MODULARIZATION AND SIMULATION TECHNIQUES FOR HEAT BALANCE BASED ENERGY AND LOAD CALCULATION PROGRAMS: THE EXPERIENCE OF THE ASHRAE LOADS TOOLKIT AND ENERGYPLUS

Richard K. Strand, School of Architecture, University of Illinois at Urbana-Champaign, Champaign IL, USA
Curtis O. Pedersen, Dept. of Mechanical and Industrial Engineering, University of Illinois at Urbana-Champaign, Urbana IL, USA
Drury B. Crawley, US Department of Energy, Washington DC, USA

ABSTRACT

Through sponsorship of the Loads Toolkit and coming changes to the Handbook of Fundamentals, ASHRAE has taken the lead in promoting a heat balance based approach as the "preferred" method for thermal load and energy analysis calculations. Building on previous ASHRAE research and to some extent the BLAST (Building Loads Analysis and System Thermodynamics) program, one of the goals of the Loads Toolkit research project is to obtain a heat balance based load calculation procedure that is relatively simple in structure where various algorithms, such as different exterior convection coefficient calculation techniques among many others, can be hooked into the heat balance without any restructuring. One of the keys to achieving this goal is the adaptation of legacy versions of a heat balance based approach a nd their modularization using a modern programming language such as FORTRAN90. This process was not a trivial task, and the insight gained in this re-engineering process in a small-scale (single zone) environment provided ideas for modularizing a larger-scale (multiple zone) program such as EnergyPlus. This paper gives an overview of the challenges faced in modularizing the heat balance algorithms in both the Loads Toolkit and EnergyPlus. In addition, it provides an analysis of the resulting heat balance routines in each project and suggestions for the developers of other simulation programs as well as those interested in working with the Loads Toolkit and EnergyPlus.

INTRODUCTION

During the late 1990s, ASHRAE Technical Committee 4.7 (Energy Calculations) set out to complete its effort to provide the simulation community with calculation algorithms that could be used to model various aspects of buildings from its primary and secondary systems to its envelope. The last component in the toolkit series was the loads calculation toolkit. The two key features of the Loads Toolkit (RP-987) were the modular nature of the programming as was the case with the previous primary and secondary equipment toolkits and the reliance on strict heat balance procedures. The requirement for a heat balance based algorithm was based on the wide agreement within the simulation community that a heat balance approach should, in theory, be the most accurate with the fewest built-in assumptions. While in the past, the heat balance procedures have been less highly regarded in part due to their slower run-times, the current computing power and speed has risen significantly in the past decade, and the concerns about execution times have been overcome for the most part.

During the period when the research goals and work statements were being finalized in ASHRAE Technical Committee 4.7, ASHRAE Technical Committee 4.1 (Load Calculation Data and Procedures) was completing a revision of the Cooling Load Manual that was also based on the heat balance approach. This work was part of RP-875 (Pedersen 1997) that also included sample software demonstrating an implementation of a heat balance-based load calculation algorithm. In the eyes of both Technical Committee 4.1 and 4.7, the goal of Cooling Load Manual and the Loads Toolkit was the widest possible dissemination of the project results and maximum re-use of the information and/or algorithms by the simulation public. The similar goals led to the use of RP-875 (Pedersen 1998) programming as a starting point for RP-987 (Loads Toolkit).

In the big picture, the EnergyPlus project (Crawley 2001) and its load calculation engine have very similar goals to RP-875 and RP-987. The EnergyPlus project also began in the late 1990s as an effort to combine the best features of BLAST and DOE-2, two popular energy analysis programs used internationally, into a single program. The goal was to provide the simulation community with a single platform for model development, saving costs and time in the never-ending process of keeping the programs up to speed with the rapid changes in the HVAC industry and maximizing the impact a new program might have on the architectural and engineering community. In addition, the development team identified algorithm accuracy and program enhancements by a wide variety of simulation experts as essential for the success of the project. All of these goals led the team to pursue a modularized heat balance approach. The heat balance approach would provide the most fundamentally based foundation to build a program while a modular version of the heat balance would allow more rapid modifications to the program by researchers both inside and outside the original program development team.

