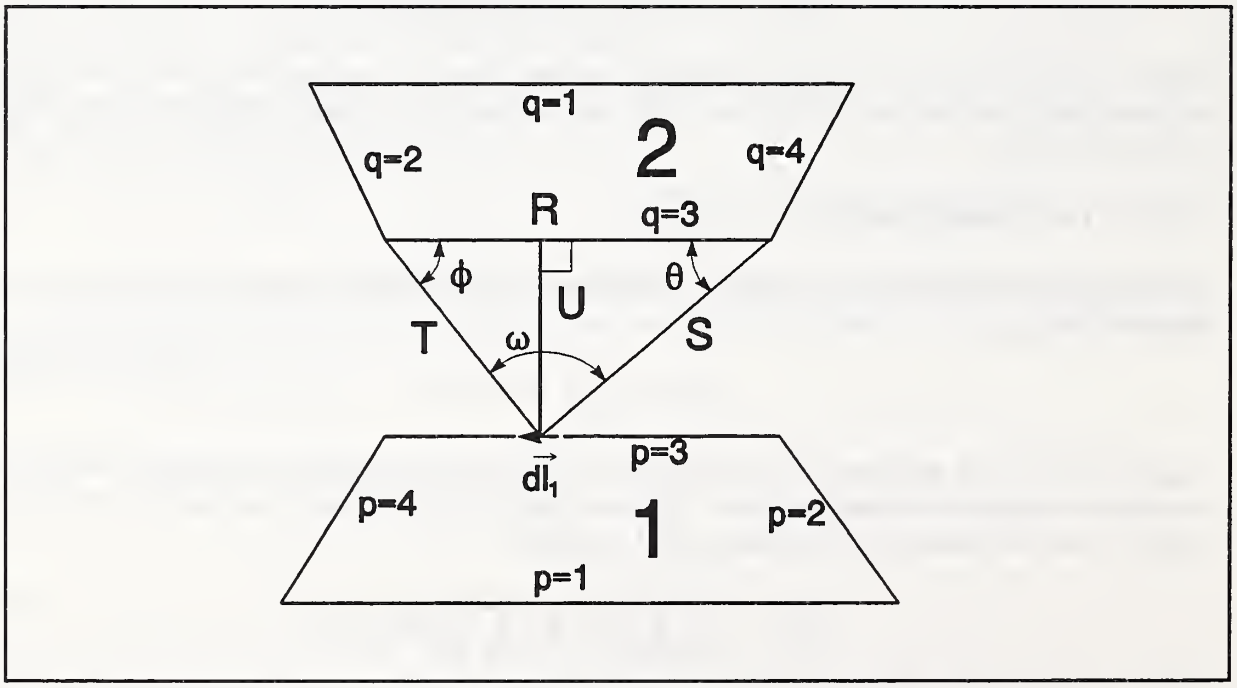


Figure D.6 Symbols for Mitalas and Stephenson Method

D.3.1.3 Mitalas and Stephenson Method

Mitalas and Stephenson [1966] present a method where one of the integrals
in equation (D.6) has been integrated analytically. The terms in the following equation are defined in figure D.5.

$$A_1F_{1,2} = \frac{1}{2\pi}\sum_{p=1}^{4}\sum_{q=1}^{4} \Phi_{pq} \oint_{C_1} (T\cos\phi \ln T + S\cos\theta \ln S + U\omega - R)\, dl_1 \qquad \text{(D.8)}$$

where S, T, U, ϕ, θ, and ω are functions of l, the location on p. Again, this is integrated numerically by dividing each edge into small elements, v_i, to give

$$A_1F_{1,2} = \frac{1}{2\pi}\sum_{p=1}^{4}\sum_{q=1}^{4} \Phi_{pq} \sum_{i=1}^{m} (T\cos\phi \ln T + S\cos\theta \ln S + U\omega - R)\, v_i \qquad \text{(D.9)}$$

Tests indicate that the highest accuracy is achieved by using the numerically evaluated integral on the polygon with the shortest edges.

D.3.1.4 Performance of the Basic Methods

The computational methods represented by equations (D.5), (D.7), and (D.9) will be referred to as the area integration method (AI), the line integration method (LI), and the Mitalas and Stephenson method (MS), respectively. Figure D.7 shows the measured execution times for computing view factors between two parallel squares for these three methods. (Note: the Cyber 855 computer on which these tests were run requires about 1.2 central processor (CPU) seconds to do 10^6 additions or multiplications and 10.0 CPU seconds for 10^6 logarithm calculations.) The edges of both squares are divided into n line elements for LI and MS, and the squares are divided into n^2 subsurfaces for AI. The important point to note is the shape of the curves which can also be determined by reviewing the equations. From the limits of the summation signs, it is obvious that AI is an $O(n^4)$ algorithm, LI is $O(n^2)$, and MS is $O(n)$. Other computers would give different execution times and possibly different values of n where one algorithm is faster than another, but the general shape of the curves will not change. Shapiro [1983] shows how a vectorizing computer influences performance.

The accuracy of the computer implementations can be tested against analytic solutions for simple configurations. Figure D.7 shows the accuracy as a function of n for two parallel, directly opposed, unit squares with unit separation. In this case the LI method is about four times as accurate as the AI method. The MS method is about twice as accurate as LI because the line integral is solved analytically on half of the edges. The LIg and MSg methods are discussed in the next section. Figure D.8 shows the accuracy of the view factor methods for two adjacent perpendicular squares. The AI method is very inaccurate in this case. The LI method cannot be used because the (ln r) term is undefined at $r = 0$, which occurs when the integration line elements coincide on the common edge.

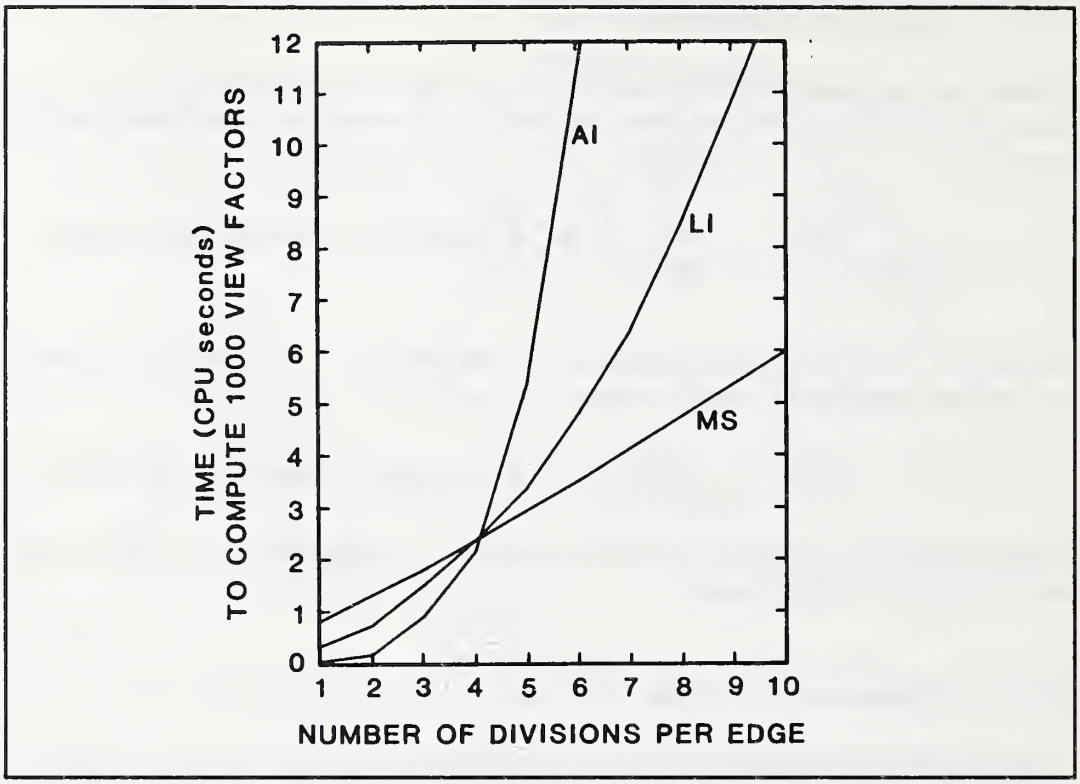


Figure D.7 Computer Execution Times for the Three Basic Algorithms

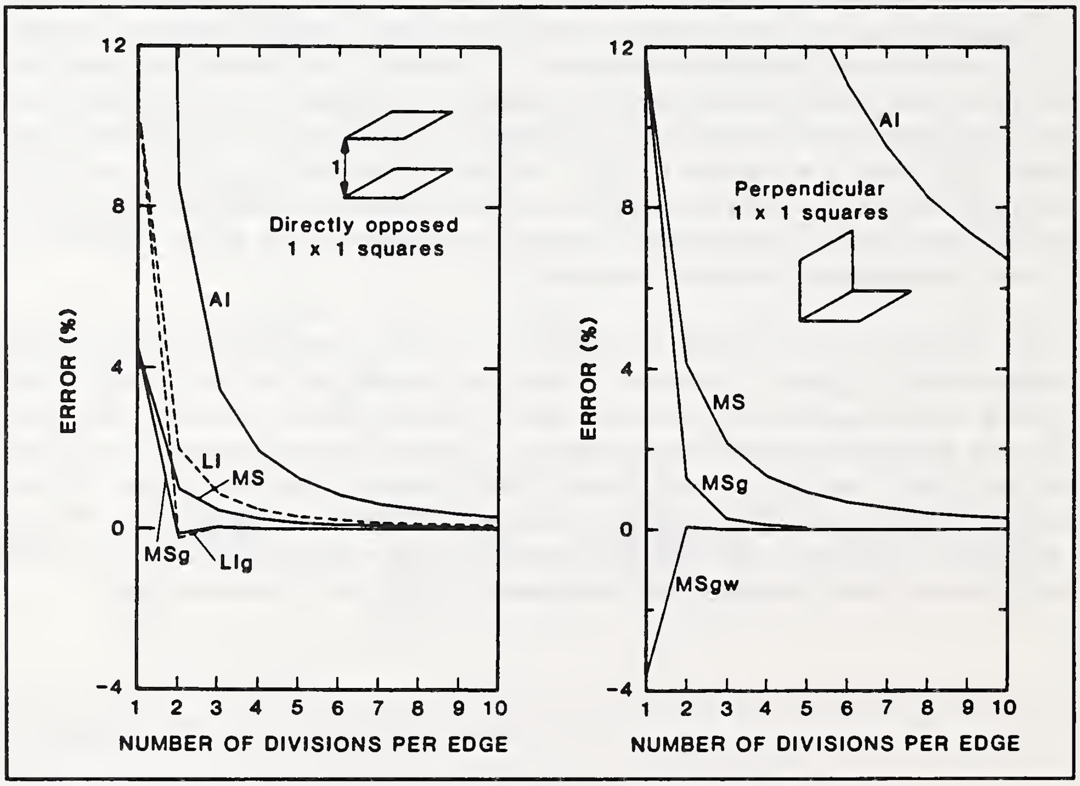
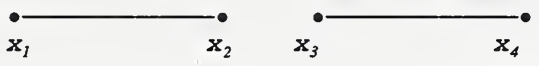


Figure D.8 Algorithm Accuracy for Two Basic Configurations

D.3.1.5 Algorithm Improvements

A significant improvement in the accuracy of the LI and MS algorithms can be made by using Gaussian integration which divides each edge into elements of different lengths. The lengths of the elements and the point within each element where the function is evaluated are specified as part of the Gaussian integration method [Conte & De Boor, 1972; Abramowitz & Stegun, 1964]. The results of this modification are shown by the lines labeled LIg and MSg in figures D.7 and D.8. Note the very rapid convergence for the case of parallel squares: the error is less than 10^{-6} for 4 or more edge divisions. Execution times are not changed.

The MSg method is considerably less accurate for the case of adjacent surfaces than it is for parallel surfaces. Most of this error occurs during integration along the common edge, but that error can be removed by solving the integration for colinear edges exactly. The sketch below shows two colinear lines whose ends are defined by one-dimensional coordinates. In this case the line integral in equation (D.9) can be solved exactly to give expression (D.10).



$a^2 = (x_2-x_1)^2 \qquad b^2 = (x_4-x_3)^2 \qquad c^2 = (x_3-x_1)^2$

$d^2 = (x_4-x_1)^2 \qquad e^2 = (x_3-x_2)^2 \qquad f^2 = (x_4-x_2)^2$

$$[(d^2\ln d^2-d^2+c^2-c^2\ln c^2+e^2\ln e^2-e^2+f^2-f^2\ln f^2)/4-ab] \qquad \text{(D.10)}$$

Two edges are colinear when $\Phi_{pq} = \pm 1$ and U = 0. When the ends of the two edges match exactly, the edges are coincident as well as colinear, so expression (D.10) reduces to

$$[(R^2\ln R^2-R^2)/2-R^2] \qquad \text{(D.11)}$$

where R is the length of the edge. This modification of the MSg method is labeled MSgw in figure D.8. The MSgw method is about twice as accurate for the case of adjacent squares as MSg is for parallel squares. At n = 2 the errors are only -.12% and -.22%, respectively. Execution time for MSgw is not much different from MS because time lost testing for colinear edges is recovered by the simple line integration.

Lipps [1983] mentions the use of Gaussian integration for the MS method in his paper describing the computer implementation of a Nusselt sphere method. He found that the MSg method is an order of magnitude more efficient for most structures in spite of poor convergence for surfaces having a common edge. Expressions (D.10) and (D.11) remove that convergence problem to provide an even greater advantage for the MSgw method over Lipps' implementation of the Nusselt sphere method.

Figures D.9 and D.10 show the accuracy of the different methods for parallel and perpendicular unit squares, respectively, at different separation distances. Both figures show that as the separation distance increases, either fewer edge divisions or a simpler method may be used to obtain a given level of accuracy.

