DEVELOPMENT OF A VISUAL TOOL

FOR HVACSIM+

By

ADITYA VARANASI

Bachelor of Engineering

Osmania University

Hyderabad, India

1998

Submitted to the Faculty of the
Graduate College of the
Oklahoma State University
in partial fulfillment of
the requirements for
the Degree of
MASTER of SCIENCE
December, 2002

DEVELOPMENT OF A VISUAL TOOL

FOR HVACSIM+

Thesis Approved:

D. S. Fisher

Thesis Advisor

JD Spitler

Timothy J. Pettibone

Dean of the Graduate College

ACKNOWLEDGEMENTS

This thesis grew out of a series of dialogues with my advisor Dr. Dan Fisher and my committee member Dr. Ron Delahoussaye.

I am indebted to Dr. Fisher for all his support during slack times. He always showed me a way out. His comments on chapter drafts are themselves a course in critical thought upon which I will always draw.

Through his Socratic questioning, Dr. Delahoussaye brought me closer to the reality I had initially perceived, eventually enabling me to grasp its rich complexity.

I am grateful to my committee member Dr. Jeff Spitler who was at the center of our friendships, turning water into wine with his music and incredible imagination.

I am also thankful to the Federal Highway Administration (FHWA) for providing continuous support for the work.

Many thanks to Xioabing Liu, Ray Young and Mahadevan Ramamoorthy for providing decisive and energetic support all the way.

I am forever grateful to my parents whose foresight and values paved the way for a privileged education, and to my brother who gently offered unconditional support at each turn of the road.

TABLE OF CONTENTS

Chapter Page

Chapter Page

Chapter **Page**

Chapter **Page**

LIST OF TABLES

Table **Page**

Table **Page**

LIST OF FIGURES

Figure **Page**

Figure **Page**

Figure **Page**

Figure **Page**

NOMENCLATURE

n = Number of scalars in a vector

$J(.)$ = Jacobian matrix

x^{*} = Solution vector

x^{k} = Intermediate value of x

$|\cdot|$ = Absolute value

$\|,\|$ = Euclidean norm

e_{t} = Error tolerance in the equation solver

e_{r} = Relative error tolerance

e_{a} = Absolute error tolerance

t_{g} = Time interval

e_{sol_vec} = Error tolerance applied to a solution vector

$e_{residual\_vec}$ = Error tolerance applied to the residual vector

P = Pressure variable

$C.$ = Control signal

T = Temperature variables

W = Mass flow rate

Figure | Page

CHAPTER 1

INTRODUCTION AND LITERATURE REVIEW

1.1. Introduction

The quest for higher building energy efficiency in the world has encouraged engineers to focus on the relationship between design variables and energy performance. Assessment of building energy performance is fundamental in making decisions regarding energy-efficient design of buildings and in quantifying the impact of energy conservation measures. Evaluation of energy characteristics of buildings serves as a critical base for developing building energy standards and assessing their effectiveness.

Simulation of building heating and cooling systems has become a very useful tool in designing energy-efficient and cost-effective systems. However, as described by Spitler (2001), system simulation is an essential design tool for ground-source heat pump (GSHP) systems, particularly hybrid GSHP systems that incorporate additional heat rejection components, such as cooling towers, ponds, and heated pavement systems. It is fortuitous that the same simulation tools useful for designing building heating and cooling systems are also useful for designing similar thermal systems, such as "Smart Bridge" (Chiasson and Spitler 2001), which are not connected to buildings.

HVACSIM+, which stands for 'HVAC SIMulation PLUS other systems', is a non-proprietary simulation package developed at the National Institute of Standards and Technology (NIST), Gaithersburg, Maryland, U.S.A. It is capable of modeling HVAC (heating, ventilation and air-conditioning) systems, HVAC controls, the building model, and energy management systems. In HVACSIM+ both the specifications by the user and the internal representation of HVAC elements are represented in terms of individual components like fan, duct, heating coil, boiler, pump, pipe, etc., which are connected to form complete systems. This allows the users to develop new models and introduce them in the package to simulate them in various configurations. This capability has been extended by the development of component models for ground loop heat exchangers (Yavuzturk and Spitler 2000), heated pavements (Chiasson, et al. 2000a), water-to-water heat pumps (Jin and Spitler 2002, Ramamoorthy 2001), and ponds (Chiasson, et al. 2000b; Ramamoorthy, et al. 2001). The modular component based modeling environment, coupled with the available component models makes this an ideal tool for design of heated bridge deck systems. Unfortunately, the existing HVACSIM+ user interface makes the program unnecessarily difficult to use, even for experienced users.

1.2. Thesis objective and scope

This study aims at the development of a Visual Modeling Tool for HVACSIM+. The study is coupled with the implementation of new convergence metrics in the tool. The main objectives of this study can be summarized as follows:

- To design, implement and validate a new tool that provides a better user-interface for HVACSIM+.
- To design, implement and validate new convergence criteria in order to achieve improved convergence speed and minimized residuals in the outputs.

Chapter 2 of this thesis conducts a detailed review of HVACSIM+ and the various programs in the package. The main program, MODSIM, is explained in detail. The numerical methods used in the package are also discussed.

Chapter 3 introduces the problems in HVACSIM+ and discusses the solutions implemented to tackle the shortcomings of the user-interface and the convergence criteria in the equation solver. This chapter explains the design and implementation of the new convergence metrics. The performance evaluation results of the new criteria are also presented in this chapter.

Chapter 4 provides the technical overview of the user-interface in the visual tool. It discusses the architecture, file formats, configuration and features of the tool. It presents a detailed description of the source code structure in the visual tool.

Chapter 5 is intended to validate the visual tool by subjecting it to specific test cases. The examples given in the HVACSIM+ Users Guide are selected for the study. These example simulations are reproduced using the tool and the obtained results are compared with the actual output.

1.3. Background

Simulation programs, intended for research, have been around for many years but their development into usable tools has not kept pace with other software commonly used by engineers – spreadsheets, CAD, etc. The large gap between what is available in the market and what is actually used and comprehended by the design professionals suggests that implementation, rather than just technical capability, is key to increasing the use of simulation tools and, hence, energy efficiency.

Many building-energy analysis programs were developed in USA and Europe. There are more than two hundred programs available in USA and over a hundred programs in Europe and elsewhere. Table 1.3-1 presents a list of few programs currently being used.

Program	Reference Source	Country of use
APEC ESP-II	Wickham, 1985	USA
ASEAM	Ohadi, Meyer, and Pollington, 1989	USA
BESA	BESA, 1993	Canada
BLAST	BLAST, 1991	USA
BUNYIP	Moller and Wooldridge, 1985	Australia
DOE-2	Birdsall, et al., 1990; LBL, 1981	USA
EnergyPlus	Crawley, et al., 2000	USA
ESP-r	ESRU, 2000	UK
HVACSIM+	Clark, 1985	USA
SPARK	Buhl, et al., 1993	USA
TRACE 600	Trane Company, 1992a & b	USA
TRNSYS	TRNSYS, 1988	USA

Table 1.3-1. *List of few Building Energy Simulation tools*

These packages have different levels of detail and input requirements. They come from research institutions, equipment manufacturers and private consulting firms. Some of the programs are public domain programs and others are proprietary programs. Apart from energy analysis, many of them also allow for standard HVAC design load calculations.

Building-energy analysis programs have undergone a slow evolution since arrival nearly three decades ago. The simulation techniques are rapidly changing with decreasing cost and increasing flexibility of computer systems. In the 1970's, the cost of conducting an energy analysis study was high. Energy programs in those days were developed on mini- or mainframe computers and were inaccessible to the vast majority of potential users. In the 1980's and 1990's, introduction of microcomputers and emergence of microcomputer versions of the simulation programs made it more affordable and accessible to carry out detailed energy studies. Many of the current simulation programs are succeeding generations of some previous ones.

Considerable amount of literature is available regarding studies performed on existing energy analysis tools. Some of the important works are presented below.

1.4. HVACSIM+

Park, Clark and Kelly (1985) give an overview of the HVACSIM+ package. The architecture and the main programs in the package are discussed. The paper discusses the features of MODSIM and introduces the numerical methods implemented in it. The techniques to solve stiff differential equations and data interpolation are discussed. The paper also presents a detailed discussion of the building shell and zone models used in HVACSIM+. The discussion on HVACSIM+ in this paper includes the solving techniques used, convergence criteria and numerical instability of the methods used in the equation solver.

Hensen (1995) gives an overview and examples of various approaches to system simulation in buildings. Advantages and disadvantages of different methods are described. The author classifies the existing software tools in four levels, as LEVEL A-D. In the case of LEVEL A, design specification and representation of plant systems is purely conceptual and only the room processes are considered. In the case of LEVEL B, the specification by the user is in terms of real systems like variable-air-volume, constant-volume zone re-heat system, residential wet central heating, etc. Examples of simulation systems operating on this level are DOE-2 and BLAST. In the case of LEVEL C both the specification by the user and the internal representation is in terms of individual plant components like fan, duct, heating coil, boiler, pump, pipe, etc., which are connected to form complete systems. Two main approaches can be distinguished in terms of individual component models. They are the input-output based approach and the conservation equation based approach. The input-output based approach represents each part of the system as an equivalent input-output relationship. These are connected to comprise the whole system in such a way that the output from one component is fed into the next as an input. TRNSYS is a well known example implementing this approach. In the conservation equation based approach HVAC system modeling is achieved by a modular, component-wise approach, involving representation of parts of a system by discrete nodal schemes and by the derivation of energy and mass flow equation sets that are solved simultaneously for each time step. An example of a conservation equation based system is HVACSIM+. In the case of LEVEL D, the specification is in the same form as in LEVEL C but the internal representation is based on the category of the variables. ESP-r is an example of a LEVEL D software tool.

Sahlin (1996) reviews many energy analysis tools like TRNSYS, HVACSIM+, and SPARK and describes the design of a general-purpose simulation environment, IDA, and the Neutral Model Format (NMF), a program independent language for modeling of dynamical systems using differential-algebraic equations with discrete events. IDA and NMF are used to effectively develop special purpose GUI-based tools that (1) are easy to use for end users without simulation expertise, and that (2) have good prospects for long term maintenance and reuse. IDA and NMF also serve as a general modeling and development environment for the sophisticated user.

Yosuke, Xiangyang, and Nobuo (1999) describe the development of a configuration tool for HVACSIM+. The tool is based on Object Modeling Technique (OMT), which is used to analyze the information of a HVACSIM+ simulation. The purpose of the tool is to replace HVACGEN, the pre-processing program of HVACSIM+

and provide an integrated environment to run simulations. The software architecture consists of the system model framework and the user interface framework. These frameworks are object class groups for describing the design information, including connections between components and the hierarchy of components, and the screens which allow users to edit this information, respectively. Some of the aspects in which the work differs from the implementation strategies followed in this thesis are presented in the Table 1.4-1.

	Work done by Yosuke et al.	**Current thesis study**
Programming technique	Uses Object Oriented Programming technique which is difficult to understand and code.	Uses structured programming technique which is easy to understand and code.
Programming language	Smalltalk	Visual Basic
Description of Blocks	Incorporates description of a component connection diagram for each block. This procedure is difficult to maintain and understand and causes the users to work with a pool of diagrams.	Single diagram describes the system and displays all components and connections between them. This procedure is easy to maintain and understand.
Component models	New component models can be registered and existing ones can be edited.	Component models cannot be added or edited.

Table 1.4-1. Table showing the differences between the work done by Yosuke et al. and this thesis.

Sowell and Haves (2000) present a discussion on the efficiencies of methods employed in solution of building simulation models and provide means for benchmarking them. Several simulation packages such as HVACSIM+, TRNSYS and SPARK are compared with each other. Specifically, HVACSIM+ and SPARK are studied and comparison analysis is presented in terms of simulation execution times.

Ramamoorthy (2001) presents a discussion on two energy simulation packages, TRNSYS and HVACSIM+. The author discusses the structure of the two packages, their relative merits and demerits, and steps to convert component models from TRNSYS to HVACSIM+. According to the author, the simultaneous non-linear equation solver is supposed to obtain a self-consistent solution for each time step in HVACSIM+, as compared to a relatively simple successive substitution algorithm in TRNSYS. The user-interface of TRNSYS is better than HVACSIM+ making it easier to work with. TRNSYS is more suited for using hourly time steps, whereas HVACSIM+ is suited for both shorter and ionger time steps if the variable step algorithm in HVACSIM+ can be effectively exploited.

1.5. Simulation Methodology

A variety of mathematical algorithms are implemented in existing building energy analysis tools to solve systems of nonlinear algebraic equations. It is important to study them in order to develop efficient diagnostic tools, specifically, when applying these algorithms to non-traditional applications. A few selected works are presented below.

Powell (1970) describes his hybrid method, which combines the features of Newton and Steepest descent methods. The method is tested and techniques for the approximation of the Jacobian are suggested. Readers are referred to Chapter 2 for a detailed description of Powell's hybrid method.

Hiebert (1982) performed an evaluation of the existing mathematical software that solves systems of nonlinear equations. Eight FORTRAN applications were considered for the study: C05NAF, BRENT, HYBRD, NS01A, QN, SOSNLE, ZONE, and ZSYSTM. These programs implemented three different methods: Brown or Brent, quasi-Newton, and Powell's hybrid. The aim of the work was to evaluate the code written and to find the relative merits and demerits of the algorithms implemented. A defined set of test cases was used to evaluate and make conclusions regarding the suitability of the software for specific problems. The following observations were made by the author:

- The initial guess has little effect on the performance of the different codes.
- Poor function scaling has essentially no effect on the performance of the Brown and Brent codes. This is due to the fact that the functions values are supplied one at a time.
- The hybrid method has not necessarily been an improvement over the quasi-Newton method, especially when poor variable or function scaling is present.
- Work needs to be done on how to handle poor variable scaling.

Dennis and Schnabel (1983) explain various aspects of solving nonlinear system of equations in their book, titled 'Numerical Methods for Unconstrained Optimization and Nonlinear Equations'. The book is a compilation of works done by several pioneers in the field. It explains the fundamentals of the calculus behind solving nonlinear equations and describes the Newton's method and several of its variants in detail. In addition, Jacobian approximation methods such as the Secant method and the Broyden's method are discussed in detail. The work, designed to be a text book, mentions examples and gives brief performance analysis for each method discussed.

Shacham (1990) describes a variable order method for solving nonlinear algebraic equations. The variable order method is a modified form of the Improved Memory Method (IMM), which was earlier proposed by the same author. The IMM method uses continued fractions to pass an inverse interpolating polynomial through all the previously calculated points, to find a new estimate of the solution. The modified form allows the number of previously calculated points, which governs the order of the method, to be changed. The algorithms discussed in the paper focus on the implementation of the method, the approach to attain global convergence and the stopping criteria. The method was subjected to ninety test cases and solutions for most of them were obtained in less than twenty iterations, proving that it is robust and reliable. It was also observed that by selecting more previously calculated points, the rate of convergence of the method can be increased. However, as the order increases, the improvement in the solution is moderate.

Buzzi-Ferraris and Tronconi (1993) describe an improved convergence criterion in the solution of nonlinear algebraic equations. The inadequacies in the existing convergence criteria, which require that either the sum of squares or the weighted sum of squares of the residuals decreases after each iteration, motivate the development of the new criterion. The new criterion introduces scaling of residuals to overcome the problem in existing criteria that is related to the fact that, if the residuals have widely differing orders of magnitudes, they tend to appear in the convergence criterion with incorrect and unbalanced weights. The improved criterion determines convergence using a modified merit function which equals exactly to the squared distance between the current values of the unknown and the solution vector in the case of linear system of equations. The authors provide a geometrical interpretation of the criterion which shows that the method introduces natural scaling or weighting factors for the residuals in the objective function, although it cannot be proven to guarantee global convergence. In addition, the evaluation of the criterion is shown to be computationally inexpensive. The authors end their discussion recommending the implementation of the criterion in general-purpose equation solvers.

Paloschi (1994) describes a hybrid continuation algorithm to solve algebraic nonlinear equations that combines Newton or Quasi-Newton and Continuation methods in such a way that the continuation strategy will be used only when the other convergence criteria fail. The continuation strategy is used only as a last resort because it is computationally expensive compared to Quasi-Newton method. The author refers to several papers and indicates that the existing methods lack in robustness and proposes the use of a continuation or homotopy method that is specially designed to improve robustness. The strategy is governed by the following homotopy function

$$h(x,H) = H\,f(x) + (1 - H)\,g(x)$$

where $g(x)$ is a function such that $g(x) = 0$ has a known solution. It is selected as $g(x) = f(x) - f(x_k)$. H is the homotopy parameter that allows defining a family of functions where $h(x,H)$ is solved easily and $h(x,1) \equiv f(x)$. The equation $h(x,H) = 0$ defines a path $x(H)$ which can lead to the solution. Three hundred and twenty seven examples that cover many different problems and reflect different scaling conditions are used to the test the method. It is found that the algorithm solves almost every problem and is proved to be robust. Finally, the author compares the performance of the method with the HYBRD (MINPACK) code, which is the previous version of the routine SNSQ used in MODSIM, and finds out that the HYBRD code is capable of solving only two hundred test cases.

Schonwalder and Sans (1995) present a method for the steady state analysis and optimization of nonlinear autonomous electrical circuits having unknown final oscillation periods in their steady state response. The approach involves discretizing the response and solving the associated system of nonlinear algebraic equations. Most iterative algorithms prove useless since autonomous circuits have infinite solutions leading to difficulties in convergence. The authors discuss the use of two robust solution approaches, namely, the Newton's method and the Fast Simulated Diffusion (FSD) method. Since the Newton's method is known to be locally quadratically convergent, the authors also suggest the use of line search algorithms or trust region methods that help in maintaining the Newton's directions or the radii of regions around the solutions, which decide the maximum step lengths. Better performance can be attained from trust region methods since they provide multi-dimensional solution search. The motivation behind the use of FSD method can be attributed to the fact that many techniques reject any tentative steps that result in an increase in the associated objective function, thus, making it difficult for the techniques to escape from local minima. The FSD method generates two kinds of steps called the descendent steps, which help find local minima and random steps, which ensure that the whole design space may eventually be reached. Appropriately reducing the length of random steps to zero, using simulated annealing technique, ensures that the algorithm will converge. The authors finally present application examples to validate the proposed approach.

Spedicato (1997) investigates the performance of several Newton like methods, namely Newton's method, the ABS Huang method, the ABS row update method and six Quasi-Newton methods. Thirty one families of problems with dimensions n=10, 50, 100 and two starting points are selected to test the algorithms. Newton's method appears to be the best in terms of number of problems solved, followed closely by the ABS Huang method. In addition, Newton's method has the smallest non-convergence region. The performance of the Quasi-Newton can be summarized as follows:

- Broyden's method and Greenstadt's second method show very poor performance.

- Greenstadt's first method, Martinez' column method, nonlinear Huang's method perform similarly.
- Thomas' method appears to be marginally more robust and fast and provides a better approximation to the Jacobian. In addition, the Thomas' method shows the smallest non-convergence region.

MacMillan (1999), in his doctoral thesis, suggests a relaxed convergence criterion to improve the convergence rate. The existing convergence criteria are compared to find out the reasons for their failure. It is shown that relaxing the conditions required to prove global convergence can improve the performance of an algorithm. It is also shown that minimizing an estimate of the distance to the minimum relaxes the convergence conditions in such a way as to improve an algorithm's convergence rate. A new line-search algorithm based on these ideas is presented that does not force a reduction in the objective function at each iteration, yet it allows the objective function to increase during an iteration only if this will result in faster convergence. Performance of the algorithm on some standard test functions is presented to illustrate that it is well defined and globally convergent.

Gould, Orban, Sartenear and Toint (2001) present a parameterized variant of the Newton's method. The Newton's method is perturbed by adding a term with a scalar parameter that is driven to zero as the iteration proceeds. The exact local solutions to the perturbed systems then form a path leading to a solution of the original system, the scalar parameter determining the progress along the path. It is shown that asymptotic convergence rate may be obtained in finding the solution and the residual vectors. Numerical experiments are used to validate the method.

CHAPTER 2

DETAILED REVIEW OF HVACSIM+

2.1. Introduction

The HVACSIM+ package has been developed primarily as a research tool for complete building system modeling. It is written in ANSI standard Fortran 77 and has enough flexibility to perform simulations of HVAC components, control systems, building models, or any combination thereof.

Documentation for HVACSIM+ consists primarily of three publications: a Reference Manual (Clark, 1985), a Users Guide (Clark and May, 1985) and a Building loads calculation Manual (Park, Clark and Kelly, 1986). The Reference Manual explains the component models distributed with HVACSIM+ and accompanying support routines. The Users Guide describes the procedure to build and run simulations. It also presents six examples of system simulations using HVACSIM+. The Building loads calculation Manual describes component models and support routines and presents sample simulations for building loads calculation. A description of the numerical methods used in HVACSIM+ is also discussed in this document.

2.2. Architecture of HVACSIM+

Operations performed by HVACSIM+ can be categorized as pre-processing, simulation and post-processing. Separate programs in HVACSIM+ perform these operations. Figure 2.2-1 shows how HVACSIM+ programs and data files interact.

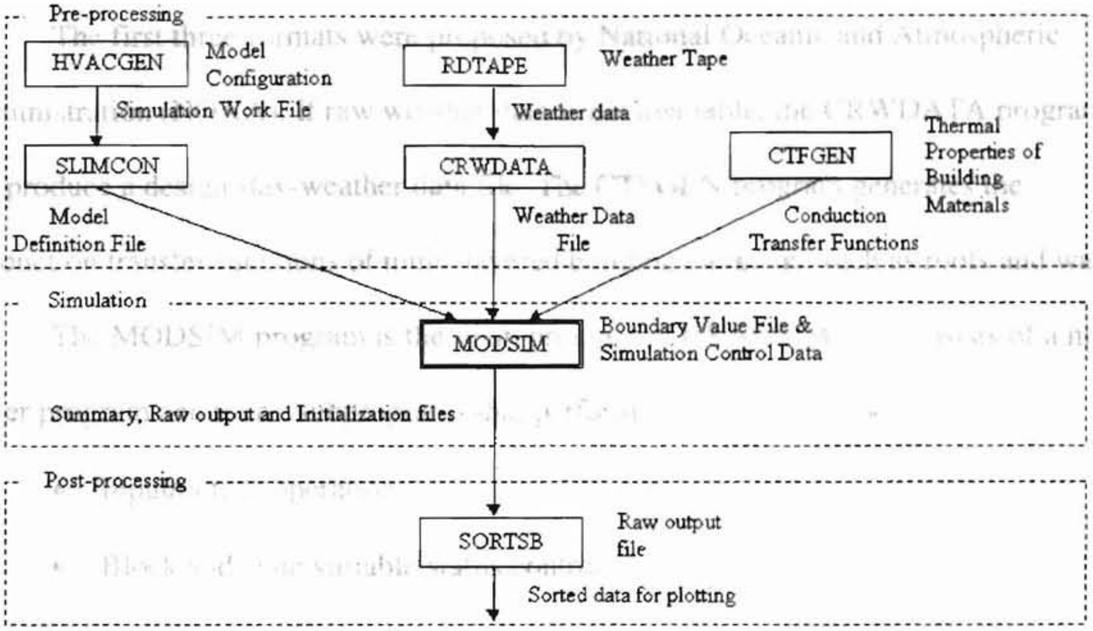


Figure 2.3.1-1. *Flow diagram of programs and data files within HVACSIM+*

The programs that perform pre-processing operations are HVACGEN, SLIMCON, RDTAPE, CRWDATA and CTFGEN. The main program MODSIM takes input from the model definition file that is created by SLIMCON from a work file generated by HVACGEN. HVACGEN is a front-end program supplied with HVACSIM+ that is used to edit a work file interactively. In generating the simulation work file, HVACGEN employs a data file containing component model information. Programs RDTAPE and CRWDATA are intended to read raw weather files and to create weather data files in a form recognized by MODSIM. The RDTAPE program can recognize unprocessed weather data files only if they are in one of the following formats:

- NOAA Test Reference Year (TRY)
- NOAA Typical Meteorological Year (TMY)
- NOAA SOLMET

- Weather Year for Energy Calculations (WYEC)

The first three formats were proposed by National Oceanic and Atmospheric Administration (NOAA). If raw weather data is not available, the CRWDATA program can produce a design-day-weather data file. The CTFGEN program generates the conduction transfer functions of multi-layered building elements, such as roofs and walls.

The MODSIM program is the main program in HVACSIM+. It consists of a main driver program and many subprograms that perform the following functions:

- Input/output operations
- Block and state variable status control
- Integration of stiff ordinary differential equations
- Solving of a system of simultaneous nonlinear algebraic equations

MODSIM reads the model definition and boundary data files and executes the simulation with user specified parameters. After a successful simulation, three data files are generated. These are the summary, raw output and initialization data files.

The post-processing program, SORTSB, is used for sorting the raw output data. The output of this program may then be used for plotting with a user-supplied graphic routine.

2.3. Modular Simulation Program (MODSIM)

MODSIM stands for MODular SIMulation. Since its inception in 1983 at the University of Wisconsin Solar Energy Laboratory, MODSIM has been enhanced significantly to include new algorithms for performance and efficiency. Important features of the current version of MODSIM are described below.

2.3.1. Hierarchical and Modular Approach

MODSIM consists of many subroutines called TYPE subroutines. These subroutines represent system components that the program can recognize. When a simulation is constructed, the components modeled by the TYPE subroutines are linked together. A UNIT in HVACSIM+ represents a component model of HVAC systems or controls, or a building element. To the user, a unit is presented as a black box that takes a set of inputs and produces a set of outputs. Using a modular approach, each UNIT is modeled in the subroutine TYPEn, where n is the index number assigned to the specific component. Components are also given UNIT numbers so that similar types can be distinguished. Each unit has distinct inputs, outputs and parameters. Each input and output variable must belong to a known category. Table 2.3.1-1 shows a list of available categories, their index numbers and default units.

Index number	**Category name**	**Category units**
1	Pressure	kPa
2	Flow Rate	kg/s
3	Temperature	°C
4	Control Signal	Non-dimensional
5	Rotation rate	Revolutions per second
6	Energy	kJ
7	Power	kW
8	Humidity	kg of water/kg of dry air

Table 2.3.1-1. *Categories of variables*

A library of component models, suitable for dynamic simulation, was developed for use with HVACSIM+. This library is called Types component library. It consists of thirty TYPE subroutines for models of HVAC components and controls. Table 2.3.1-2 shows the list of models categorized based on their functionality.

Functional category	TYPE	Model Description
Fan or Pumps	TYPE 1	Fan or Pump
Conduits	TYPE 2	Conduit (Duct or Pipe)
	TYPE 3	Inlet Conduit (Duct or Pipe)
Flow Splits and Merges	TYPE 4	Flow Merge
	TYPE 6	Flow Split
	TYPE 13	Three-way Valve with Actuator
	TYPE 17	Mixing Dampers and Merge
	TYPE 18	Plenum
	TYPE 21	'Grounded' Flow Split
Valves, Dampers and Flow Restrictors	TYPE 5	Damper or Valve
	TYPE 9	Linear Valve with Actuator
	TYPE 13	Three-way Valve with Actuator
	TYPE 17	Mixing Dampers and Merge
	TYPE 23	Steam Nozzle
	TYPE 24	Ideal Gas Nozzle
Sensors and Controller	TYPE 7	Temperature Sensor
	TYPE 8	Proportional-Integral Controller
	TYPE 16	'Sticky' Proportional Controller
	TYPE 20	High or Low Limit Controller
	TYPE 26	Control Signal Inverter
Heat Exchangers	TYPE 10	Hot Water to Air Heating Coil (Simple)
	TYPE 11	Hot Water to Air Heating Coil (Detailed)
	TYPE 12	Cooling or Dehumidifying Coil
	TYPE 25	Steam to Air Heat Exchanger
Humidifiers and Dehumidifiers	TYPE 12	Cooling or Dehumidifying Coil
	TYPE 14	Evaporative Humidifier
	TYPE 22	Steam Spray Humidifier
Conditioned Zone	TYPE 15	Room
Superblock Coupling Components	TYPE 16	'Sticky' Proportional Controller
	TYPE 19	Flow Balance Control
Building Shell	TYPE 50	Zone Envelope
	TYPE 51	Building Surface
	TYPE 53	Weather Input
Building Zone	TYPE 52	Zone Model
Smart Bridge deck heating system	TYPE 700	Slab Model: Explicit Slab Model
	TYPE 711	Heat Pump: Gang Of Water To Water Heat Pump
	TYPE 712	Heat Pump: Single Water To Water Heat Pump
	TYPE 721	GLHE Model: Hierarchical Load Aggregation Model
	TYPE 730	Flow Control: Linear Proportional Control (Heating)
	TYPE 740	Heater: 2-Mode Ideal Steady State Heater
	TYPE 750	Pump Model: Simple Pump Model

Table 2.3.1-2. *Component models arranged in functional categories*

Simulating very large systems involving many algebraic and differential equations takes considerable time and greater effort by the routines that implement the numerical methods to solve them. This problem is tackled by grouping sets of equations together into smaller subsets. MODSIM uses a hierarchical simulation structure consisting of BLOCKS and SUPERBLOCKS.

A number of functionally related units (or a single unit) form a block. The connections between the units define a set of simultaneous equations that are solved by the equation solver in MODSIM. Input variables that do not participate in the set of equations are boundary conditions.