So, the nature of the Loads Toolkit and the development of the EnergyPlus program both require a modular programming structure that will facilitate ease of understanding by potential users and more efficient modification and enhancement by developers. However, despite having the same "big picture" goals, the stark differences in the data structure and the specific goals of the two projects resulted in significantly difference formulations of a modular heat balance algorithm. These differences in themselves and the path taken to arrive at the two algorithms are interesting from a building simulation standpoint as well as from a generic standpoint of other types of simulation programs.

Most of the dilemma is created by the tension between the needs of smaller, limited audience projects such as the Loads Toolkit and larger, more generic programs such as EnergyPlus. This is most visibly seen in the classical data versus algorithm conflict that plays slightly different roles in the Loads Toolkit and EnergyPlus. In the Loads Toolkit with its more limited amount of data due to its single zone nature and its somewhat limited features, the amount of data sharing is kept to a minimum. This results in a cleaner modular structure and a simpler heat balance algorithms. In EnergyPlus with more data and more features, the amount of sharing that is required increases by several orders of magnitude, placing much larger demands on program data and requiring more complex data structures.

The remainder of this paper begins with an introduction to the heat balance approach. Using this definition as a basis, the concept of implementing this approach within the framework of single zone and multiple zone simulations is addressed. While the programs and simulation environments mentioned in this paper are clearly not the only applications of heat balance based energy and loads calculations for buildings, they do provide a view into the inner workings of a method that is never trivial to bring to a solution.

HEAT BALANCE APPROACH

The heat balance approach is an attempt to capture the fundamental thermal physics of a building envelope by applying the First Law of Thermodynamics (conservation of energy) to important points within the building geometry. The most classical formulation of the heat balance approach applies a control volume at the outside face of every building surface, at the inside face of every building surface, and around the inside air of each thermal zone defined within the building. This can be seen graphically in Figures 1 through 3.

In the outside surface heat balance, the four thermal "forces" acting on the control volume at the outer surface of each wall as shown in Figure 1 must be balanced for energy to be conserved. This assumes that the control volume itself has no mass and thus no ability to store energy. Mathematically speaking, we can formulate the following equation from this diagram:

$$Q_{SWrad} + Q_{LWrad} + Q_{conv} + Q_{cond} = 0$$

where:

Q_{SWrad} is the amount of solar radiation absorbed on the surface,

Q_{LWrad} is the amount of thermal radiation exchanged between the surface and its surroundings (including the ground, sky, air, other buildings, vegetation, etc.),

Q_{conv} is the amount of convection between the surface and the surrounding air, and

Q_{cond} is the amount of energy conducted into the wall materials.

Further details on the various components of the outside surface heat balance are available in the literature (McClellan 1997). Given known environmental conditions, the above equation breaks down into a single equation (or series of equations for multiple surfaces—one equation per surface) and two unknowns: the outside and inside surface temperatures. Both of these terms are also unknowns in the inside surface heat balance equation.

In the inside surface heat balance, the six thermal "forces" acting on the control volume at the inside surface of each wall as shown in Figure 2 must be balanced for energy to be conserved. Again, this assumes that the control volume itself has no mass and thus no ability to store energy. Mathematically speaking, we can formulate the following equation from this diagram:

$$Q_{solar} + Q_{SWlights} + Q_{LWradExch}$$
$$+ Q_{LWradIntGains} + Q_{conv} + Q_{cond} = 0$$

where:

Q_{solar} is the amount of solar radiation absorbed on the inside face of the surface,

$Q_{SWlights}$ is the amount of short wavelength radiation from lights that is absorbed by the surface,

$Q_{LWradExch}$ is the amount of net long wavelength (thermal) radiation that is exchanged with other surface in the zone,

$Q_{LWradIntGains}$ is the amount of long wavelength (thermal) radiation from internal heat gains such as people, lights, and equipment that is absorbed by the surface,

Q_{conv} is the amount of convection between the surface and the air in the zone, and

Q_{cond} is the amount of energy conducted into the wall materials.

Further details on the various components of the inside surface heat balance are available in the literature (Liesen 1997). As with the outside surface heat balance, each inside surface heat balance is a single equation (or series of equations) with the inside and outside surface temperatures as the unknowns. It should be noted that there is one other term in the inside heat balance that may also be unknown: the zone air temperature. In some cases, the zone air temperature is a known setpoint temperature. In situations where a conditioning system is not present, not operating, out of capacity, or not controlling air temperature directly (as in a radiant system), the zone air temperature may be "floating". In this case, the zone air heat balance must be solved to obtain the zone air temperature.