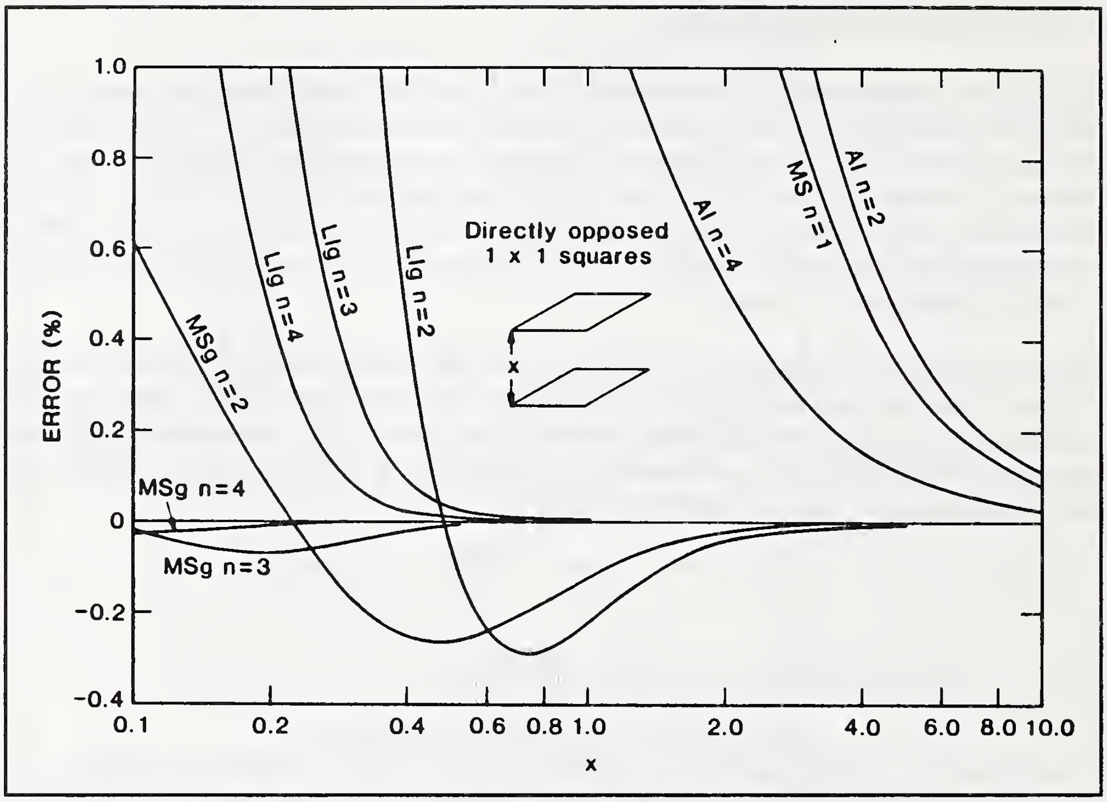


Figure D.9 Accuracy of Algorithms for Parallel Squares

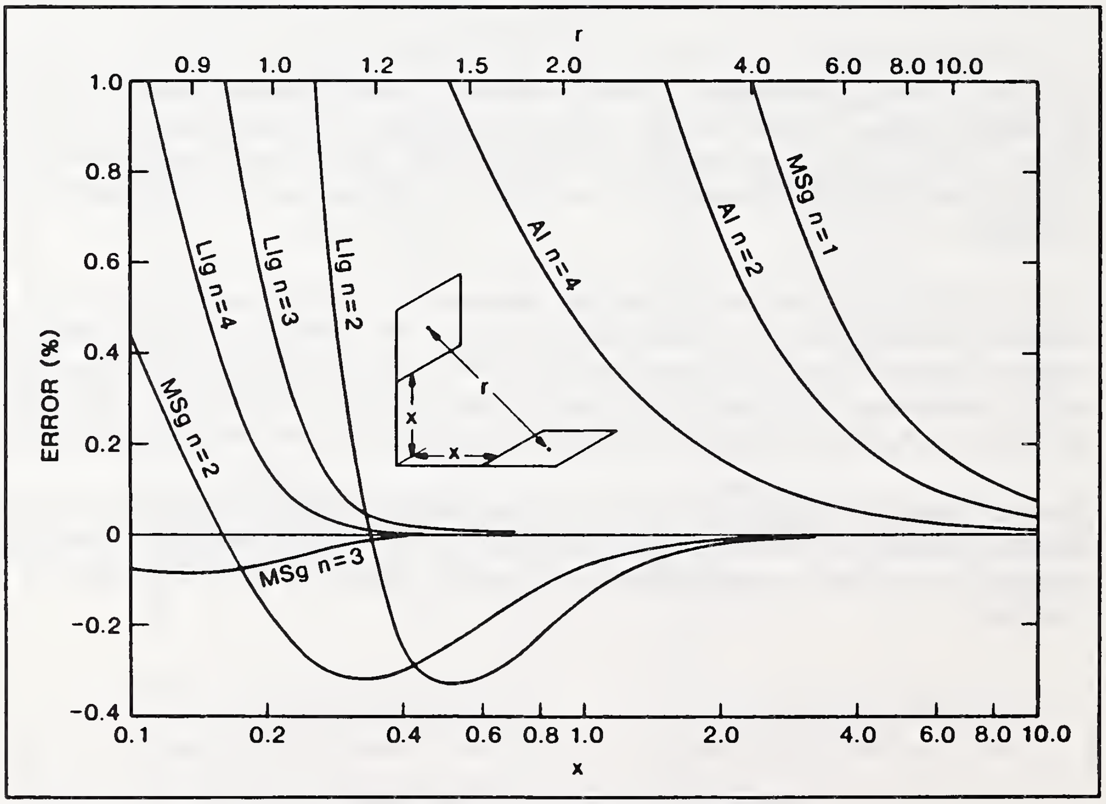


Figure D.10 Accuracy of Algorithms for Perpendicular Squares

D.3.2 Obstructed Views

Most view factor programs [Wong, 1976; Puccinelli, 1973; Shapiro, 1983; Emery, 1986] evaluate obstructed view factors by utilizing an area integration method in spite of its execution time and accuracy problems. The algorithm described below retains line integral methods for all obstructed view calculations. The methods are limited to convex polygons, i.e., those having no interior angle greater than 180°.

Figure D.11 shows a simple obstructed view configuration consisting of two opposed unit squares at unit separation with a smaller (.5 by .5) square located between them at a point 3/4 of the distance from surface 1 to surface 2. The value of $F_{1,2}$ is determined analytically to be 0.115621 .

$A_1F_{1,2}$ can be determined numerically by a form of double area integration. Consider surfaces 1 and 2 to be divided into n^2 subsurfaces which are represented by their centroids and labeled i and j, respectively. It is necessary to determine which of the views between subsurfaces i and j are blocked by the obstructing surface. This can be done in two ways. (1) The point where the line from subsurface i to subsurface j intersects with the plane of the obstructing surface is computed. Then it must be determined if the intersection point lies on the obstructing surface. This can be done with vector cross products for a convex obstructing surface. (2) Equivalently, the vertices of the obstructing surface can be projected from subsurface i onto surface 2 as a 'shadow'. A homogeneous coordinates procedure (Section D.4) can be used to determine which points j are inside the shadow. Note that both procedures require $O(n^4)$ calculations. Multiple obstructing surfaces are handled by checking all remaining unblocked views against each obstruction. When a view line intersects the edge of an obstruction, special handling may be desirable.

Once it is known which subsurface pairs have obstructed views, there are three methods that can be used to compute the total radiation interchange area. (a) The simplest method is to sum the interchange areas for each unobstructed view to determine the total interchange area using equation (D.2). (b) If the unobstructed interchange area is known (by an accurate line integral calculation), the total interchange area of each obstructed subsurface to subsurface view can be subtracted from it to determine the total interchange area. (c) The unobstructed interchange area can be used to correct for the inaccuracy in the AI calculation by summing the interchange areas for all views and scaling the sum of the unobstructed views by the sum of all views divided by the unobstructed interchange area.

A very different method can be developed using only line integral methods and shadow projections. The interchange area between subsurface i and the shadow of the obstruction can be computed by the LI or MS algorithms. Subtracting these interchange areas for shadows projected from all subsurfaces of surface 1 from the unobstructed interchange area gives the total interchange area between surfaces 1 and 2. Section D.4 describes a method for processing overlapping shadows from multiple obstructing surfaces. Note that it is necessary to divide only one of the surfaces into subsurfaces which implies an $O(n^2)$ algorithm. However, the calculations are more complex than for the AI methods.

Figure D.12 shows the results of three different calculations for the simple obstructed view in figure D.11. Method A, indicated by small squares, uses projection (2 above) and adjusting (c) to perform the double area integration. Method B, shown by small triangles, uses the line integral method with the shadow of surface 3 projected toward surface 1. Method C is like B, except the shadow is projected toward surface 2. Note that the direction of projection is very important in this case. Method C converges uniformly to the correct solution and is relatively accurate even for a small number of subsurfaces. Projection in the wrong direction causes an error similar to that of the area integration

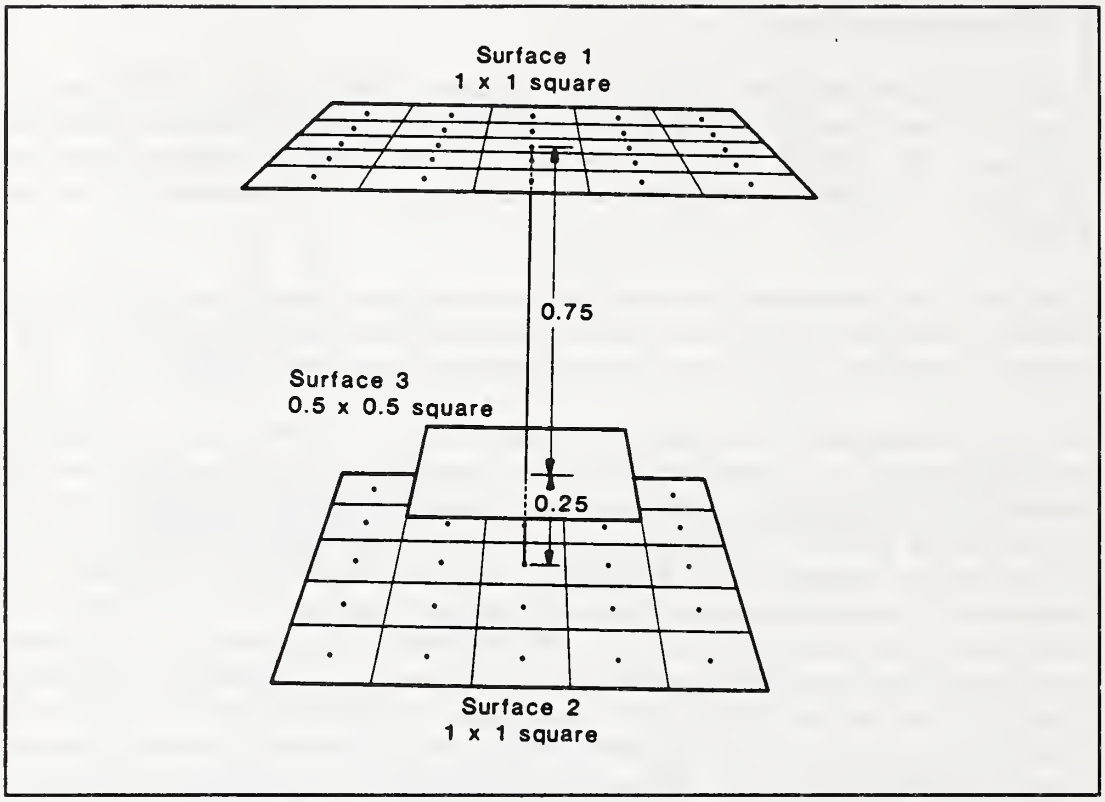


Figure D.11 A Simple Obstructed 3-D View Configuration

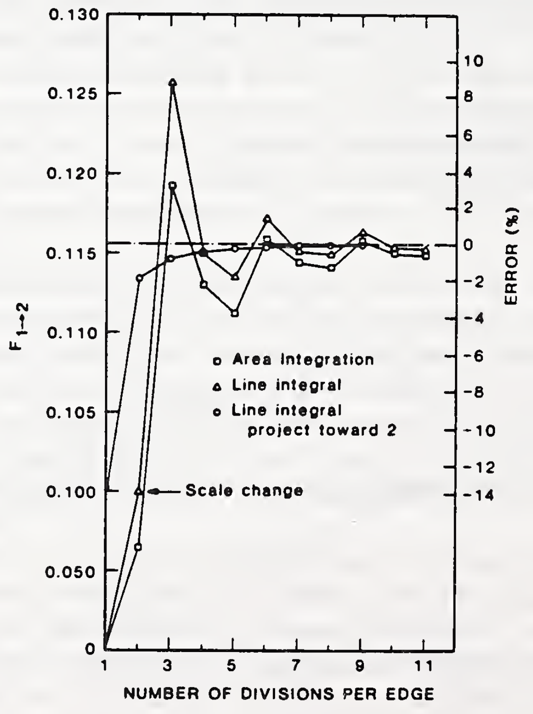


Figure D.12 Accuracy of Algorithms for Simple Obstruction

method, and that error does not reduce uniformly as the number of subsurfaces increases.

D.3.3 Overall 3-D Algorithm

The following algorithm (which is the same as the 2-D algorithm) will compute the view factors between plane, convex polygons. Non-planar surfaces must be approximated by multiple small plane polygons. A planar surface could be viewed from either side, but there is a considerable simplification of algorithm logic by considering a surface to have only one side, the active side or front, which interacts with other surfaces. Surfaces are described by the coordinates of their vertices listed in counter-clockwise sequence as seen from the active side.

```
3-D View Factor Algorithm

    Initializations.
    Set up list of possible view obstructing surfaces.

    Process rows n from 1 through N.
        Process column elements m from 1 through N.
            If m < n, view already computed, so ---------------------+
            If m = n, view = 0  ----------------+                    |
            Check for self-obstructed view; if so --+                |
            Eliminate possible obstructing surfaces.|                |
            Compute unobstructed view factor.       v                |
            If needed, compute obstructing effect.                   |
            Set Am*Fmn and An*Fnm as computed.    Set to 0.          v
        Sum all An*Fnm in row n.
```

Much of the 3-D algorithm is devoted to the efficient detection of surfaces which can be view obstructors.

D.3.3.1 Initializations

Initial values are set for some algorithm performance values and graphic scaling factors. The following initializations are made if view factors are not being recalculated. Values are computed for each surface: its area, the components of its unit normal vector, the coordinates of its centroid, and the minimum radius about the centroid which encloses the surface which is the maximum distance from the centroid to any vertex. In addition, quadrilateral surfaces are tested to insure that all four vertices lie in the same plane.

A list of potential obstructing surfaces is created so that surfaces which can never obstruct the view between any other pair of surfaces are not considered later in the algorithm when the obstructing surfaces between each pair of surfaces must be determined. This is similar to the 2-D algorithm. Surface n cannot be an obstructing surface if all other surfaces are on or in front of the plane of surface n. This is determined by evaluating the dot product of the unit normal to surface n with the vector from the centroid of n to every vertex of every surface m. If the dot product is negative, then the vertex lies behind the plane of surface n, and so surface m is a potential view obstructor. This algorithm is especially useful for surfaces forming the inside of an enclosure. In some computer programs [Shapiro, 1983; Emery, 1986] such a list of potential obstructing surfaces is entered by the user based on his knowledge of the geometry of the problem.