A number of blocks (or a single block) make up a superblock. Superblocks (or a single superblock) comprise a simulation. Connections between the blocks in a superblock define a set of simultaneous equations that are solved by the equation solver. MODSIM assumes that the couplings between superblocks in a simulation are weak enough so that each superblock can be treated as an independent subsystem. No simultaneous equations are defined between superblocks. Since superblocks evolve independently in time, the inputs to a superblock may change at a time when that superblock is not scheduled for processing. To ease this problem, a superblock input scanning option is provided. When this option is selected, all superblock inputs are scanned after each time step. If the inputs to a superblock that was not processed during the time step have changed, the superblock is called and its state is calculated based on the new values of its inputs. When the input scanning option is not selected, each superblock is called at time determined by the time-step control algorithm, regardless of any changes in its inputs at intermediate times.

In MODSIM, when state variables reach steady state, they are frozen, and removed from the system of equations until deviations in the steady-state values are encountered. Similarly, a block is deactivated if all the input variables to the block are frozen and activated back when any of them change. The user is allowed to enable or disable variable freezing in the simulation.

2.3.2. Hybrid Simulation Time Steps

Many component models for HVAC and control systems involve differential equations. When the system is unsteady, a large time step invites numerical instability, especially when the components in the system have a wide range of time constants. To prevent this instability, smaller intervals are necessary at simulation startup or during a period when sudden change occurs. After the system has stabilized, the use of long time steps is more appropriate.

The MODSIM program incorporates two different types of time steps, namely, the fixed time step and the variable time step. While defining the simulation, the user chooses the minimum and maximum time step values. If these values are different, it indicates that the simulation will use variable time steps.

Variable time steps are best suited for solving stiff differential equations that have widely varying time constants. Building models may use fixed time steps since they are based on uniformly distributed time sampling.

MODSIM employs variable-order Gear algorithms (Gear, 1971) that are proven to work efficiently in integrating stiff differential equations. The order of integration and the size of the time steps are dynamically determined during the calculation to minimize the computation required for a desired degree of accuracy. The time step and integration order are determined independently for each superblock since they are considered to be independent subsystems that proceed independently in time.

2.3.3. Time Dependent Boundary Conditions

A state variable that is external to the system being simulated can be designated as a boundary variable when the simulation work file is generated. The boundary variables may be constant or time dependent. Data for time-dependent boundary variables are stored in the boundary value file and read as the simulation progresses.

The first column in the boundary value file must always contain time values. Time intervals in the boundary value file need not be equal, since a third order Lagrangian interpolation method, explained in the Section 2.4.6, is used to find the boundary variable values at simulation time steps. The remaining columns contain values of boundary variables at the specified times. MODSIM begins all simulations at time zero regardless of the first time entry in the boundary value file.

Including in the boundary data file two different data values of a boundary variable at a given time signals a discontinuity in a boundary condition. When the change in a boundary variable is discontinuous, the routine that integrates the differential equations is reset at the time of discontinuity to bring the simulation time step to a minimum value.

2.3.4. Reports Generation

MODSIM produces three files on output, namely the Final State File (.FIN), the Output File (.OUT) and the Summary File (.SUM). The Final State File contains numbers defining the state of the simulation at the simulation stopping time. The Output File contains the raw data at each time step in the simulation. The Summary File contains a summary of the simulation including information relating to configuration, diagnostics performed and listing of reported variables at equal time intervals.

2.4. Numerical methods in MODSIM

The numerical methods employed in the MODSIM program involve techniques for solving systems of simultaneous nonlinear algebraic equations, integrating stiff ordinary differential equations, and interpolating data sampled in either a fixed period or variable time intervals. A large number of subprograms in MODSIM are related to these numerical algorithms.

2.4.1. Solving Systems of Simultaneous Nonlinear Algebraic Equations

The nonlinear equation solver is implemented in the subroutine SNSQ. The purpose of this subroutine is to find a minimum of a system of N nonlinear functions in N variables using the Powell's hybrid method.

The parameters to SNSQ are classified as input and output. Input parameters must be specified on entry to SNSQ and are not changed on exit. Output parameters need not be specified on entry and are set to appropriate values on exit from SNSQ. Table 2.4.1-1 presents a list of input and output parameters and their descriptions. Apart from the listed parameters, the SNSQ routine also takes the name of the user-supplied subroutine that calculates the functions as a parameter.

The user has the option of either providing a subroutine that calculates the Jacobian or of allowing the code to calculate it by a forward-difference approximation method. The input parameter IOPT is used to identify which method the user prefers to use. A numerical value of one indicates that the user wishes to use the former method. In this case, the name of the subroutine that calculates the Jacobian should also be passed to SNSQ as a parameter.

Parameter	Type	Input/ Output	Description
IOPT	Integer	Input	IOPT specifies how the Jacobian will be calculated. If IOPT = 1, then the user must supply the Jacobian through the subroutine JAC. If IOPT = 2, then the code will approximate the Jacobian using the forward difference method.
N	Integer	Input	It is a positive integer variable set to the number of functions and variables.
X	Real	Input and Output	It is an array of length N. On input; X must contain an initial estimate of the solution vector. On output, X contains the final estimate of the solution vector.
FVEC	Real	Output	FVEC is an array of length N that contains the functions evaluated at the output X.
FJAC	Real	Output	FJAC is an N-by-N array that contains the orthogonal matrix Q produced by the QR factorization of the final approximate Jacobian.
LDFJAC	Integer	Input	LDFJAC is a positive integer variable not less than N that specifies the leading dimension of the array FJAC.
XTOL	Real	Input	It is a non-negative number. Termination occurs when the relative error between two consecutive iterations is at most XTOL. Therefore, XTOL measures the relative error desired in the approximate solution.
MAXFEV	Integer	Input	It is a positive integer variable. Termination occurs when the number of calls to FCN is at least MAXFEV by the end of iteration.
ML	Integer	Input	ML is a non-negative integer variable that specifies the number of sub-diagonals within the band of the Jacobian matrix. If the Jacobian is not banded or IOPT=1, set ML to at least N - 1.
MU	Integer	Input	MU is a non-negative integer variable that specifies the number of super-diagonals within the band of the Jacobian matrix. If the Jacobian is not banded or IOPT=1, set MU to at least N - 1.
EPSFCN	Real	Input	It is used in determining a suitable step for the forward-difference approximation. This approximation assumes that the relative errors in the functions are of the order of EPSFCN. If EPSFCN is less than the machine precision, it is assumed that the relative errors in the functions are of the order of the machine precision. If IOPT = 1, then EPSFCN can be ignored.
DIAG	Real	Input	It is an array of length N. If MODE = 1 (see below), DIAG is internally set. If MODE = 2, DIAG must contain positive entries that serve as implicit scale factors for the variables.

Table 2.4.1-1. *List of parameters to SNSQ*

Parameter	Type	Input/ Output	Description
MODE	Integer	Input	It is an integer variable. If MODE = 1, the variables will be scaled internally. If MODE = 2, the scaling is specified by the input DIAG. Other values of MODE are equivalent to MODE = 1.
FACTOR	Real	Input	It is a positive variable used in determining the initial step bound. This bound is set to the product of FACTOR and the Euclidean norm of DIAG*X if nonzero, or else to FACTOR itself. In most cases, FACTOR should lie in the interval (0.1, 100.0). The generally recommended value is 100.0.
NPRINT	Integer	Input	It is an integer variable that enables controlled Printing of iterates if it is positive. In this case, FCN is called with IFLAG = 0 at the beginning of the first iteration and every NPRINT iteration thereafter and immediately prior to return, with X and FVEC available for printing. Appropriate print statements must be added to FCN. If NPRINT is not positive, no special calls of FCN with IFLAG = 0 are made.
NFEV	Integer	Output	It is an integer variable set to the number of calls to FCN.
NJEV	Integer	Output	It is an integer variable set to the number of calls to JAC. (If IOPT=2, then NJEV is set to zero.)
LR	Integer	Input	It is a positive integer variable not less than (N*(N+1))/2.
QTF	Real	Output	It is an output array of length N which contains the vector (Q^T)*FVEC.
INFO	Integer	Output	It is an integer variable. If the user has terminated execution, INFO is set to the (negative) value of IFLAG. Otherwise, INFO is set as follows. • INFO = 0, improper input parameters. • INFO = 1, relative error between two consecutive iterates is at most XTOL. • INFO = 2, number of calls to FCN has reached or exceeded MAXFEV. • INFO = 3, XTOL is too small. No further improvement in the approximate solution X is possible. • INFO = 4, iteration is not making good progress, as measured by the improvement from the last five Jacobian evaluations. • INFO = 5, iteration is not making good progress, as measured by the improvement from the last ten iterations.
R	Real	Output	R is an output array of length LR that contains the upper triangular matrix produced by the QR factorization of the final approximate Jacobian, stored row-wise.
WA1, WA2, WA3, WA4	Real	Output	They are work arrays of length N.

Table 2.4.1-1. (Contd.) *List of parameters to SNSQ*

2.4.2. Mathematical description of the solver

A detailed description of the solver was given by Powell (1970). This section explains the mathematical description of the methods implemented in the SNSQ subroutine.

The system of nonlinear equations can be written in vector form as:

$$f(x_1, x_2, x_3, \cdots x_n) = 0 \qquad (2.4.2\text{-}1)$$

Expanding the system in a Taylor series, and neglecting the high order terms, the linearized, approximate system becomes:

$$f(x^*) = f(x^k) + J(x^k)(x^* - x^k) = 0 \Rightarrow J(x^k)(x^* - x^k) = -f(x^*) \qquad (2.4.2\text{-}2)$$

where, x^* is the solution vector and $J(x^k)$ is a Jacobian evaluated at x^k.

The Newton step of the nonlinear system, Δx, can be expressed as:

$$\Delta x = x^{k+1} - x^k = -\left[J^T(x^k)J(x^k)\right]^{-1} J^T(x^k) f(x^k) \qquad (2.4.2\text{-}3)$$

The major advantages of Newton's method are its rapid (quadratic) convergence and its suitability with highly interacting systems because interactions between variables are fully taken into account by the Jacobian matrix. However, the algorithm requires a good initial guess for successful convergence. In addition, more computational effort is required to calculate the inverse of the Jacobian matrix. In order to reduce the computational effort, a Quasi-Newton method (Dennis and Schnabel, 1983) is used. The main idea of this method is to approximate the Jacobian at each iteration using only the current and previous values of x and $f(x)$.

The Quasi-Newton method uses Broyden's rank-one update (Broyden, 1965) instead of calculating the full Jacobian at each iteration. For the first iteration, the Jacobian is calculated either by a user-supplied subroutine or by a forward-difference approximation. The Jacobian is not recalculated until the rank-one method fails to give satisfactory progress. The method is explained below.

Let the k+1 step give the solution for the problem, in the Equation 2.4.2-1. Equation 2.4.2-2 can then be modified as:

$$f(x^{k+1}) = f(x^k) + J(x^k)(x^{k+1} - x^k) = 0 \qquad (2.4.2\text{-}4)$$

Rearranging the terms,

$$J(x^k)(x^{k+1} - x^k) = f(x^k) - f(x^{k+1}) \qquad (2.4.2\text{-}5)$$

Let A^k represent the approximation of $J(x^k)$. This equation can be rewritten as:

$$A^k(x^{k+1} - x^k) = f(x^k) - f(x^{k+1}) \qquad (2.4.2\text{-}6)$$

Equation 2.4.2-6 is referred to as the Secant equation.

Revisiting Equation 2.4.2-2,

$$f(x^*) = f(x^k) + J(x^k)(x^* - x^k) = 0 \qquad (2.4.2\text{-}7)$$

Rewriting the Equation 2.4.2-7 in a model form gives,

$$f(x) = f(x^k) + A^k(x - x^k) \qquad (2.4.2\text{-}8)$$

Let $m^k(x)$ represent the right side of the Equation 2.4.2-2. We have,

$$m^k(x) = f(x^k) + A^k(x - x^k) \qquad (2.4.2\text{-}9)$$

For the next iteration, x^{k+1}, in the above equation can be modified as:

$$m^{k+1}(x) = f(x^{k+1}) + A^{k+1}(x - x^{k+1}) \qquad (2.4.2\text{-}10)$$

Taking the difference between the equations 2.4.2-10 and 2.4.2-9 and rearranging gives,

$$m^{k+1}(x) - m^k(x) = f(x^{k+1}) - f(x^k) - A^{k+1}(x^{k+1} - x^k) + (A^{k+1} - A^k)(x - x^k) \qquad (2.4.2\text{-}11)$$

Substituting the Equation 2.4.2-6 in 2.4.2-11 and rearranging gives,

$$m^{k+1}(x) - m^k(x) = (A^{k+1} - A^k)(x - x^k) \tag{2.4.2-12}$$

Considering the Equation 2.4.2-12 as a minimizing function and defining a step, x, to be taken in order to reach the global minimum, as

$$x - x^k = \alpha(x^{k+1} - x^k) + t \tag{2.4.2-13}$$

where, α is a small value. Variable t is defined by the equation $t^T s_c = 0$. Substituting the Equation 2.4.2-12 in Equation 2.4.2-12 gives,

$$m^{k+1}(x) - m^k(x) = \alpha(A^{k+1} - A^k)(x^{k+1} - x^k) + (A^{k+1} - A^k)t \tag{2.4.2-14}$$

The behavior of the first term on the right side in the Equation 2.4.2-14 cannot be predicted since it involves $f(x)$. However, the second term can be made zero by choosing A^{k+1} such that $(A^{k+1} - A^k)t = 0$ for all t orthogonal to $(x^{k+1} - x^k)$. This is possible only when $(A^{k+1} - A^k)$ is in the form $u(x^{k+1} - x^k)^T$, $u \in \Re^n$. This can be represented as:

$$u(x^{k+1} - x^k)^T = (A^{k+1} - A^k) \tag{2.4.2-15}$$

Applying the new form of $(A^{k+1} - A^k)$ to the secant equation in the Equation 2.4.2-6, we have,

$$u(x^{k+1} - x^k)^T(x^{k+1} - x^k) = f(x^k) - f(x^{k+1}) \tag{2.4.2-16}$$

The above equation gives an expression for u. Substituting the Equation 2.4.2-16 in the Equation 2.4.2-15, we obtain,

$$A^{k+1} = A^k + \frac{\left[f(x^k) - f(x^{k+1})\right](x^{k+1} - x^k)^T}{(x^{k+1} - x^k)^T(x^{k+1} - x^k)} \tag{2.4.2-17}$$

The Equation 2.4.2-17 gives the Broyden's update. The word 'update' indicates that $J(x^{k+1})$ is not updated from scratch, rather the approximation A^k to $J(x^k)$ into an approximation A^{k+1} to $J(x^{k+1})$. For the first iteration, A^0 is chosen to be either the negative identity matrix or the finite difference method is applied to the Jacobian matrix and the matrix is inverted.

The Quasi-Newton method only requires a single pass through the system at each iteration. It accurately reflects the interaction of variables and does not need to perform inversions of matrices at each step. The calculation of second derivatives is also eliminated.

The convergence of the algorithm is only superlinear (not quadratic), because of which more iterations are needed for the algorithm to converge. However, the cost per step of Broyden's method is substantially less.

2.4.2.1. Powell's Hybrid Method

The major shortcoming of the Quasi-Newton method is the necessity of a good initial guess. To improve this property Powell's Hybrid method (Powell, 1970) is employed in the SNSQ routine. The hybrid step, also called the "dogleg" step, is a combination of quasi-Newton and steepest gradient step.

In the Powell's Hybrid method, Equation 2.4.2-3 is modified as:

$$\Delta x = -\left[J^T(x^k)J(x^k) + \mu^k I\right]^{-1} J^T(x^k) f(x^k) \qquad (2.4.2.1\text{-}1)$$

where, I is a unit matrix and μ^k is a non-negative parameter. As μ^k is increased the algorithm approaches the steepest descent method with small steps:

$$\Delta x \cong -\frac{1}{\mu^k} J^T(x^k) f(x^k)\text{, for large } \mu^k \qquad (2.4.2.1\text{-}2)$$

while as μ^k is decreased to zero the algorithm becomes Gauss-Newton.

The value of μ^k is based on an objective function, also called the minimization function. The objective function used in the solver is the sum of squares of residuals. A residual may be defined as the difference between the function evaluated at the calculated solution and the function evaluated at the actual solution. From the Equation 2.4.2-1, we can observe that the function evaluated at the actual solution is zero. The objective function can be expressed as follows:

$$F(x) = \sum_{i=1}^{n} [f_i(x)]^2 \qquad (2.4.2.1\text{-}3)$$

The value of μ^k is calculated in such a way that the following inequality is satisfied.

$$F(x^{k+1}) < F(x^k) \qquad (2.4.2.1\text{-}4)$$

The algorithm begins with μ^k set to some small value. If a step does not yield a smaller value for $F(x)$, then the step is repeated with μ^k multiplied by some factor $v > 1$. Eventually $F(x)$ should decrease, since the algorithm takes small steps in the direction of steepest descent. If a step produces a smaller value of $F(x)$, then μ^k is divided by v for the next step. This enables the algorithm to approach Gauss-Newton, thus providing faster convergence. The method thus provides a good compromise between the speed of Newton's method and the guaranteed convergence of steepest descent.

Figure 2.4.2.1-1 shows the Powell's hybrid step. The figure presents an example performance surface, between two parameters, on which the method is applied. The black dot represents the global minimum of the surface, which is (10, 3). The gray dot represents the initial guess, which is (-3, -3). The dashed arrow represents the direction taken for small μ^k, which corresponds to the Gauss-Newton direction. The solid arrow represents the direction taken for large μ^k, which corresponds to the steepest descent direction. The gray curve represents the hybrid step for all intermediate values of μ^k.

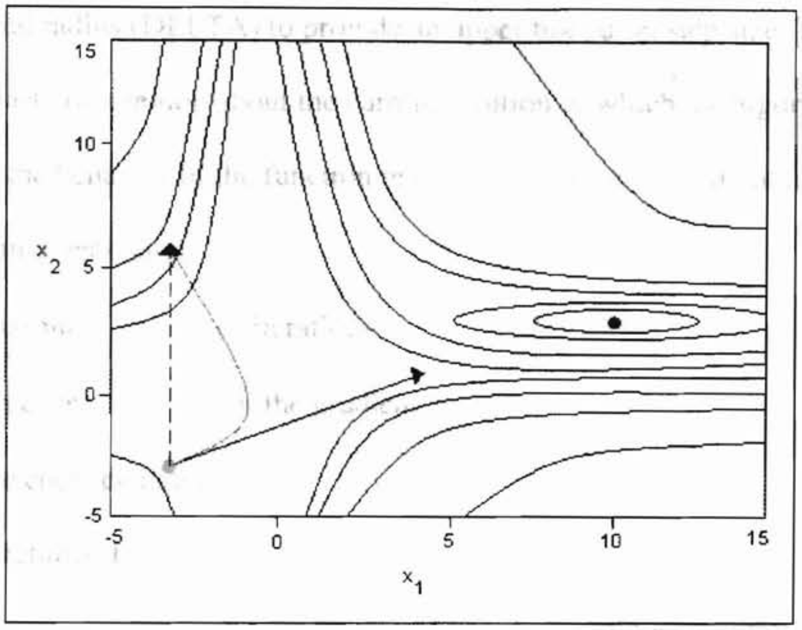


Figure 2.4.2.1-1. *Powell's hybrid step*

The HVACSIM+ equation solver implements Powell's hybrid method by using a specialized algorithm called the DOGLEG algorithm that utilizes a combination of steepest-descent and Newton steps in the process of minimizing a function. As long as the gradient steps are relatively large, they are used. However, since gradient steps tend to perform poorly in valleys on the performance surface, Newton steps are also used. Newton steps, however, perform poorly with data points far away from the minimum. Hence, the algorithm uses a bound on the minimum step size and provides a compromise step, called a "dogleg" step, that combines the gradient and Newton steps. Depending on the performance of the DOGLEG algorithm the bound is either increased or decreased for future iterations.

The DOGLEG algorithm requires the following inputs:

- Function $f(x)$
- Starting values x_0
- An initial radius (DELTA) to provide an upper bound on step size. The radius is an estimate of a region about the current solution in which the algorithm can predict the behavior of the function reasonably well. A default value of 1.0 is used as an initial estimate.
- The maximum number of iterations
- Convergence tolerance for the gradient
- Relative coefficient change

The algorithm returns the step that it decided to take (Newton, steepest-gradient or dogleg). The acceptability of the Newton step is based on Euclidean norms. The Euclidean norm is defined below.

The Euclidean norm, also called least-squares norm, of a vector $v = (v_1, v_2, v_3, ...v_n)^T \in \Re^n$ is expressed as,

$$\|v\| = \left(\sum_{i=1}^{n} (v_i)^2 \right)^{1/2} \qquad (2.4.2.1\text{-}5)$$

The dogleg algorithm is shown in the Figure 2.4.2.1-2. At the beginning of each iteration, the Gauss-Newton direction is calculated. The step-bound limitation is then checked. If the Newton's step is not acceptable, a gradient step is employed. If the gradient step also fails to improve the minimization function, shown in the Equation 2.4.2.1-4, a dogleg step is attempted. If all the three steps fail, the algorithm repeats with the same solution. The bound radius (DELTA) is then increased or decreased and a new iteration begins.

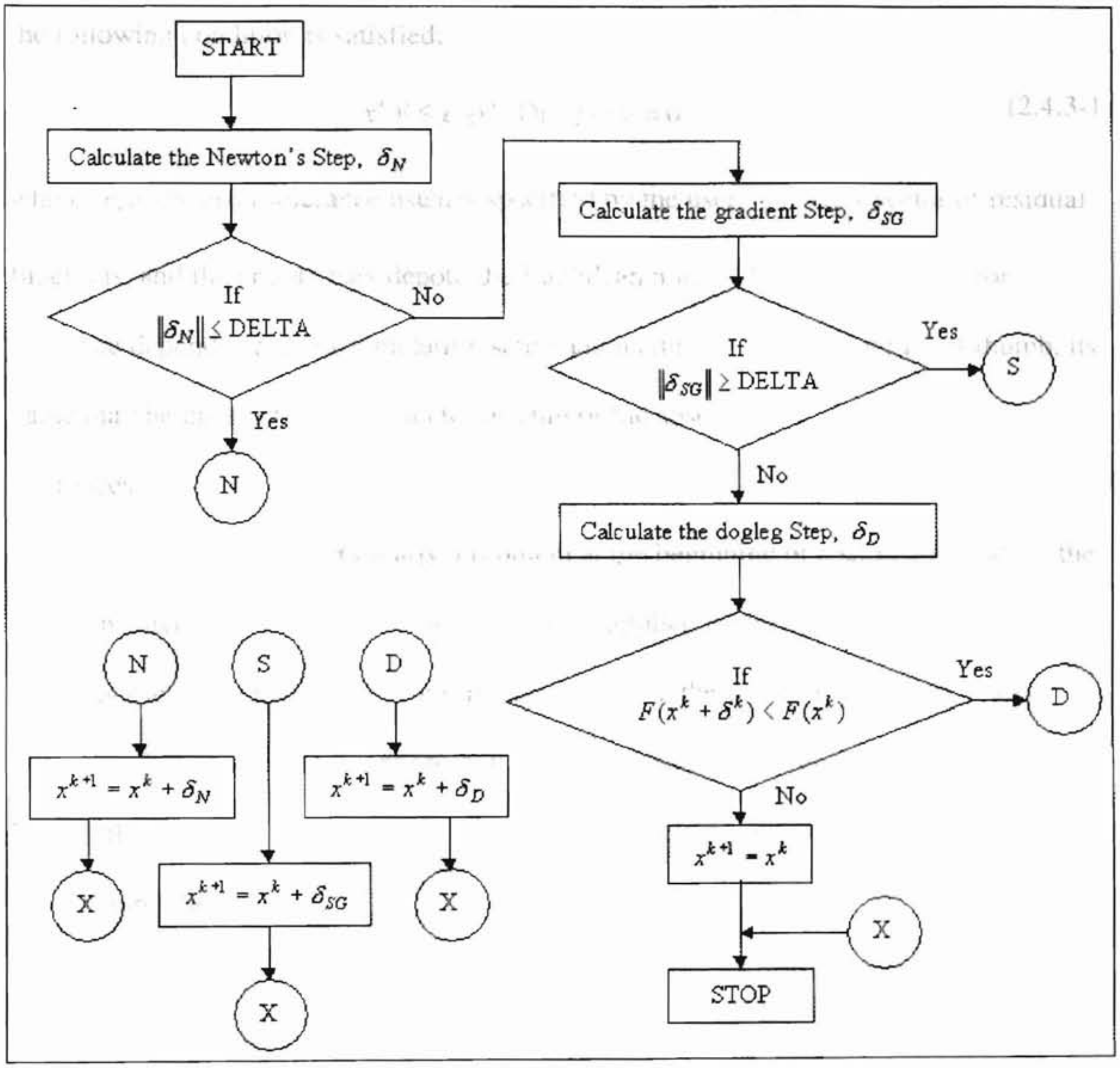

START
Calculate the Newton's Step, δ_N
If
$\|\delta_N\| \leq$ DELTA
No
Yes
N
Calculate the gradient Step, δ_{SG}
If
$\|\delta_{SG}\| \geq$ DELTA
Yes
S
No
Calculate the dogleg Step, δ_D
If
$F(x^k + \delta^k) < F(x^k)$
Yes
D
No
$x^{k+1} = x^k$
X
STOP
N
S
D
$x^{k+1} = x^k + \delta_N$
$x^{k+1} = x^k + \delta_D$
$x^{k+1} = x^k + \delta_{SG}$
X
X
X

Figure 2.4.2.1-2. *DOGLEG Algorithm*

The initial choice of DELTA can profoundly affect the performance of the algorithm – different values sometimes lead to finding different local minimizers. Too small or too large a value can cause the algorithm to spend several function evaluations in the first iteration to decide whether to decrease or increase DELTA.

2.4.3. Convergence Criteria in the Equation Solver

The convergence test implemented in the solver selects x^k as a solution vector if the following condition is satisfied:

$$\left\|(x^{k+1} - x^k)\right\| \le e_t \left\|x^k\right\| \text{Or } \left\|f(x)\right\| = 0 \qquad (2.4.3\text{-}1)$$

where, e_t is the error tolerance usually specified by the user, $f(x)$ is a vector of residual functions, and the double bars denote the Euclidean norms. The value of the error tolerance depends upon the simulation setup and its initial values. As a rule of thumb, its value may be greater than or equal to the sum of the absolute and relative error tolerances.

Convergence is particularly a problem at the beginning of a simulation, when the equation solver must work from the set of user-supplied initial conditions. If the initial conditions are a large distance from the final solution, the equation solver may fail to converge. The choice of the error tolerance for the equation solver is an important factor. Even if the solutions are convergent, an improperly selected error tolerance can result in unacceptably large errors.

The problem of slow and uncertain convergence to a self-consistent set of initial conditions is unavoidable, but the consequences can be minimized by the use of the initialization option. This permits a series of simulations to be run from the same initial state, without repeating an inaccurate and computationally expensive startup transient at the beginning of each run.

The convergence test in the solver can be rewritten as $\left\|x^{k+1} - x^k\right\| \le e_t \left\|x^k\right\|$, where, x_k is the solution vector and e_t is the error tolerance. This test is based on Euclidean norms. Another kind of norm, called the Infinity norm, is used in the field of optimization. The Infinity norm, represented as $\|.\|_\infty$, may be defined on a vector $v = (v_1, v_2, \ldots v_n)^T \in \Re^n$ as,

$$\|v\|_\infty = \max_{1 \le i \le n} |v_i| \tag{2.4.2.1-2}$$

If, for example, $x_{k+1} - x_k = [\gamma, \ldots, \gamma]^T$, where, γ is any scalar quantity, then

$$\|x_{k+1} - x_k\| = \sqrt{n}|\gamma| \text{ and } \|x_{k+1} - x_k\|_\infty = |\gamma|,$$

where, *n* is the number of variables. If *n* is large, the Euclidean norm of the gradient can be large even if γ is small. This can distort the convergence test and so it is preferable to use the Infinity norm when large problems are solved. The use of norms in the convergence test can have other side effects. Suppose that:

$$x^{k-1} = [1.55564 \quad 0.00082 \quad 0.0000078]^T$$

$$x^k = [1.55552 \quad 0.00011 \quad 0.0000011]^T$$

If we had chosen $e_t = 10^{-3}$ then

$$\|x^{k-1} - x^k\|_\infty = \|[0.00012 \quad 0.00071 \quad 0.0000067]^T\| = 0.0007201 \le 1.5555.10^{-3},$$

and x^k would pass this test. If, for an application, it were important that the significant digits of all variables be accurate, then x^k would not be satisfactory since its second and third components are still changing in their first significant digit. The use of norms emphasizes the larger components in a vector and so the smaller components may have poor relative accuracy. This effect can be rectified by non-dimensionalizing the solution vector. This is achieved by scaling the solution vector, which can ensure that all the variables in the transformed problem would have approximately the same relative accuracy.

A scale vector, represented here as d, where $d = (d_1, d_2, d_3, ...d_n)$, can profoundly affect the performance of the optimization algorithms. Vector, d, should be such that the elements $|d_i x_i|, 1 \leq i \leq n$, are in comparable units. A reasonable choice of d is obtained by guessing the upper bounds, represented as ξ_i, on $|x_i^*|$, where x^* is the solution vector, and setting d_i equal to $1/\xi_i$. Several methods have been suggested to estimate the scale vector. Gay (1990) summarized the available methods and concluded that selection of a scaling method is problem specific.

Scaling in the equation solver is computed by estimating the upper bound of the solution vector, ξ, and computing the scale as $1/\xi$. There is no user interaction required to perform scaling.