The zone air heat balance has two possible formulations depending on whether or not the storage of energy in the air itself is taken into account. From Figure 3, it can be seen that in the absence of accounting for energy storage in the zone air, a heat balance equation would be similar to the following:

$$Q_{conv} + Q_{convIntGains} + Q_{infil} + Q_{sys} = 0$$

where:

Q_{conv} is the amount of convection between the all of the surfaces in the zone and the zone air,

$Q_{convIntGain}$ is the amount of heat convected from internal gains such as people, lights, and equipment,

Q_{infil} is the amount of heat gained or lost due to infiltration, and

Q_{sys} is the amount of heat added to or subtracted from the space due to a space conditioning system.

For most cases, the quasi-steady equation shown above is adequate for solving the zone air heat balance. However, when one attempts to integrate the heat balance solution with a primary and secondary equipment simulation to allow feedback between the equipment and the heat balance, the mass of the zone air actually helps to stabilize the solution. In that case, rather than setting the four "driving forces" to zero, one accounts for the energy storage capacity of the air with the addition of one term:

$$C\frac{dT}{dt} = Q_{conv} + Q_{convIntGains} + Q_{infil} + Q_{sys}$$

where C is the product of the mass of the zone air and the specific heat of air. Used in conjunction with a third order finite difference approximation for the derivative term and time steps between 0.1 and 0.25 hours, this method of calculating the zone air heat balance has been shown to be stable (Taylor 1991).

Thus, the basic heat balance approach is to set up energy balance equations using control volumes at the inside and outside faces of each surface in a particular zone as well as a control volume around the zone air. For a zone with N surfaces, this produces a system of 2N+1 equations with 2N+1 unknowns. While a solution can be found for this system of equations, the actual solution process is not trivial and generally involves either an iterative scheme or a more complex solution algorithm. Some of these details will be discussed in the next sections.

SINGLE ZONE IMPLEMENTATION

In part, the ASHRAE RP-875 project began with the process of taking the heat balance concepts from the BLAST (BLAST Support Office 1995) heat balance engine that has been shown to be a robust and capable simulation algorithm for over two decades. A new program was then written whose scope was limited to a single zone with 12 well-defined surfaces (4 walls, ceiling/roof, floor, internal mass, and windows for the walls and ceiling/roof) for a single simulation day. In the process of creating this "new" program which was called HBFort, the research team broke the heat balance algorithm into elements that defined distinct categories of environmental effects (such as convection, sky models, etc.) and then hooked them together again using the heat balance approach outlined in the preceding section. The hope was that the newly organized structure would allow for swapping between algorithms (such as selecting alternate sky models in successive runs) easily. While the program that resulted was successful in fulfilling its goals, the team realized during the project that there were many lessons that were learned that could benefit future projects.

One area in which problems were noted was during the "testing" phase. Substantial work had been done in the past to verify the BLAST program, but this work was not valid for HBFort. As a result, a significant amount of work was invested to compare the results between BLAST and HBFort. This process was time consuming and tedious, even considering the substantial amount of run automation that was already in existence (Strand 1996). Reflections on these problems were responsible in part for some of the software development procedures used in both the ASHRAE Loads Toolkit and EnergyPlus.

Another area where the team realized there was room for improvement in HBFort was modularity and data structure. Algorithms were separated into distinct "modules" in HBFort. In fact, the heat balance routines were relatively clean in comparison to BLAST in that very few calculations existed within the heat balance routines other than the surface temperature equations themselves. Most of the actual calculations were done by calling the appropriate subroutine for the various components of the heat balance. The advantage to this structure was the ability to easily switch between different component calculations algorithms. For example, selecting between the BLAST and the Brown sky models was now simply a matter of making the appropriate subroutine call.

However, despite the clearer separation of algorithms, data structures were generally declared as "public" to all modules or were available through the "USE" statements that are required in FORTRAN90 to grant access from one module to another. While this did not necessarily present a problem for a limited scope program such as HBFort, such a data structure was not modular and would not work for either the ASHRAE Loads Toolkit or EnergyPlus.