D.3.3.2 Self-obstruction tests

The view between a pair of surfaces, n and m, may be obstructed by one of the surfaces. This is called 'self obstruction'. Figure D.13 shows four surfaces: B is entirely in front of A, D is entirely behind A, and C is partly in front and partly behind. These conditions can be determined by the dot product test mentioned in the previous paragraph. If the dot products of the surface n unit normal with all the vectors from the centroid of n to the vertices of m are zero or negative, then surface m is entirely behind n and the view is completely self-obstructed. If m is not entirely behind n, then it is also necessary to check if n is behind m. This test very quickly determines that there is no view between most of the surfaces which form the outside of an object.

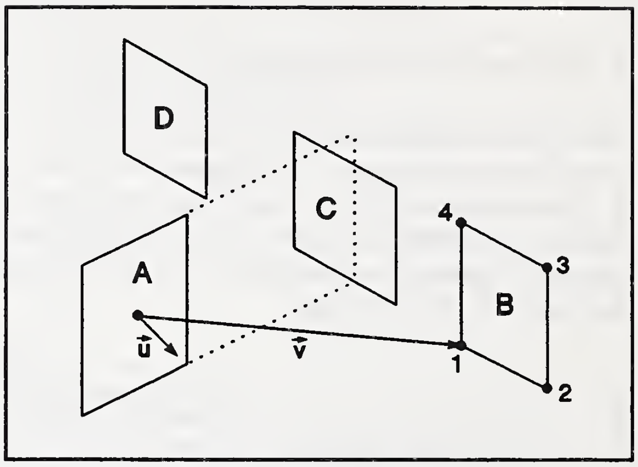


Figure D.13 Self-Obstruction Tests

It is possible that a surface is only partially behind the plane of the other surface, in which case it is necessary to remove, or 'clip', the portion of the one surface which lies behind the other. The values of the dot products can be used in this polygon clipping process. Note that clipping can increase the number of vertices by one, so sufficient storage for an added vertex must be available.

D.3.3.3 Eliminating Potential Obstructions

After a pair of surfaces pass the self-obstruction test, it is necessary to determine if any other surfaces obstruct the view. This involves a series of tests. It is important that these tests be arranged to eliminate non-obstructing surfaces as quickly and easily as possible because of the potential $O(N^3)$ execution time penalty.

One simple test is based on the surface radius values computed during initialization. That radius, R_k, and the coordinates of the centroid, (x_k, y_k, z_k), determine a sphere which completely surrounds the surface k. The radius, R_C, of a cylinder centered on the line between the centroids of surfaces n and m which must enclose both surfaces is also known: R_C is the larger of R_n and R_m. The minimum distance, D, from the centroid of k to the line connecting the centroids of n and m is given by the cross product of the vector from n to k with the unit vector from n to m: $D = | V_{nk} \times U_c |$. If $D^2 > (R_C + R_k)^2$, k cannot obstruct the view because the sphere containing k lies outside the cylinder. This test is referred to as the 'cylinder radius test'. It is most useful when there are many similar size surfaces as will often occur in the critical large N case. When surfaces n and m have significantly different radii, it is advantageous to use a slightly more complex test based upon a cone rather than a cylinder.

Another simple test involves determining the minimal box containing both surfaces n and m. If surface k is entirely outside that box, it cannot be an obstructing surface.

There are several surface orientation relationships where surface k cannot obstruct the view from surface n to surface m: (1) k entirely behind m (m cannot see k), (2) k entirely behind n (n cannot see k), (3) m and n entirely in front of k, and (4) m and n entirely behind k. These four relationships can be determined by the dot product tests. In the first two it is only necessary to find the first vertex of a surface which fails a test to determine that the relationship is not satisfied. These tests also eliminate surfaces n and m as potential view obstructors. The last two tests check all vertices in order to determine whether k can see m, k can see n, or k can see both. These relationships are used in the automatic determination of direction of projection described below. If it were known that the direction of projection were from m to n, or if the direction does not matter as when using the area integration method, the following two tests could also be used to eliminate potential shadowing surfaces: (5) m entirely in front of k, and (6) n entirely behind k (k cannot see m).

The final obstruction test determines whether the shadow of the obstructing surfaces will be projected from n or from m. This test is based on the idea that more accurate view factors are usually computed by projecting the obstruction toward the nearest surface. The distances between the center of n and the center of k are computed for each k which can see n. The minimum and average distances are determined. Similar distances are determined relative to m. Shadows will be projected toward the surface with the smallest combined average and minimum distance. This algorithm may not be optimal for all configurations.

D.3.3.4 Computing the Unobstructed View Factor

The unobstructed radiation interchange area between surfaces n and m is computed using the LIg or MSgw algorithms. The number of edge divisions, ND, is allowed to vary in order to reduce execution time. The value of ND is based on the relative distance, T, between the surfaces. T is equal to $S/(R_n + R_m)$ where S is the distance between the centroids of n and m. ND is smaller at larger T. The MSgw method is always used for $T < 1$ because it allows the possibility of a colinear edge. Also, when using the MSgw method, the larger of the two surfaces is chosen to have the analytic edges.

D.3.3.5 Computing the Obstructing Effect

The effect of obstructions is handled by the shadow projection and line integration method described in section D.3.1. It begins with a coordinate transformation of surfaces n, m, and the probable obstructing surfaces such that the surface toward which the shadows will be projected, n2, lies in the $Z = 0$ plane with its centroid at the origin. The surface from which shadows are projected, n1, lies entirely above the $Z = 0$ plane. Any portions of obstructing surfaces which lie below the $Z = 0$ plane are removed by clipping to prevent projection of false shadows. Surface n1 is then divided into subsurfaces. Again, the number of divisions is variable to improve execution speed. It is based on the unobstructed interchange area and a user input parameter such that higher values of the interchange area and the parameter give more subsurfaces. Shadows of all obstructing surfaces are projected from each subsurface onto the $Z = 0$ plane. If necessary, the obstructing surface is clipped to prevent an upward projection of its shadow. Each shadow is reduced to that portion which overlaps n2 and it is then combined with previous shadows to eliminate overlaps with them. The radiation interchange area between the subsurface and each shadow polygon is computed by LIg or MSgw and subtracted from the unobstructed interchange area. This is done for shadow projections from all subsurfaces of n1.

D.4 Numerical Processing of Convex Polygons

The processing of geometric data in both the 2D and 3D view factor algorithms is based on conventional vector calculations which are familiar to most engineers and on homogeneous coordinate (HC) techniques which have found extensive application in computer graphics but are unfamiliar to most engineers. This section reviews the fundamental properties of two-dimensional HC and describes a method for processing overlapping convex polygons.

D.4.1 Homogeneous Coordinates

HC are described in most textbooks about the mathematics of computer graphics, such as [Newman & Sproull, 1973; Pavlidis, 1982]. Points and lines in HC are represented by a single form which allows simple vector operations between those forms. A point (X, Y) is represented by a three element vector (x, y, w) where x = wX, y = wY, and w is any real number except zero. A line is also represented by a three element vector (a, b, c). The directed line (a, b, c) from point (x_1, y_1, w_1) to point (x_2, y_2, w_2) is given by

$$\begin{aligned}(a,b,c) &= (x_1,y_1,w_1) \times (x_2,y_2,w_2) \\ &= (y_1w_2-y_2w_1, w_1x_2-w_2x_1, x_1y_2-x_2y_1)\end{aligned} \qquad \text{(D.12)}$$

The sequence in the cross product is a convention to determine sign.
The condition that a point (x, y, w) lie on a line (a, b, c) is that

$$(a,b,c)\cdot(x,y,w) \equiv ax+by+cw = 0 \qquad \text{(D.13)}$$

Normalize a point by dividing its coordinates by w. Then if

$$(a,b,c)\cdot(\frac{x}{w},\frac{y}{w},1) > 0 \qquad \text{(D.14)}$$

the point lies to the left of the line, and if it is less than zero, the point lies to the right of the line (as viewed from the starting point of the line). The intercept (x, y, w) of line (a_1, b_1, c_1) and line (a_2, b_2, c_2) is given by

$$(x,y,w) = (a_1,b_1,c_1)\times(a_2,b_2,c_2) \qquad \text{(D.15)}$$

Note that the use of HC as outlined above provides a consistent method and notation for defining points and lines, for determining intercepts, and for determining whether a point lies to the left, to the right, or on a line. Normalization provides a means of transforming between HC and Cartesian coordinates. Thus, if (X, Y) is the Cartesian coordinate pair of a point, its HC description is (X, Y, 1). Similarly, the HC point (x, y, w) can be transformed to the Cartesian point (x/w, y/w).

The area, A, of a plane polygon consisting of n sequential vertices (X_1, Y_1), (X_2, Y_2), ..., (X_n, Y_n) is given by [Selby, 1974]

$$A = (x_1y_2+x_2y_3+\ldots+x_{n-1}y_n+x_ny_1-y_1x_2-y_2x_3-\ldots-y_{n-1}x_n-y_nx_1)/2 \qquad \text{(D.16)}$$

If the HC of the vertices have all been computed with the same value of w, then area is expressed more simply in terms of the c coordinates of the edges as

$$A = \frac{1}{2w^2}\sum_{i=1}^{n} c_i \qquad \text{(D.17)}$$

The area is positive if the vertices are in counter-clockwise sequence and negative if clockwise.

A problem may occur during computer implementation because of the critical importance of the value zero in determining the relationship of a point to a line. Computer arithmetic with 'real' numbers is subject to round-off error, so the tests against zero in equations (D.13) and (D.14) may fail. To prevent this the tests are done against some small number, typically 10^{-5} times the area of the surfaces involved in the calculation.

D.4.2 Intersection of Convex Polygons

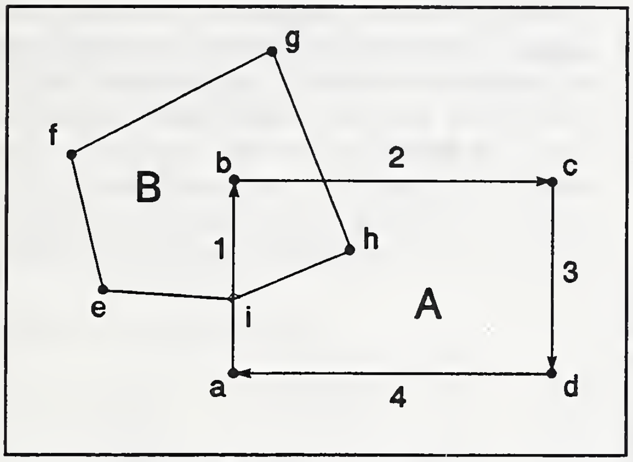


Figure D.14 Overlapping Polygons

The following algorithm has been developed to determine the intersection (or overlap) of two convex polygons. Consider polygons A and B in figure D.14. Polygon A consists of vertices a-b-c-d in clockwise sequence and the directed edges 1-2-3-4. Polygon B consists of vertices e-f-g-h-i, also in clockwise sequence. The first step in determining the overlap of A and B involves testing all vertices of B against edge 1 of A as shown in figure D.15.

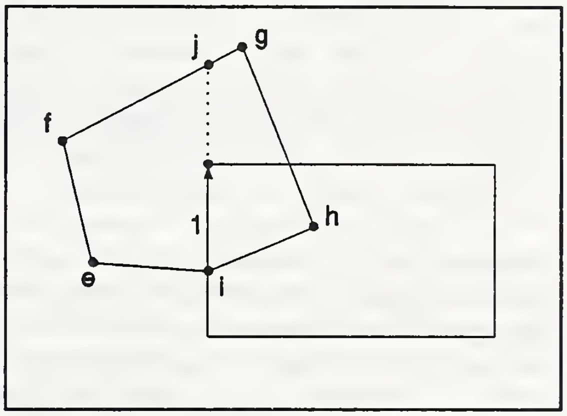


Figure D.15 First Edge Cut

Vertices e and f are to the left of edge 1; vertex g is to the right of edge 1. Since f and g are on opposite sides of edge 1, edge 1 must intercept edge f-g. That intercept can be computed by equation (D.15) and will be labeled j. Vertex h is also to the right of edge 1, and vertex i lies on edge 1. The portion of polygon B which lies to the left of edge 1 (vertices e-f-j-i) cannot overlap polygon A. The portion to the right of edge 1 (vertices j-g-h-i) may overlap depending on the results of tests against the remaining edges. Note that both portions of B must also be convex polygons.

Figure D.16 shows the situation for the reduced part of polygon B which must now be tested against edge 2 of polygon A. Vertices j and g are to the left of edge 2; vertex h is to the right of edge 2; so another intercept is computed and labeled k. Vertex i is right of edge 2. Since the next vertex, j, is left of the edge, another intercept (labeled l) is computed. The polygon to the left of edge 2 (vertices l-j-g-k) cannot overlap polygon A. The polygon to the right of edge 2 (vertices k-h-i-l) may overlap Polygon A.

Tests against edges 3 and 4 in figure D.17 show that vertices k-h-i-l are all to the right of both edges, so polygon k-h-i-l is the intersection of polygons A and B and is labeled polygon C.

If at any time during the tests against an edge of polygon A, it is found that all vertices of B are left of the edge, then A and B do not overlap. If all vertices of B are on or to the right of all edges of A, then B is within A. If the area of C is equal to the area of A, then A is within B. Note that it is necessary to allow storage for an indefinite number of vertices in polygon C. This number is usually small because C must be convex. Also note that it is critical to maintain the order of the vertices as the new polygons

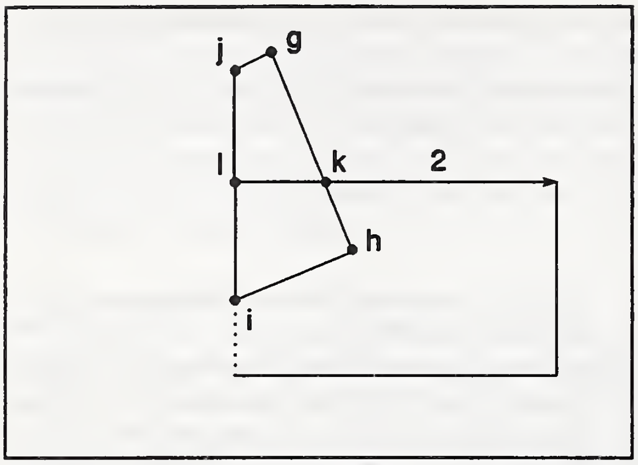


Figure D.16 Second Edge Cut

are outside C (which is called B'). B' consists of polygon e-f-j-i, which is the part of B to the left of side 1, plus polygon l-j-g-k, which is to the left of side 2. So, in this case, the union of A and B can be defined by three convex polygons. If A and B do not overlap, the union consists simply of polygons A and B. If A is within B, or B is within A, the union is simply the larger polygon.