In order to solve the system of equations we need a starting value and, since $f(x)$ is not scale invariant, we also need a way to measure the scale of the residuals $(r(x) = y - f(x))$ at the beginning of the computations. We also may need to re-measure the scale as the computation continues.

Since the initial conditions supplied by the user are usually not good enough, the convergence is particularly a problem at the beginning of the simulation. If the initial conditions are too far from the actual solution, the equation solver may not converge. In some cases, errors made by the solver can cause erratic solutions for several time steps, even if the solutions are convergent. The choice of the error tolerance for the equation solver is an important factor that influences this behavior.

Whenever the solution does not converge, MODSIM shows a warning message "Iteration not making good progress". This message is shown when the steps taken in the gradient direction, while searching for a global minimum in the domain, exceed 10. The number of iterations, which limit the gradient step bounds, are used by the algorithm to check the progress of simulation. For certain simulations, a closer investigation of the residuals, printed in the diagnostic report, revealed that the solution failed to converge even though it had satisfied all the stringent convergence criteria. In addition, the program did not show any warning messages.

The convergence criteria used in MODSIM do not directly consider the magnitude of the residuals. In a system that simulates mainly the temperatures (which are mostly in the range of –10 to 60 degrees centigrade), a residual greater than 1 degree affects the results significantly. Moreover, in computing the heat transfer rates, where the temperature differential is multiplied by high flow rates, the effect of high residuals is very significant. The only limiting criterion of the solver, to check whether the optimum solution is reached or not, is the step size used by the solution algorithm.

The convergence properties of the equation solver are also dependent on the Block/Superblock structure. The user must carefully define blocks and superblocks to obtain good convergence since they determine the equations sets to be solved during a simulation. This fact limits the flexibility of simulation construction.

2.4.4. Integration of Stiff Ordinary Differential Equations

The algorithm, implemented in MODSIM, to integrate differential equations is based on a method developed by Brayton, Gustavson and Hachtel (1972). This method uses variable time steps and variable-order-integration techniques to solve sets of differential equations and hence can significantly reduce the amount of computer time required for dynamic simulations. This section highlights the important aspects of this method.

A general system of differential equations may contain higher order derivatives, but such a system may be reduced to first-order form by introducing new variables. A system of first-order differential algebraic equations can be expressed as:

$$f(x,\dot{x},t)=0 \tag{2.4.4-1}$$

where, x is a state vector that is a function of time, t, and $\dot{x}$ is the derivatives of x. If the solution vector $x(t)$ of the above equation had been obtained at previous discrete times, $t=t_n$, $t=t_{n-1}$,..., and $t=t_{n+1-p}$, then the solution x_{n+1} at the current time, $t=t_{n+1}$, satisfies:

$$f(x_{n+1},\dot{x}_{n+1},t_{n+1})=0 \tag{2.4.4-2}$$

For stiff differential equations, having widely varying time constants, the p-th order backward differentiation formula is given as:

$$\dot{x}_{n+1}=-\frac{1}{h}\sum_{i=0}^{p}a_i x_{n+1-i} \tag{2.4.4-3}$$

where, a_i are constants and h is the step size $(t_{n+1}-t_n)$.

Substituting equation 2.4.5-3 in 2.4.5-2 yields a set of nonlinear algebraic equations in x_{n+1} at time t_{n+1}. The obtained system of equations is solved by the equation solver. If the solution is converged, the iteration is terminated and the order, p, and step size, h, are updated as follows.

The truncation error of p-th order backward differential formula is given as,

$$E_p = \frac{h}{t_{n+1} - t_{n-p}} \left(x_{n+1} - x_{n+1}^{g} \right) \qquad (2.4.4\text{-}4)$$

where, x_{n+1}^{g} is the guess value of x_{n+1}. The guess value is formulated using the same regressor expression in Equation 2.4.4-3. The guess value is expressed as,

$$x_{n+1}^{g} = \sum_{i=0}^{p+1} \gamma_i x_{n+1-i} \qquad (2.4.4\text{-}5)$$

where, γ_i are constants.

If the truncation error satisfies the condition,

$$E_p \leq \frac{3h}{t_{if}} \left(e_r \left| x_{n+1} \right| + e_a \right) \qquad (2.4.4\text{-}6)$$

where, t_{if} is the time interval, e_r is the relative error tolerance and e_a is the absolute error tolerances, the step size and the order are acceptable. The time interval and the error tolerances are specified by the user.

To update the step size and the order, the above condition is applied for orders $p+1$, p and $p-1$. The order corresponding to the largest time step size is selected. The computation of $(p-1)$th order differential is skipped if p is 1, in which case, only the step sizes at p and $p+1$ are compared. The maximum order of p is limited to six.

2.4.5. Interpolation of data

In the Section 2.3.3, it was mentioned that HVACSIM+ interpolates the data in the boundary value file to find the boundary values at simulation times. This section describes the application of Lagrangian interpolation techniques to perform interpolation of data.

During a simulation, the MODSIM program might take time step values for which boundary values are not specified in the boundary value file. In order to find the data values for the boundary variables the third order Lagrangian interpolation method is used. This method uses the simulation time to interpolate the data. It is expressed as follows.

$$x(t) = \sum_{i=0}^{4} \left[\prod_{\substack{j=1 \\ i \neq j}}^{4} \left(\frac{t - t_j}{t_i - t_j} \right) \right] x_i \tag{2.4.5-1}$$

where, $x(t)$ is the interpolated boundary variable at time t, and $x_i = x(t_i)$.

CHAPTER 3

HVACSIM+ ENHANCEMENTS

Although HVACSIM+ was developed over two decades ago, a number of problems exist that make the program difficult to use. These problems, as documented by Clark (1985) include:

- complexity of the user interface
- difficulty in debugging the simulation
- improper convergence of the equation solver

3.1. Development of the user-interface

The user interface of HVACSIM+ may be considered primitive. The program interacts with the user through a combination of command and menu-driven systems. The menu driven program control has a hierarchy of menus that the user can select by typing one or more characters to direct the program. However, if the user makes an error while configuring a system, there is no way that it can be corrected immediately. The user has to either abort the process or supply arbitrary values to get to the next command prompt. Aborting the process discards all changes that have been made. Once the user returns to the main menu, the EDIT command can be chosen to edit the changes in the system configuration. This is usually time consuming and requires patience.

Each input and output variable in a simulation is given an index number. The user uses this number to identify the variable that needs to be processed for setting connections or setting initial values. Keeping track of the indices is required in order to represent the system correctly in a physical sense. Assigning numbers to variables in large systems is tedious and time consuming. Repetition of indices, or connectivity of one category with a different category (e.g. control signal with pressure) distorts the whole system configuration. The simulation work file, created using HVACGEN, and the model definition file, generated by SLIMCON, are also difficult to understand.

Using an event-driven approach instead of the question-and-answer-based logic can rectify the problems with the user-interface. In the question-and-answer method, a central controlling program determines the sequence of operations for an application. Event-driven programming determines the sequence of operations for an application by the user's interaction with the application interface (forms, menus, buttons, etc.). For example, rather than having a main procedure that executes a series of processes, an event-driven application remains in the background until the user picks the processes to be performed.

An event is an action recognized by an element of a graphical user interface. Some examples include a 'mouse click' on a button or a 'key press' on the keyboard. Events can also relate to actions that trigger processes. For example, a 'simulation ended' event at the end of a simulation could, among other things, close any open files.

Event-driven programming and graphical user interfaces (GUIs) are related since forms and the graphical interface objects on the forms serve as the skeleton for the entire application. In an event-driven application, small programs are attached to events associated with objects. In this way, the behavior of the application is determined by the interaction of a number of small manageable programs rather than one large program.

The primary advantages of event-driven programs are the following:

- Flexibility - Events rather than a sequential program control the flow of the application. Thus, the user does not have to conform to the programmer's understanding of how tasks should be executed.
- Robustness - Event-driven applications tend to be more robust since they are less sensitive to the order in which users perform activities. In conventional programming, the programmer has to anticipate virtually every sequence of activities the user might perform and define responses to these sequences.

As discussed in Chapter 2, the main programs in HVACSIM+ that provide an interface for building and executing a simulation are the HVACGEN and MODSIM programs. Both these programs use the conventional question-and-answer based approach discussed above to accept input from the user. HVACGEN is used to create an initial work file that is processed by the SLIMCON program to create a definition file. The MODSIM program reads the definition and the boundary value files and simulates the system. The user should use all three programs to run a simulation.

The user interface for HVACSIM+ has been improved by creating a Visual Modeling Tool to replace HVACGEN and adding residual reporting feature in MODSIM.

3.1.1. Visual Modeling Tool

The visual modeling tool has been developed primarily for research purposes and is intended to carry out dynamic simulation studies. A detailed description of the tool is given in Chapter 4. The features of the tool can be summarized as follows:

- The tool uses icons and pictures to represent units, boundary variables and connections between them.

- The tool imitates the functionality of HVACGEN using an event-driven approach but also goes a step forward and makes calls to SLIMCON and MODSIM programs in the background.
- The tool eliminates the need for the user to use three different programs to build and run a simulation.
- The user is not required to keep track of variable indices.
- The tool provides facilities to plot the simulation output.
- An online help system supports the tool.

3.1.2. Residual Reporting

During a simulation, HVACSIM+ does not use any validation tests to determine if the results being generated are reasonable. For simulations that run several hours such tests would allow the users to decide whether to continue with the simulation or not. Since no such test is available, the user is forced to wait for the simulation to end, analyze the results and re-run the simulation with any corrections.

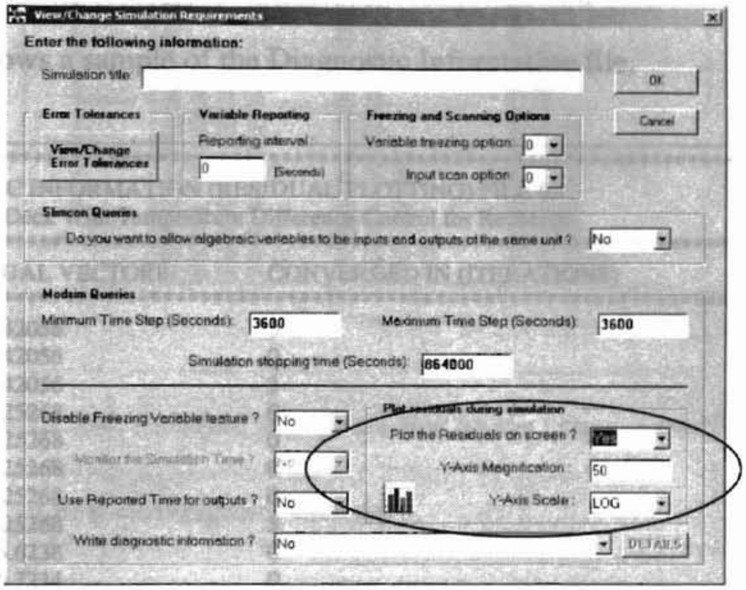
The solution algorithm and its associated subroutines are more difficult to debug, since the code lacks structure and logical flow. The GOTO statement has been used extensively in the code, which makes the code difficult to comprehend. Variables are named inconsistently. The source code also lacks necessary documentation.

The only reprieve during the debugging phase is the diagnostic report that can be printed out for a specified duration in the simulation. The diagnostic report can be made to print information such as the Jacobian, residual information, "hybrid step" size of the optimization algorithm (indicating the progress of the iteration), limiting convergence criteria and intermediate solution vector. This information, although useful, is insufficient to find the source of most problems.

In order to give a useful picture of the simulation progress, the interface is capable of plotting residuals while the simulation is running. This has been achieved by adding a child window in which the plotting is performed. The user can use the menu to switch from the simulation status window to the plot window.

The residual plot feature can only be activated if MODSIM is called from the visual tool. To activate the feature the user is required to interact with the form shown in Figure 3.1.2-1. Complete details about this dialog are presented in the Chapters 4 and 5. The user can decide not to show the plot only at this stage. Once the plot window is activated, there is no way to close it.

The application plots the Euclidean norm of the residual vector against time on a bar graph. The scale on the y-axis, which shows the norms, can be linear or log. The scale on the x-axis, which shows the time step values and iterations within them, is always linear. Each time step is allotted a random color that remains the same for all iterations belonging to that time step. A filled circle represents convergence of the algorithm over a bar. Figure 3.1.2-2 shows a sample plot of residuals generated during a simulation. The plot shows that during the initial stages of the simulation the residuals are high. This is because of poor initialization of variables. After a good guess is made, the algorithm performs better. After the first two time steps, the residuals in the plot are zero. This is probably because of no loads in the simulation or because the algorithm is performing very efficiently. The peaks start appearing when there is considerable effort required by the equation solver to find a solution. As the iterations for a time step proceed, the residuals should reduce. Some time steps indicate convergence although the residual values are not zero.


View/Change Simulation Requirements
OK
Cancel
View/Change Error Tolerances
DETAILS

Enter the following information:

Simulation title:

Error Tolerances

Variable Reporting

Reporting interval: 0 (Seconds)

Freezing and Scanning Options

Variable freezing option: 0

Input scan option: 0

Simcon Queries

Do you want to allow algebraic variables to be inputs and outputs of the same unit ? No

Modsim Queries

Minimum Time Step (Seconds): 3600

Maximum Time Step (Seconds): 3600

Simulation stopping time (Seconds): 864000

Disable Freezing Variable feature ? No

Monitor the Simulation Time ? No

Use Reported Time for outputs ? No

Write diagnostic information ? No

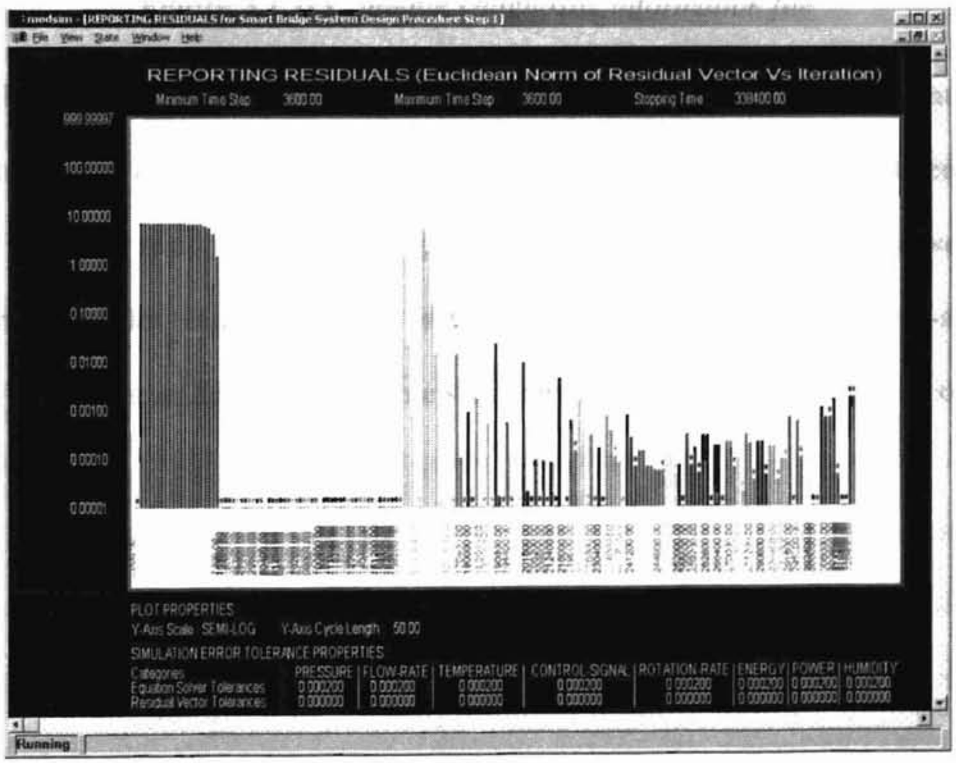
Plot residuals during simulation

Plot the Residuals on screen ? Yes

Y-Axis Magnification: 50

Y-Axis Scale: LOG

Figure 3.1.2-1. *Simulation Requirements dialog used to initiate residual reporting*


modsim - [REPORTING RESIDUALS for Smart Bridge System Design Procedure Step 1]
File View State Window Help
999.99997
100.00000
10.00000
1.00000
0.10000
0.01000
0.00100
0.00010
0.00001
Running

REPORTING RESIDUALS (Euclidean Norm of Residual Vector Vs Iteration)

Minimum Time Step 3600.00 Maximum Time Step 3600.00 Stopping Time 338400.00

PLOT PROPERTIES

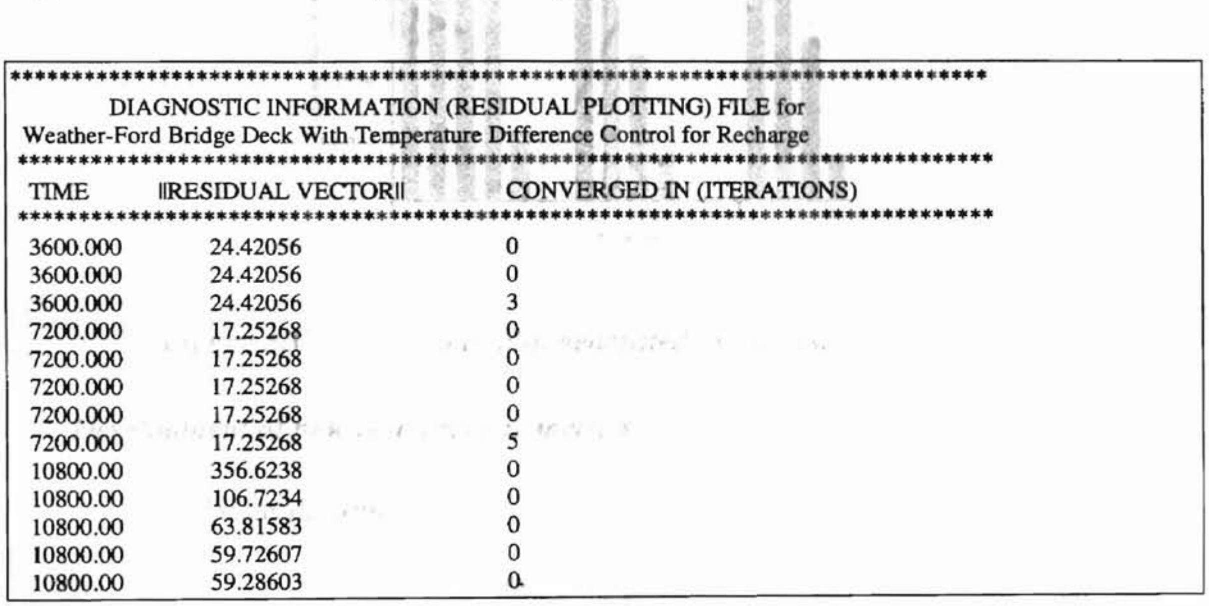
Y-Axis Scale: SEMI-LOG Y-Axis Cycle Length: 50.00

SIMULATION ERROR TOLERANCE PROPERTIES

Categories	PRESSURE	FLOW-RATE	TEMPERATURE	CONTROL-SIGNAL	ROTATION-RATE	ENERGY	POWER	HUMIDITY
Equation Solver Tolerances	0.000200	0.000200	0.000200	0.000200	0.000200	0.000200	0.000200	0.000200
Residual Vector Tolerances	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000

Figure 3.1.2-2. *Sample residual reporting screen (LOG scale)*

The program also generates an output file that contains the plotting data, including the x-axis and y-axis values from the residuals plot discussed above. This file has the file extension '. DIG'. It will be referred to as the Diagnostic Information file. Figure 3.1.2-3 shows a sample of the Diagnostic Information file.



```
*******************************************************************************
        DIAGNOSTIC INFORMATION (RESIDUAL PLOTTING) FILE for
 Weather-Ford Bridge Deck With Temperature Difference Control for Recharge
*******************************************************************************
 TIME        IIRESIDUAL VECTORII          CONVERGED IN (ITERATIONS)
*******************************************************************************
 3600.000        24.42056             0
 3600.000        24.42056             0
 3600.000        24.42056             3
 7200.000        17.25268             0
 7200.000        17.25268             0
 7200.000        17.25268             0
 7200.000        17.25268             0
 7200.000        17.25268             5
 10800.00        356.6238             0
 10800.00        106.7234             0
 10800.00        63.81583             0
 10800.00        59.72607             0
 10800.00        59.28603             0
```

Figure 3.1.2-3. *Sample Diagnostic information file*

The visual tool consists of a routine that can read the Diagnostic Information file and regenerate the residuals plot. Figure 3.1.2-4 shows a sample residual plot generated by visual tool. The residual plot is a bar-plot of Euclidean norm of the residual vector versus time. The plot features log scale on the y-axis and a linear scale on the x-axis. Each bar represents an iteration for a time step, between two tick marks, shown on the x-axis.

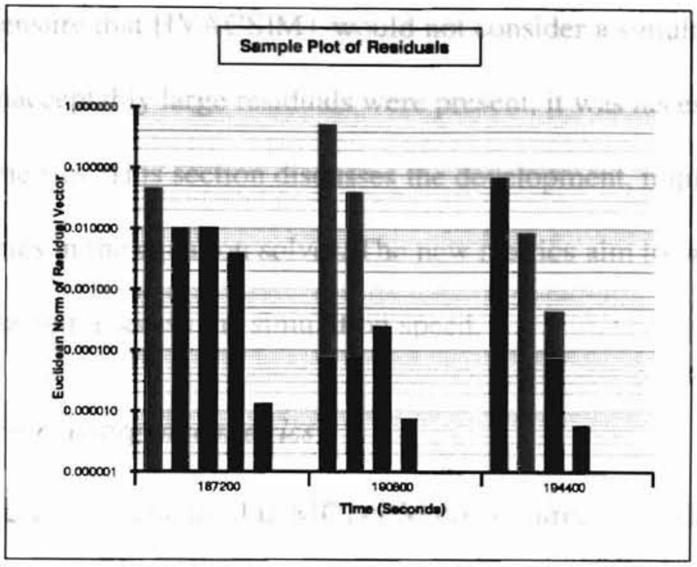


Figure 3.1.2-4. *Residual plot generated by the visual modeling tool*

3.2. Development of new convergence metrics

Since the initial conditions supplied by the user are usually poor, the convergence is often a problem particularly at the beginning of the simulation. If the initial conditions are too far from the actual solution, the equation solver may not converge. In some case, errors made by the solver can cause erratic solutions for several time steps, even if the solutions are convergent. The choice of the error tolerance for the equation solver is an important factor that influences this behavior.

Whenever the solution does not converge, MODSIM shows the warning message "Iteration not making good progress". This message is shown when the steps taken in the gradient direction, while searching for a global minimum in the domain, exceed 10. The number of iterations limits the gradient step bounds. For certain simulations, a closer investigation of the residuals, printed in the diagnostic report, revealed that the solution failed to converge even though it had satisfied all the stringent convergence criteria and did not show any warning messages.

In order to ensure that HVACSIM+ would not consider a simulation time step converged when unacceptably large residuals were present, it was necessary to establish new convergence metrics. This section discusses the development, implementation and testing of new metrics in the equation solver. The new metrics aim to provide good convergence criteria and increase the simulation speed.

3.2.1. Design of new convergence metrics

The convergence criteria used in MODSIM do not directly consider the absolute value of the residuals. In a system that simulates mainly temperatures (which are mostly in the range of –10 to 60 degrees centigrade), a residual greater than 1 degree affects the results significantly. Moreover, in computing the heat flux, where the temperature differential is multiplied by high flow rates, the effect of high residuals is very significant. The only limiting criterion of the solver is the step size.

The major shortcoming of the existing HVACSIM+ convergence criteria is that it does not consider the absolute value of the residuals. The effect of not taking magnitudes was explained in the Section 3.1.3. This fact motivated the development of new convergence criteria.

The convergence criteria, described in the Section 2.4.4 of Chapter 2, are represented as:

$$\left\|(x^{k+1}-x^{k})\right\| \le e_{t}\left\|x^{k}\right\| \text{Or } \left\|f(x)\right\| = 0 \qquad (3.2.1\text{-}1)$$

where, x is the solution vector, $f(x)$ is the residual vector, and e_t is the error tolerance.

For complex systems, the residual functions will rarely be equal to a numerical value of zero. It is unlikely that the calculated value of the functions would ever be exactly zero because of rounding errors in computer calculations. Even if there were no rounding errors, no algorithm is guaranteed to find such a solution in a finite amount of time.

As an alternative, a tolerance value can be used for the residual functions. This modifies $f(x)$ in the Equation 3.3.1-1 as:

$$\|f(x)\| \le e_{residual} \qquad (3.2.1\text{-}2)$$

where, $e_{residual}$ is a small value. However, identifying an approximately small value for the tolerance may not be easy. Equation 3.3.1-2 faces another problem when the magnitude of the solution vector can vary over a wide range. Suppose that the measurement units were changed. For example, suppose that instead of measuring the objective in terms of kilometers it was measured in terms of millimeters. This would cause the function to be multiplied by 10^6 and hence would cause the norm to be multiplied by 10^6. Simply changing the units would make the convergence test much more difficult to satisfy unless the error tolerance was also changed to reflect the change in units.

During the simulation, the solution vector holds values for different kinds of variables irrespective of their units. For example, a solution vector might contain variables that are temperature and pressure variables. Having a single tolerance value that is applied to all variables may lead to inaccurate results. Using separate tolerance values for different categories of variables could solve this problem by allowing the user to assign different convergence criteria to different types of variables. For example, mass flow rates in a system may be less important than temperatures. A large tolerance may be applied to variables that represent mass-flow-rates and smaller tolerance to temperature variables. This way the algorithm would spend more time on important variables. Let e_{tol_vec} be an array that contains the tolerance values to be applied on the solution vector. Each element in e_{tol_vec} is based on the category of the variable in the solution vector. Let $e_{residual\_vec}$ be an array that contains the tolerance values to be applied to the residual vector. Each element in $e_{residual\_vec}$ is based on the category of the variable in the residual vector. The new convergence criteria may be represented as:

$$\left\|(x^{k+1}-x^{k})\right\| \le \left\|e_{tol\_vec}x^{k}\right\| \text{ Or } \left|f(x)\right| \le e_{residual\_vec} \tag{3.2.1-3}$$

From the Equation 3.3.1-3, it can be observed that the residual vector is no longer normalized. The absolute value of each scalar in the residual vector is compared with the corresponding tolerance value in the $e_{residual\_vec}$ vector. In addition, if the vector, e_{tol_vec}, contains the same values and the vector, $e_{residual\_vec}$, contains zeros then the Equation 3.3.1-3 represents the Euclidean norm, given in the Equation 3.3.1-1.

3.2.2. Performance evaluation of the new convergence metrics

In order to test the new convergence metrics two systems were selected: the Geothermal Smart Bridge system and the Single zone air handler system with a cooling coil. These systems were simulated using the new convergence criteria and the output generated was analyzed.

The Geothermal Smart Bridge system was developed as a result of ongoing research at Oklahoma State University. The simulation models a bridge deck heating system that will eliminate preferential icing on bridges and reduce bridge maintenance costs. This system makes use of ground source heat pump technology and hydronic circuits embedded in the concrete bridge deck. The system is comprised of two parts, namely, the load-side and the source-side. The load-side sub-system consists of the bridge deck. The source-side consists of a ground loop heat exchanger (GLHE) usually in the form of a vertical borehole field. Heat transfer from the earth to the fluid takes place in the ground loop heat exchanger. A gang (bank) of water-to-water heat pumps connects the load-side and source-side sub-systems. A controller is used to determine the number of heat pumps to be operated in the system. The controller also sends a designed value of mass flow rates of the fluid to the load-side and source-side sub-systems. In addition, the GLHE model supplies the average temperature of the fluid in the exchanger to the controller.

During the development of Smart Bridge technology, several configurations were simulated. One such configuration, called "Design Procedure Step 2", will be used as a test case. In this procedure, only the load-side sub-system is simulated with constant source-side conditions. A constant value of $-3°C$ is supplied to the heat pump as the source-side entering fluid temperature. The average fluid temperature in the GLHE model, supplied to the controller, is set at $-100°C$. The Figure 3.2.2-1 shows the system schematic for this configuration.

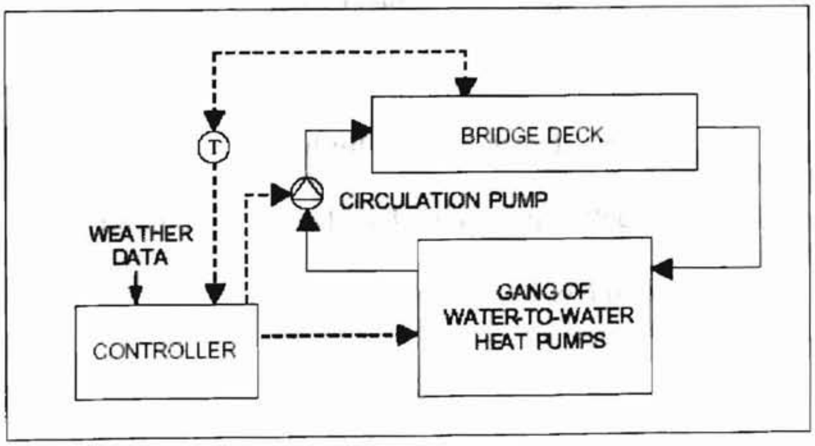


Figure 3.2.2-1. *Geothermal Smart Bridge (Design Procedure Step 2)*

The components and their functions are listed below:

- TYPE 700: Slab model – This model represents the bridge deck, the temperature of which is maintained at a specific value.
- TYPE 711: Gang of Water to Water Heat Pumps model – This model simulates a series of heat pumps. The number of heat pumps in the series is determined by the TYPE 730, the linear proportional controller.
- TYPE 750: Simple Pump model – This type is similar to TYPE 1 Pump that is distributed with HVACSIM+.
- TYPE 730: Linear Proportional Controller for heating – This model determines the number of heat pump pairs that should be operated based on the conditions of the bridge deck.