Finally, HBFort also made the research team aware of some of the struggles that the original developers of BLAST experienced in trying to get a heat balance solution to converge for all cases. In BLAST, a modified successive substitution (also known as Gauss-Seidel) approach was used to achieve simulation convergence. The modifications to this process include special versions of the outside heat balance equations for surfaces that reacted quickly to environmental changes (very low or no thermal mass/capacitance constructions) or surfaces that reacted more slowly. Within the successive substitution solution, iterations occurred mainly in the inside heat balance. Temperatures at the inside surface were assumed to be converged when the maximum change in any inside surface temperature from the previous iteration to the current iteration was less than 0.01°C.

In HBFort, the iteration scheme was modified slightly to avoid the necessity of multiple forms of the outside heat balance equation. To resolve this issue, additional iterations were allowed between the inside and outside surface heat balances. The basic form of the iterative scheme is:

```
Day Iteration Loop
   Hour Loop (1 to 24)
      Surface Iteration Loop
         Outside Surface Heat Bal
         Inside Surface Heat Bal
      End Surface Iteration Loop
      Zone Air Heat Balance
   End Hour Loop
End Day Iteration Loop
```

The surface heat balances successively run through each surface contained in the user input model. The number of surface iteration loops and the number of day loops were considered user input rather than fixed. As a result, stability was not necessarily guaranteed and convergence to a steady periodic solution was also not assured. Nevertheless, using reasonable input values, fairly accurate simulation results could be obtained for a single zone over one simulation day in a very short amount of time (typically around 1 second execution time) for fixed one-hour time steps.

While the purpose of the ASHRAE Loads Toolkit (RP-987) was different from the HBFort program (RP-875), HBFort did serve as a better starting point for the development of the ASHRAE Loads Toolkit. This was because the Toolkit was intended to be a collection of modules that could be used by other researchers rather than a full-featured program. In any case, the RP-987 research team used the successive substitution algorithm and structure of HBFort as the framework for testing the Toolkit modules.

The Toolkit development process used a modified "evolutionary reengineering" (ER) process that was developed for EnergyPlus. In ER, changes are made incrementally so that there is always a working version of the code. As changes are made, comparisons with the results of previous versions are made to avoid the introduction of errors that are possible when code is recreated or one "starts over".

The main improvement in the Toolkit was the improved modularity of the code over the parts of HBFort. Whereas HBFort contained data modules (a modern version of common blocks) that allowed the sharing of data between program groups, Toolkit eliminated these structures because of the requirement that the Toolkit merely be a collection of modeling modules rather than an actual program. Thus, the design of the Toolkit was different, and some ER was used to achieve the desired modular structure in both data and algorithms.

The resulting Toolkit code, while bearing some resemblance to the HBFort algorithms, is significantly different in data structure. Whereas HBFort relied on data modules, the Toolkit passes data as parameters in the subroutine call statements. Any "global" data required in individual modules is read in since each must be a self-contained unit that can be compiled and used separately (or integrated with other simulation packages). An example of a property that was required by more than a single module is the facing angle of a surface that is required by solar routines to determine the amount of solar radiation incident on the surface, convection routines to determine along with wind speed and direction the amount of convection to/from the surface from/to the outside air, etc.

The successful modularization of the heat balance technique in the Toolkit also led to the ability to more easily investigate solution techniques other than the standard successive substitution algorithm that had been used by BLAST and HBFort. Toolkit also investigated a Newton-Raphson and "steady periodic" scheme to solving the heat balance equations for a 24 hour period (Asmundsson 2000).

In the Newton-Raphson solution method, the iterative loops are similar to the ones shown for the successive substitution approach shown above except that inside the hour loop the Newton-Raphson scheme iterates among all of the heat balances including the zone air heat balance. This algorithm converges quadratically provided the initial guess for the system is "good enough". This however can be somewhat problematic for both the successive substitution and the Newton-Raphson scheme and is still a matter of study in the various heat balance formulations.