This technique for processing polygons can be used in assessing solar heat gains in a building. Shadows may sometimes involve partially transparent surfaces. The fraction of light shining through a surface is its transparency. When two shadows overlap, the effective transparency of the overlap is the product of the transparencies of the two shadow casting surfaces. When determining the union of polygons representing partially transparent shadows, it is necessary to retain information on the transparency of each overlap or outside polygon as it is computed. For example, if A and B in figure D.14 have transparencies T_A and T_B, the union of A and B would consist of polygon C with transparency T_AT_B plus the two convex polygons representing B' with transparency T_B plus two convex polygons representing A' with transparency T_A.

are created. Vertices are stored in a linked list data structure. The small number of vertices in each polygon limits the usefulness of vectorization of these calculations on high-performance computers.

D.4.3 Union of Convex Polygons

In addition to computing the intersection of two polygons, shadowing calculations commonly need to determine the union (or combined area) of two overlapping polygons. Both processes can be done with essentially the same algorithm. The union of polygons A and B in figure D.14 consists of all of A plus the portion of B which

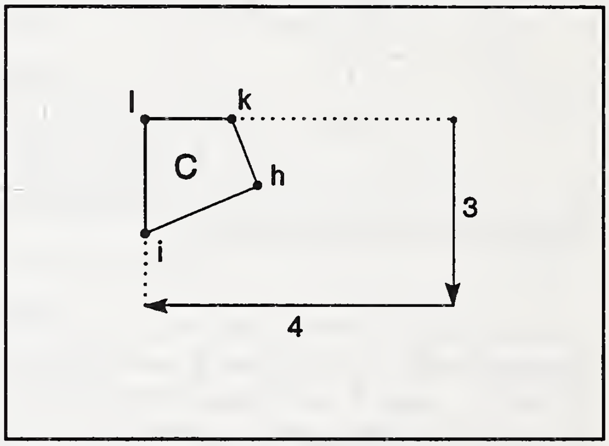


Figure D.17 Check Edges 3 and 4

The polygon union process tends to create more polygons than the number of polygons being joined, at least for a relatively small number of polygons. As more polygons are joined to the union of all previous polygons, it becomes more likely that they are totally within that union and so are less likely to add additional polygons. Computation time is strongly influenced by the number of overlapping polygons. An alternative algorithm which processes non-convex polygons using HC and an interesting edge classification method [Putnam & Subrahmanyam, 1986] was developed and tested. It avoids the increasing number of polygons during the union process, but was sufficiently more complex to require more execution time except for cases involving a large number of initial polygons. The algorithm for convex polygons is generally faster for the number of surfaces typically defined by users in the present generation of building energy analysis programs. In such cases, on conventional computers, it is quite fast. The primary advantages of the method are accuracy and generality.

D.5 Post-Processing Operations

The algorithms described in the previous sections require convex polygons. When it is necessary to describe a non-convex surface, VLITE provides two alternatives based on view factor algebra.

D.5.1 Combined Surfaces

It is possible to define a complex surface in terms of simple convex polygons as shown in figure D.18. Surface 2 and 3 combine to form a single surface 2*. In this case

$F_{1,2*} = F_{1,2} + F_{1,3}$ (D.18)

Combining surfaces is very easy.

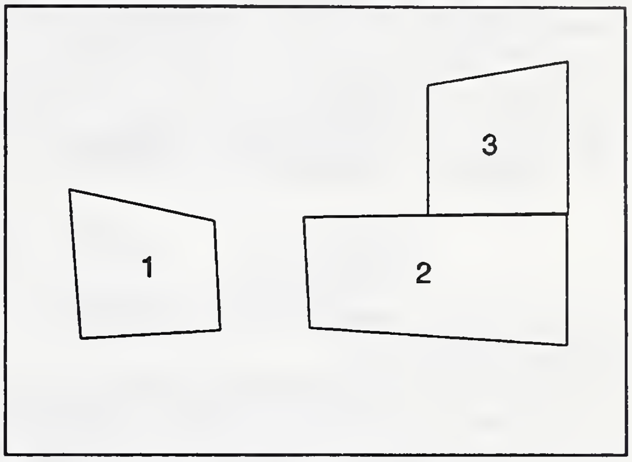


Figure D.18 Combined Surfaces

D.5.2 Included Surfaces

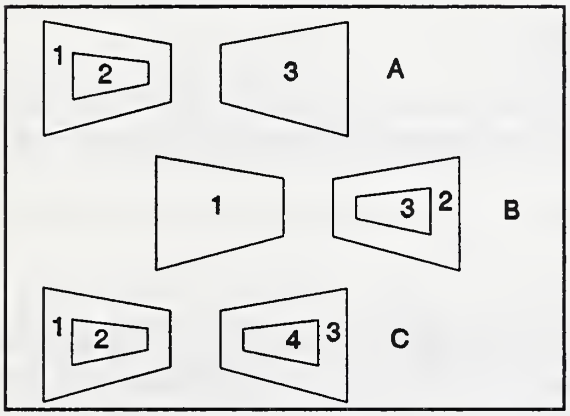


Figure D.19 Included Surfaces

A common non-convex surface in buildings is a wall with windows or a door in it. It is possible to describe the wall as a combination of convex surfaces, but it is easier to describe the wall as a single rectangle with included subsurfaces. Figure D.19 shows three subsurface configurations.

In case A surface 1* consists of surface 1 combined with subsurface 2. Interchange factors $A_{1*}F_{1*,3}$ and $A_2F_{2,3}$ are known. Then $A_1F_{1,3} = A_{1*}F_{1*,3} - A_2F_{2,3}$ (D.19)

Case B is the reverse of case A giving $A_1F_{1,2} = A_1F_{1,2*} - A_1F_{1,3}$ (D.20)

Case C is a combination of cases A and B. The values of $A_{1*}F_{1*,3*}$, $A_{1*}F_{1*,4}$, and $A_2F_{2,4}$ are known.

from A: $A_1F_{1,4} = A_{1*}F_{1*,4} - A_2F_{2,4}$ (D.21a)

from B: $A_2F_{2,3} = A_2F_{2,3*} - A_2F_{2,4}$ (D.21b)

finally: $A_1F_{1,3} = A_{1*}F_{1*,3*} - A_2F_{2,3} - A_1F_{1,4}$ (D.21c)

Tests indicate that multiple subsurfaces can be handled by first doing all the type A reductions, and then doing all the type B reductions, This requires that the subsurfaces be listed after the base surfaces. When both included surfaces and combined surfaces exist, the included surfaces are processed first.

D.5.3 Normalization

The numerical integration techniques presented above are fairly accurate, but they are not exact. When the surfaces form an enclosure, the sum of the view factors from any surface should equal one. If it does not, some energy will be either spuriously lost or created during the heat transfer calculations. The reciprocity relationship ($A_1F_{1,2} = A_2F_{2,1}$) allows the calculation and storage of only half the view factors. VLITE stores the view factors as shown in figure D.20. This guarantees reciprocity, but does not insure that

$$\sum_{j=1}^{N} A_i F_{i,j} = A_i \qquad \text{(D.21)}$$

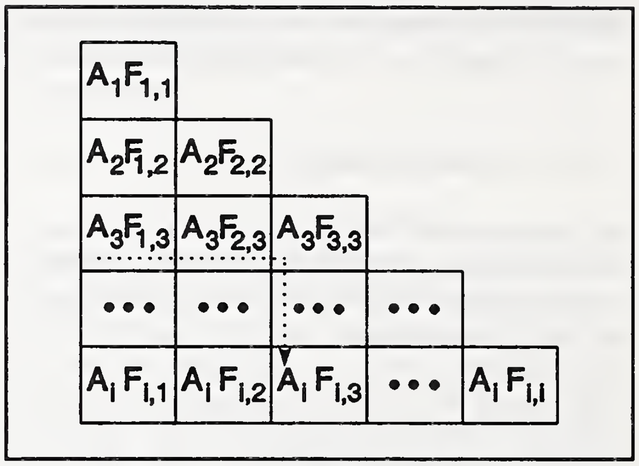


Figure D.20 VLITE View Factor Array

A simple iterative procedure has been found to satisfy equation (D.21):

(1) the sum of $A_iF_{i,j}$ is computed for each row;
(2) that sum is divided by A_i to create an adjustment factor;
(3) all $A_iF_{i,j}$ in the row are divided by that adjustment factor;
(4) steps 1 to 3 are repeated until the adjustment factors for all rows are sufficiently close to one. Note that because of the triangular storage scheme, the factors in a row include the factors in the column under the diagonal.

D.5.4 HLITE input

VLITE can produce the view factor matrix link (B.3.4.7) used by HLITE. This matrix is printed in the VLITE output file and must be moved to the HLITE input file by a text editor. A link name will have to be added.

The surface areas used in computing the view factors must match the surface areas used in HLITE. The areas are reported as comments in the view factor matrix to assist you in this check. An area error will require a change in the HLITE NDF or a change in the VLITE input to compute a new set of view factors. The surface names must also match HLITE surface names, but any name error can be corrected by editing.

D.5.5 Emissivity

It is also possible to produce a listing of radiation interchange factors which include the effects of surface emissivity. These are calculated using an algorithm from Hottel [1967, pp 93-100]. This calculation is also done in HLITE from the surface view factors and emissivities.

APPENDIX E

VLITE Input and Output

E.1 Introduction

The execution of VLITE is controlled interactively, but the primary input and output data are kept in ASCII files. The input file contains the description of the problem geometry. This file is created by the user with a text editor. The output file will contain the computed view factors and other data. This data may be edited and used as input for other programs; a special provision has been made for creating data for the HLITE program. Several aspects of the simulation are controlled interactively, and information is displayed on the screen indicating the progress of the calculations.

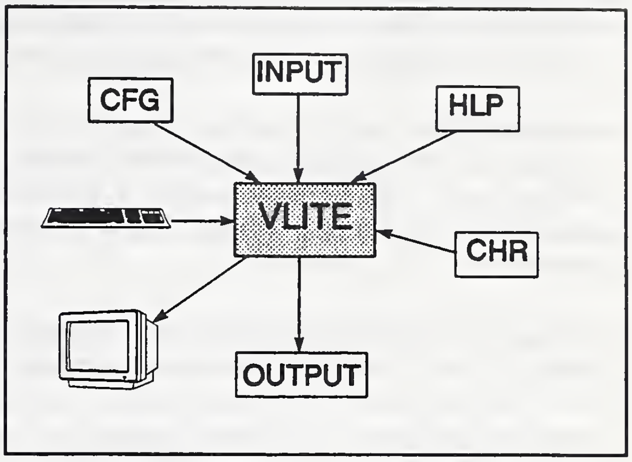


Figure E.1 VLITE Files

There are three other files associated with the execution of VLITE.

VLITE.HLP contains the interactive help information. This file is optional.

CHRSET.VGA or CHRSET.EGA contain the bit patterns for characters used during the graphic display.

VLITE.CFG is an optional configuration file. It determines how the screen appears during processing. This is normally handled by default, but the user can create a custom configuration file by using the MAKECFGG program. The configuration file controls the size of the cursor and the foreground and background colors for various types of messages for both text and graphics modes. The cursor blink rate and the size of the window frames can also be set for graphics mode.

Run VLITE with all files in the current working directory.

E.2 Execution Control

The execution of VLITE is controlled interactively by various keyboard entries (no mouse). You may have to press a key to end a pause, enter a file name or a number, or enter a 'y' or 'n' in response to a question. When VLITE requires an entry, help can usually be obtained by pressing the 'F1' function key. Default values are often provided; if still in doubt, use the default.

The first user choice in VLITE refers to displaying a description of the program. The first time you run VLITE, answer yes to this question. You will also get a description of the interactive input process.

You then enter the names of the input and output files. You may choose to echo the input file as it is processed. This is useful in identifying the inputs which are causing error messages.

You may choose to display a picture of the problem as described by the geometry file. This is the best way of insuring that the geometry has been correctly described. 3-D problems require you to define a view point which is used to determine the direction from which the surfaces are viewed. If an input error is revealed, you should not continue the view factor calculations.

It is important to answer the question of enclosure correctly. It is true if the surfaces form an enclosure, and you are solving for the view factors inside that enclosure. This information is used by VLITE to improve the accuracy of the computed view factors for enclosures.

Some simulation control parameters may then be set. Higher values of the "output control parameter" cause more intermediate calculations to be written to the output file. Higher values of the "surface division factor" cause greater number of subsurfaces and edge divisions to be used in the view factor calculations. The "direction of projection" influences the calculations, but not in an always predictable manner.

During the view factor calculation, the number of the surface being processed is displayed on the screen. This tells how far the calculation has progressed, which can be reassuring on a very long run.

After the calculation of the view factors, you may list them to the screen or to the output file. The amount of data listed is interactively controlled.

If some factors appear to be inaccurate, you may recompute them using a higher surface division factor or the opposite projection.

After you are satisfied with the computed view factors, several post-processing operations are done. Subsurfaces and combinable surfaces are processed. If the surfaces form an enclosure, the view factors are adjusted to insure conservation of energy. These view factors may be reported.

You may direct the creation of view factor data in the form (a **vfm** link) required by the HLITE program. This data can be copied from the VLITE output file to the HLITE input file with a text editor. Add the link name to the HLITE data.

Finally, surface emissivities may be combined with the view factors to create "script F" view factors. You must enter the surface emissivities individually. A default value of 0.9 is provided.

Now try running VLITE with each of the following input files: ROOM.VS3, LAMP2.VS2, and PLENUM.VS1. These will provide examples of the three different geometries. They should execute quickly. Use the F1 help key liberally.

E.3 Input File Format

E.3.1 Description

The input file for the VLITE program is line based. That is, each line of the input file contains specific information in a specific order. Each line begins with a single character which specifies the kind of data on the rest of the line. Remaining data elements are separated by one or more blanks or a comma. The different data line types will now be described according to the first character on the line.

(! /) A '!' or '/' indicates a comment which reports information to the user but is not used by the program. Use comments to document special features described by the input file.

(T t) A 'T' or 't' indicates a title. The remaining information on this line is transferred to all intermediate files and is echoed in all view factor reports. This is usually the first line of the input file.

(G g) The geometry line indicates the type of geometry being described by the rest of the input file. It consists of a 'G' or 'g' followed by a single parameter: 3 for a 3-D problem, 2 for a 2-D problem, or 1 for a 1-D (area information only) problem. This line determines how the vertex and surface lines will be interpreted.

(* E e) A '*' or 'E' or 'e', as in 'End', indicates end-of-information which terminates reading the input file. Any information on the file after this line will not be read. This allows you to store additional information, such as different geometric descriptions, at the end of the input file. The '*' is for consistency with the HLITE input file.