The design consists of two superblocks. The flow controller that decides the number of heat pumps is the only unit in Superblock #2. All the other components are grouped together in Superblock #1. A design data set, consisting of one snow event, was created for the simulation to be used as boundary values. The equation sets in both Superblocks consist of only temperature variables. The average bridge-surface temperature was used for analyzing the results.

The single zone air handler with a cooling coil is discussed in Section 5.1.5 of Chapter 5. The system consists of multiple blocks. Block 1 represents a cooling coil control loop, block 2 represents an air-handling unit, and block 3 represents a room configuration with a supply zone, a conditioned zone and a return duct.

The new convergence criteria were applied to the selected test cases. The overall execution time and residuals were inspected during the simulations. The test results and inferences made from them are presented below.

Since the new convergence criteria are conditioned by the error tolerances for the solution and the residual vectors, only they were changed in the tests. The primary aim was to show that the results generated are acceptable when different limits are applied to the residuals. The improvement in the simulation speed is automatically achieved when the convergence criteria are not rigid. Comparative analysis was performed using the results obtained with the original criteria.

All the simulations were run on a computer using an AMD-Athlon Processor running at 750 MHz and having 128 MB RAM. The simulation execution times were estimated by making two calls to the Fortran function, SECNDS. The first call sets the timer to zero seconds and the second call estimates the time taken since the first call. The code to initialize, simulate and report results is placed between the two call statements.

The visual modeling tool was used to plot the residuals generated during the simulations. However, the plots have been modified so that each time step is drawn with a different color shade making the plots more readable.

3.2.3. Simulation results for the Smart Bridge system

In order to conduct a comparison analysis, the Smart Bridge model was first simulated using the original convergence criteria. The error tolerance applied to the solution vector was selected as 0.0002. The simulation time step was fixed at 3600 seconds (1 hour). A simulation stopping time of 338400 seconds (approximately 4 days) was selected.

The simulation execution time was 45.2 seconds. The residuals in the simulation are zero until the snow event begins at 169200 seconds, as shown in Figure 3.2.3-1. This is because the heat pumps are not running when there is no snow or ice formation on the bridge and hence, there are no equations to solve in the system. The residuals are high during the first few iterations of every time step taken during the snow event. As the iterations for a time step proceed, the equation solver is getting closer to finding the solution for that time step. This can be observed by the reduction of the residuals. The solution does not converge when the time is 216000 seconds, as shown in Figure 3.2.3-2. The solutions converge for all other time steps.

Figures 3.2.3-1 and 3.2.3-2 show bar-plots of Euclidean norm of the residual vector versus time when the snow event begins and during the time step when the algorithm does not converge, respectively. The plots feature log scale on the y-axis and a linear scale on the x-axis. Each bar represents an iteration for a time step, between two tick marks, shown on the x-axis. Time steps can be distinguished by the color of the bars.

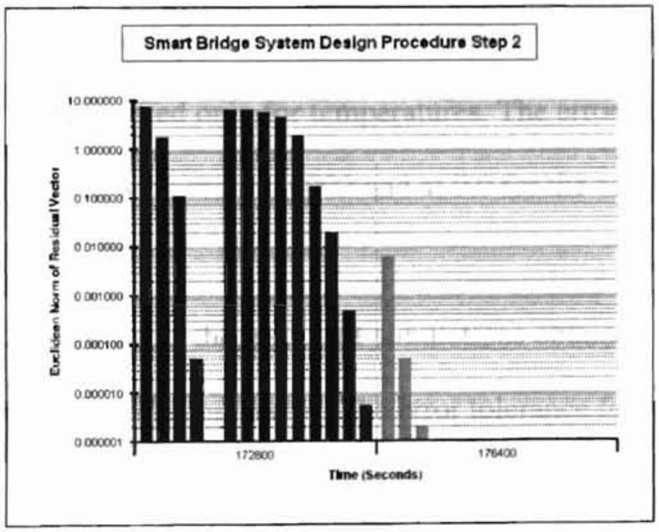


Figure 3.2.3-1. *Plot of residuals when the snow event begins*

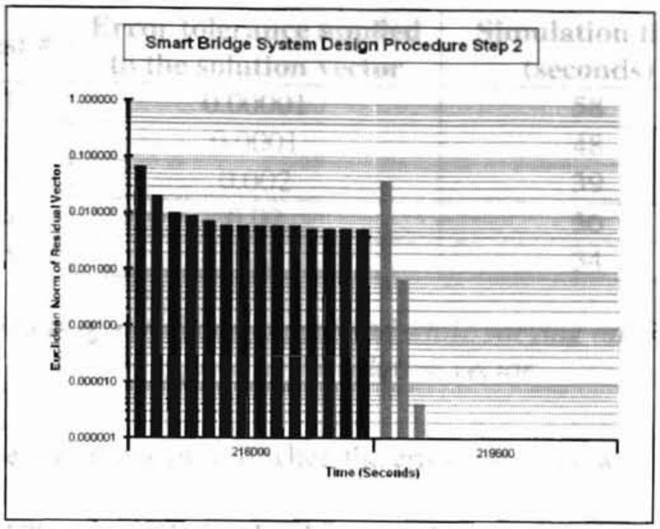


Figure 3.2.3-2. *Plot showing the time step (216000 seconds) when the solution did not converge*

The Smart Bridge model was then simulated using various test conditions. Three test environments were selected, which include:

- varying only the error tolerance applied to the solution vector
- varying only the error tolerance applied to the residual vector
- varying both the error tolerance for the equation solver and the error tolerance for the residual vector

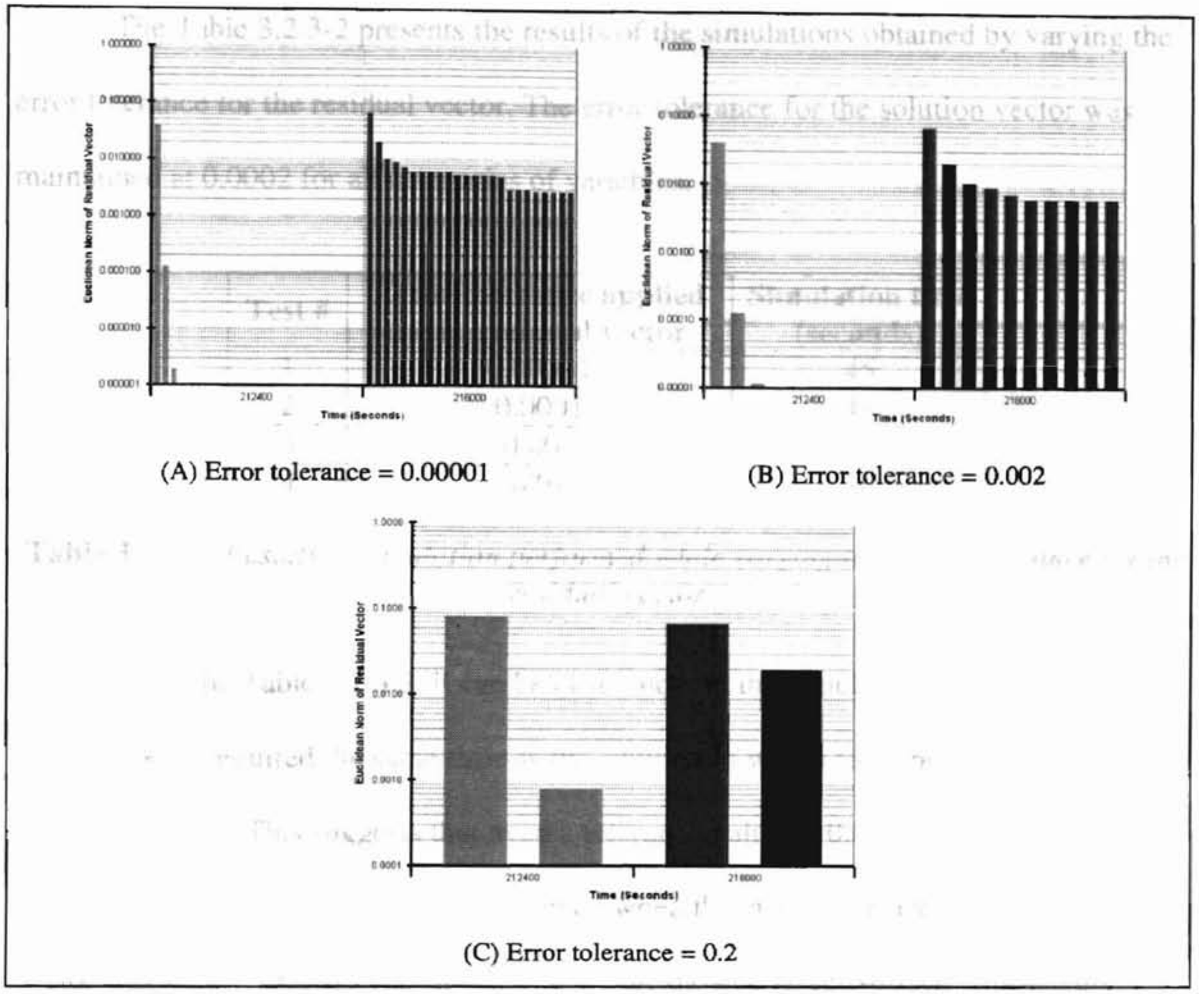
Since the equation sets defined for the model include only temperature variables, the error tolerances were specified only for temperatures. The error tolerances for all other categories of variables were set at 0.0002 and at 0.0 for the equation solver and for the residual vector, respectively.

The Table 3.2.3-1 presents the results of simulations obtained by varying only the error tolerance applied to the solution vector. The error tolerance for the residual vector was maintained at zero for all categories of variables.

Test #	Error tolerance applied to the solution vector	Simulation time (seconds)
1	0.00001	58
2	0.0001	48
3	0.002	39
4	0.02	36
5	0.2	34

Table 3.2.3-1. *Results of simulation performed while varying only the error tolerance applied to the solution vector*

Convergence was not achieved when the error tolerance was set to 0.00001. As the error tolerance was reduced to 0.002, the simulation output matched the output generated by using the original criteria with the error tolerance for the solution vector was set at 0.0002. When the error tolerance was further reduced, the simulation run time was also reduced since convergence was achieved within the first three iterations for all time steps. Figure 3.2.3-3 shows three residual plots for different error tolerances for the time steps, 212400 and 216000 seconds. The error tolerance values for plots (A), (B), and (C) are 0.00001, 0.002, and 0.2, respectively. Figure 3.2.3-4 shows the plot of the average bridge-surface temperature against time for the original criteria (error tolerance of 0.0002) and the new criteria with the error tolerances value set at 0.2. As shown in the figure, a less restrictive error tolerance results in a larger error band around the deck temperature. The fluctuations begin at the time 280800 seconds and end at about 334800 seconds. At all other times, the simulation converges to the same solution for both criteria.



(A) Error tolerance = 0.00001

(B) Error tolerance = 0.002

(C) Error tolerance = 0.2

Figure 3.2.3-3. *Plot of residual for different error tolerance for the solution vector*

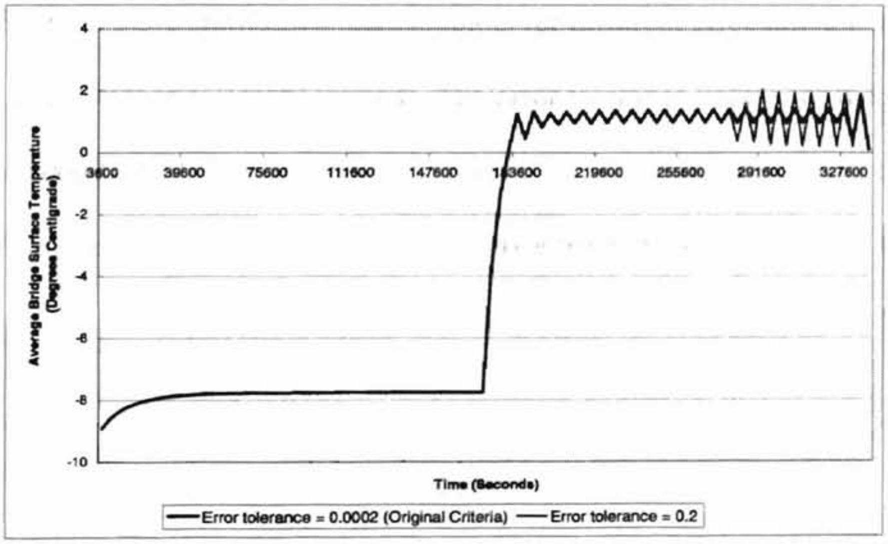


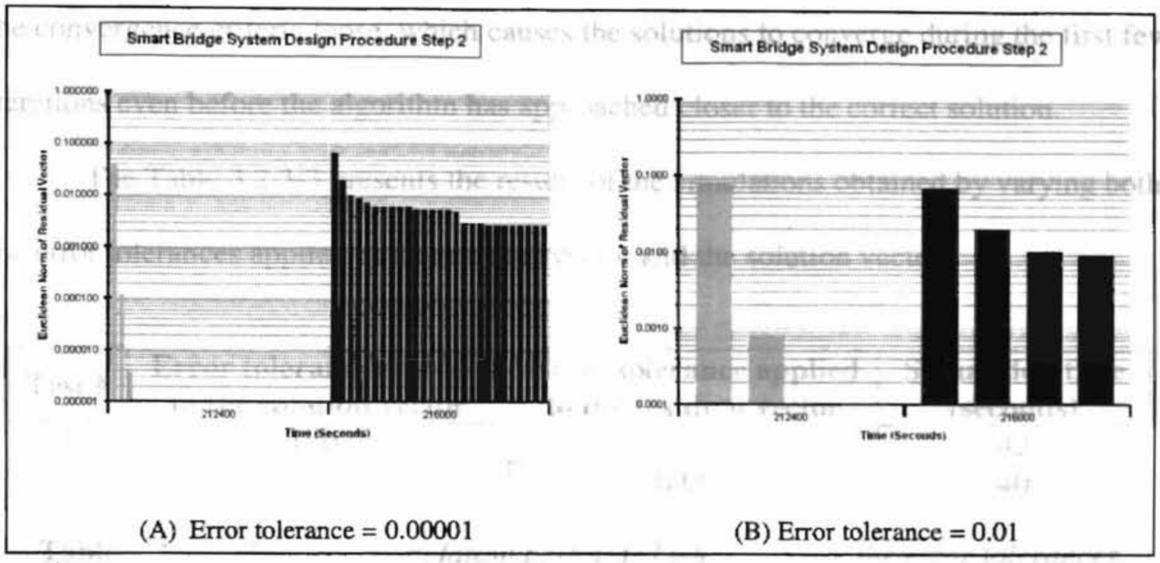
Figure 3.2.3-4. *Plot of the average bridge-surface temperature against time when the error tolerance for the solution vector is varying*

The Table 3.2.3-2 presents the results of the simulations obtained by varying the error tolerance for the residual vector. The error tolerance for the solution vector was maintained at 0.0002 for all categories of variables.

Test #	Error tolerance applied to the residual vector	Simulation time (seconds)
1	0.00001	45
2	0.0001	41
3	0.001	40
4	0.01	36

Table 3.2.3-2. *Results of simulation performed while varying the error tolerance for the residual vector*

From the Table 3.2.3-2, it can be observed that the simulation with error tolerance set at 0.00001 required the same time as the simulation with a tolerance of 0.0 (the original criteria).. This suggests that even a tolerance value of 0.00001 is rigid to achieve convergence. Convergence was not achieved when the error tolerance for the residual vector was set at 0.00001. The simulation output showed larger surface temperature fluctuations when the error tolerance was set at 0.01, suggesting that the value selected was high. Figure 3.2.3-5 shows two residual plots for different error tolerances at the time steps taken at 212400 and 216000 seconds. The error tolerance values for the plot (A), and (B) are 0.00001, and 0.01, respectively. Figure 3.2.3-6 shows the plot of the average bridge-surface temperature against time for the error tolerances, 0.00001 and 0.01.



(A) Error tolerance = 0.00001

(B) Error tolerance = 0.01

Figure 3.2.3-5. *Plot of residual for different error tolerance for the residual vector*

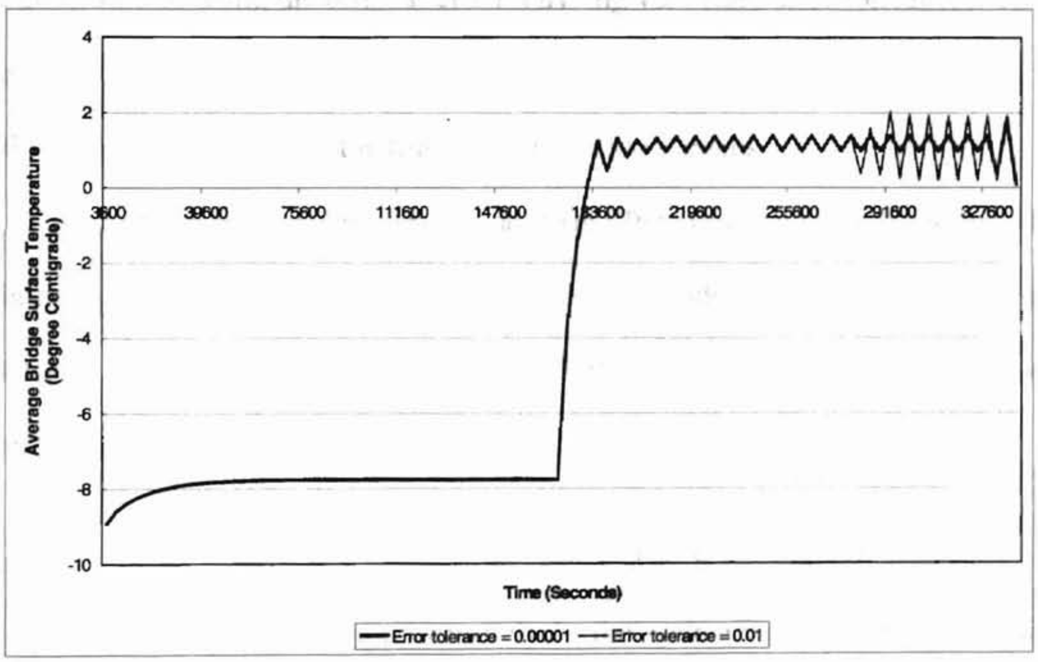


Figure 3.2.3-6. *Plot of average bridge-surface temperature against time when error tolerance for the residual vector is varying*

It can be observed from the Figures 3.2.3-4 and 3.2.3-6 that setting a tolerance value of 0.2 for the solution vector or a tolerance value of 0.01 for the residual vector produces similar results. Using unreasonably high values for the error tolerances makes the convergence criteria loose, which causes the solutions to converge during the first few iterations even before the algorithm has approached closer to the correct solution.

The Table 3.2.3-3 presents the results of the simulations obtained by varying both the error tolerances applied to the residual vector and the solution vector.

Test #	Error tolerance applied to the solution vector	Error tolerance applied to the residual vector	Simulation time (seconds)
1	0.002	0.0001	42
2	0.002	0.001	40

Table 3.2.3-3. *Results of simulation performed while varying the error tolerances*

The output generated through the tests shown in the above table matched the output generated by using the original criteria having very rigid error tolerances. The simulation running time was reduced. Similar to the previous test conditions, the algorithm did not converge for the time step at 216000 seconds.

The saving in the simulation run time is significant for annual simulations. Using the original criteria, the simulation would take 4210 seconds (1.17 hours, approximately) to end. Using the conditions for Test #2 shown in the Table 3.2.3-3, the simulation would need only 3728 seconds (1.04 hours, approximately).

3.2.4. Simulation of Single zone air handler system with a cooling coil

In order to conduct a comparison analysis, the system was first simulated using the original convergence criteria. The error tolerance for the solution vector was selected as 0.0005. Variable time step is used in the simulation with a minimum time step of 0.1 seconds and a maximum time step of 300 seconds. A simulation stopping time of 86400 seconds was selected.

The simulation execution time was recorded as 37.2 seconds for 7554 overall iterations. During many time steps, the equation solver demonstrated its characteristic of reporting convergence although large residuals were present in the solutions. The Table 3.2.4-1 shows a list of selected time steps, the Euclidean norm of the residual vector when the solver reported convergence and the number of iterations taken for that time step. The data in the table was extracted from the Diagnostic Information file generated during the simulation. The data in the table illustrates the convergence problem mentioned above.

Time (seconds)	Norm of Residual vector	Number of iterations
7858.3	0.11028	19
54940.8	0.00759	270
56140.8	0.00376	24
71740.8	0.007326	29
80440.8	0.031	37

Table 3.2.4-1. *Performance of the simulation at selected time steps*

The solution vector consists of only pressure, mass flow rate and temperature variables. The system was simulated using various combinations of error tolerances applied to these variable categories. The first set of tests involves varying the error tolerance applied to the solution vector for pressures, mass flow rates and temperatures. The second set of tests involves varying the error tolerance applied to the residual vector for these categories.

In the first set of tests, the error tolerance applied to the residual vector was maintained at zero and the error tolerance applied to the solution vector was changed. A tolerance of 0.005 was applied to each category in the solution vector and the performance of the solver was analyzed while maintaining the tolerance for other categories at 0.0005. A summary of the test results is shown in Table 3.2.4-2. The table displays the error tolerance value, category on which the test is applied and simulation run time for the test. The table also displays the total number of iterations performed in the simulation.

Test #	Error tolerance applied to the solution vector	Categories selected	Simulation time (seconds)	Number of iterations
1	0.005	Pressure	27.8	5453
2	0.005	Flow Rate	30	6052
3	0.005	Temperature	25.1	4934
4	0.05	Pressure and Flow rate	27.4	5460
5	0.05	Flow rate and Temperature	24	4579

Table 3.2.4-2. *Results of simulation performed while varying only the error tolerance applied to the solution vector*

From the above table, it can be observed that by reducing the error tolerance on pressures and temperatures we achieve faster convergence compared to reducing tolerance on flow rates. The solution vector contains more pressures than any other categories of variables. Thus, by reducing the tolerance on the pressure variables, the solver required less effort to perform computations. In addition, the presence of only one temperature in the solution vector influenced faster convergence in the simulation when the tolerance value for temperatures was increased.

The simulation output matched the original output for the first three test cases. Test #4 and Test #5 generated results that showed fluctuations in output at irregular time intervals.

In the second set of tests, the error tolerance applied to the solution vector was maintained at 0.0005 for all categories of variables and the error tolerance applied to the residual vector was changed. A summary of the test results is shown in Table 3.2.4-3. The table displays the error tolerance value, category on which the test was applied and simulation run time for that test. The table also displays the total number of iterations performed in the simulation. The error tolerance values for categories not part of the tests were maintained at 0.01.

Test #	Error tolerance applied to the residual vector	Categories selected	Simulation time (seconds)	Number of iterations
1	0.01	All categories	24.34	4721
2	0.1	Pressure	17.26	3111
3	0.1	Flow rate	24.73	4721
4	0.1	Temperature	24.33	4677
5	0.1	Pressure and Flow rate	20.33	3729
6	0.1	Flow rate and Temperature	23.9	4677

Table 3.2.4-3. *Results of simulation performed while varying only the error tolerance for the residual vector*

Convergence was achieved in all the test cases shown in Table 3.2.4-3. However, the solver reported convergence when the residuals were large. When the error tolerance was set at 0.01, as in Test #1, the simulation generated output that matched the output generated when the tolerance was set at 0.0. The simulation run time was the shortest when the tolerance on the pressure variables was set at 0.1. When the tolerance of both the pressure and flow rate variables was controlled, the simulation showed fastest convergence with only 3729 overall iterations.

The two test sets show that the new convergence criteria cannot rectify the convergence problem in the equation solver. However, they provide means to control the simulation to yield similar results with shorter run times.

CHAPTER 4

TECHNICAL OVERVIEW OF THE USER INTERFACE

4.1. Concept

The Visual Modeling Tool, discussed in the Chapter 3, provides a powerful graphical user interface for HVACSIM+. The interface provides:

- a workspace for representing information about a HVAC system.
- a means to define the environment of such a system in terms of boundary values.
- facilities to simulate the system.
- a means to plot the results.

4.2. Architecture

The core components of the application include graphical administration tools, the boundary file editor, the system simulator, and the plotting tool.

The graphical administration tools create components in the workspace, provide sufficient graphical information on the component icons, perform various editing operations like cut, copy and paste, and remove icons from the workspace. These tools also handle operations like dragging components around the workspace, to provide a visually well-organized system configuration.

The boundary file editor tries to mimic a traditional spreadsheet organizer having a matrix of rows and columns, where columns represent boundary conditions and rows represent data values for all the boundary conditions at specific times. It provides facilities for creating boundary value files, which supply input data to the simulation. Additional support allows for associating columns of data to boundary conditions in the simulation, and handling editing operations on the data between applications (like Microsoft Excel) using the windows clipboard as an intermediate buffer.

The simulation manager analyzes the workspace for basic errors and sets up an environment for calling back-end applications like SLIMCON and MODSIM. The simulation manager identifies errors from MODSIM and interprets these errors for the user. It also modifies the output files to add additional information, like variable names assigned by the user, to clearly differentiate between them in these files.

The major purpose of the plotting tool is to plot the data in the output files. The user is allowed to select reported variables and the tool reads the variable data and the associated time values from the output file.

4.3. Data Format

All the information entered by the user is written into an input file with the default file extension of '. DAT'. The user can save his work any time while using the visual tool using the 'Save' or 'Save As' options under the 'File' menu. This input file has a special file format, shown in the Figure 4.3-1, that include the following:

- Header: A header identifying that the file belongs to the application.
- Block and Superblock details: Information about the number of blocks and superblocks in the workspace.
- Unit Information: Information about the various units involved including the location of the icons in the workspace, superblock and block numbers, type

number, initial and parameter values, reporting status of all the input and output ports, and variables names.

- Boundary condition information: Information about the time-dependent and constant boundary conditions involved including their location in the workspace, name, and constant value.
- Connection information: Information about the connections involved including the originating port identities and ending port identities.
- Simulation information: Simulation details including the title, error tolerances, simulation times, and answers to questions related to diagnostics.
- Boundary value information: Information relating to the boundary file such as the header names of columns and the data values in the grid.

After execution of the simulation, MODISM generates three output files, discussed in the sections 2.3.4. The application appends the raw output file with additional information required for plotting the results. The format of the additional information is shown in the Figure 4.3-2. The other two files generated are not modified by the application. The plotting routine can also read output files without having the additional information. Each reported variable is identified with its Unit and TYPE numbers. For example, an inlet temperature for a fan or pump may be represented as Unit 1, TYPE 1: Inlet Temperature.