In the Toolkit configuration, there was the ability to experiment somewhat with different starting points. One way to provide both the successive substitution and Newton-Raphson schemes with a valid starting point was to solve the system of equations using a steady periodic solution procedure (Asmundsson 2000). In the steady periodic approach, the order of the simulation loops is reorganized. This is based on the fact that if one is looking solely for the steady periodic solution (the same day repeated over and over again) one can actually solve the system for each surface over 24 hours and then iterate among the surfaces rather than iterating over the surfaces every hour. The simulation loops in the steady periodic solution are then:

```
Convergence Loop
   Surface Loop
      Solve Heat Bals over 24 hours
   End Surface Iteration Loop
End Convergence Loop
```

No iteration is required among the heat balance routines in this case because the interconnection between these equations (linked via the conduction transfer function or response factor terms) can be solved algebraically. The use of the steady periodic solution to establish a starting point for other solution schemes that will allow the simulation of more than a single day is made possible by the modular nature of the Toolkit as well as the one zone limitation of the Toolkit heat balance formulation.

The modular nature of the Toolkit also allowed the investigation of different solution models for various terms in the heat balance such as convection models, sky models, radiant exchange algorithms, etc. Asmundsson (2000) investigated the effect of interior radiant exchange schemes on solution convergence speed and found that the speed of the solution was somewhat dependent on the radiant exchange algorithm in particular and that the more sophisticated Newton-Raphson approach was not necessarily quicker than the successive substitution strategy.

The simpler, less cumbersome nature of simple one zone heat balance formulations allows a variety of studies to be done including the modularization of code, investigation of various solution/iteration procedures, and trial of different component models such as radiant exchange algorithms. While this is convenient and provides good insight into how to apply various new strategies to more complex programs, multiple zone programs such as EnergyPlus present their own set of unique challenges that must be addressed.

MULTIPLE ZONE IMPLEMENTATION

Despite the differences between single zone programs such as HBFort and multiple zone programs such as

EnergyPlus, the origins of the two programs are the same—both programs stem from the BLAST program. HBFort was derived directly from the BLAST heat balance routines. EnergyPlus used IBLAST, a research version of the BLAST program that integrated loads, systems, and plant simulations and allowed for sub-hourly time steps (Taylor 1991), as a starting point. The heat balance portion of IBLAST was nearly identical to BLAST so both HBFort (and thus the Loads Toolkit) and EnergyPlus have the same "DNA" from the perspective of programming code.

Unfortunately, that is where most of the similarities end. While some of the pieces and component models within the heat balance are comparable, the data structure and the sheer volume of data are vastly different. In the Toolkit, one could easily pass data as parameters from one module to another and also re-read the same data into different modules without negative side effects. The same cannot be said for EnergyPlus. Passing data as parameters can get tedious when the modules gets several levels deep (as opposed to only one level deep in the Toolkit). In addition, reading the same data into several different places in the program can be problematic. The Toolkit will remain fairly static in its input structure, but EnergyPlus could be required to alter its input data structure somewhat in the future. This would require extensive knowledge of all of the locations where a particular piece of data was being read into the program.

As a result of the many layers in a full-featured simulation program and the desire to avoid multiple reads of the same input data structures, EnergyPlus is more like HBFort than Toolkit when one views the internal data structure. One key difference between the data structure of EnergyPlus and HBFort, however, is that the EnergyPlus code is very careful to encapsulate data and limit access to it. Global access to all data, while appearing to be convenient at first glance, can become dangerous in the long run. Any data that is accessible can also be modified at any point without restriction. This can lead to major problems and quite unexpected results. As a result, the goal for EnergyPlus code was to keep data available only to local modules as much as possible and to reluctantly allow a variable to "graduate" from being a private variable within a subroutine to being a private variable within a module. After that, a variable was allowed to be public as part of a data module if warranted and generally only to avoid multiple reads of input data or transferring large amounts of data via parameters through multiple module layers.

With these goals in mind, the process of taking the IBLAST heat balance code and transforming it into the more modular and significantly more understandable and readable EnergyPlus code followed the "evolutionary reengineering" (ER) process mentioned briefly in the previous section. ER is about making small improvements and changes to the code and insuring that the changes have not caused any significant shifts in the output. The reasons for this strategy include the desire to eliminate human errors that can easily leak into the code during any modification process and the desire to keep the level of confidence in the program results as high as previous version that had been the subject of extensive verification studies. The logic is that if the old and the new version produce effectively the same results and the old version was "tested", then the results of the new version were "verified".