(V v) The vertex lines give the coordinates of the vertices. Each line consists of a 'V' or 'v', the vertex number, and the coordinates of the vertex. The vertices must be numbered in order starting at 1. 2-D geometry requires the X and Y coordinates of the vertex; 3-D geometry requires the X, Y, and Z coordinates.

(S s) The surface lines define the radiatively participating surfaces. Each line consists of an 'S' or 's', the surface number, the vertex pointers, a pointer for obstruction surface inclusion, a pointer for simple surface inclusion, a pointer for surface combination, and the surface name. The surfaces must be numbered in order beginning at 1.

3-D geometry requires the vertex numbers of four vertices. The vertex numbers are listed in counter-clockwise sequence as viewed from the front of the surface. If the surface is a triangle, the fourth vertex number should be zero. Quadrilateral surfaces must be convex, i.e. no interior angle greater than 180°. 2-D surfaces have only two vertices because each surface is represented by a straight line. The vertex at the right (when viewed from the active side of the line) end of the line is listed first.

Obstruction surface inclusion means that the surface being described is included as part of an obstructing surface for the purposes of evaluating inter-surface view obstructions. The pointer refers to the number of an obstructing surface as defined in the next section. A value of 0 indicates no inclusion. Note that the included surface must lie in the plane of and entirely within the larger obstruction surface.

Simple surface inclusion means that the surface is included as part of a larger surface for the purpose of evaluating view factors (see also D.5.2). For example, a rectangular wall which includes a window can be described by setting the surface inclusion pointer of the window to the surface number of the wall. Note that the included surface must lie in the plane of and entirely within the including surface; the simple inclusion algorithm requires that the included surface have a surface number that immediately follows its including surface; and the 'enclosure error' values are not meaningful until the included surfaces have been treated in post-processing.

Surface combination allows several surfaces to be combined into a single surface in the final view factors (see also D.5.1). This is necessary for describing a non-convex surface; until post-processing it must be treated as two or more convex surfaces. The wall around the window could have been described as four rectangles which will be combined during post-processing. Note that this causes a renumbering of the radiation interchange areas, and the sequence of combining surfaces is not important.

The surface name helps to identify surfaces in the reports. A name may have no imbedded blanks; use '_', '-', or selective capitalization instead. If view factors are being computed for the HLITE program, the surface names must match the corresponding surface node names.

(O o) The obstruction surface lines define the surfaces for which view factors will not be computed but which may obstruct the views between pairs of radiative surfaces. Each line consists of an 'O', the surface number, and the vertex pointers. This data is similar to the first part of the surface lines. The obstruction surfaces must be numbered in order starting after the last radiating surface continuing the surface number sequence. Obstruction surfaces can be used to represent several regular surfaces to simplify the obstruction calculation.

E.3.2 Input File Examples

The following examples are based on the test room for the study of the lighting/HVAC interaction.

E.3.2.1 2-D Lamp

The first example is a two tube fluorescent luminaire. A two dimensional model is appropriate for the elongated geometry of this device. Figure E.2 shows how the following input file describes the luminaire.

The thermal network of the luminaire consists of three surfaces: the lamp, the housing, and the lens. Surfaces 2, 3, and 4 combine to form the housing. Surfaces 5 through 28 combine to form the lamp. The two lamps are composed of many short line segments to approximate their cylindrical shape.

```
T  2 tube luminaire
G  2
V    1   0.00   0.00
V    2   1.00   0.00
V    3   1.00   0.417
V    4   0.00   0.417
V    5   0.250  0.229
V    6   0.281  0.221
V    7   0.304  0.200
V    8   0.313  0.167
V    9   0.304  0.135
V   10   0.281  0.112
V   11   0.250  0.104
V   12   0.219  0.112
V   13   0.196  0.135
V   14   0.187  0.167
V   15   0.196  0.200
V   16   0.219  0.221
V   17   0.750  0.229
V   18   0.781  0.221
V   19   0.804  0.200
V   20   0.813  0.167
V   21   0.804  0.135
V   22   0.781  0.112
V   23   0.750  0.104
V   24   0.719  0.112
V   25   0.696  0.135
V   26   0.687  0.167
V   27   0.696  0.200
V   28   0.719  0.221                   ┌──────────── surface combinations here
S    1    1    2    0    0    0    AcrLens
S    2    2    3    0    0    0    housing
S    3    3    4    0    0    2    housing
S    4    4    1    0    0    2    housing
S    5    5    6    0    0    0    tube
S    6    6    7    0    0    5    tube-1
S    7    7    8    0    0    5    tube-1
S    8    8    9    0    0    5    tube-1
S    9    9   10    0    0    5    tube-1
S   10   10   11    0    0    5    tube-1
S   11   11   12    0    0    5    tube-1
S   12   12   13    0    0    5    tube-1
S   13   13   14    0    0    5    tube-1
S   14   14   15    0    0    5    tube-1
S   15   15   16    0    0    5    tube-1
S   16   16    5    0    0    5    tube-1
S   17   17   18    0    0    5    tube-2
S   18   18   19    0    0    5    tube-2
S   19   19   20    0    0    5    tube-2
S   20   20   21    0    0    5    tube-2
S   21   21   22    0    0    5    tube-2
S   22   22   23    0    0    5    tube-2
S   23   23   24    0    0    5    tube-2
S   24   24   25    0    0    5    tube-2
S   25   25   26    0    0    5    tube-2
S   26   26   27    0    0    5    tube-2
S   27   27   28    0    0    5    tube-2
S   28   28   17    0    0    5    tube-2
End of data
```

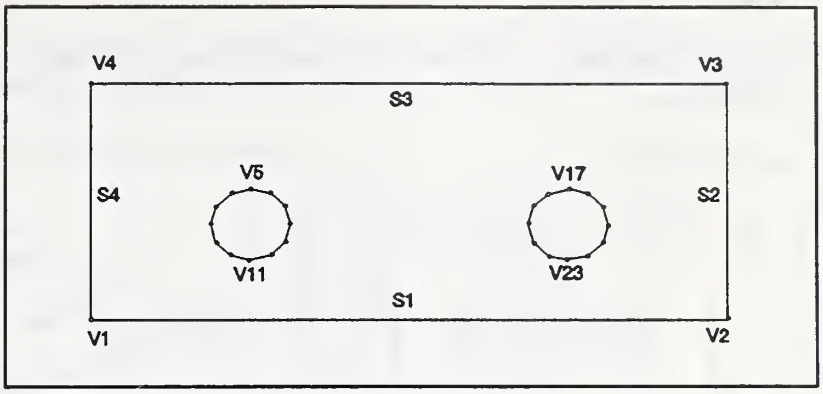


Figure E.2 Luminaire Configuration

E.3.2.2 Test Room

The test room with its four luminaires in the ceiling provide a very simple geometry for view factor calculations.

file ROOM.VS3:

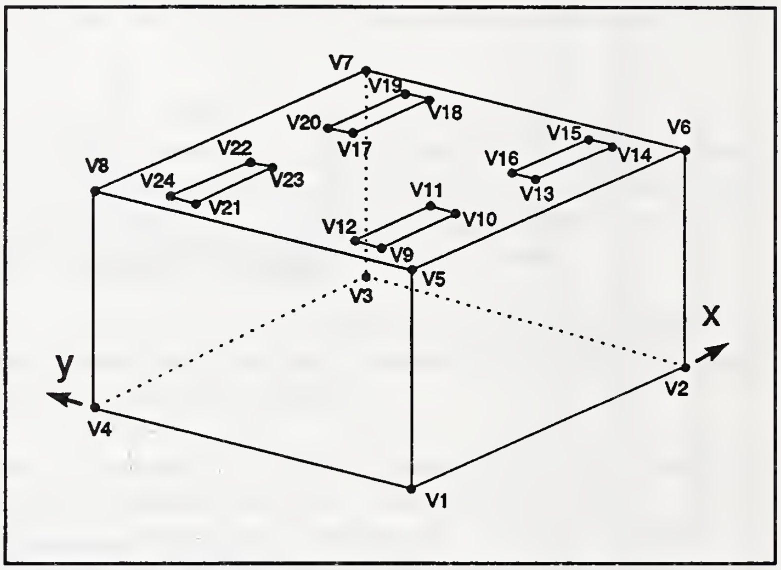


Figure E.3 Test Room Configuration

```
T  data for room geometry
G  3
V    1    0.       0.      0.
V    2    0.      12.25    0.
V    3   13.875   12.25    0.
V    4   13.875    0.      0.
V    5    0.       0.      8.
V    6    0.      12.25    8.
V    7   13.875   12.25    8.
V    8   13.875    0.      8.
V    9    0.94     2.0     8.
V   10    4.94     2.0     8.
V   11    4.94     3.0     8.
V   12    0.94     3.0     8.
V   13    8.94     2.0     8.
V   14   12.94     2.0     8.
V   15   12.94     3.0     8.
V   16    8.94     3.0     8.
V   17    0.94     9.0     8.
V   18    4.94     9.0     8.
V   19    4.94    10.0     8.
V   20    0.94    10.0     8.
V   21    8.94     9.0     8.
V   22   12.94     9.0     8.
V   23   12.94    10.0     8.
V   24    8.94    10.0     8.
                                   ------------- included surfaces here
                                   |   ---------------- surface combinations here
                                   |   |
S    1    1    4    3    2    0    0   0   Floor
S    2    1    5    8    4    0    0   0   W-Wall
S    3    1    2    6    5    0    0   0   S-Wall
S    4    2    3    7    6    0    0   0   E-Wall
S    5    3    4    8    7    0    0   0   N-Wall
S    6    5    6    7    8    0    0   0   SspCeil
S    7    9   12   11   10    0    6   0   AcrLens
S    8   13   16   15   14    0    6   7   lens-2
S    9   17   20   19   18    0    6   7   lens-3
S   10   21   24   23   22    0    6   7   lens-4
End of data
```

The only geometric complication is the four luminaires in the ceiling. Here they are modeled by subtracting the lens surfaces, 7 to 10, from the ceiling surface, 6, and then combining the four lens surfaces into a single lens surface.

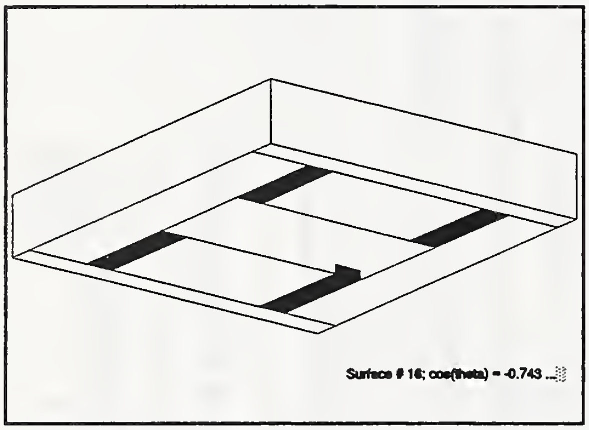
E.3.2.3 Plenum

The test room plenum is a very complex space. The input file uses many surfaces to represent the suspended ceiling, the luminaire housings, and the structural steel.

```
T  data for plenum geometry - HLITE test case
G  3
V   1   0.      0.     10.5
V   2   0.     12.25   10.5
V   3  13.875  12.25   10.5
V   4  13.875   0.     10.5
V   5   0.      0.      8.
V   6   0.     12.25    8.
V   7  13.875  12.25    8.
V   8  13.875   0.      8.
V   9   0.94    2.0     8.
V  10   4.94    2.0     8.
V  11   4.94    3.0     8.
V  12   0.94    3.0     8.
V  13   8.94    2.0     8.
V  14  12.94    2.0     8.
V  15  12.94    3.0     8.
V  16   8.94    3.0     8.
V  17   0.94    9.0     8.
V  18   4.94    9.0     8.
V  19   4.94   10.0     8.
V  20   0.94   10.0     8.
V  21   8.94    9.0     8.
V  22  12.94    9.0     8.
V  23  12.94   10.0     8.
V  24   8.94   10.0     8.
V  25   0.94    2.0     8.333
V  26   4.94    2.0     8.333
V  27   4.94    3.0     8.333
V  28   0.94    3.0     8.333
V  29   8.94    2.0     8.333
V  30  12.94    2.0     8.333
V  31  12.94    3.0     8.333
V  32   8.94    3.0     8.333
V  33   0.94    9.0     8.333
V  34   4.94    9.0     8.333
V  35   4.94   10.0     8.333
V  36   0.94   10.0     8.333
V  37   8.94    9.0     8.333
V  38  12.94    9.0     8.333
V  39  12.94   10.0     8.333
V  40   8.94   10.0     8.333
V  41   0.94    0.      8.
V  42   0.94   12.25    8.
V  43  12.94    0.      8.
V  44  12.94   12.25    8.
V  45   2.0     1.0     9.0
V  46   2.0     1.0    10.0
V  47   2.0    11.0    10.0
V  48   2.0    11.0     9.0
V  49   3.0     1.0     9.0
V  50   3.0     1.0    10.0
V  51   3.0    11.0    10.0
V  52   3.0    11.0     9.0
V  53   4.0     1.0     9.0
V  54   4.0     1.0    10.0
V  55   4.0    11.0    10.0
V  56   4.0    11.0     9.0
V  57   5.0     1.0     9.0
V  58   5.0     1.0    10.0
V  59   5.0    11.0    10.0
V  60   5.0    11.0     9.0
V  61   6.0     1.0     9.0
V  62   6.0     1.0    10.0
V  63   6.0    11.0    10.0
V  64   6.0    11.0     9.0
V  65   7.9     1.0     9.0
V  66   7.9     1.0    10.0
V  67   7.9    11.0    10.0
V  68   7.9    11.0     9.0
V  69   8.9     1.0     9.0
V  70   8.9     1.0    10.0
V  71   8.9    11.0    10.0
V  72   8.9    11.0     9.0
V  73   9.9     1.0     9.0
V  74   9.9     1.0    10.0
V  75   9.9    11.0    10.0
V  76   9.9    11.0     9.0
V  77  10.9     1.0     9.0
V  78  10.9     1.0    10.0
V  79  10.9    11.0    10.0
V  80  10.9    11.0     9.0
V  81  11.9     1.0     9.0
V  82  11.9     1.0    10.0
V  83  11.9    11.0    10.0
V  84  11.9    11.0     9.0
S   1   1   2   3   4   0   0   0   UnderSide
S   2   1   4   8   5   0   0   0   S-Plenum
S   3   1   5   6   2   0   0   0   W-Plenum
S   4   2   6   7   3   0   0   0   N-Plenum
S   5   3   7   8   4   0   0   0   E-Plenum
S   6   5  41  42   6   0   0   0   PlenumSC
S   7  43   8   7  44   0   0   6   SC-2
S   8  41  43  14   9   0   0   6   SC-3
S   9  20  23  44  42   0   0   6   SC-4
S  10  10  13  24  19   0   0   6   SC-5
S  11  12  11  18  17   0   0   6   SC-6
S  12  16  15  22  21   0   0   6   SC-7
S  13  25  26  27  28   0   0   0   housing
S  14  12   9  25  28   0   0  13   housing-1
S  15   9  10  26  25   0   0  13   housing-1
S  16  10  11  27  26   0   0  13   housing-1
S  17  11  12  28  27   0   0  13   housing-1
S  18  29  30  31  32   0   0  13   housing-2
S  19  16  13  29  32   0   0  13   housing-2
S  20  13  14  30  29   0   0  13   housing-2
S  21  14  15  31  30   0   0  13   housing-2
S  22  15  16  32  31   0   0  13   housing-2
S  23  33  34  35  36   0   0  13   housing-3
S  24  20  17  33  36   0   0  13   housing-3
S  25  17  18  34  33   0   0  13   housing-3
S  26  18  19  35  34   0   0  13   housing-3
S  27  19  20  36  35   0   0  13   housing-3
S  28  37  38  39  40   0   0  13   housing-4
S  29  24  21  37  40   0   0  13   housing-4
S  30  21  22  38  37   0   0  13   housing-4
S  31  22  23  39  38   0   0  13   housing-4
S  32  23  24  40  39   0   0  13   housing-4
S  33  45  46  47  48   0   0   0   Steel
S  34  45  48  47  46   0   0  33   steel
S  35  49  50  51  52   0   0  33   steel
S  36  49  52  51  50   0   0  33   steel
S  37  53  54  55  56   0   0  33   steel
S  38  53  56  55  54   0   0  33   steel
S  39  57  58  59  60   0   0  33   steel
S  40  57  60  59  58   0   0  33   steel
S  41  61  62  63  64   0   0  33   steel
S  42  61  64  63  62   0   0  33   steel
S  43  65  66  67  68   0   0  33   steel
S  44  65  68  67  66   0   0  33   steel
S  45  69  70  71  72   0   0  33   steel
S  46  69  72  71  70   0   0  33   steel
S  47  73  74  75  76   0   0  33   steel
S  48  73  76  75  74   0   0  33   steel
S  49  77  78  79  80   0   0  33   steel
S  50  77  80  79  78   0   0  33   steel
S  51  81  82  83  84   0   0  33   steel
S  52  81  84  83  82   0   0  33   steel
End of data
```