MODSIM also generates the Diagnostic information file, mentioned in Section 3.2.2. This file is not modified by the visual tool and is used only to plot the residuals.

```
            ********************************************
Header      * VISUAL MODELING TOOL FOR HVACSIM+  *
            *                  Input File              *
            ********************************************
            <Blank line>
Block       <Number of Superblocks>
&           <Number of Blocks in each Superblock on separate lines>
Superblock  <Active Superblock number>
Details     <Active Block number>
            <Number of units involved in the simulation>
Unit        [List of all units involved in the following format for each unit]
Information <X-position>, <Y-position>, <Superblock>,<Block>,<Type number>, <Initial values>, <Parameter
            values>, <Reporting status; 0 or 1>
                    <Variables names, each on a separate line>
            <Number of boundary conditions in the simulation>
Boundary    [List of all boundary conditions involved in the following format for each condition]
Condition   <X-Position>, <Y-Position>, <Is it a constant condition; 0 or 1>, <Category index>, <Constant
Information value>, <Name>
            [For constant conditions, Name is *.]
            <Number of connections in the simulation>
Connection  [List of all connections involved in the following format for each connection]
Information <From unit>, <From port>, <From category>,  <To unit>, <To port>, <To category>
            [Connections involving boundary conditions have –1 for the From port and From category values.]
            <Simulation title>
            <Relative error tolerance>
            <Absolute error tolerance>
            <Tolerance for simultaneous equations>
            <Error tolerance for time interval>
Simulation  <Reporting interval>
Information <Variable freezing option>
            <Variable scanning option>
            <Minimum time step value>
            <Maximum time step value>
            <Simulation stopping time>
            <Answers, one in each line, to questions related to diagnostics>
            <Total number of headers defined>
            [List of header names in the following format:]
Boundary    <Header name>
Value       <Total number of rows>
Information <Total number of columns>
            [Boundary value data in the following format:]
            <Value in each column; comma delimited>
```

Figure 4.3-1. *Input file structure*

```
[The raw output listing]
*****************************************************
<Simulation title>
<Date and time of creation>
<Location of the corresponding input file>
<Number of Superblocks>
[For each Superblock, the reported variable information is represented in the following format:]
<Number of Reported variables in the Superblock>
<Variable name, one on each line>
```

Figure 4.3-2. *Format of additional information in the raw output file*

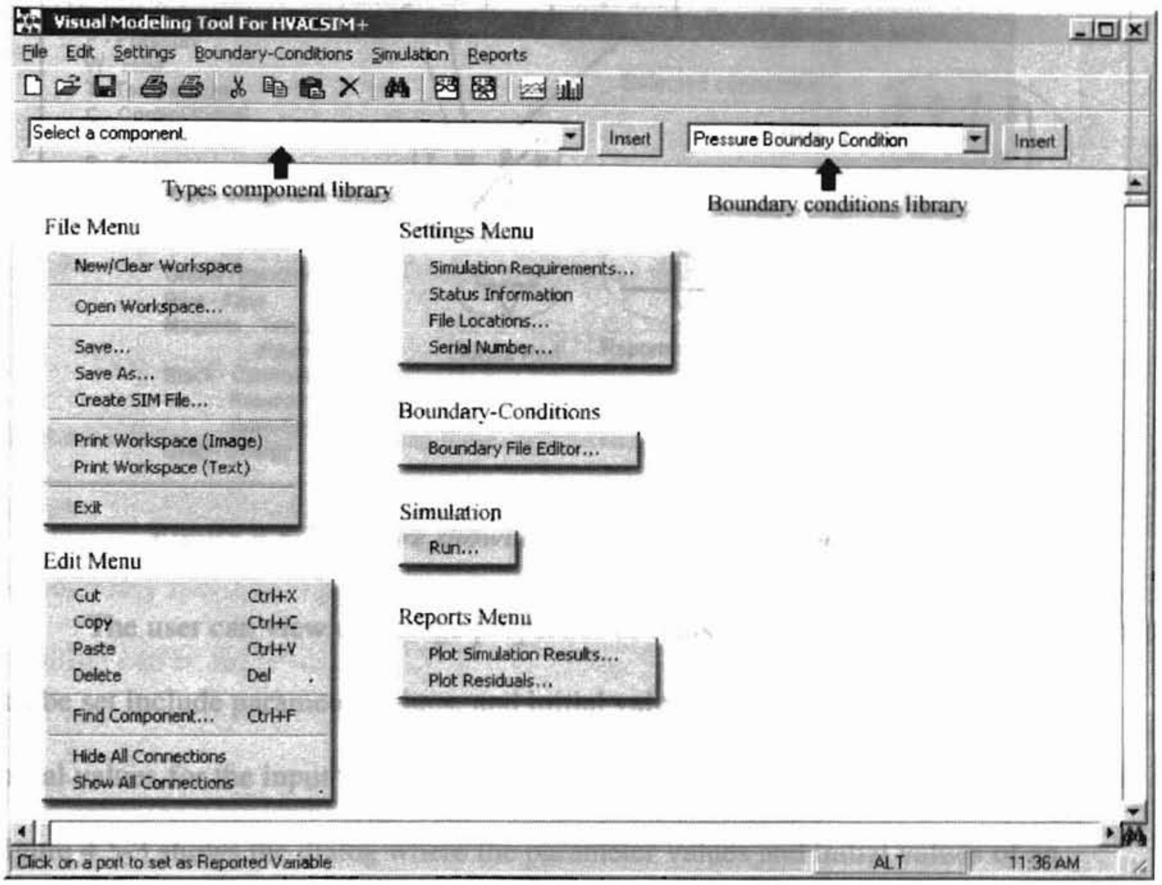
4.4. Configuration

Configuration of the product is carried out by identifying the correct Types component library file (TYPAR.DAT) and copying the file to the application directory before starting the application. The Types component library contains information about all the types which SLIMCON and MODSIM can identify. The application reads this file while starting and prepares a list containing the type numbers and descriptions of each TYPE in the file.

Since the application is based on an event-driven approach, discussed in the Section 3.2, there is no sequence of steps to follow to configure a simulation. However, a task tree is presented in the Appendix B to help build a simulation. The task tree may be used as a checklist to ensure that the simulation is completely configured. The tree lists the processes to be performed and shows options, in square brackets, on how to perform them.

4.5. Workspace

The main window consists of a comprehensive menu that supports the important operations. The Figure 4.5-1 shows the available menu and toolbar options. Operations such as opening, saving and printing a workspace can be performed using the File menu. The Edit menu options provide facilities to perform edition operations such as cut, copy, paste, delete and find. The Edit menu options work with units, boundary conditions and connections selected by the user. For large and complicated simulations, involving many components, the Hide All Connections and Show All Connections menu options can be used to hide or show the connections in the workspace. The Settings menu can be used to set the simulation requirements, the locations of MODSIM and SLIMCON and the serial number for the workspace. The Boundary File Editor can be accessed through the Boundary-Conditions menu. A simulation can be initiated and executed using the Simulation menu. The simulation output can be plotted using the options in the Reports menu.



Click on a port to set as Reported Variable

Figure 4.5-1. *Interface menu and toolbars*

The Types component library and the Boundary conditions library shown in the Figure 4.5-1 can be used to insert HVACSIM+ models and boundary conditions in the workspace, respectively. These components are visually represented as icons in the workspace. The Figure 4.5-2 shows the various components and their icon descriptions using an example.

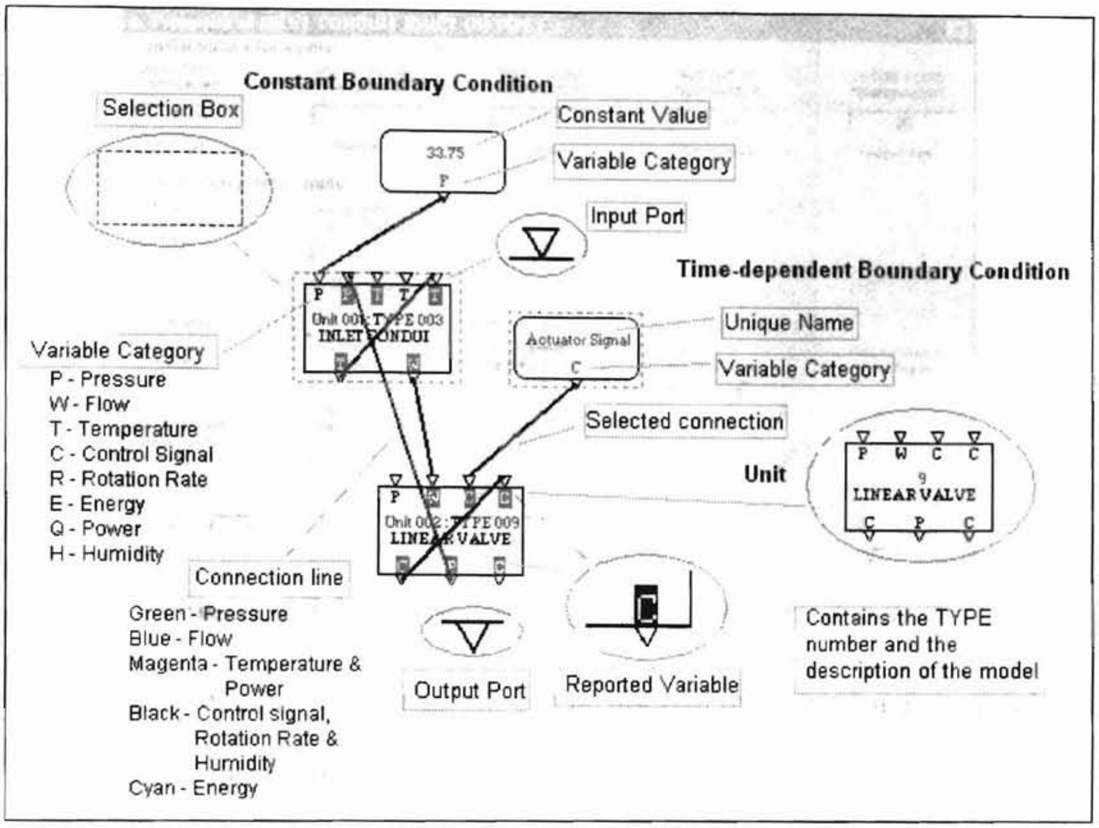
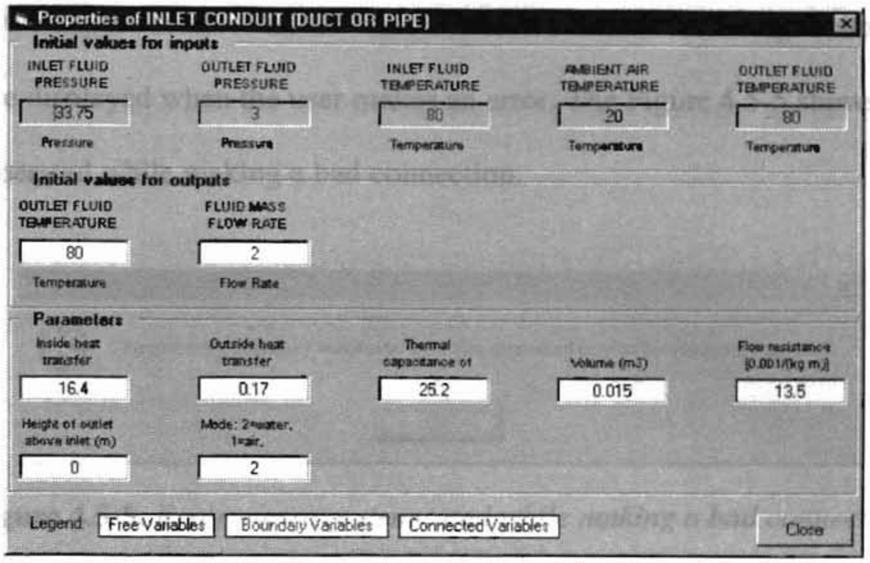


Figure 4.5-2. *Figure showing the description of component icons*

The user can view or change properties of units in the workspace. Properties that can be set include parameter values, and initial values for output and input variables. Initial values for the input variables can be set using constant boundary conditions. The Figure 4.5-3 shows the dialog where the parameter values and initial values of an example unit have been set.



Properties of INLET CONDUIT (DUCT OR PIPE)

Initial values for inputs

INLET FLUID PRESSURE: 33.75 (Pressure)

OUTLET FLUID PRESSURE: 3 (Pressure)

INLET FLUID TEMPERATURE: 80 (Temperature)

AMBIENT AIR TEMPERATURE: 20 (Temperature)

OUTLET FLUID TEMPERATURE: 80 (Temperature)

Initial values for outputs

OUTLET FLUID TEMPERATURE: 80 (Temperature)

FLUID MASS FLOW RATE: 2 (Flow Rate)

Parameters

Inside heat transfer: 16.4

Outside heat transfer: 0.17

Thermal capacitance of: 25.2

Volume (m3): 0.015

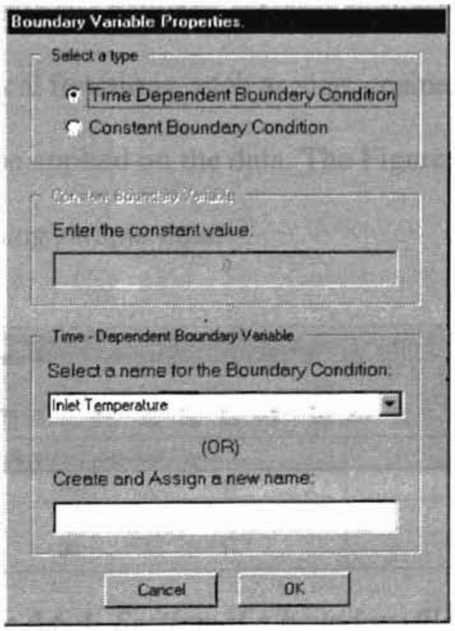
Flow resistance [0.001/(kg m)]: 13.5

Height of outlet above inlet (m): 0

Mode: 2=water, 1=air.: 2

Figure 4.5-3. *Dialog used to set the parameters of units*

Two kinds of boundary conditions can be inserted in the workspace, namely, the constant boundary conditions and time-dependent boundary conditions. The interface provides an efficient way of attaching time-dependent boundary variables to columns in the boundary file. The Figure 4.5-4 shows the dialog where the properties of a boundary condition can be set or modified.



Boundary Variable Properties

Select a type

● Time Dependent Boundary Condition

○ Constant Boundary Condition

Constant Boundary Variable

Enter the constant value: 0

Time - Dependent Boundary Variable

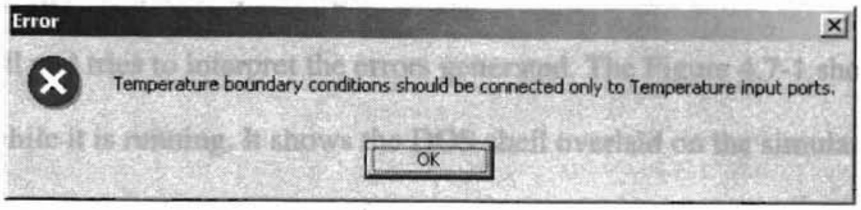
Select a name for the Boundary Condition: Inlet Temperature

(OR)

Create and Assign a new name:

Figure 4.5-4. *Dialog used to set the properties of boundary conditions*

The workspace features an efficient and effective error handling. Suitable error messages are displayed when the user makes an error. The Figure 4.5-5 shows an error message generated while making a bad connection.



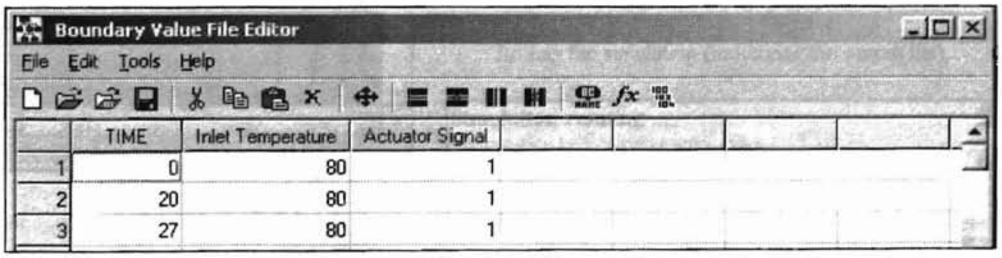
Temperature boundary conditions should be connected only to Temperature input ports.

Figure 4.5-5. *Error message generated while making a bad connection*

4.6. Boundary File Editor

This section describes the features of the boundary file editor and its sub-components.

The boundary file editor represents boundary files as spreadsheets. It provides functions to create and extract data from boundary value files into the workspace. Edition operations like cut, copy and paste can be performed on the data in the editor. The editor supports copy and paste operations between external applications like Microsoft Excel. In addition, several mathematical functions, such as sine, cosine, average, maximum, and minimum, are available to be applied on the data. The Figure 4.6-1 shows a section of the boundary file editor containing sample data.



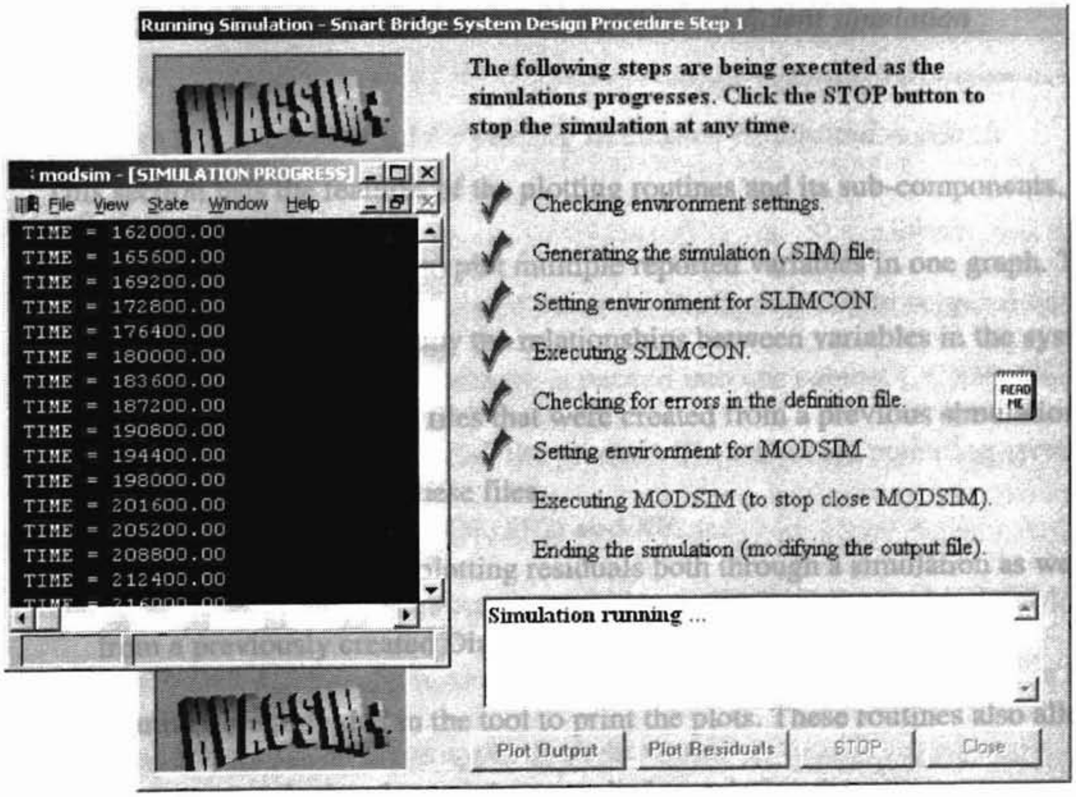
	TIME	Inlet Temperature	Actuator Signal
1	0	80	1
2	20	80	1
3	27	80	1

Figure 4.6-1. *Section of a boundary file editor*

4.7. Simulation

This section describes the features of the simulation manager and its sub-components.

The program efficiently manages the execution of SLIMCON and MODSIM in the DOS shell and tries to interpret the errors generated. The Figure 4.7-1 shows a simulation while it is running. It shows the DOS shell overlaid on the simulation manager. Efficient error handling is performed to interpret any errors generated. The Figure 4.7-2 shows an error message generated when boundary values are not identified while running a simulation. There is also no need for the user to open the output files and study them manually since the interface can plot the graphs immediately after the simulation ends.



Running Simulation - Smart Bridge System Design Procedure Step 1
HVACSIM+
modsim - [SIMULATION PROGRESS]
File
View
State
Window
Help
Plot Output
Plot Residuals
STOP
Close
READ ME

The following steps are being executed as the simulations progresses. Click the STOP button to stop the simulation at any time.

✓ Checking environment settings.

✓ Generating the simulation (.SIM) file.

✓ Setting environment for SLIMCON.

✓ Executing SLIMCON.

✓ Checking for errors in the definition file.

✓ Setting environment for MODSIM.

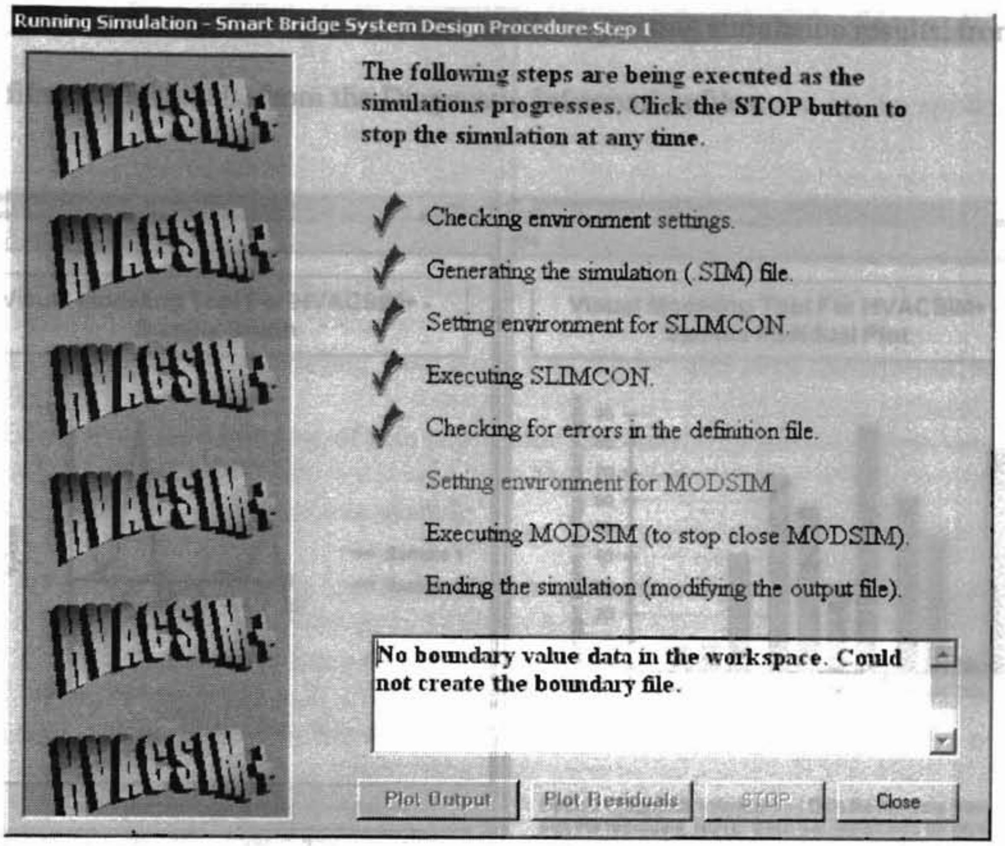
Executing MODSIM (to stop close MODSIM).

Ending the simulation (modifying the output file).

Simulation running ...

```
TIME = 162000.00
TIME = 165600.00
TIME = 169200.00
TIME = 172800.00
TIME = 176400.00
TIME = 180000.00
TIME = 183600.00
TIME = 187200.00
TIME = 190800.00
TIME = 194400.00
TIME = 198000.00
TIME = 201600.00
TIME = 205200.00
TIME = 208800.00
TIME = 212400.00
TIME = 216000.00
```

Figure 4.7-1. *Figure showing the concurrent working of DOS shell and the interface*


Running Simulation - Smart Bridge System Design Procedure Step 1

The following steps are being executed as the simulations progresses. Click the STOP button to stop the simulation at any time.

✓ Checking environment settings.
✓ Generating the simulation (.SIM) file.
✓ Setting environment for SLIMCON.
✓ Executing SLIMCON.
✓ Checking for errors in the definition file.
Setting environment for MODSIM.
Executing MODSIM (to stop close MODSIM).
Ending the simulation (modifying the output file).

No boundary value data in the workspace. Could not create the boundary file.

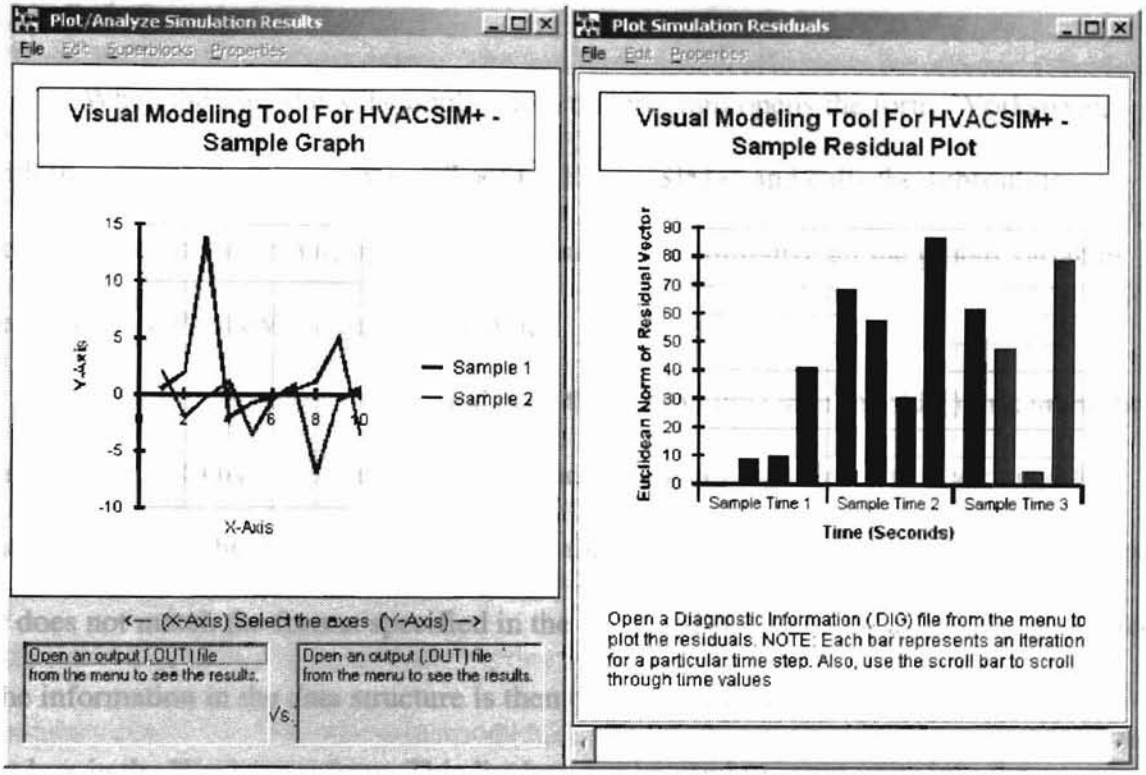
Figure 4.7-2. *Error message while running a deficient simulation*

4.8. Plotting

This section lists the features of the plotting routines and its sub-components.

- The tool allows the user to plot multiple reported variables in one graph. This feature can be used to study the relationships between variables in the system.
- The tool can read output files that were created from a previous simulation. It can also plot data from these files.
- The interface facilitates plotting residuals both through a simulation as well as from a previously created Diagnostic information file (. DIG).
- Routines are provided in the tool to print the plots. These routines also allow the user to scale the plots on the paper before printing them.

The Figures 4.8-1 shows the forms used for plotting simulation results, from the output files and residuals, from the Diagnostic Information files.


Plot/Analyze Simulation Results
File Edit Superblocks Properties
Visual Modeling Tool For HVACSIM+ - Sample Graph
15
10
5
0
-5
-10
Y-Axis
2
6
8
10
X-Axis
Sample 1
Sample 2
<— (X-Axis) Select the axes (Y-Axis) —>
Vs.
Plot Simulation Residuals
File Edit Properties
Visual Modeling Tool For HVACSIM+ - Sample Residual Plot
Euclidean Norm of Residual Vector
90
80
70
60
50
40
30
20
10
0
Sample Time 1
Sample Time 2
Sample Time 3
Time (Seconds)

Open an output (.OUT) file from the menu to see the results.

Open an output (.OUT) file from the menu to see the results.

Open a Diagnostic Information (.DIG) file from the menu to plot the residuals. NOTE: Each bar represents an iteration for a particular time step. Also, use the scroll bar to scroll through time values

Figure 4.8-1. *Forms used for plotting simulation results and residuals*

4.9. Installation and maintenance

An installation script was developed using the Packaging wizard supplied with Microsoft Visual Basic 6.0 PE. The software is packed into one cabinet (. CAB) file. The application SETUP.EXE is used to install the product. The supported operating systems are Windows 95/98, NT 4.0 (SP6), 2000 (SP2) and XP.

During installation, the Types component library (TYPAR.DAT), MODSIM and SLIMCON are installed in the application directory. Examples discussed in Chapter 5 are installed in the Samples folder in the application directory.

To accommodate future models, the user replaces the TYPAR.DAT file and identifies the latest versions of MODSIM and SLIMCON to be used by the application.

4.10. Code Description

When the user starts the application, the program opens the form, Workspace, with the caption 'Visual Modeling Tool for HVACSIM+' and calls the subroutine Form_Load. The main purpose of this subroutine is to initialize all the global variables defined in the PublicVariables.bas module.

After initialization, the method calls the subroutine OpenTYPAR() that takes the path of TYPAR.DAT and creates a data structure containing information about all the models found in the file. This file resides in the application directory. If the file is empty or does not match the format specified in the HVACSIM+ Manuals, an error is generated. The information in the data structure is then used to build the Types component library list box in the Workspace form. This list box can be used to insert units into the workspace. The Boundary conditions list box is then filled with available category descriptions like pressure, temperature, control signal, flow, etc. The program then calls the Form_Resize() subroutine that prepares the controls like scroll bars, picture boxes, menu and toolbars in the window.

The canvas on which the components are inserted is a picture box. Connections between ports are not drawn on this picture box but are drawn on a user-defined Active-X control defined in LineCntl.ctl. Control arrays of picture boxes are created to draw the icons for units and boundary conditions respectively. Units, boundary conditions and connections between ports have individual data structures that logically represent data in their corresponding controls. Panning of the workspace is handled by Timer controls that work with the scroll bars.

After all the initialization has taken place, the application has no messages to process because its queue is empty. The application then idly waits for a menu selection, toolbar selection or any other message of user input, in the form of a message in its queue. In order for the user to start a session, either of the following must be performed: add components to the workspace, load an existing workspace file, create a boundary file or plot the results of a previous simulation.

The toolbar in the Workspace form can be used to add units and boundary conditions to the workspace. The user makes his choice from the combo box and clicks on the accompanying button to eventually call either the CreateUnit() or the CreateBC() subroutine that create icons and add entries to the data structure of either a unit or a boundary condition. Both these methods take a location in the workspace where the icons need to be created as one of the arguments. Additionally, CreateUnit() uses the type number to obtain information such as the number of input and output ports and port descriptions from the structure created from the file TYPAR.DAT. The subroutine CreateBC() takes the boundary condition type as its next argument. The argument either contains numeric zero, which is the default, indicating that it is a time-dependent condition, or a numeric one, indicating a constant condition. This value is then added to the data structure representing the boundary conditions in the workspace. Unlike CreateUnit() this method does not draw an icon but only creates it and makes a data structure entry. After creating the condition, there is always a need to call either DrawTimeDepBC() or DrawConstantBC() subroutine to draw the icon. These methods take the index number of the condition in the control array and either a constant value or a name depending on its type and draw the corresponding icons.