The ER process began with the baseline code and a baseline set of input files that served as a "testing suite". As changes were made to make the program more modular, the entire test suite was exercised and the results compared to the last version. Careful attention was made not only to small changes between versions but also to any pronounced "drift" over a series of changes. This helped increase the confidence of the research team in the modularization process. The main focus of the ER process was modularization without significant changes to the algorithms. One simulation detail that remained the same throughout the procedure was the successive substitution method as well as the iteration scheme, all of which are substantially the same as in BLAST.

In some cases, progress was literally made on a "line-by-line" basis to root out potential errors. In other cases, problems relating to original variable definitions (real vs. double precision) or to the choice of compiler were exposed. In one classic situation, the order of a single calculation (changing say X = A + B to X = B + A) actually ended up producing slightly different answers.

The most interesting dilemma encountered during the modularization process using ER relates to the choice of internal radiant algorithms. Asmundsson (2000) found that, using the Toolkit code, it was possible to apply exact radiant exchange principles using approximate view factors rather than the MRT method (Walton 1980) and receive more accurate results without any significant execution speed penalty. However, when switching to this more exact radiant exchange formulation (Hottel 1967) from the MRT based method in EnergyPlus, some significant stability problems appeared. This problem was linked to thermal initial conditions of the simulation. It appears that the MRT based method smoothed out any large changes that occurred while the simulation was just starting and that this caused the instability with the exact interior radiant exchange formulation.

Numerous methods for resolving this instability were tried within EnergyPlus so that the more accurate radiant exchange algorithm could be kept. In the end, the simulation technique that proved successful in maintaining stability was including a damping factor in the inside heat balance equations. This term effectively slowed down the changes in temperature between iteration with the damping term being reduced to zero when the solution converged. It is not clear how much the time step (which can be defined by the user for between 1 and 6 time steps per hour) in EnergyPlus affects the noted instability. Thus, this is an area for further research.

CONCLUSIONS

With the acceptance of the heat balance approach to calculating thermal load in buildings, the interest in the application of heat balance based procedures is certain to grow. The ASHRAE Loads Toolkit provides researchers with a set of tools to construct their own heat balance based simulation programs while EnergyPlus provides researchers with a programming framework that can be enhanced to include modules for new technologies and improved calculational algorithms. These two projects, as described in the previous sections, show that while heat balance based procedures may appear simple in concept that their application in actual code can result in some interesting problems that require creative solutions. However, some of the experiences of these two projects and concepts such as evolutionary reengineering provide evidence that heat balance based procedures can be modularized and successfully implemented.

ACKNOWLEDGEMENTS

The authors of this paper wish to acknowledge the American Society of Heating, Refrigerating, and Air-Conditioning Engineers for their support of RP-875 and RP-987 and the United States Department of Energy for their support and direction of the EnergyPlus project. In addition, the authors would like to thank the following members of their research team who significantly contributed to one, two, or all three of the projects highlighted in this paper: Dan Fisher, Jakob Asmundsson, Rich Liesen, Linda Lawrie, and Fred Winkelmann.

REFERENCES

Asmundsson, J.M., „A study of solution techniques and process models for heat balance based thermal load calculations“, M.S. Thesis, Department of Mechanical and Industrial Engineering, University of Illinois at Urbana-Champaign, 2000.

BLAST Support Office, „BLAST User's Manual“, University of Illinois at Urbana-Champaign, 1995.

Crawley, D.B., L.K. Lawrie, F.C. Winkelmann, and C.O. Pedersen, „EnergyPlus: New Capabilities in a Whole-Building Energy Simulation Program“, International Building Performance Simulation Association, Conference Proceedings of Building Simulation 2001, Rio de Janeiro, 2001.

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

Liesen, R.J. and C.O. Pedersen, „An Evaluation of Inside Surface Heat Balance Models for Cooling Load Calculations“, ASHRAE Transactions, Volume 103, Part 2, 1997.

McClellan, T.M. and C.O. Pedersen, „Investigation of Outside Heat Balance Models for Use in a Heat Balance Cooling Load Calculation Procedure“, ASHRAE Transactions, Volume 103, Part 2, 1997.