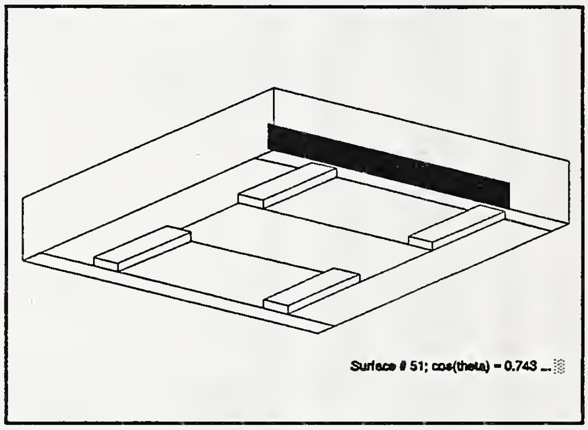
This complex configuration provides an excellent example of the graphic output provided by VLITE to check the geometric correctness of the input file. This visual check allows you to see the surfaces from any view point. Figures E.4 through E.7 are taken from the screen as surfaces are displayed one at a time.

Figure E.4 shows how 7 surfaces are used to form the floor of the plenum which include cut-outs for the four luminaires. Included surfaces will not be here because the surfaces in this plane (the lenses) are not part of the plenum radiant interchange. This figure includes the first luminaire housing surface. Figure E.5 includes all the housing surfaces and the first surface representing the structural steel.



```
Surface # 16; cos(theta) = -0.743 ...
```

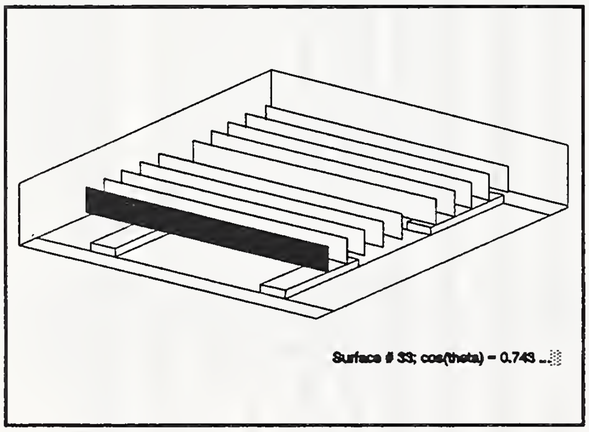
Figure E.4



```
Surface # 51; cos(theta) = 0.743 ...
```

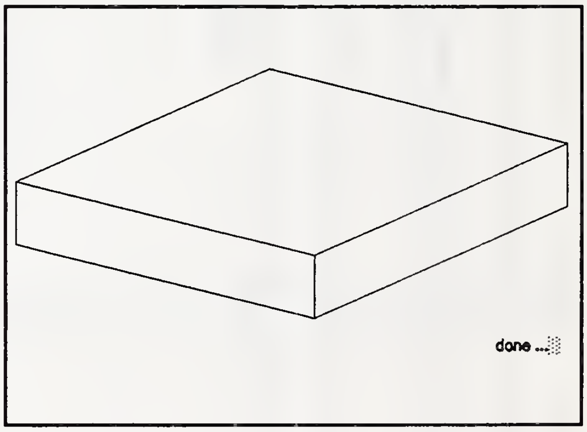
Figure E.5

Figure E.6 includes all the structural steel surfaces. Figure E.7 shows the final 3 surfaces which hide all the surfaces behind them.



```
Surface # 33; cos(theta) = 0.743 ...
```

Figure E.6



```
done ...
```

Figure E.7

The lower right corner of each screen shows the number of the surface being displayed. The color of the surface (which cannot be shown here) indicates whether the surface is facing toward or away from the viewer. This check should be sufficient to guarantee the accuracy of the geometric model.

E.3.2.4 1-D Plenum Model

The complexity of the plenum model indicates a need for a simpler view factor calculation. A special model is provided called the 1-dimensional model. Each surface is described by a single number, its area. This is a model which computes view factors with very limited accuracy but it tends to give reasonable results in the heat balance calculations. The following input file describes the 1-D model of the plenum.

file PLENUM.VS1:

```
T  data for simplified plenum geometry - HLITE test case
G  1
S  1 169.9688  UnderSide
S  2  34.6875  S-Plenum
S  3  30.6250  W-Plenum
S  4  34.6875  N-Plenum
S  5  30.6250  E-Plenum
S  6 153.9688  PlenumSC
S  7  29.3200  housing
S  8 200.0000  Steel
*
```

For 1-D geometry this line consists of the 'S' or 's', the surface number, the surface area, and the surface name. There are no vertices or obstructions.

Results using full 3-D computation:

```
row     1       2       3       4       5       6       7       8
 1  .000000 .066128 .055179 .068052 .055004 .367762 .046495 .341380
 2  .324029 .000000 .063521 .019422 .063180 .302814 .058125 .168909
 3  .306246 .071947 .000000 .073266 .007816 .303065 .051786 .185875
 4  .333454 .019422 .064685 .000000 .064448 .312225 .053458 .152308
 5  .305271 .071561 .007816 .072997 .000000 .302494 .051836 .188025
 6  .405979 .068221 .060281 .070341 .060167 .000000 .040313 .294699
 7  .269532 .068766 .054091 .063244 .054143 .211695 .002755 .275774
 8  .290119 .029295 .028462 .026416 .028791 .226872 .040429 .329616
```

Results using simple 1-D computation:

```
row     1       2       3       4       5       6       7       8
 1  .000000 .058506 .051600 .058521 .051585 .284519 .046918 .448352
 2  .286678 .000000 .039095 .044338 .039084 .215565 .035547 .339693
 3  .286381 .044281 .000000 .044017 .038800 .214002 .035289 .337230
 4  .286751 .044338 .038862 .000000 .039093 .215620 .035556 .339779
 5  .286299 .044268 .038800 .044279 .000000 .213941 .035279 .337134
 6  .314085 .048565 .042566 .048577 .042554 .000000 .047712 .455942
 7  .271983 .042055 .036860 .042065 .036849 .250551 .000000 .319637
 8  .381029 .058916 .051638 .058931 .051624 .351004 .046859 .000000
```

E.4 Output File Format

The following output data was produced while computing view factors for the plenum (see E.3.2.3). Numbers, e.g. **(3)**, refer to comments at the end of the output.

```
Project:  data for plenum geometry - HLITE test case                          (1)
                                                                              (2)
Time to read data file:   0.16
   52 total surfaces 'NTOT'
   52 radiating surfaces 'NSRF'                                               (3)
   84 vertices 'NVRT'

Time for initializations:   0.05
   40 possible view obstructing surfaces 'NPOS'                               (4)
 List of potential view obstructing surfaces:
   13   14   15   16   17   18   19   20   21   22
   23   24   25   26   27   28   29   30   31   32
   33   34   35   36   37   38   39   40   41   42
   43   44   45   46   47   48   49   50   51   52

    1 output control parameter 'LIST'                                         (5)
 4.00 surface division factor 'DIVF'
    0 row control parameter 'IROW'
    0 column control parameter 'ICOL'
    1 enclosure designator 'ENCL'
    1 projection parameter 'PDIR'

Surface pairs where F(i,j) must equal zero:  716                              (6)
Surface pairs without obstructing surfaces:   91
Surface pairs with obstructing surfaces:     519
Total number of obstructions considered:    5588
                                                                              (7)
          1    2    3    4    5    6    7    8    9   10   11   12   13   14
 NOS:    26   41   36   33   16   35   38   33   12   20   32   30   24  143
 NDli:    0   97   42    0    0    0    0    0    0    0    0    0    0    0
 NDms:    0   22  245   78   35    0    0    0    0    0    0    0    0    0


  53.85 seconds to compute view factors.

 data for plenum geometry - HLITE test case

 Row:     1   UnderSide               0.993954     0.006046                   (8)
.000000 .064994 .055044 .067225 .054835 .023512 .023211 .050605 .061876 .087533
.060032 .059956 .009555 .000065 .000489 .000193 .001027 .009548 .000194 .000488
.000064 .001024 .009777 .000066 .001023 .000200 .000598 .009742 .000201 .001020
.000065 .000597 .016079 .016665 .016194 .016677 .016440 .016677 .016588 .016677
.016677 .020992 .020993 .016675 .016677 .016585 .016677 .016435 .016677 .016186
.016665 .015961
 ...
 Row:    52   steel                   1.001175     0.001175
.271288 .024352 .000000 .020047 .491183 .000000 .092647 .008663 .009132 .000000
.000000 .055119 .000000 .000000 .000000 .000000 .000000 .012542 .000000 .000487
.000000 .001344 .000000 .000000 .000000 .000000 .000000 .012542 .000000 .001344
.000000 .000487 .000000 .000000 .000000 .000000 .000000 .000000 .000000 .000000
.000000 .000000 .000000 .000000 .000000 .000000 .000000 .000000 .000000 .000000
.000000 .000000
```

```
Summary:

Max enclosure error:  0.015548                                                    (9)
RMS enclosure error:  0.005739
Largest errors [row, error]:
      12  0.015548
      11  0.015528
       7  0.012876
       6  0.012282
      47  0.011104
      38  0.010882
       2  0.007728
      10  0.007402
      42  0.006553
      43  0.006354

 New,   Old surface numbers                                                      (10)
   1:   1
   2:   2
   3:   3
   4:   4
   5:   5
   6:   6   7   8   9  10  11  12
   7:  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30
31  32
   8:  33  34  35  36  37  38  39  40  41  42  43  44  45  46  47  48  49  50
51  52
Number of surfaces reduced to 8.

   0.05 seconds to combine surfaces.

Combined view factors:
 data for plenum geometry - HLITE test case

 Row:     1   UnderSide              0.993954     0.006046                      (11)
.000000 .064994 .055044 .067225 .054835 .366726 .045934 .339196
 Row:     2   S-Plenum               0.992272     0.007728
.318468 .000000 .063463 .019216 .063084 .302434 .057514 .168091
 Row:     3   W-Plenum               1.005830     0.005830
.305492 .071882 .000000 .073573 .007921 .307212 .052008 .187742
 Row:     4   N-Plenum               0.997183     0.002817
.329401 .019216 .064956 .000000 .064679 .313422 .053166 .152343
 Row:     5   E-Plenum               1.005234     0.005234
.304333 .071453 .007921 .073259 .000000 .306446 .052026 .189797
 Row:     6   PlenumSC               1.003665     0.003665
.404836 .068135 .061106 .070611 .060953 .000000 .040471 .297554
 Row:     7   housing                0.997029     0.002971
.266282 .068043 .054323 .062899 .054342 .212528 .002747 .275865
 Row:     8   Steel                  1.001775     0.001775
.288264 .029153 .028748 .026422 .029063 .229070 .040442 .330613

Summary:

Max enclosure error:  0.007728
RMS enclosure error:  0.004886
Largest errors [row, error]:
       2  0.007728
       1  0.006046
       3  0.005830
       5  0.005234
       6  0.003665
       7  0.002971
       4  0.002817
       8  0.001775
```

```
Surfaces:                                                                      (12)
  n   bsn   csn    area    name
  1    0     0  1.6997e+02  UnderSide
  2    0     0  3.4688e+01  S-Plenum
  3    0     0  3.0625e+01  W-Plenum
  4    0     0  3.4688e+01  N-Plenum
  5    0     0  3.0625e+01  E-Plenum
  6    0     0  1.5397e+02  PlenumSC
  7    0     0  2.9320e+01  housing
  8    0     0  2.0000e+02  Steel

AFNORM: 0  err: 0.008107                                                       (13)
AFNORM: 1  err: 0.003055
AFNORM: 2  err: 0.000362
AFNORM: 3  err: 0.000051
AFNORM: 4  err: 0.000019
AFNORM: 5  err: 0.000003
AFNORM: 6  err: 0.000000
  7 iterations

   0.00 seconds to normalize factors.

 data for plenum geometry - HLITE test case

 Row:     1   UnderSide              1.000000     0.000000                     (14)
.000000 .066128 .055179 .068052 .055004 .367762 .046495 .341380
 Row:     2   S-Plenum               1.000000     0.000000
.324029 .000000 .063521 .019422 .063180 .302814 .058125 .168909
 Row:     3   W-Plenum               1.000000     0.000000
.306246 .071947 .000000 .073266 .007816 .303065 .051786 .185875
 Row:     4   N-Plenum               1.000000     0.000000
.333454 .019422 .064685 .000000 .064448 .312225 .053458 .152308
 Row:     5   E-Plenum               1.000000     0.000000
.305271 .071561 .007816 .072997 .000000 .302494 .051836 .188025
 Row:     6   PlenumSC               1.000000     0.000000
.405979 .068221 .060281 .070341 .060167 .000000 .040313 .294699
 Row:     7   housing                1.000000     0.000000
.269532 .068766 .054091 .063244 .054143 .211695 .002755 .275774
 Row:     8   Steel                  1.000000     0.000000
.290119 .029295 .028462 .026416 .028791 .226872 .040429 .329616

Summary:

Max enclosure error:  0.000000
RMS enclosure error:  0.000000


link vfm 8   !  data for plenum geometry - HLITE test case                     (15)
UnderSide                  ! area: 169.9688
.000000 .066128 .055179 .068052 .055004 .367762 .046495 .341380
S-Plenum                   ! area: 34.6875
.324029 .000000 .063521 .019422 .063180 .302814 .058125 .168909
W-Plenum                   ! area: 30.6250
.306246 .071947 .000000 .073266 .007816 .303065 .051786 .185875
N-Plenum                   ! area: 34.6875
.333454 .019422 .064685 .000000 .064448 .312225 .053458 .152308
E-Plenum                   ! area: 30.6250
.305271 .071561 .007816 .072997 .000000 .302494 .051836 .188025
PlenumSC                   ! area: 153.9688
.405979 .068221 .060281 .070341 .060167 .000000 .040313 .294699
housing                    ! area: 29.3200
.269532 .068766 .054091 .063244 .054143 .211695 .002755 .275774
Steel                      ! area: 200.0000
.290119 .029295 .028462 .026416 .028791 .226872 .040429 .329616
```

Comments:

(1) The title line is echoed to the head of the output file.