Connections are made by holding the left mouse button on the output port of a unit or a boundary condition, dragging the mouse to an input port on a unit and leaving the mouse button on it. When the user clicks on an input port, the MouseDown event of the corresponding component is triggered. In the handler for this event, some module level variables are set to distinguish between a click on a unit or a boundary condition. A temporary data structure is filled with details related to the output port and will be used when the connection is completed. While the user drags the mouse, the DragOver event of the workspace canvas picture is called repeatedly. In its handler, a red colored line is drawn from the output port selected to the current mouse position. If the user disengages the mouse on anything other than an input port, the connection is terminated. If the mouse is released on an input port of a unit then the DragDrop event of that unit is triggered. In the handler for this event the input port details are filled in the above mentioned structure and the validity of the connection is checked. If the connection is found valid then a line is drawn between input and output ports with a pre-defined color that represent the category of the ports involved. Validity of connections is decided based on the categories of ports and the number of previous connections with the ports involved.

Units can be grouped into blocks and superblocks by pressing the right mouse button on their icon and selecting the 'Grouping' option from the popup-menu. This option opens the GroupingNumbers form, which allows the user to create new blocks and superblocks and assign them to units.

The SimulationRequirements form is used to accept simulation details like title, reporting interval, simulation times and diagnostic information from the user. To enter the error tolerances for the simulation, the user clicks on the 'Error Tolerances' button, which opens the ErrorTolerances form. To specify the diagnostics that need to be performed during the simulation, the user clicks the 'Details' button, which open the DiagnosticInformation form. The DiagnosticInformation form provides a list of available options for the user to choose.

The properties of a boundary condition can be set or changed by double clicking the boundary condition icon in the workspace, which opens the BCSetProperties form.

Various operations can be performed by using mouse buttons over unit icons. Double clicking on these icons opens the UnitProperties form where the user can assign parameter values and initial values for outputs. This form can also be accessed by right clicking and selecting the 'Properties of the component' option in the popup menu. Variable names can be assigned to a port by right clicking the port and selecting the 'Variable Names' option. This option opens the VariableNames form. The popup menu can also be used to make connections between ports on units. Setting of reported variables is handled in the MouseDown event of the unit icon.

Editing operations that can be performed on components in the workspace use several miscellaneous subroutines defined in the Workspace form. Subroutines that begin with the expressions CutSelected, PasteSelected and CopySelected are used for cut, paste and copy operations on selected components respectively. They work only with units and boundary conditions since there is no logical meaning for these operations on connections. Subroutines that begin with the word Delete are used for deletion of components in the workspace. The entire workspace can be used to insert components. There are tools for finding components in the workspace and hiding connections so that only the units and boundary conditions can be visualized and are accessed from the menu in the Workspace form. The form FindComponent presents a list of available units and boundary conditions in the workspace. The user selects a component and clicks on the accompanying button to set focus on it. This form can also be activated by double clicking on the blue colored picture provided at the bottom right corner of the Workspace window. Hiding or showing connections is achieved by setting the visibility of the user-defined Active-X control on which the connections are drawn.

The Save menu option calls the subroutine CreateSimFile() to create an input file (. SIM) for SLIMCON and is accessed by the Save menu option.

The File Locations menu option loads the FilePaths form, which is used to identify different versions of SLIMCON, MODSIM and GFILE for the simulation being built.

The boundary file editor is contained in the BCFileEditor form. It consists of useful tools to create boundary value files. Microsoft's Hierarchical Flex grid is used to host the boundary values. The major advantage of this control is the capability of holding large volumes of data. This form is unloaded only when the application is exited which means that the instance of the window remains in the memory even after the user closes it. This preserves the boundary data in the grid and eliminates the use of additional memory, in the form of an array, to hold the data.

The form BCHeaderNames is used to create new unique boundary condition names that can later be associated with boundary variables. The user can also modify the existing names and change the column position of a name without affecting the performance of the simulation.

The form BCExtractColumn is used to read a column of data from an existing boundary value file and write into the grid. It presents to the user the first ten lines of the file. The user then selects the desired column from a list of boundary condition names.

The form BCInsertFunction provides mathematical functions that can work with the data in the grid. The user chooses the input and output cells and the function to apply.

The form BCInsertSeries provides tools to insert linear series into columns in the editor. The user specifies the initial, final and step values. The algorithm generates the series data and writes the data into the specified column.

The forms SimulationPreRun and SimulationRun are used to execute the simulation. The SimulationPreRun form allows the user to specify a name for the output files and asks the user the location of boundary values. The location of data is either a boundary value file or the grid in the boundary file editor. The SimulationRun form actually executes the simulation by setting the environment and calling the DOS based SLIMCON and MODSIM. This form uses Timer controls to proceed from one step in the simulation to the other.

Two kinds of reports are generated by the application, namely, the plot of simulation results and the plot of simulation residuals. The form GraphView is used to plot the simulation results. The form DiagGraphView is used to plot the residuals. The forms GraphPrint and GraphPlotProperties can be used to print the plots. These two forms can be opened using the Print option in the File menu in these forms.

CHAPTER 5

EXAMPLES

5.1. Simulation of Examples from HVACSIM+ documentation

The following sections discuss the implementation of example models explained in the HVACSIM+ Users Guide provided by National Bureau of Standards, Gaithersburg, Maryland U.S.A (Clark and May, 1985).

5.1.1. Example 1: Inlet pipe and linear valve system

This system represents an inlet conduit and a linear control value system as shown in the Figure 5.1.1-1.

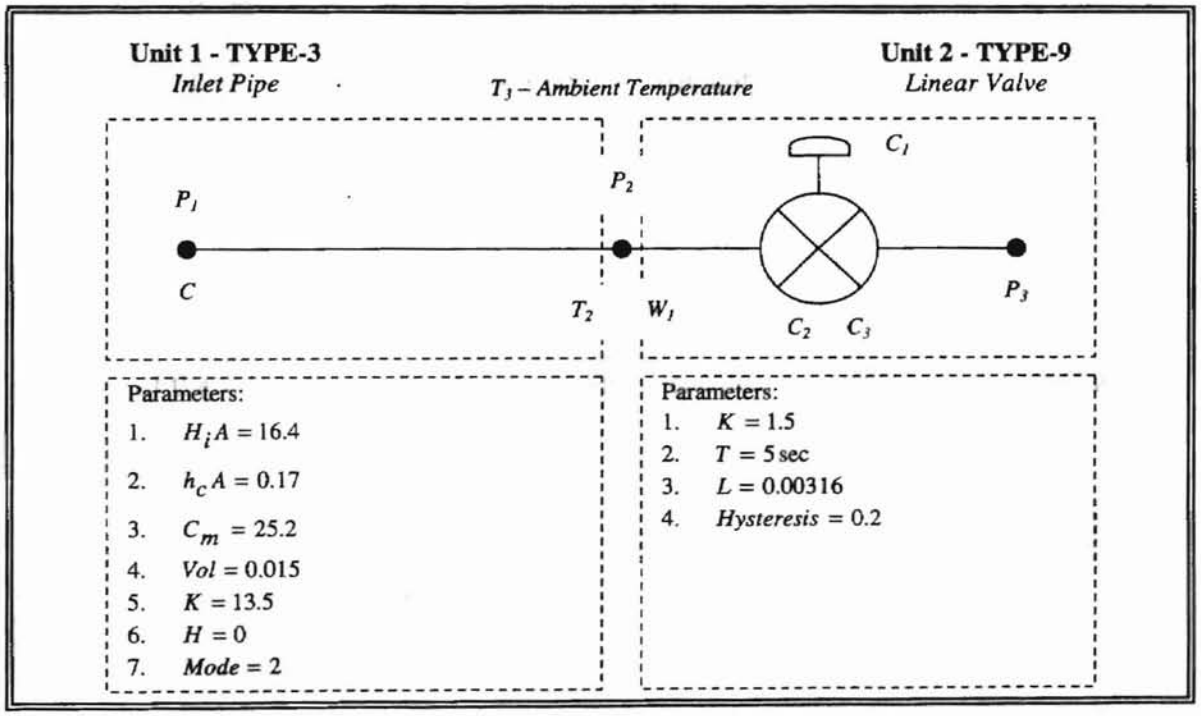


Figure 5.1.1.1-1. *Example 1: System sohematic diagram*

5.1.1.1. System Description

This example consists of two units: the TYPE 3 inlet pipe and the TYPE 9 valve. The component model TYPE 3 belongs to the functional category of Conduits. Component model TYPE 9 belongs to the functional category of Valves, Dampers and Flow Restrictors. The flow rate for the inlet conduit is calculated from inlet and outlet pressures. The inlet pressure for linear valve is calculated from flow rate and outlet pressure. Ten state variables are identified in the figure: three pressures, one flow rate, three temperatures, and three control signals. Parameter values are indicated beneath the sketch of the system. Descriptions of the units involved can be found in Appendix A.

Component models generally relate pressure losses to flow rates using equations of the following form:

$$\Delta P = KW^2 \qquad (5.1.1.1\text{-}1)$$

where ΔP = pressure drop across the device (kPa)

K = flow resistance or pressure loss coefficient ($g^{-1}m^{-1}$)

= K for valve + K for pipe

W = mass flow rate (kg/s)

The installed characteristic of a valve depends on the inherent characteristic and the authority of the valve. Using the flow-resistance parameter values shown in the Figure 5.1-1, the valve authority in this simulation is given by

$$a = \frac{K_{valve}}{K_{valve} + K_{pipe}} = \frac{1.5}{13.5 + 1.5} = 0.10 \text{ or } 10\%. \qquad (5.1.1.1\text{-}2)$$

The desired maximum flow rate, when the valve is open, determines the difference between the inlet and outlet pressures, P_1 and P_3:

$$P_1 - P_3 = \left(K_{valve} + K_{pipe}\right)W^2 = 15W^2 \qquad (5.1.1.1\text{-}3)$$

Thus, if a maximum flow rate of 1.5 kg/s is desired P_1 and P_3 should be assigned values such that P_1 minus P_3 equals 33.75.

The minimum (leakage) flow rate, when the valve is closed, can be expressed as a function of the leakage parameter, L, and the valve authority, a. To a good approximation, the relative leakage is given by

$$\frac{W_{min}}{W_{max}} = L\,a^{-0.5} \qquad (5.1.1.1\text{-}4)$$

The value of L in this simulation was chosen to yield a leakage flow ratio of 0.01, or a turndown ratio of 100.

The transport delay in the pipe, T_x, is given by

$$T_x = \frac{D.Volume}{W_1} \qquad (5.1.1.1\text{-}5)$$

where D is the density of the fluid. At a flow rate of 1.5 kg/s, the transport delay is ten seconds.

Any state variable, which is the dependent variable in a differential equation on an output of one unit and an input to one or more other units, will be solved simultaneously by the main simulation program. This statement involves a subtle and somewhat confusing distinction: in cases where an output of a unit is also an input to the same unit, the status of the variable cannot be determined at a glance. Two examples of variables that fit this description are the control variable C_2 and the temperature T_2. The control C_2 is solved simultaneously because of its involvement in a differential equation. If the actuator time constant, second parameter of TYPE 9 model, were changed to one second or less, however, C_2 would be determined within the TYPE 9 subroutine and would cease to be a simultaneously solved variable. On the other hand, the temperature T_2 is solved within the TYPE 3 subroutine in this simulation and is not considered a simultaneous variable. If the MODE (seventh parameter of the pipe model) were made negative, the equation solver would solve the differential equation involving T_2 and T_2 would become a simultaneous variable. Thus, the number of simultaneous equations and the number of differential equations in a simulation may depend on parameter values. In this simulation, there are three simultaneously solved variables: P_2, W_1 and C_2.

State variable C_3 is an output but not an input, so it is not solved simultaneously. Its sole purpose is to provide the user with information about the state of the system. T_2 is also considered an informative output in this system.

The remaining variables (P_1, T_1, T_3, P_3 and C_1) are inputs but not outputs, and serve as boundary conditions. Boundary variables may be either constant (time independent) or time dependent. In this example, the inlet and outlet pressures P_1 and P_3 and the ambient temperature T_3 will be constant boundary conditions, retaining their initial values throughout the simulation.

5.1.1.2. Creating the system model

The application begins with a new workspace. The user can begin creating his model at this stage. If the user is already working on a simulation, he can save his work and clear the workspace to begin a new simulation. The system is now ready to accept the various entities like units, boundary conditions and connections between units.

The user can save his work at any time during the process of creating the simulation. This is important for a novice trying to create a large simulation involving many units. In addition, the user can use the various editing operations such as cut, copy, paste and delete on the entities in the workspace while establishing the simulation input.

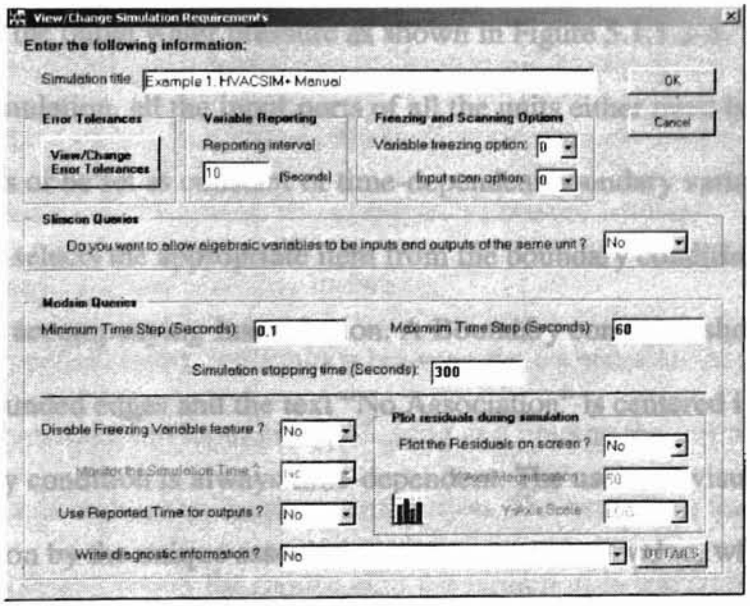
The user can initially set the simulation requirements by selecting the SETTINGS->SIMULATION REQUIREMENTS menu option. A dialog pops up asking the user the simulation title, various error tolerances, reporting interval, freezing option and scanning option for the simulation, questions related to required diagnostics and simulation times, as shown in Figure 5.1.1.2-1. The various error tolerances in HVACSIM+ are:

- The relative and absolute error tolerances, which are used in integrating differential equations and choosing the time step size. They are also used in deciding when to freeze or unfreeze state variables.
- An error tolerance used by the equation solver
- The time interval over which the integral of a differential equation must satisfy the error criterion defined by relative and absolute tolerances.

The error tolerance used by the equation solver is discussed in Section 2.4.3 of Chapter 2. The relative and absolute error tolerances and the time interval are discussed in Section 2.4.4 in the same chapter. The Figure 5.1.1.2-2 displays a dialog showing the error tolerances selected for the simulation.

When the dialog pops up fields like error tolerances, simulation times and diagnostic questions are filled with default values. The HVACSIM+ Users Guide suggests a number that is the square root of the machine precision for the error tolerance in simultaneous equations and a large value, such as 10.0 seconds, for time interval. A large value for the time interval results in greater accuracy and shorter time steps. A value of ten seconds has been chosen for this example since it is a simple simulation. The variable freezing option explained in Section 2.3.1 of Chapter 2, serves no purpose in this simulation since there is only one block of units involved. The variable scan option explained in Section 2.3.1 of Chapter 2, also does not serve any purpose since only one superblock is involved in the system. Figure 5.1.1.2-1 shows this dialog box displaying the values set for the pipe-valve simulation.

The user inserts the TYPE 3 and TYPE 9 components by individually selecting from the Types Component Library list and clicking the accompanying 'Insert' button. The inserted units show up as rectangular boxes with inlet and outlet ports as shown in Figure 5.1.1.2-5. The TYPE number, description of the unit, and variable symbols are shown in these boxes. For every component the input ports point into the rectangle and output ports point away from the rectangle.


View/Change Simulation Requirements
Enter the following information:
OK
Cancel
Error Tolerances
View/Change Error Tolerances
Variable Reporting
Freezing and Scanning Options
Simcon Queries
Modsim Queries
Plot residuals during simulation
DETAILS

Simulation title: Example 1. HVACSIM+ Manual

Reporting interval: 10 (Seconds)

Variable freezing option: 0

Input scan option: 0

Do you want to allow algebraic variables to be inputs and outputs of the same unit ? No

Minimum Time Step (Seconds): 0.1

Maximum Time Step (Seconds): 60

Simulation stopping time (Seconds): 300

Disable Freezing Variable feature ? No

Monitor the Simulation Time ? No

Use Reported Time for outputs ? No

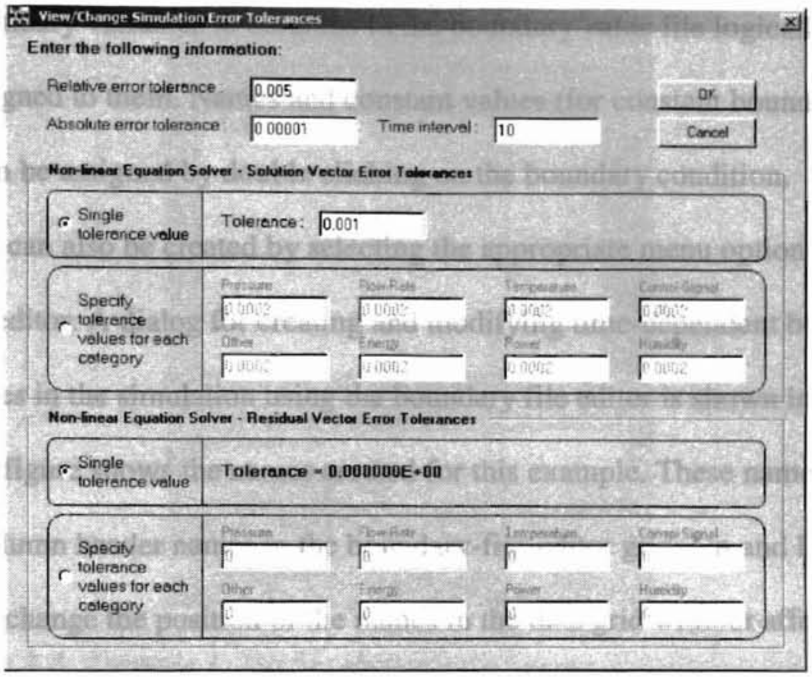
Plot the Residuals on screen ? No

Y-axis Magnification: 50

Y-axis Scale: LOG

Write diagnostic information ? No

Figure 5.1.1.2-1. *Example 1: Dialog showing the simulation requirements*



View/Change Simulation Error Tolerances

Enter the following information:

Relative error tolerance: 0.005

Absolute error tolerance: 0.00001

Time interval: 10

Non-linear Equation Solver - Solution Vector Error Tolerances

● Single tolerance value — Tolerance: 0.001

○ Specify tolerance values for each category

Pressure	Flow Rate	Temperature	Control Signal
0.0002	0.0002	0.0002	0.0002
Other	**Energy**	**Power**	**Humidity**
0.0002	0.0002	0.0002	0.0002

Non-linear Equation Solver - Residual Vector Error Tolerances

● Single tolerance value — Tolerance = 0.000000E+00

○ Specify tolerance values for each category

Pressure	Flow Rate	Temperature	Control Signal
0	0	0	0
Other	**Energy**	**Power**	**Humidity**
0	0	0	0

Figure 5.1.1.2-2. *Example 1: Dialog showing the simulation requirements*

The user can now insert boundary variables into the simulation. Both the conduit and the valve have one time-dependent boundary condition in the simulation categories of temperature and control-signal respectively. In addition, three constant boundary conditions are required to set the initial values for inlet fluid pressure, the ambient temperature, and the outlet water pressure as shown in Figure 5.1.1.2-5.

In any simulation, all the input ports of all the units either must be connected to other output ports or be set as constant or time-dependent boundary variables.

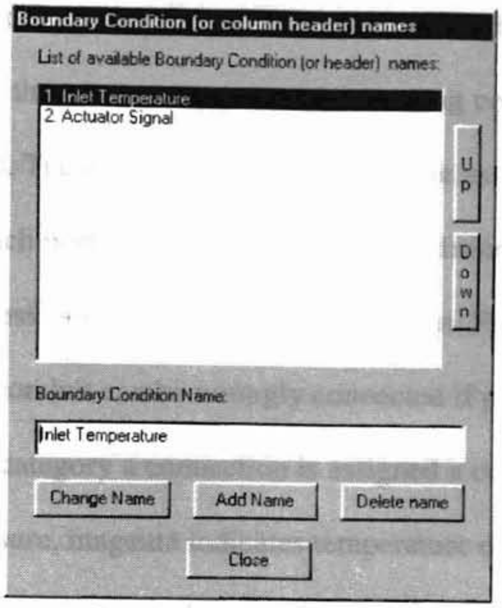
The user selects the appropriate item from the boundary conditions library list and clicks on the accompanying Insert button. A Boundary condition shows up as a rectangle with rounded edges and the text "No Association" is centered in it. A newly inserted boundary condition is always time-dependent. The user can visually identify a boundary condition by the unique association name or constant value, which he/she assigns, and the letter corresponding to its type in the bubble. In order to associate time-dependent boundary variables to columns in the boundary value file logically, unique names are assigned to them. Names and constant values (for constant boundary conditions) can be assigned by double clicking on the boundary condition.

Names can also be created by selecting the appropriate menu option in the boundary file editor. A dialog for creating and modifying time-dependent boundary condition names in the simulation using the boundary file editor is shown in Figure 5.1.1.2-3. The figure shows the names created for this example. These names also become the column header names in the boundary-file-editor grid. Up and Down buttons can be used to change the position of the names in the data grid without affecting the simulation setup, as long as the associations are correct.

Names created using the boundary file editor can be assigned to corresponding boundary conditions by double clicking on a boundary condition and selecting the previously created name from a list. The Figure 5.1.1.2-4 shows the property dialogs of time-dependent and constant boundary conditions. The picture also shows the appearance of the boundary conditions after they have been assigned a name or a constant value.

It should be noted that a boundary condition could be connected to an input port only if the category matches. In addition, the application does not check if the assigned name is appropriate for that variable. For example, a pressure boundary condition can be named 'Temperature' without affecting its nature.

The user can then insert connections between the components of the simulation. The user identifies output and input ports. Clicking and holding the left mouse on the output port when it is highlighted begins a connection. The user drags the mouse on to the input port and finally leaves the mouse after the input port is highlighted.

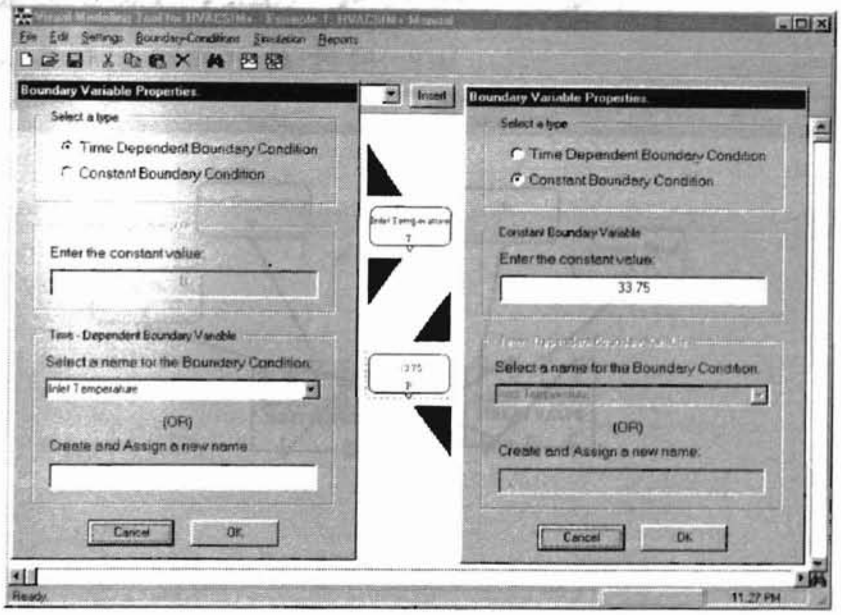


List of available Boundary Condition (or header) names:

1. Inlet Temperature
2. Actuator Signal

Boundary Condition Name: Inlet Temperature

Figure 5.1.1.2-3. *Example 1: Dialog showing the names for the boundary conditions*


File Edit Settings Boundary-Conditions Simulation Reports
Insert
Boundary Variable Properties
Inlet Temperature
33.75
Cancel
OK
Ready
11:27 PM

Boundary Variable Properties

Select a type
● Time Dependent Boundary Condition
○ Constant Boundary Condition

Enter the constant value:

Time - Dependent Boundary Variable
Select a name for the Boundary Condition: Inlet Temperature
(OR)
Create and Assign a new name:

Boundary Variable Properties

Select a type
○ Time Dependent Boundary Condition
● Constant Boundary Condition

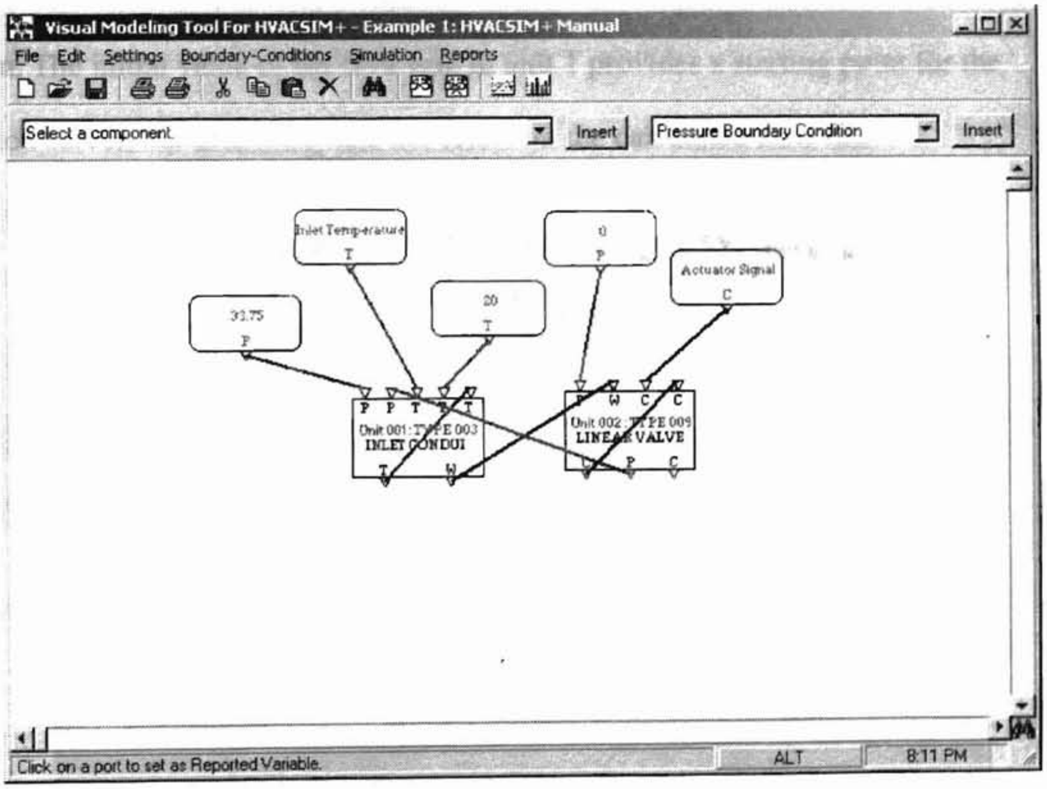
Constant Boundary Variable
Enter the constant value: 33.75

Select a name for the Boundary Condition:
(OR)
Create and Assign a new name:

Figure 5.1.1.2-4. *Example 1: Setting the properties of boundary conditions*

In addition to the five boundary conditions, this simulation involves four additional connections as shown in the Figure 5.1.1.2-5. The tool tips on the inlet and outlet ports can be helpful in making these connections. These tips display the complete description of ports. The tool tip is displayed when the user highlights a port by moving the mouse pointer on it. If the user attempts to make a wrong connection, an error message will be displayed. The user should be careful while assigning connections since the units do not know which ports should be boundary conditions or connections to other ports. There is always a possibility of mix up of connections. For example, the three input temperatures on the inlet conduit can be wrongly connected if proper care is not taken.

Depending on the category a connection is assigned a color. Dark blue indicates a flow, green indicates pressure, magenta indicates temperature or power and black indicates control-signal, rotation rate or humidity.

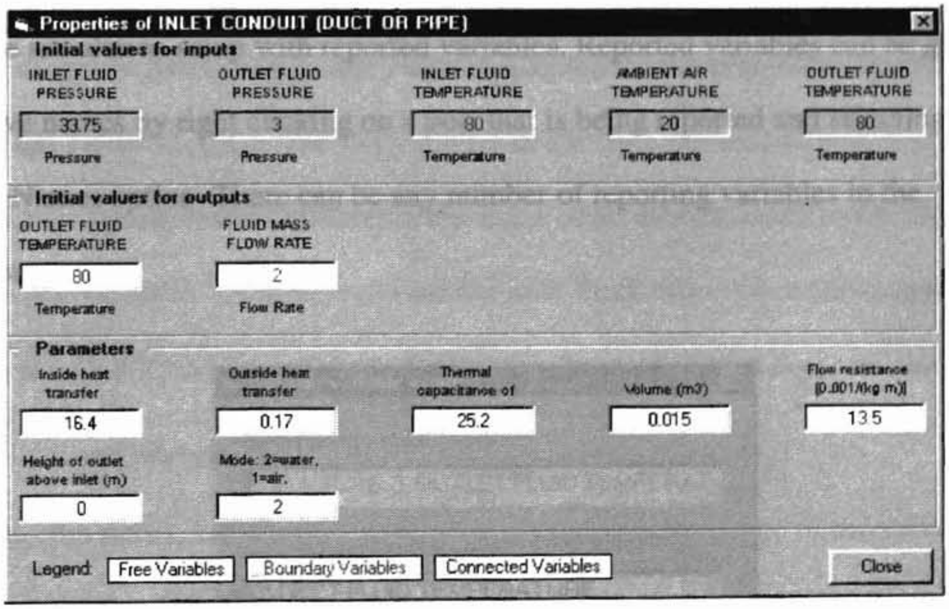


Click on a port to set as Reported Variable.