Pedersen, C.O., D.E. Fisher, and R.J. Liesen, „Development of a Heat Balance Procedure for Cooling Loads“, ASHRAE Transactions, Volume 103, Part 2, 1997.

Pecersen, C. O., D.E. Fisher, J.D. Sp;itler, and R.J.Liesen, Cooling and Heating Load Calculation Principles, 1998, ASHRAE.

Strand, R.K., D.E. Fisher, P.C. Lindstrom, „Dry Bulb Economizer Comparison“, U.S. Army Construction Engineering Research Laboratories Project Final Report, Champaign, IL, 1996.

Taylor, R.D., C.O. Pedersen, D. Fisher, R. Liesen, and L. Lawrie, „Impact of simultaneous simulation of building and mechanical systems in heat balance based energy analysis programs on system response and control“, International Building Performance Simulation Association, Conference Proceedings of Building Simulation 1991, Nice, France, 1991.

Walton, G.N., „A New Algorithm for Radiant Interchange in Room Loads Calculations“, ASHRAE Transactions, Volume 86, Part 2, 1980.

Walton, G.N., „Thermal Analysis Research Program Reference Manual“, National Bureau of Standards, NBSSIR 83-2655, 1983.

FIGURES

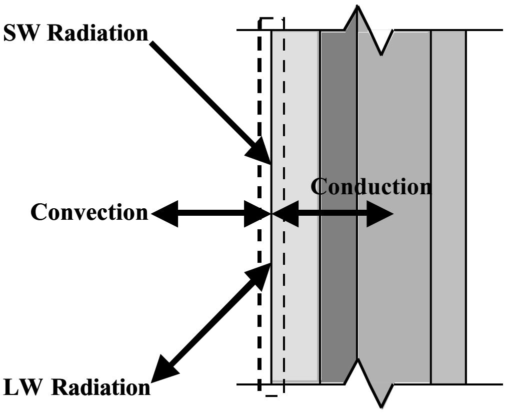

Figure 1. Outside Surface Heat Balance.

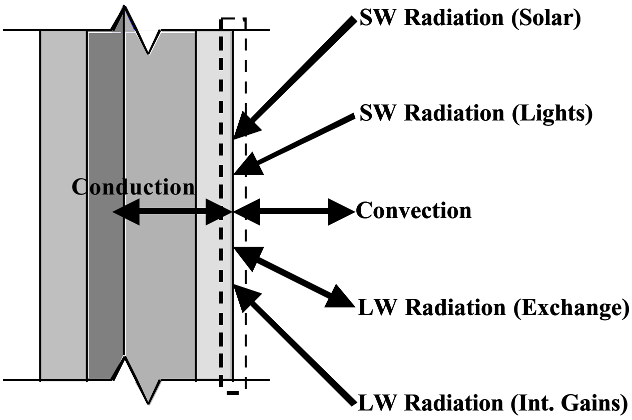

Figure 2. Inside Surface Heat Balance.

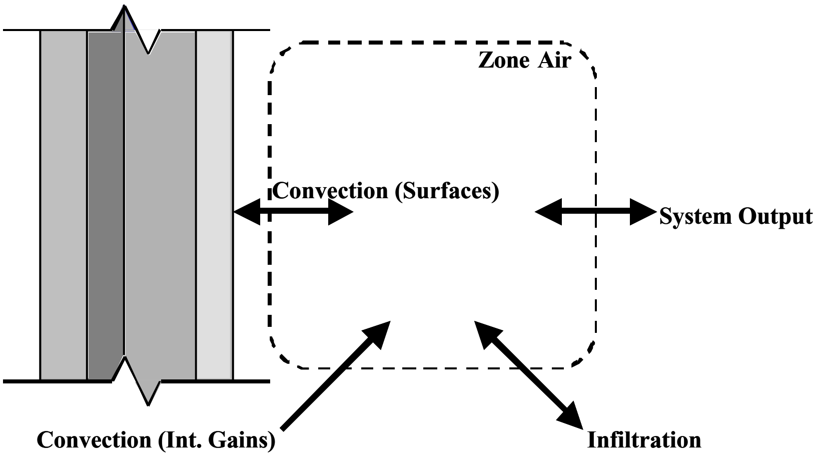

Figure 3. Zone Air Heat Balance.