(2) The echo of the input file would appear here (see E.3.2.3).

(3) Report the number of vertices and surfaces. NTOT equals NSRF plus obstruction surfaces.

(4) The potential view obstructing surfaces are determined during initialization (see D.2.3 and D.3.3.3), and the total number of surfaces and individual surface numbers are reported.

(5) The values of the interactive simulation controls parameters (see E.2) are reported allowing recreation of the calculations.

(6) This is a summary of the types of calculations done. Some factors are determined to be zero by surface orientation tests; for views between some pairs of surfaces there are no obstructions; for others there may be one or more obstructions. Execution time is strongly related to the surface division factor and the total number of obstructions.

(7) Further details on the calculations are given. Line NOS: means that 26 views involve 1 obstruction, 41 involve 2 obstructions, ..., and 143 views involve 14 or more obstructions. Line NDli: indicates that obstructed views were computed with the line integral method (D.3.1.2) using 2 divisions per edge (4 subsurfaces) 97 times and 3 divisions per edge (9 subsurfaces) 42 times. Line NDms: indicates that the Mitalas-Stephenson method (D.3.1.3) was used with 2 divisions per edge (4 subsurfaces) 22 times, ..., and 5 divisions per edge (25 subsurfaces) 35 times.

(8) The detailed report for all surfaces includes the row (or surface) number, name, sum of the view factors, and (if the surface form an enclosure) $| 1 - \Sigma F |$. This line is followed by the individual view factors.

(9) The summary report includes the maximum and average (RMS) values of $| 1 - \Sigma F |$, and the ten largest values together with the row where they occur. These values may be used to guide you in recomputing some view factors.

Several of the following reports were generated during post-processing by requesting a dump.

(10) This report indicates that some surfaces have been combined reducing the total number of surfaces to 8.

(11) This view factor report is generated by requesting the dump.

(12) The revised areas of the combined surfaces should be checked for agreement with expected values.

(13) Normalization (for an enclosure) of the view factor matrix proceeds by a series of iterations (see D.5.3) until the error value is sufficiently small.

(14) Note that the reported view factors now sum to 1.000000 for each row.

(15) The HLITE input for a **vfm** link (B.3.4.7) can the transferred to the NDF with an editor. Be sure to add a name for the link.

APPENDIX F

VLITE Test Cases

F.1 2-D Tests

There are several sample inputs and outputs for 2-D problems on the distribution disk. The following discussion of those examples will help explain the use of the VLITE for 2-D problems.

F.1.1 TEST1.VS2, unobstructed views

This simple test consists of four surfaces in the shape of a rectangular cavity as shown in figure F1.

This geometry is specified in file TEST1.VS2 as shown below. Note especially the sequence of the vertex pointers for each surface.

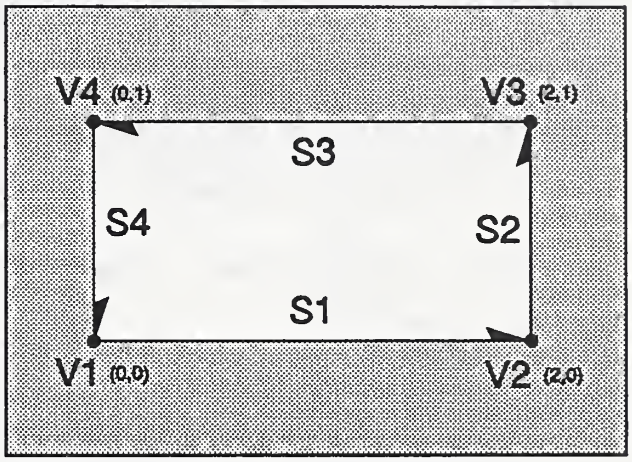


Figure F1. Geometry for TEST1.VS2

```
T  TEST1:  4 surface enclosure
G  2
V  1    .00   .00
V  2   2.00   .00
V  3   2.00  1.00
V  4    .00  1.00
S  1   1  2  0  0  0   south_surface
S  2   2  3  0  0  0   east_surface
S  3   3  4  0  0  0   north_surface
S  4   4  1  0  0  0   west_surface
End of data
```

The length of sides 1 and 3 is 2.0; the length of sides 2 and 4 is 1.0; the length of the diagonals is $\sqrt{5}$. Substituting the appropriate values into eqn D.2 gives $F_{1,3} = 0.61803399$, $F_{1,2} = 0.19098301$, and $F_{2,4} = 0.23606798$. Other values can be obtained by symmetry or by reciprocity.

$F_{1,1} = 0.$	$F_{1,2} = 0.19098301$	$F_{1,3} = 0.61803399$	$F_{1,4} = 0.19098301$
$F_{2,1} = 0.38196601$	$F_{2,2} = 0.$	$F_{2,3} = 0.38196601$	$F_{2,4} = 0.23606798$
$F_{3,1} = 0.61803399$	$F_{3,2} = 0.19098301$	$F_{3,3} = 0.$	$F_{3,4} = 0.19098301$
$F_{4,1} = 0.38196601$	$F_{4,2} = 0.23606798$	$F_{4,3} = 0.38196601$	$F_{4,4} = 0.$

Since the surfaces form an enclosure, the view factors in each row sum to 1.

```
1  .000000 .190983 .618034 .190983   1.000000
2  .381966 .000000 .381966 .236068
3  .618034 .190983 .000000 .190983
4  .381966 .236068 .381966 .000000
```

F.1.2 TEST2.VS2, simple obstruction

This simple test consists of four surfaces as shown in figure F2.

This geometry is specified in file TEST2.VS2 as shown below. Again note the sequence of the vertices for each surface. The obstructions consist of a pair of surfaces facing in opposite directions so as to block the view in either direction.

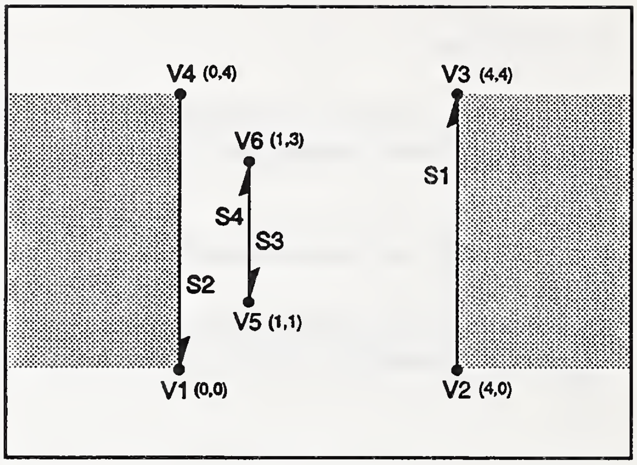


Figure F2. Simple Obstruction

```
T  TEST2:  4 surface test - simple obstruction - no enclosure
G  2
V    1    .00   .00
V    2   4.00   .00
V    3   4.00  4.00
V    4    .00  4.00
V    5   1.00  1.00
V    6   1.00  3.00
S    1    2    3    0    0    0    east_surface
S    2    4    1    0    0    0    west_surface
S    3    6    5    0    0    0    east_obstruction
S    4    5    6    0    0    0    west_obstruction
End of data
```

The vertices have been chosen so that view factors can be computed analytically. The unobstructed view $F_{1,2'}$ = 0.41421356 and $F_{1,3}$ = .27009076. Therefore, $F_{1,2}$ = .14412280 (= $F_{1,2'}$ - $F_{1,3}$). The complete set of view factors is:

$F_{1,1}$ = 0.	$F_{1,2}$ = 0.14412280	$F_{1,3}$ = 0.27009076	$F_{1,4}$ = 0.
$F_{2,1}$ = 0.14412280	$F_{2,2}$ = 0.	$F_{2,3}$ = 0.	$F_{2,4}$ = 0.43701602
$F_{3,1}$ = 0.54018152	$F_{3,2}$ = 0.	$F_{3,3}$ = 0.	$F_{3,4}$ = 0.
$F_{4,1}$ = 0.0	$F_{4,2}$ = 0.87403205	$F_{4,3}$ = 0.	$F_{4,4}$ = 0.

Using a division factor of 4, VLITE computed the following view factors:

```
row     1       2       3       4
 1  .000000 .144022 .270091 .000000
 2  .144022 .000000 .000000 .437016
 3  .540181 .000000 .000000 .000000
 4  .000000 .874032 .000000 .000000
```

Only the value of $F_{1,2}$ (and $F_{2,1}$) involves obstruction and is therefore computed by a numerical integration. You can make multiple runs to see how various division factors affect the computed value. The value as computed above is quite good.

F.1.3 TEST3.VS2, complex obstructions

Figure F3 shows the configuration of a more complex test consisting of 8 surfaces. This is an extension of TEST2.VS2 so the input file is not listed here. The geometry is arranged to test the obstruction calculations in some limiting cases.

This test includes surfaces that cross each other. The view factors in this configuration are difficult to compute analytically.

Surfaces 5 and 6 cross surfaces 7 and 8. This is not a typical behavior for surfaces, and VLITE will display warning messages when intersecting surfaces are encountered.

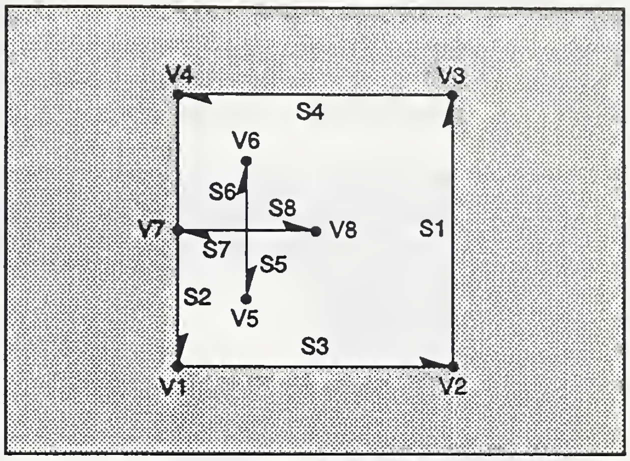


Figure F3. Complex 2D Obstruction

The following analytic view factors are expected:

row	1	2	3	4	5	6	7	8
1	.00000000	.14412280	.29289322	.29289322	.21437180	.00000000	.02785948	.02785948
2	.14412280	.00000000	.17793860	.17793860	.00000000	.30901700	.09549150	.09549150
3	.29289322	.17793860	.00000000	.????????	.????????	.????????	.23020740	.00000000
4	.29289322	.17793860	.????????	.00000000	.????????	.????????	.00000000	.23020740
5	.42874360	.00000000	.????????	.????????	.00000000	.00000000	.14644661	.14644661
6	.00000000	.61803400	.????????	.????????	.00000000	.00000000	.14644661	.14644661
7	.05571896	.19098301	.46041481	.00000000	.14644661	.14644661	.00000000	.00000000
8	.05571896	.19098301	.00000000	.46041481	.14644661	.14644661	.00000000	.00000000

Using a division factor of 4, VLITE computed the following view factors:

row	1	2	3	4	5	6	7	8	Σ $F_{i,j}$
1	.000000	.144022	.292893	.292893	.214539	.000000	.027859	.027860	1.000066
2	.144022	.000000	.184578	.184578	.000000	.323418	.095492	.095492	1.027578
3	.292893	.184578	.000000	.207107	.069591	.022268	.242044	.000000	1.018481
4	.292893	.184578	.207107	.000000	.069591	.022268	.000000	.242044	1.018480
5	.429077	.000000	.139182	.139182	.000000	.000000	.146447	.146447	1.000334
6	.000000	.646835	.044536	.044536	.000000	.000000	.146447	.146447	1.028801
7	.055719	.190983	.484088	.000000	.146447	.146447	.000000	.000000	1.023683
8	.055719	.190983	.000000	.484087	.146447	.146447	.000000	.000000	1.023682

After normalization:

row	1	2	3	4	5	6	7	8	Σ $F_{i,j}$
1	.000000	.142863	.292190	.292191	.217245	.000000	.027756	.027756	1.000000
2	.142863	.000000	.179116	.179116	.000000	.313820	.092542	.092542	1.000000
3	.292190	.179116	.000000	.202122	.068938	.021730	.235903	.000000	1.000000
4	.292191	.179116	.202122	.000000	.068938	.021730	.000000	.235903	1.000000
5	.434490	.000000	.137876	.137876	.000000	.000000	.144879	.144879	1.000000
6	.000000	.627641	.043461	.043461	.000000	.000000	.142719	.142719	1.000000
7	.055511	.185084	.471807	.000000	.144879	.142719	.000000	.000000	1.000000
8	.055511	.185085	.000000	.471806	.144879	.142719	.000000	.000000	1.000000

F.1.4 TEST4.VS2, 100 Surface Test

TEST4.VS2 uses the problem configuration in figure F4 which shows a rectangle made up of 40 surfaces enclosing many small squares and triangles which total 60 obstructing surfaces. This test exists to establish a benchmark execution time for a non-trivial two dimensional problem. The values of the view factors are not of concern. The data file is not listed here to save space.