Figure 5.1.1.2-5. *Example 1: Arrangement of entities in the simulation*

Care should be taken while making connections with boundary conditions. A boundary condition of a certain category can only be connected to input ports of the same category. More than one boundary condition (of similar category) can have the same name and a boundary condition can be connected to many input ports of the same category. Such variables share the same column in the boundary value file, discussed in Section 2.2.4.

The user can now enter the initial values for inputs, outputs and parameters of the units involved in the simulation by double clicking on a component. The initial values selected for each unit are shown in the Figures 5.1.1.2-6 and 5.1.1.2-7. It is important to recognize that a boundary variable will retain its initial value throughout the simulation. For example, in this simulation flow is driven by the difference between the inlet and outlet pressures. This pressure difference determines the maximum flow rate through the system. The initial value of the flow rate in unit 1 provides a starting point for the equation solver, but ultimately has not effect on the calculated flow rate.



Properties of INLET CONDUIT (DUCT OR PIPE)

Initial values for inputs

INLET FLUID PRESSURE	OUTLET FLUID PRESSURE	INLET FLUID TEMPERATURE	AMBIENT AIR TEMPERATURE	OUTLET FLUID TEMPERATURE
33.75	3	80	20	80
Pressure	Pressure	Temperature	Temperature	Temperature

Initial values for outputs

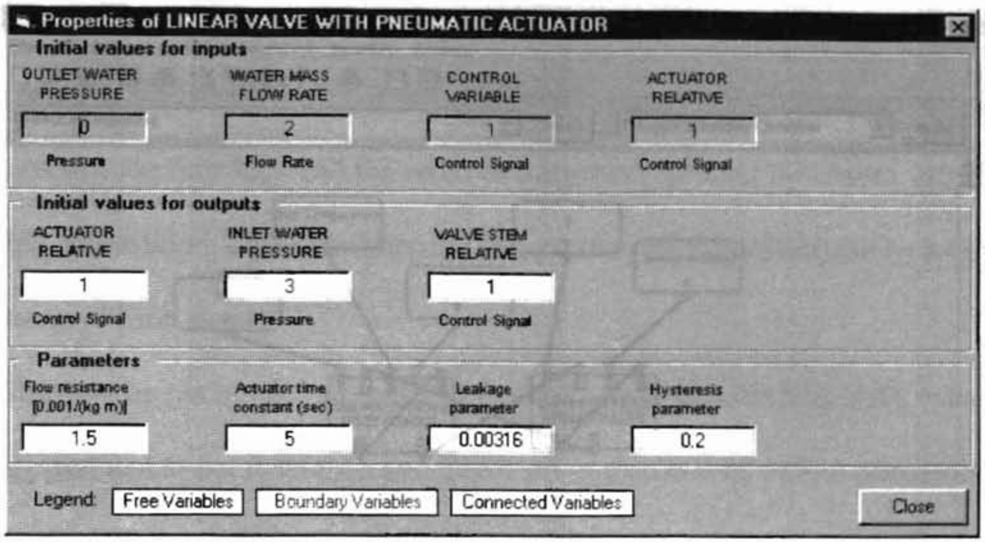
OUTLET FLUID TEMPERATURE	FLUID MASS FLOW RATE
80	2
Temperature	Flow Rate

Parameters

Inside heat transfer	Outside heat transfer	Thermal capacitance of	Volume (m3)	Flow resistance [0.001/(kg m)]
16.4	0.17	25.2	0.015	13.5

Height of outlet above inlet (m)	Mode: 2=water, 1=air,
0	2

Figure 5.1.1.2-6. *Example 1: Initial values for unit 1*



Properties of LINEAR VALVE WITH PNEUMATIC ACTUATOR

Initial values for inputs

OUTLET WATER PRESSURE	WATER MASS FLOW RATE	CONTROL VARIABLE	ACTUATOR RELATIVE
0	2	1	1
Pressure	Flow Rate	Control Signal	Control Signal

Initial values for outputs

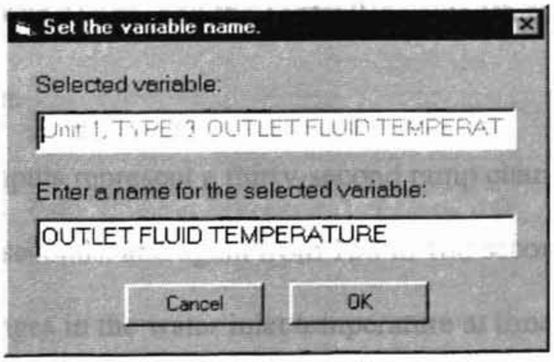
ACTUATOR RELATIVE	INLET WATER PRESSURE	VALVE STEM RELATIVE
1	3	1
Control Signal	Pressure	Control Signal

Parameters

Flow resistance [0.001/(kg m)]	Actuator time constant (sec)	Leakage parameter	Hysteresis parameter
1.5	5	0.00316	0.2

Figure 5.1.1.2-7. *Example 1: Initial values for unit 2*

The user can now select the reported variables for the simulation. Reported variables can be set or unset by clicking on the desired port while holding the ALT key. The Figure 5.1.1.2-8 shows a dialog where a variable is being named. Reported variables are highlighted in dark gray background and white foreground. The Figure 5.1.1.2-9 shows the simulation setup with reported variables. Reported variables can be given descriptive names by right clicking on a port that is being reported and selecting the Variable Names option. There can be any number of reporting variables in the simulation.



Set the variable name.

Selected variable:

Unit 1, TYPE 3 OUTLET FLUID TEMPERAT

Enter a name for the selected variable:

OUTLET FLUID TEMPERATURE

Figure 5.1.1.2-8. *Example 1: Setting variable names*

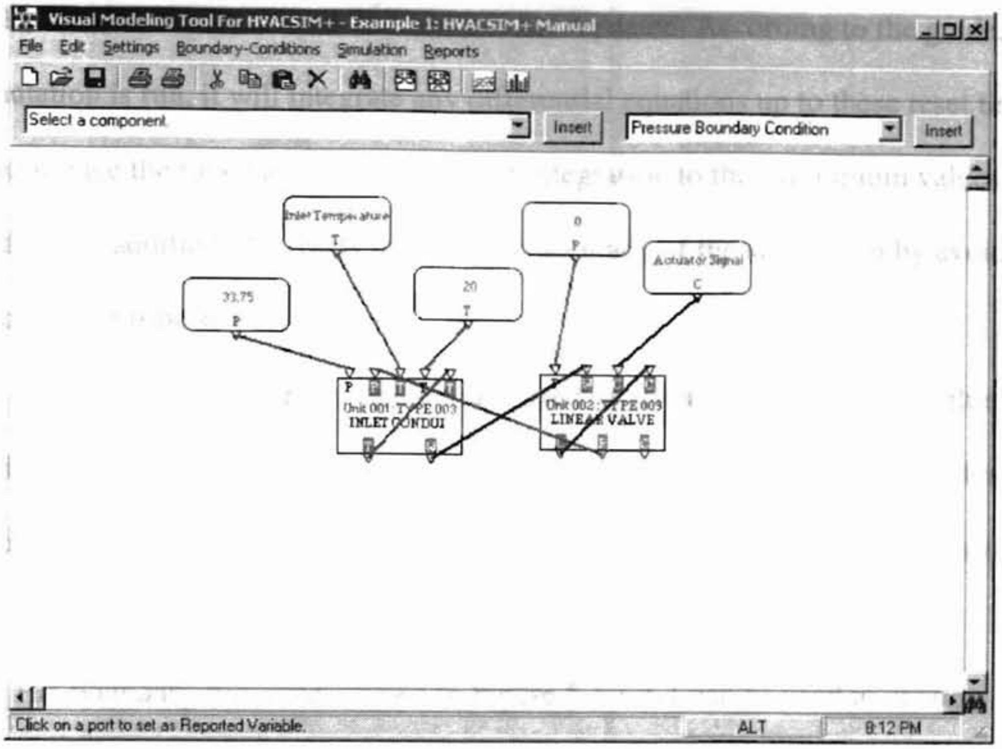


Figure 5.1.1.2-9. *Example 1: Screenshot showing the reported variables selected for the simulation*

5.1.1.3. Preparing the Boundary Value File

A complete listing of boundary values selected for the boundary variables is shown in the Table 5.1.1.3-1. The first column in the table represents time. The other columns in the file represent variables in the same order as mentioned in the .SIM file. The boundary variables in the example are the inlet fluid temperature (third input on unit 1), which is called T_1 in our design, and the controller value (third input on unit 2), which is called C_1 in our design.

The simulation inputs represent a thirty-second ramp change in the valve position from time 30 to time 60 seconds, and again from 150 to 180 seconds. These are followed by five degrees step changes in the water inlet temperature at times 210 and 255 seconds. A part of data in the table uses the "reset convention" of having repeated time values. For example, the time values 30 and 255 seconds are repeated. According to the guide, when the simulation is run, it will integrate any differential equations up to these reset times, and then reduce the time step and the order of integration to their minimum values before proceeding. In addition, the resets improve the accuracy of the simulation by avoiding overshooting of time steps.

Third-order LaGrange interpolation is used to generate the boundary values. The data is interpolated to the reset time and a new set of data is read before proceeding. This avoids interpolation across a step change or the around the corner of a discontinuous change in slope.

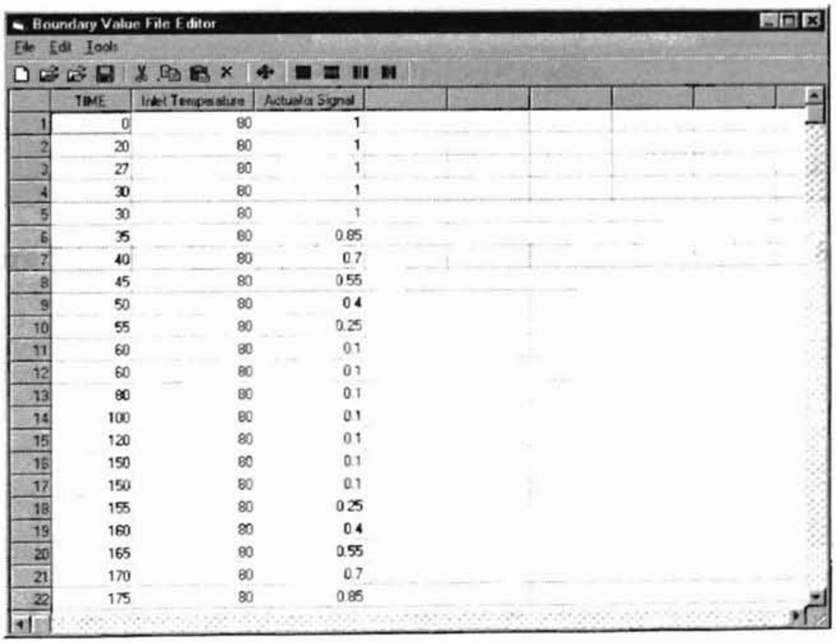
The boundary file editor shown in Figure 5.1.1.3-1 can be used to create a boundary file. The user enters data in the grid as shown in Table 5.1.1.3-1 by copying to or from an open Microsoft Excel worksheet or by directly typing in the cells. The first column in the grid is reserved for time data. The headers for the columns can be changed by double clicking on the dark faced cells in the top row or by selecting the appropriate menu option. The columns in the grid are associated with boundary conditions on the various units through the header names. Editing operations like cut, copy, paste and delete can be performed on the cells. Data can also be copied to or from an open Microsoft Excel worksheet.

Time	Temperature	Control Signal
0.00	80.00	1.00
20.00	80.00	1.00
27.00	80.00	1.00
30.00	80.00	1.00
30.00	80.00	1.00
35.00	80.00	0.85
40.00	80.00	0.70
45.00	80.00	0.55

Table 5.1.1.3-1. *Example 1: Boundary values for the simulation*

Time	Temperature	Control Signal
50.00	80.00	0.40
55.00	80.00	0.25
60.00	80.00	0.10
60.00	80.00	0.10
80.00	80.00	0.10
100.00	80.00	0.10
120.00	80.00	0.10
150.00	80.00	0.10
150.00	80.00	0.10
155.00	80.00	0.25
160.00	80.00	0.40
165.00	80.00	0.55
170.00	80.00	0.70
175.00	80.00	0.85
180.00	80.00	1.00
180.00	80.00	1.00
190.00	80.00	1.00
200.00	80.00	1.00
210.00	80.00	1.00
210.00	75.00	1.00
220.00	75.00	1.00
230.00	75.00	1.00
255.00	75.00	1.00
255.00	80.00	1.00
270.00	80.00	1.00
300.00	. 80.00	1.00
450.00	80.00	1.00
600.00	80.00	1.00

Table 5.1.1.3-1(Contd.). *Example 1: Boundary values for the simulation*



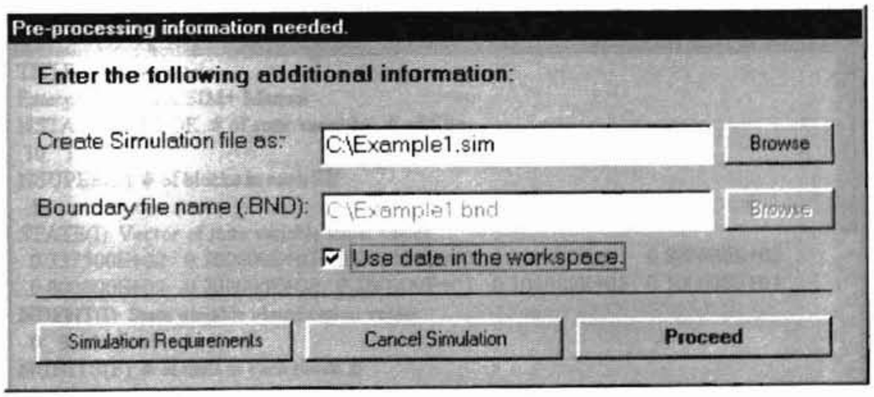
	TIME	Inlet Temperature	Actuator Signal
1	0	80	1
2	20	80	1
3	27	80	1
4	30	80	1
5	30	80	1
6	35	80	0.85
7	40	80	0.7
8	45	80	0.55
9	50	80	0.4
10	55	80	0.25
11	60	80	0.1
12	60	80	0.1
13	80	80	0.1
14	100	80	0.1
15	120	80	0.1
16	150	80	0.1
17	150	80	0.1
18	155	80	0.25
19	160	80	0.4
20	165	80	0.55
21	170	80	0.7
22	175	80	0.85

Figure 5.1.1.3-1. *Example 1: Boundary file editor*

5.1.1.4. Running the simulation

After entering all the required data, the simulation can be executed from the interface. The Figure 5.1.1.4-1 shows the initial dialog that pops up when the user selects SIMULATION->RUN from the menu to run the simulation. The simulation file name that will be created is entered. The boundary file that will be used is also identified. Checking the check box "Use data in workspace" as shown in Figure 5.1.1.4-1 makes the system use the data in the grid for creating the boundary file.

Figure 5.1.1.4-2 shows the final screen after the simulation has successfully been executed. If there is an error, the simulation will stop and the process step where the error occurred will be highlighted in red. Figure 5.1.1.4-3 shows a partial listing of the definition file generated after SLIMCON was executed.



Pre-processing information needed.

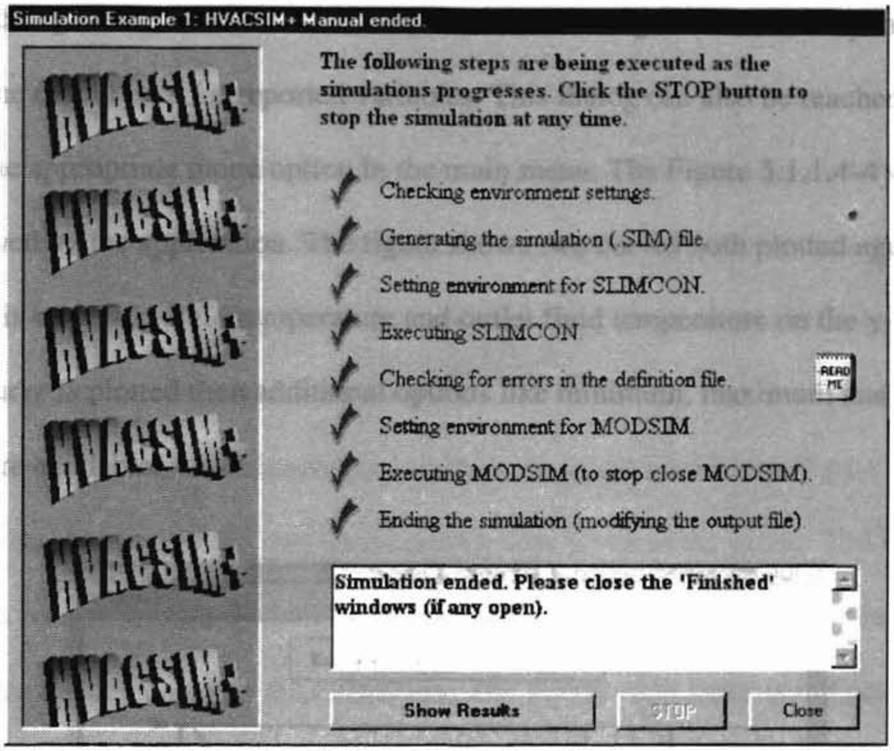
Enter the following additional information:

Create Simulation file as: C:\Example1.sim

Boundary file name (.BND): C:\Example1.bnd

☑ Use date in the workspace.

Figure 5.1.1.4-1. *Example 1: Initial screen before the simulation begins*

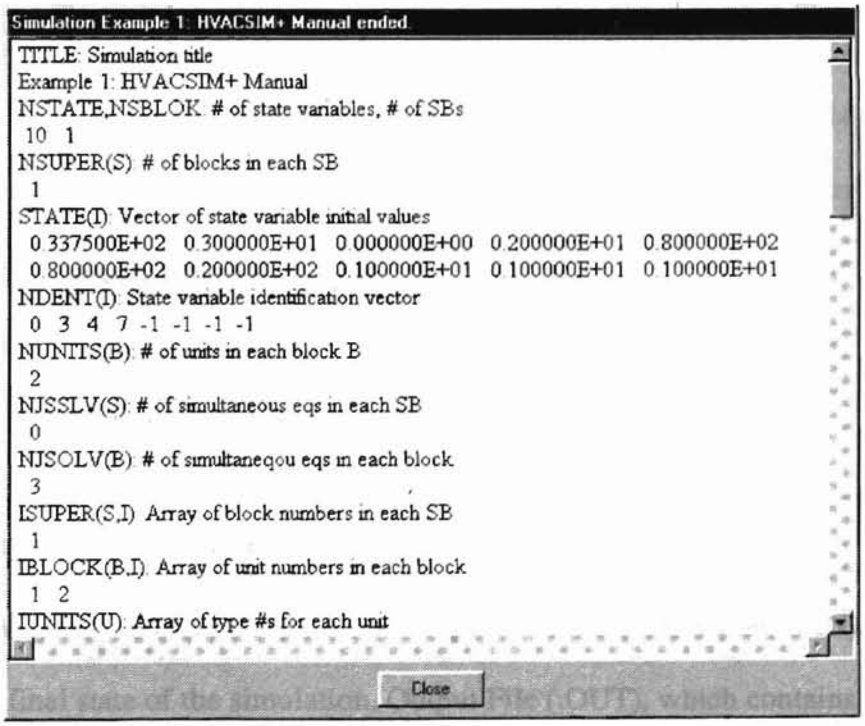


The following steps are being executed as the simulations progresses. Click the STOP button to stop the simulation at any time.

- ✓ Checking environment settings.
- ✓ Generating the simulation (.SIM) file.
- ✓ Setting environment for SLIMCON.
- ✓ Executing SLIMCON
- ✓ Checking for errors in the definition file.
- ✓ Setting environment for MODSIM.
- ✓ Executing MODSIM (to stop close MODSIM).
- ✓ Ending the simulation (modifying the output file).

Simulation ended. Please close the 'Finished' windows (if any open).

Figure 5.1.1.4-2. *Example 1: Screen showing the end of simulation*



```
TITLE: Simulation title
Example 1: HVACSIM+ Manual
NSTATE,NSBLOK: # of state variables, # of SBs
 10  1
NSUPER(S): # of blocks in each SB
  1
STATE(I): Vector of state variable initial values
  0.337500E+02  0.300000E+01  0.000000E+00  0.200000E+01  0.800000E+02
  0.800000E+02  0.200000E+02  0.100000E+01  0.100000E+01  0.100000E+01
NDENT(I): State variable identification vector
  0  3  4  7 -1  -1  -1  -1
NUNITS(B): # of units in each block B
  2
NJSSLV(S): # of simultaneous eqs in each SB
  0
NJSOLV(B): # of simultanegou eqs in each block
  3
ISUPER(S,I): Array of block numbers in each SB
  1
IBLOCK(B,I): Array of unit numbers in each block
  1  2
IUNITS(U): Array of type #s for each unit
```

Figure 5.1.1.4-3. *Example 1: Partial listing of the Definition file generated during simulation*

Clicking on the 'Show Results' button, shown in Figure 5.1.1.4-2, opens a dialog that plots the output data for reported variables. This dialog can also be reached by selecting the appropriate menu option in the main menu. The Figure 5.1.1.4-4 shows a plot displayed by the application. The figure shows two curves both plotted against time on the x-axis with inlet fluid temperature and outlet fluid temperature on the y-axis. If only one curve is plotted then additional options like minimum, maximum and standard deviation are available.

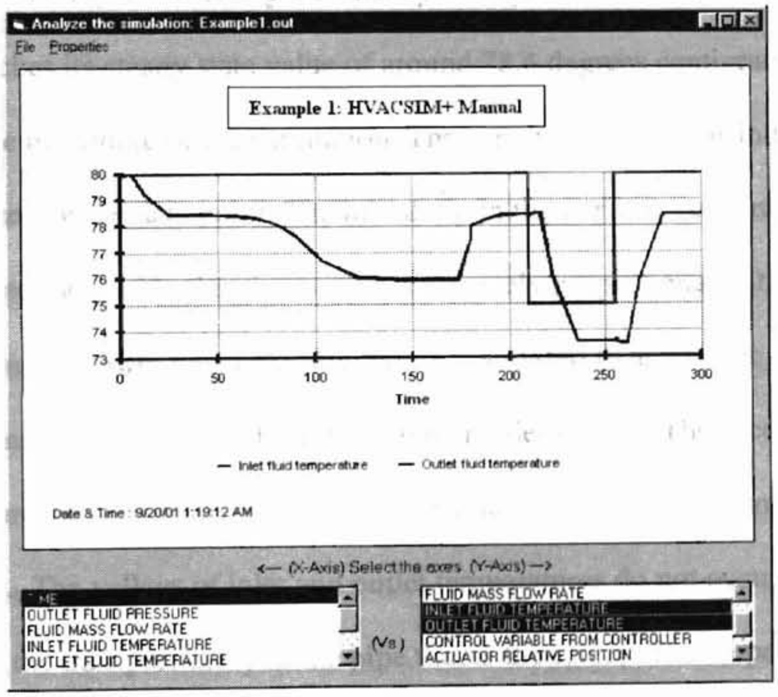


Figure 5.1.1.4-4. *Example 1: Plot of reported variables*

5.1.1.5. Analyzing the simulation results

HVACSIM produces three files on output. The Final State File (.FIN), which contains the final state of the simulation, Output File (.OUT), which contains the raw data at each time step in the simulation and Summary File (.SUM), which contains a summary of simulation including configuration, diagnostic information and interpolated listing of reported variables at equal time intervals.

The results created by the interface were compared to results generated by an older version of HVACSIM+. The files generated using the new system matched the files generated by the older version exactly. Since the test was designed to check input manipulation only, an exact match was expected.

The inlet and outlet temperatures are plotted against time in the Figure 5.1.1.5-1. The curves titled old represent data generated by files given in the guide. The outlet temperature reaches its steady state value of around 78.4 degrees centigrade in 25 seconds from the beginning of the simulation. The steady state value is indicated in the figure as a marker for the data point. The initial dip in the curve is accounted for the fact that the time constant used in the pipe model is inversely proportional to the flow rate. The response time is slower when the flow rate is decreased than when it is increased. In addition, the transport time of the fluid in the pipe model is long. The second dip in the outlet temperature curve is accounted for the step changes in the inlet temperature (shown as a dotted line). The valleys of inlet and outlet temperatures do not occur at the same time because of the transport delay in the pipe that is the effect of the pipe's thermal capacitance.

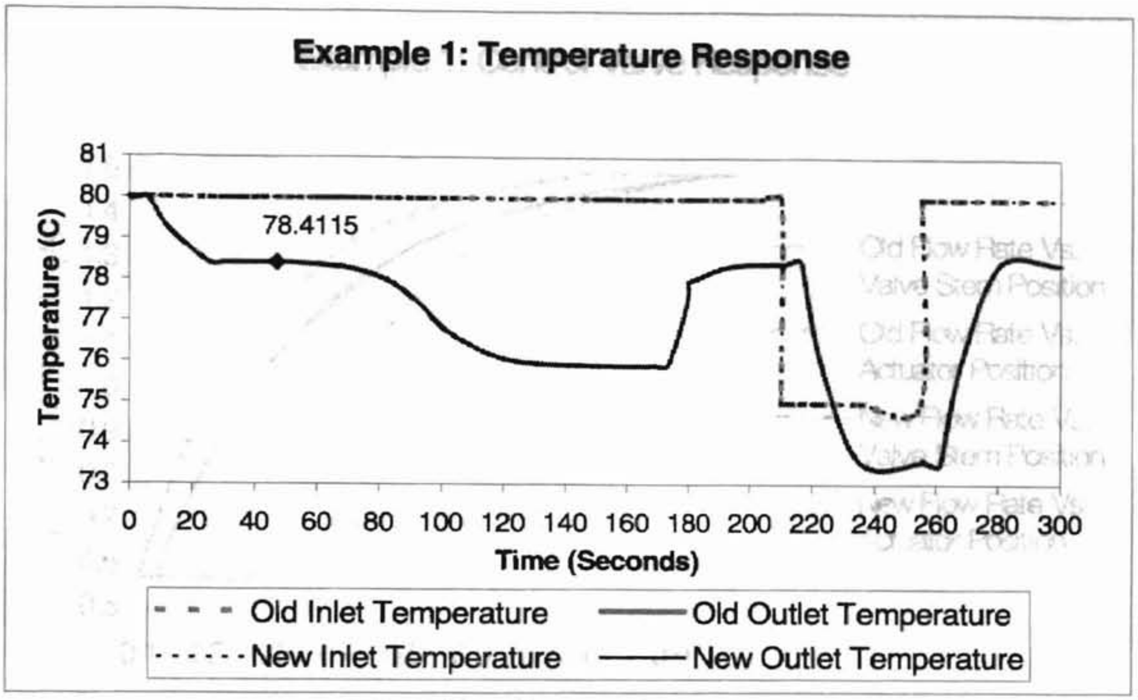


Figure 5.1.1.5-1. *Example 1: Temperature response*

The control valve response is plotted in the Figure 5.1.1.5-2. The control signals valve stem and actuator positions are plotted against flow rate. The solid lines in the figure show the mass flow rate through the valve as a function of the actual valve position as the valve opens and closes respectively. The characteristic of the TYPE 9 valve used in the simulation depends on its authority over the system in which it is installed. The difference between the two solid lines is due to the numerical error in the computation. The dashed lines in the figure show the mass flow rate through the valve as a function of the actuator position. The actuator position differs from valve stem position due to the valve hysteresis parameter of 0.2 specified in the simulation and is unavoidable due to mechanical slack in their linkage.