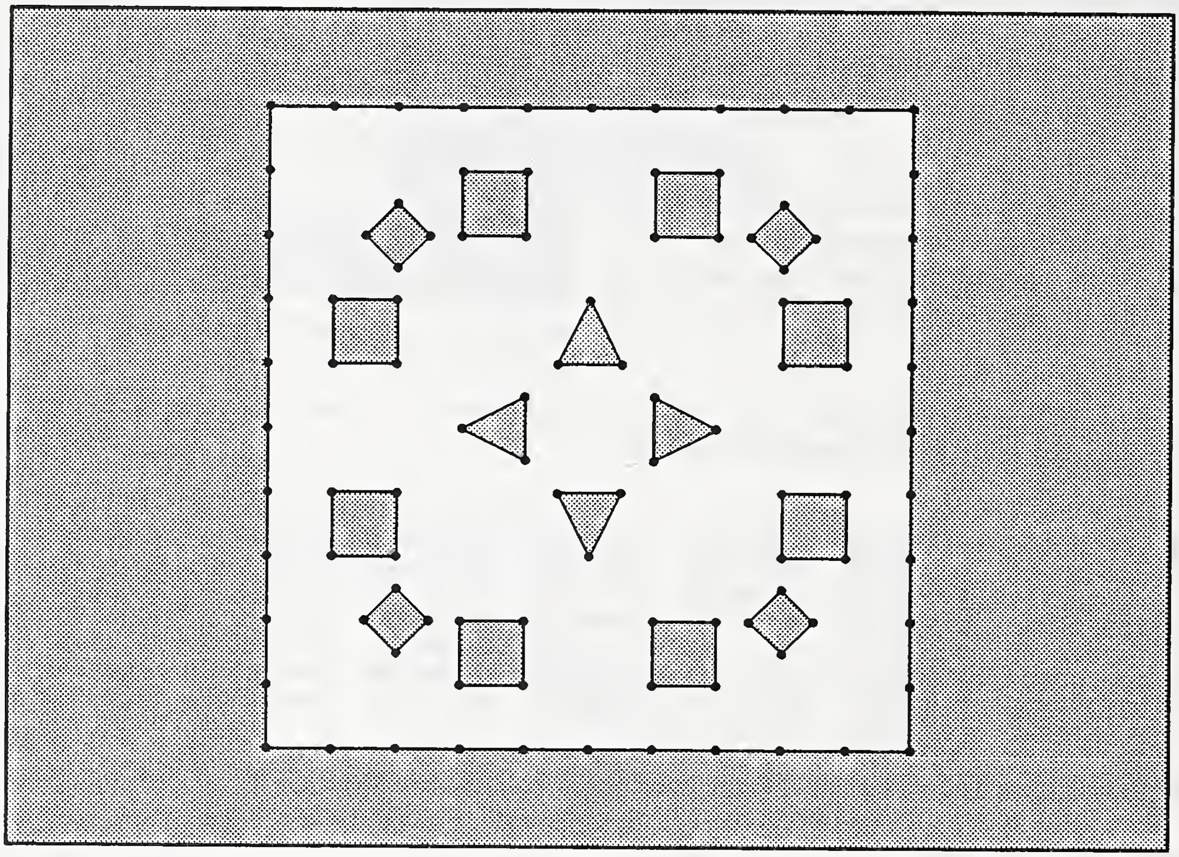


Figure F4. 100 Surface 2D Benchmark

Typical execution times:

machine	view factors	normalize	include ϵ
33 MHz 80486 PC-compatible	8.63 s	0.55 s	2.20 s
5 MHz 8088/87 PC-compatible	225 s	27 s	273 s

F.2 3-D Tests

There are several sample inputs and outputs for 3-D problems on the distribution disk. The following discussion of those examples will help explain the use of the VLITE for 3-D problems.

F.2.1 TEST5.VS3, unobstructed views

This is one of the simplest view factor calculations. The geometry described in file TEST5.VS3 is a simple cube with a length of 4 on the edges.

```
T  TEST5:  Simple cube; No view obstructions
G  3
V 1    0.   0.   4.
V 2    0.   0.   0.
V 3    0.   4.   0.
V 4    0.   4.   4.
V 5    4.   4.   4.
V 6    4.   4.   0.
V 7    4.   0.   0.
V 8    4.   0.   4.
S 1    1    2    3    4    0    0    0   surface_1
S 2    5    6    7    8    0    0    0   surface_2
S 3    1    4    5    8    0    0    0   surface_3
S 4    8    7    2    1    0    0    0   surface_4
S 5    7    6    3    2    0    0    0   surface_5
S 6    5    4    3    6    0    0    0   surface_6
End of data
```

Using a division factor of 4, VLITE computed the following view factors:

```
row    1       2       3       4       5       6          Σ F_ij
 1  .000000 .199829 .200044 .200044 .200044 .200044    1.000005
 2  .199829 .000000 .200044 .200044 .200044 .200044    1.000005
 3  .200044 .200044 .000000 .200044 .199829 .200044    1.000004
 4  .200044 .200044 .200044 .000000 .200044 .199829    1.000005
 5  .200044 .200044 .199829 .200044 .000000 .200044    1.000005
 6  .200044 .200044 .200044 .199829 .200044 .000000    1.000004
```

Hottel [1967, p 59] reports the following analytic value for the view factor between adjoining surfaces in a cube: $F_{adj} = 0.20004376$. Therefore, the view factor between opposite surfaces should be $F_{opp} = 1 - 4 F_{adj} = 0.19982496$. VLITE comes extremely close to these values.

F.2.2 TEST6.VS3, obstructed view

This is one of the simplest obstructed view factor calculations. The geometry described in file TEST6.VS3 is similar to that shown in figure D.11.

```
T  TEST6:  4 surface test case for simple view obstruction.
G  3
! SURFACE 1: 4 X 4 SQUARE
V 1     0.   0.   4.
V 2     0.   0.   0.
V 3     0.   4.   0.
V 4     0.   4.   4.
! SURFACE 2: 4 X 4 SQUARE
V 5     4.   4.   4.
V 6     4.   4.   0.
V 7     4.   0.   0.
V 8     4.   0.   4.
! SURFACE 3/4: 1 X 1 SQUARE
V 9     3.   1.   1.
V 10    3.   3.   1.
V 11    3.   3.   3.
V 12    3.   1.   3.
S 1     1    2    3    4   0   0   0   surface_1
S 2     5    6    7    8   0   0   0   surface_2
S 3     9   10   11   12   0   0   0   obstruct_3
S 4    10    9   12   11   0   0   0   obstruct_4
END
```

By running a series of tests with different division factors, you can duplicate the data presented in figure D.12. The line on the output file labeled NDms indicates how many edge divisions were used to compute the obstructed view factor. You can change the number of edge divisions only by changing the "Division Factor", and these two quantities are not directly related. The analytic solution is $F_{1,2} = 0.115621$.

Division Factor	NDms	$F_{1,2}$
1.0	2	0.112347
2.0	2	0.112347
3.0	3	0.114651
4.0	3	0.114651
4.3	4	0.115082
5.0	5	0.115279
6.0	6	0.115380
7.0	7	0.115444
8.0	9	0.115514
9.0	11	0.115549

F.2.3 TEST7.VS2, 106 Surface Test

TEST7.VS2 uses a fairly simple configuration consisting of a 3 $\times$ 4 $\times$ 5 box (composed entirely of 1 $\times$ 1 squares) which encloses two unit cubes. Therefore, there are 94 surfaces on the enclosure and 12 surfaces which can obstruct views. This test exists to establish a benchmark execution time for a non-trivial three dimensional problem. The values of the view factors are not of concern. The data file is not listed here to save space.

Typical execution times:

machine	view factors	normalize	include ϵ
33 MHz 80486 PC-compatible	28.85 s	0.27 s	2.64 s
5 MHz 8088/87 PC-compatible	1294 s	21 s	290 s

F.3 Post-Processing Tests

The use of normalization has been demonstrated in the prior tests. The following tests demonstrate the use of combined and included surfaces. 2-D geometries are used in order to have simple drawings, but these features are most useful for 3-D geometries. Two input files have been created to represent identical geometries. In TEST8, 16 individual surfaces, indicated in italics in Figure F5, are combined into 10 surfaces, indicated by normal digits. In TEST9, 10 surfaces are described with some overlapping, or being "included" in, others.

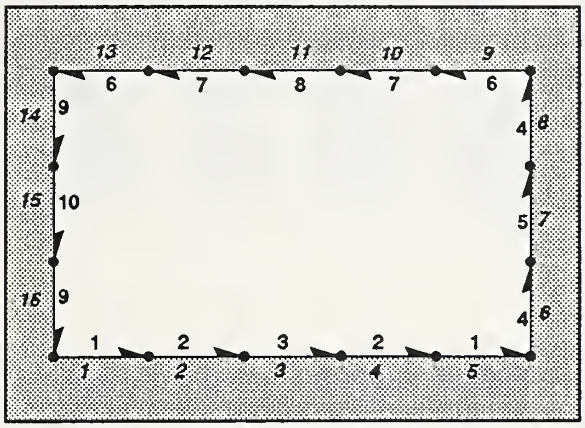


Figure F5. Combined Surfaces

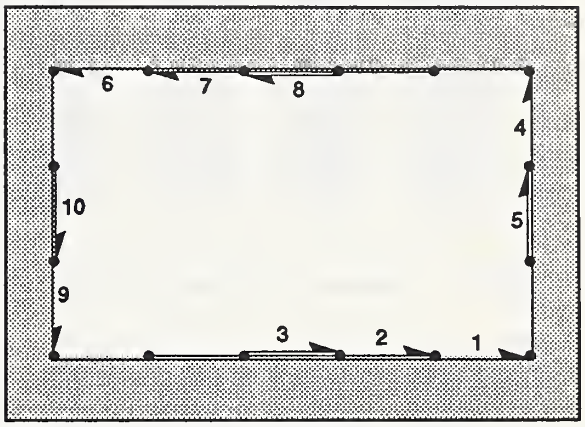


Figure F6. Included Surfaces

```
T  TEST8:  Combine surfaces; Match TEST9
G  2
V    1    .00   .00
V    2   1.00   .00
V    3   2.00   .00
V    4   3.00   .00
V    5   4.00   .00
V    6   5.00   .00
V    7   5.00  1.00
V    8   5.00  2.00
V    9   5.00  3.00
V   10   4.00  3.00
V   11   3.00  3.00
V   12   2.00  3.00
V   13   1.00  3.00
V   14    .00  3.00
V   15    .00  2.00
V   16    .00  1.00
S    1    1   2   0   0   0   surface_1
S    2    2   3   0   0   0   surface_2
S    3    3   4   0   0   0   surface_3
S    4    4   5   0   0   2   surface
S    5    5   6   0   0   1   surface
S    6    6   7   0   0   0   surface_4
S    7    7   8   0   0   0   surface_5
S    8    8   9   0   0   6   surface
S    9    9  10   0   0   0   surface_6
S   10   10  11   0   0   0   surface_7
S   11   11  12   0   0   0   surface_8
S   12   12  13   0   0  10   surface
S   13   13  14   0   0   9   surface
S   14   14  15   0   0   0   surface_9
S   15   15  16   0   0   0   surface_10
S   16   16   1   0   0  14   surface
End of data
```

```
T  TEST9:  Include surfaces; Match TEST8
G  2
V    1    .00   .00
V    2   1.00   .00
V    3   2.00   .00
V    4   3.00   .00
V    5   4.00   .00
V    6   5.00   .00
V    7   5.00  1.00
V    8   5.00  2.00
V    9   5.00  3.00
V   10   4.00  3.00
V   11   3.00  3.00
V   12   2.00  3.00
V   13   1.00  3.00
V   14    .00  3.00
V   15    .00  2.00
V   16    .00  1.00
S    1    1   6   0   0   0   surface_1
S    2    2   5   0   1   0   surface_2
S    3    3   4   0   2   0   surface_3
S    4    6   9   0   0   0   surface_4
S    5    7   8   0   4   0   surface_5
S    6    9  14   0   0   0   surface_6
S    7   10  13   0   6   0   surface_7
S    8   11  12   0   7   0   surface_8
S    9   14   1   0   0   0   surface_9
S   10   15  16   0   9   0   surface_10
End of data
```

The key point is that a complicated surface can be described by using combined or included surfaces. Since these two tests describe the same geometry, the view factors after post-processing should be identical. The results of the tests, printed below, indicate this is true except for some minor round-off differences.

```
 TEST8:  Combine surfaces; Match TEST9

row   1       2       3       4       5       6       7       8       9      10
 1 .000000 .000000 .000000 .191435 .060258 .199074 .200633 .096908 .191435 .060258
 2 .000000 .000000 .000000 .118907 .080935 .200633 .259186 .140498 .118907 .080935
 3 .000000 .000000 .000000 .106912 .074543 .193816 .280996 .162278 .106912 .074543
 4 .191435 .118907 .053456 .000000 .000000 .191435 .118907 .053456 .178840 .093563
 5 .120515 .161869 .074543 .000000 .000000 .120515 .161869 .074543 .187126 .099019
 6 .199074 .200633 .096908 .191435 .060258 .000000 .000000 .000000 .191435 .060258
 7 .200633 .259186 .140498 .118907 .080935 .000000 .000000 .000000 .118907 .080935
 8 .193816 .280996 .162278 .106912 .074543 .000000 .000000 .000000 .106912 .074543
 9 .191435 .118907 .053456 .178840 .093563 .191435 .118907 .053456 .000000 .000000
10 .120515 .161869 .074543 .187126 .099019 .120515 .161869 .074543 .000000 .000000
```

```
 TEST9:  Include surfaces; Match TEST8

row   1       2       3       4       5       6       7       8       9      10
 1 .000000 .000000 .000000 .191435 .060258 .199074 .200633 .096908 .191435 .060258
 2 .000000 .000000 .000000 .118907 .080934 .200633 .259185 .140498 .118907 .080935
 3 .000000 .000000 .000000 .106912 .074543 .193816 .280996 .162278 .106913 .074543
 4 .191435 .118907 .053456 .000000 .000000 .191435 .118907 .053456 .178840 .093563
 5 .120516 .161869 .074543 .000000 .000000 .120515 .161869 .074543 .187126 .099020
 6 .199074 .200633 .096908 .191435 .060258 .000000 .000000 .000000 .191435 .060258
 7 .200633 .259185 .140498 .118907 .080935 .000000 .000000 .000000 .118907 .080934
 8 .193816 .280996 .162278 .106913 .074543 .000000 .000000 .000000 .106912 .074543
 9 .191435 .118907 .053456 .178840 .093563 .191435 .118907 .053456 .000000 .000000
10 .120515 .161870 .074543 .187126 .099020 .120516 .161869 .074543 .000000 .000000
```

TEST10.VS2 and TEST11.VS2 do similar tests for a larger number of surfaces and a slightly more complex geometry.