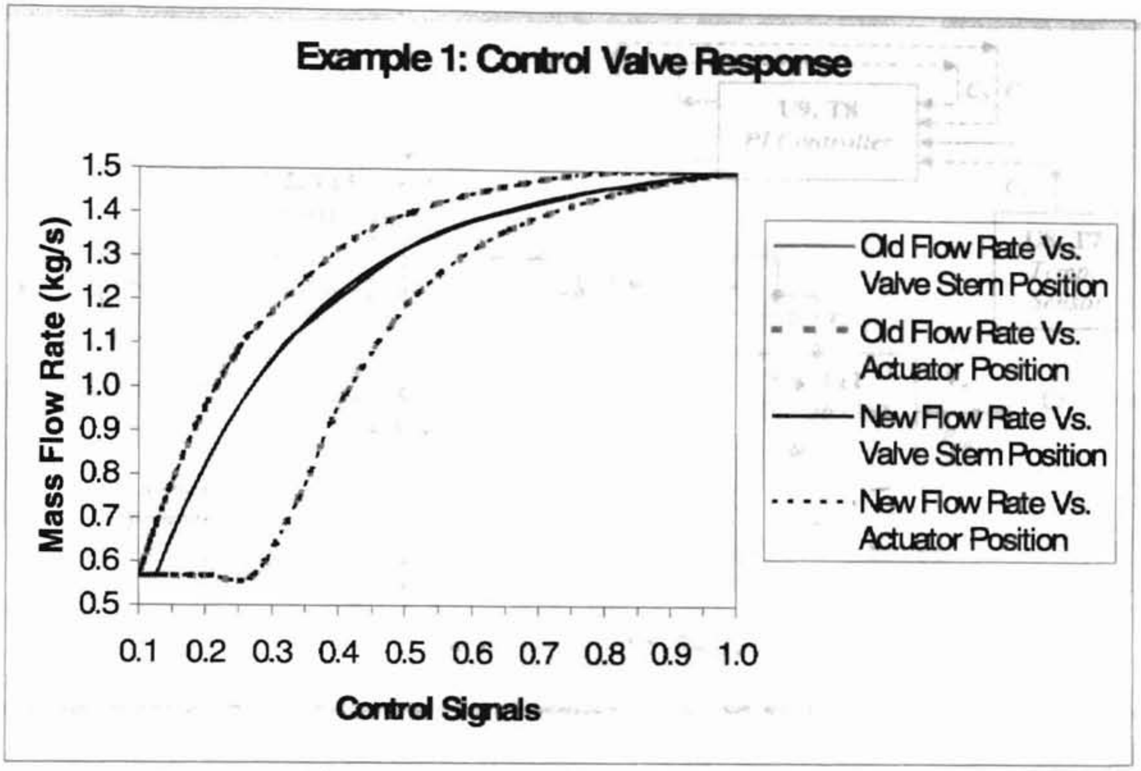


Figure 5.1.1.5-2. *Example 1: Control valve response*

5.1.2. Example 2: Heating Coil Control Loop, Variable Inlet Water Temperature Control

This example represents a system in which the hot water flow rate through a heat exchanger is kept nearly constant, and control is achieved by varying the temperature of the hot water entering the coil. The system schematic is shown in the Figure 5.1.2-1.

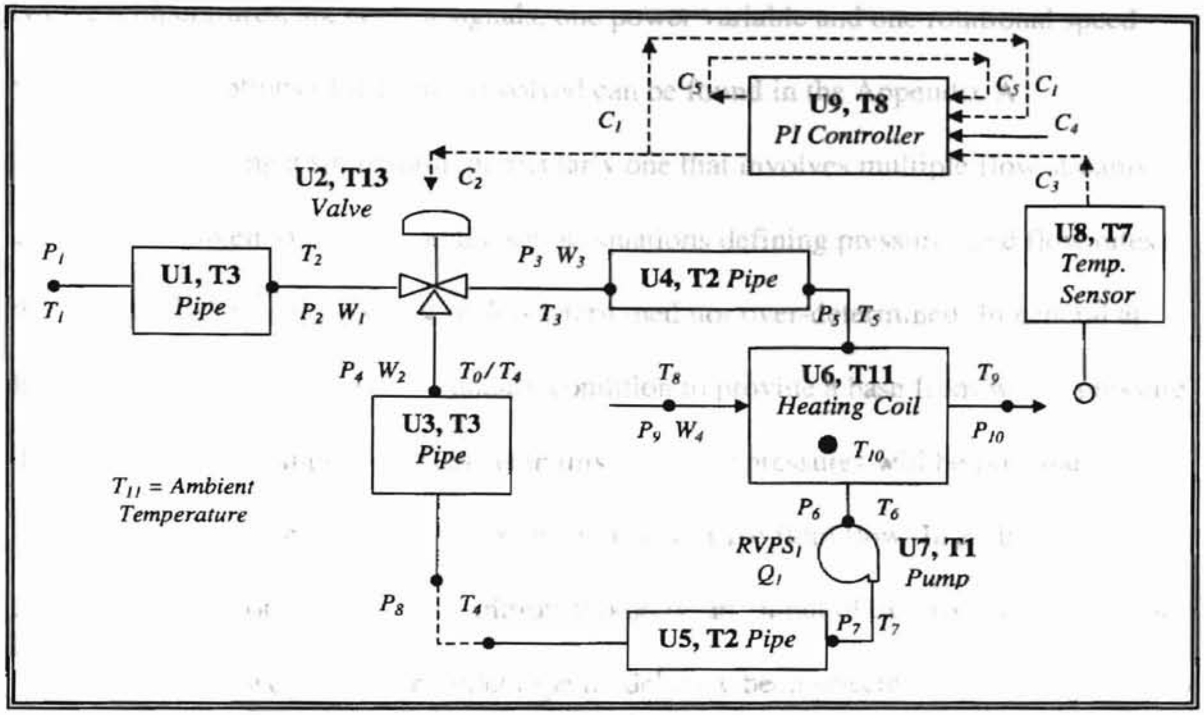


Figure 5.1.2-1. *Example 2: System setup*

5.1.2.1. Model Description

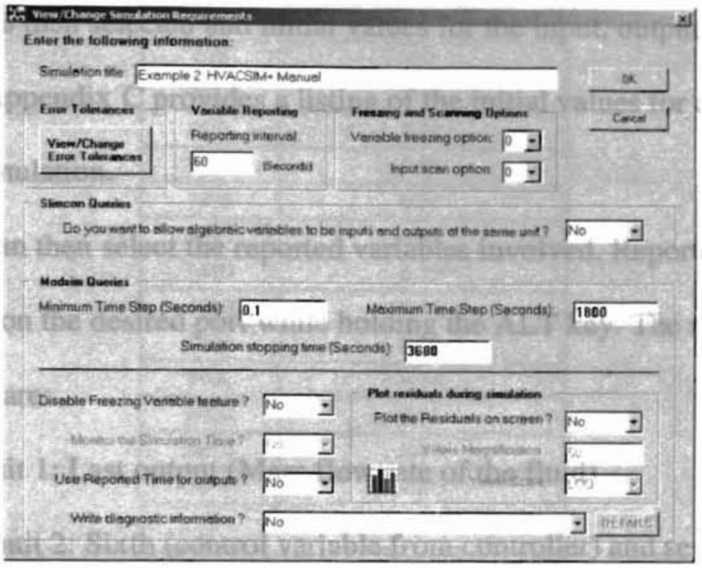
This example consists of nine units- TYPE 3 Inlet Conduit, TYPE 13 Three-way Valve Model, TYPE 2 Conduit, TYPE 11 Hot water to air heating coil, TYPE 1 Fan or Pump, TYPE 7 Temperature Sensor and TYPE 8 Proportional-Integral Controller. There are two each of TYPE 3 and TYPE 2 components. The categorization of TYPE 3 component is given in the earlier example. Component model TYPE 13 belongs to the functional category of Valves, Dampers and Flow Restrictors. TYPE 2 model belongs to the functional category of Conduits. TYPE 11 model belongs to the functional category of Heat Exchangers. TYPE 1 belongs to the functional category of Fans and Pumps. TYPE 7 and TYPE 8 models belong to the functional category of Sensors and Controllers. The inlet pressure is calculated from the flow rate and outlet pressure. Thirty five state variables are identified in the figure: eleven pressures, four flow variables, twelve temperatures, six control signals, one power variable and one rotational speed variable. Descriptions of the units involved can be found in the Appendix A.

In designing a simulation, particularly one that involves multiple flow streams, care must be taken to ensure that the set of equations defining pressures and flow rates throughout the system is neither under-determined nor over-determined. In general at least one pressure should be a boundary condition to provide a base from which pressure difference can be computed. Often, as in this case, two pressures will be boundary conditions, providing a fixed pressure difference to drive fluid flow. In addition, each flow rate that is not a boundary condition should be an output of exactly one component. In this example, two units of the inlet pipe model have been selected to calculate the flow rates W_1 and W_2. Flow rate W_3 is calculated in the valve model, unit two, as the sum of flows W_1 and W_2. On the airside of the unit 6, heating coil, pressure P_{10} and flow W_4 are boundary conditions that determine pressure P_9.

In unit 3, TYPE 3 pipe model is used only to calculate flow rate W_2 in the simulation. Thermal losses, thermal capacitance effects, and the transport delay have been turned off by setting the first four parameters of the unit to zero.

5.1.2.2. Creating the simulation

The user begins by typing the simulation title, the various error tolerances, reporting interval, freezing and scanning options, required diagnostics and simulation times by selecting the appropriate menu option. The Figure 5.1.2.2-1 shows the dialog displaying the simulation requirements. The Figure 5.1.2.2-2 shows the error tolerances selected for the simulation.



View/Change Simulation Requirements

Enter the following information:

Simulation title: Example 2: HVACSIM+ Manual

Error Tolerances

Variable Reporting

Reporting interval: 60 (Seconds)

Freezing and Scanning Options

Variable freezing option: 0

Input scan option: 0

Simcon Queries

Do you want to allow algebraic variables to be inputs and outputs of the same unit ? No

Modsim Queries

Minimum Time Step (Seconds): 0.1

Maximum Time Step (Seconds): 1800

Simulation stopping time (Seconds): 3600

Disable Freezing Variable feature ? No

Monitor the Simulation Time ? No

Use Reported Time for outputs ? No

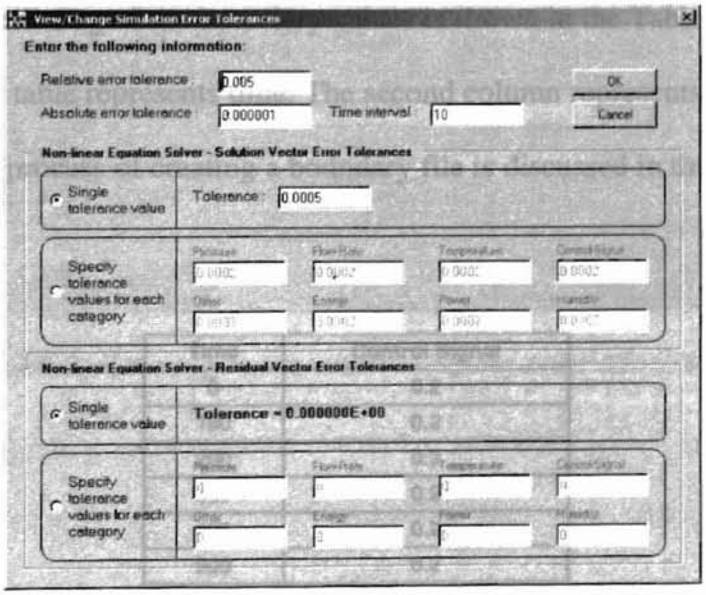
Write diagnostic information ? No

Plot residuals during simulation

Plot the Residuals on screen ? No

Figure 5.1.2.2-1. *Example 2: Dialog showing the simulation requirements*

The user then inserts the components involved in the system and makes connections between them. The workspace diagram for this example can be found in the Appendix C. It shows the various units involved and the connections between them along with the reported variables in the simulation. The reported variables can be named suitably. The only boundary condition in the simulation is named 'Control Variable'.



View/Change Simulation Error Tolerances

Enter the following information:

Relative error tolerance: 0.005

Absolute error tolerance: 0.000001

Time interval: 10

Non-linear Equation Solver - Solution Vector Error Tolerances

● Single tolerance value — Tolerance: 0.0005

○ Specify tolerance values for each category — Pressure: 0.0002, Flow Rate: 0.0002, Temperature: 0.0002, Control Signal: 0.0002, Other: 0.0003, Energy: 0.0002, Power: 0.0002, Humidity: 0.0002

Non-linear Equation Solver - Residual Vector Error Tolerances

● Single tolerance value — Tolerance = 0.000000E+00

○ Specify tolerance values for each category — Pressure: 0, Flow Rate: 0, Temperature: 0, Control Signal: 0, Other: 0, Energy: 0, Power: 0, Humidity: 0

Figure 5.1.2.2-2. *Example 2: Dialog showing the simulation requirements*

Each unit is then selected and initial values for the input, output and parameters are entered. The Appendix C provides a listing of the initial values for all the units involved in the simulation.

The user can then select the reported variables involved. Reported variables can be set by clicking on the desired port while holding the ALT key. The reported variables for the simulation are:

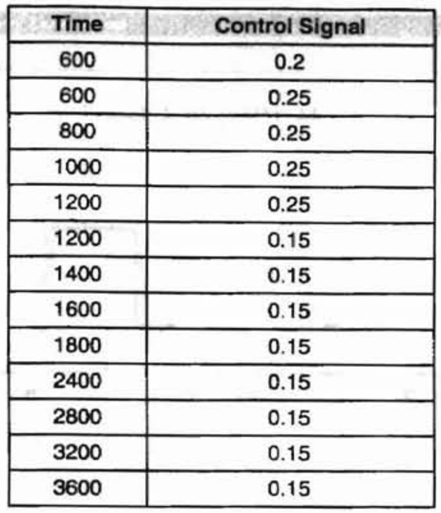
1. TYPE 3 unit 1: Last output (Mass flow rate of the fluid).
2. TYPE 13 unit 2: Sixth (control variable from controller) and seventh (actuator relative position) inputs and second output (outlet mass flow rate).
3. TYPE 11 unit 6: Fourth input (outlet air temperature).
4. TYPE 7 unit 8: Second input (sensor output).
5. TYPE 8 unit 9: Second (set point for controlled variable) and third (integral portion of control signal) inputs.

5.1.2.3. Preparing the Boundary Value File

A complete listing of the boundary values is shown in the Table 5.1.2.3-1. The first column in the table represents time. The second column represents control variable 4 in the design. The process of creating a boundary file is discussed in the previous example.

Time	Control Signal
0	0.2
100	0.2
200	0.2
300	0.2
400	0.2
500	0.2

Table 5.1.2.3-1. *Example 2: Boundary values for the simulation*

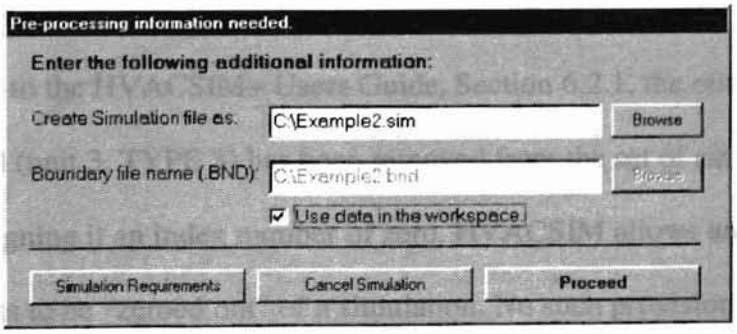


Time	Control Signal
600	0.2
600	0.25
800	0.25
1000	0.25
1200	0.25
1200	0.15
1400	0.15
1600	0.15
1800	0.15
2400	0.15
2800	0.15
3200	0.15
3600	0.15

Table 5.1.2.3-1 (Contd.). *Example 2: Boundary values for the simulation*

5.1.2.4. Running the simulation

After entering all the required data, the simulation was run. Figure 5.1.2.4-1 shows the initial dialog where the simulation file details can be entered. The data in the boundary file editor's grid is used to create the boundary value file. Figure 5.1.2.4-2 shows a plot of two reported variables, sensor output and set point controlled variable, with time.



Pre-processing information needed.

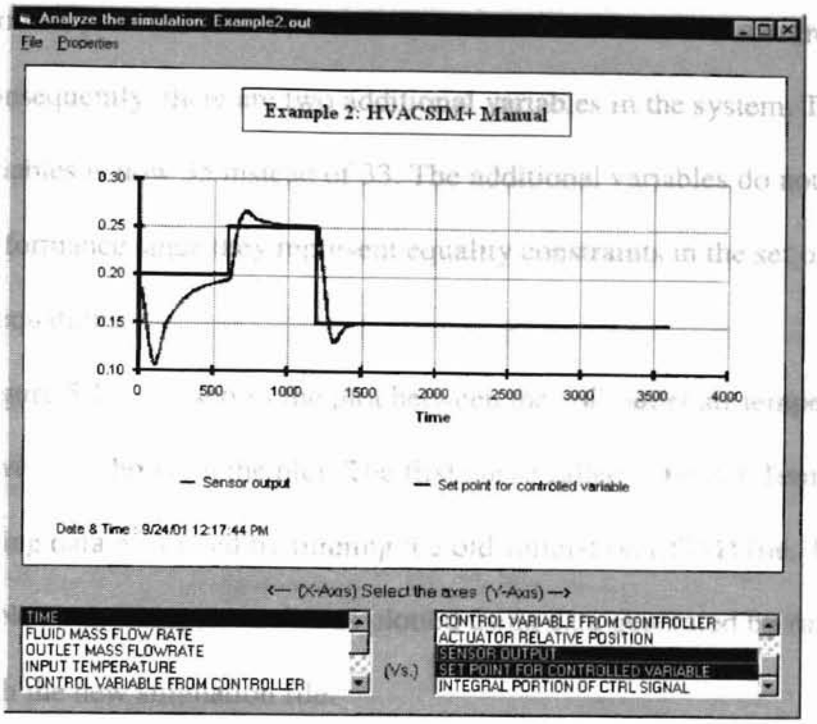
Enter the following additional information:

Create Simulation file as: C:\Example2.sim

Boundary file name (.BND): C:\Example2.bnd

☑ Use data in the workspace

Figure 5.1.2.4-1. *Example 2: Screen showing the simulation requirements*


Analyze the simulation: Example2.out
File Properties
Example 2: HVACSIM+ Manual
0.30
0.25
0.20
0.15
0.10
0
500
1000
1500
2000
2500
3000
3500
4000
Time
— Sensor output
— Set point for controlled variable
<-- (X-Axis) Select the axes (Y-Axis) -->
TIME
FLUID MASS FLOW RATE
OUTLET MASS FLOWRATE
INPUT TEMPERATURE
CONTROL VARIABLE FROM CONTROLLER
(Vs.)
CONTROL VARIABLE FROM CONTROLLER
ACTUATOR RELATIVE POSITION
SENSOR OUTPUT
SET POINT FOR CONTROLLED VARIABLE
INTEGRAL PORTION OF CTRL SIGNAL

Date & Time : 9/24/01 12:17:44 PM

Figure 5.1.2.4-2. *Example 2: Plot of reported variables*

5.1.2.5. Analyzing the simulation results

The simulation was recreated using the older version of HVACSIM+, and, as expected the output files generated using the new system matched exactly with their older versions.

Referring to the HVACSIM+ Users Guide, Section 6.2.1, the outlet temperature in the pipe model (unit 3, TYPE 3) has been removed from the set of simultaneous equations by assigning it an index number of zero. HVACSIM allows any number of inputs and outputs to be "zeroed out" of a simulation. No such provision is available in this version of HVACSIM. To implement such variables in the simulation no connections are established from or to these variables. In addition, this version of HVACSIM does not allow variables other than ambient temperature to have the same variable number. The old design for this simulation assigns the number eight to both inlet fluid pressure (unit 3, TYPE 3) and outlet pressure (unit 5, TYPE 2). The two variables are now treated separately. Consequently, there are two additional variables in the system. The total number of variables is now 35 instead of 33. The additional variables do not affect the simulation performance since they represent equality constraints in the set of simultaneous equations.

The Figure 5.1.2.5-1 shows the plot between the coil outlet air temperature and time. Two curves are shown in the plot. The first curve called "Old Air Temperature" was plotted using data generated by running the old simulation (.SIM) file. The second curve called "New Air Temperature" was plotted using data generated by running the simulation with the new simulation file.

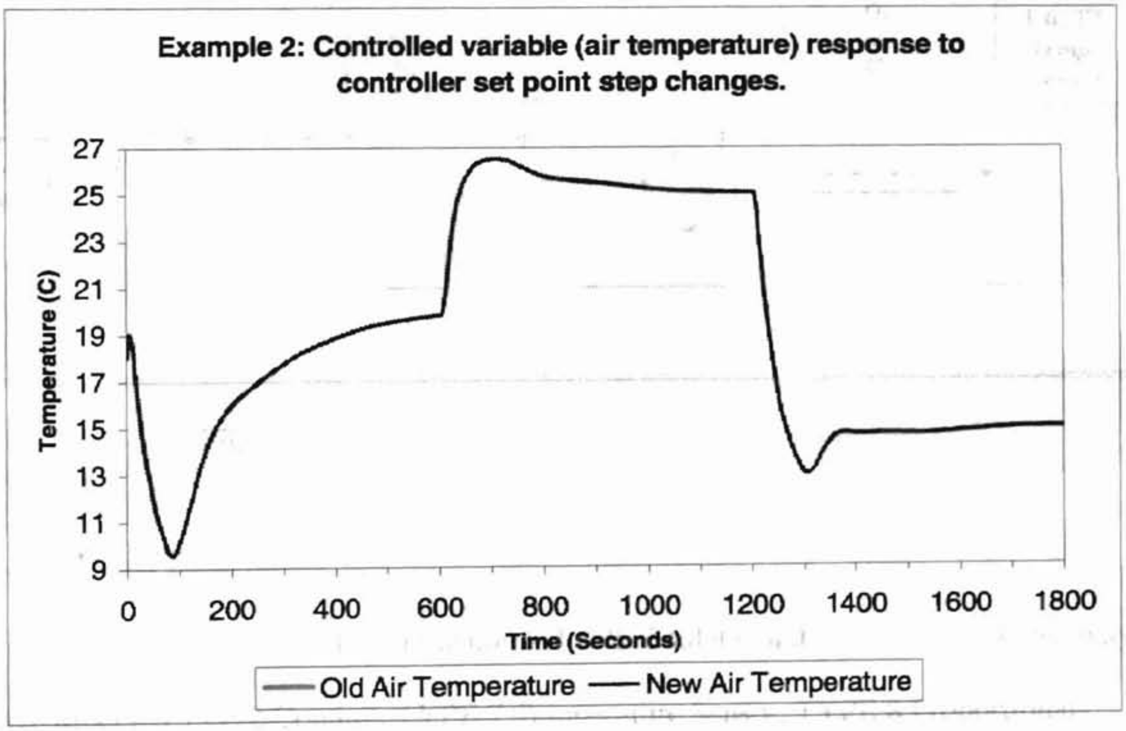


Figure 5.1.2.5-1. *Example 2: Plot between coil outlet air temperature and time*

5.1.3. Example 3: Cooling Coil Control Loop with Variable Water Flow Rate

This example represents a system in which the inlet water temperature remains constant while the flow rate is controlled. Unlike Example 2, this simulation does not include a pump. Flow is driven by the constant difference between the inlet fluid pressure of the pipe and the outlet pressure of the valve. The system is represented schematically in the Figure 5.1.3-1.

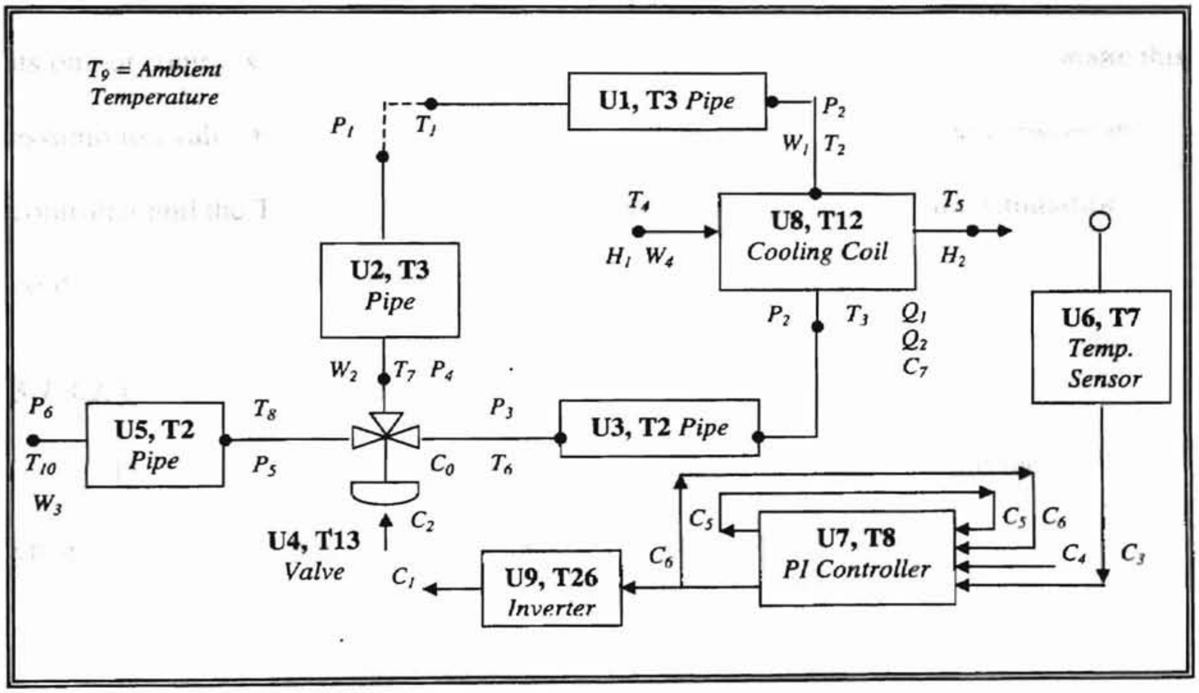


Figure 5.1.3-1. *Example 3: System schematic diagram*

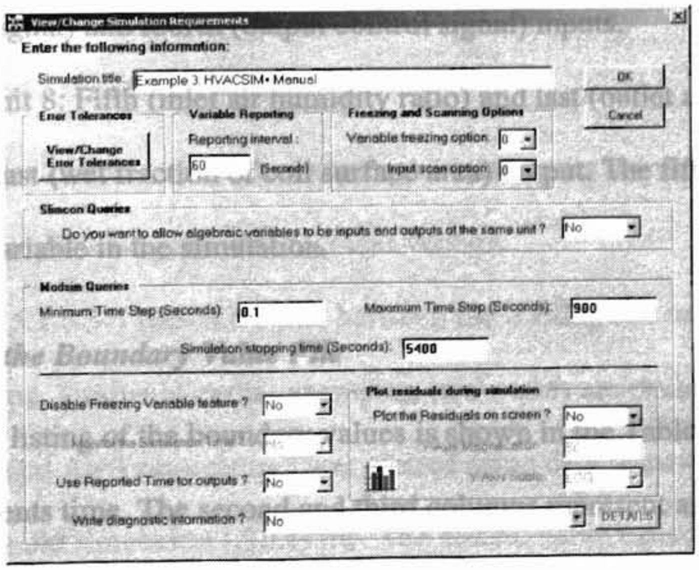
5.1.3.1. Model Description

This example consists of nine units- TYPE 3 Inlet Conduit, TYPE 13 Three-way Valve Model, TYPE 2 Conduit, TYPE 7 Temperature Sensor, TYPE 8 Proportional-Integral Controller, TYPE 12 Cooling or Dehumidifying Coil and TYPE 26 Control Signal Inverter. There are two each of TYPE 3 and TYPE 2 components. The categorization of TYPE 3, TYPE 13, TYPE 2, TYPE 7 and TYPE 8 components is given in the earlier example. The TYPE 12 component is a heat exchanger where there is no pressure drop. The TYPE 26 component has no fluid flow through it and belongs to the category of sensors and controllers. Thirty-four state variables are identified in the figure: seven pressures, four flow variables, eleven temperatures, eight control signals, two power variables and two humidity variables. Descriptions of the units involved can be found in the Appendix A.

The TYPE 8 Proportional and Integral controller model assumes that increasing its output control signal will cause the input controlled variable to increase. To make this assumption valid, the TYPE 26 Control Signal Inverter model was added between the controller and the Three-way valve model. A Heating Coil, if used in the simulation, could replace both TYPE 12 and TYPE 26 models.

5.1.3.2. Creating the simulation

Figure 5.1.3.2-1 shows the dialog displaying the simulation requirements. The error tolerances selected are the same as the previous example.



View/Change Simulation Requirements

Enter the following information:

Simulation title: Example 3 HVACSIM+ Manual

Error Tolerances

Variable Reporting

Reporting interval: 60 (Seconds)

Freezing and Scanning Options

Variable freezing option: 0

Input scan option: 0

Simcon Queries

Do you want to allow algebraic variables to be inputs and outputs of the same unit? No

Modsim Queries

Minimum Time Step (Seconds): 0.1

Maximum Time Step (Seconds): 900

Simulation stopping time (Seconds): 5400

Disable Freezing Variable feature? No

Use Reported Time for outputs? No

Write diagnostic information? No

Plot residuals during simulation

Plot the Residuals on screen? No

Figure 5.1.3.2-1. *Example 3: Dialog showing the simulation requirements*

The workspace diagram included in Appendix C represents the simulation in the application. It shows the various units involved and the connections between them along with the reported variables in the simulation. The two boundary conditions are named 'Air Inlet Temperature' and 'Humidity Ratio' respectively.

Double clicking on a unit displays a dialog where the user enters the initial values for the input and output variables and parameters. The parameters for the cooling coil model were determined for a specific piece of equipment in a different simulation. These parameter values were then fed into this simulation. The Appendix C gives a listing of the initial values for all the units involved in the simulation.

The reported variables involved in the system are:

1. TYPE 3 unit 1: Last output (mass flow rate of the fluid).
2. TYPE 13 unit 4: Sixth (control variable from controller) and seventh (actuator relative position) inputs.
3. TYPE 7 unit 6: Last input (sensor output)
4. TYPE 8 unit 7: Second (set point for controlled variable), third (integral portion of control signal) and fourth (output control signal) inputs.
5. TYPE 12 unit 8: Fifth (inlet air humidity ratio) and last (outlet air humidity ratio) inputs and last (wet fraction of coil surface area) output. The fifth input is also the boundary variable in the simulation.

5.1.3.3. Preparing the Boundary Value File

A complete listing of the boundary values is shown in the Table 5.1.3.3-1. The first column represents time. The second and third columns represent air inlet temperature and humidity ratio.

Time	Air Inlet Temperature	Humidity Ratio
0	25	0.016
450	25	0.016
900	25	0.016
1350	25	0.016
1800	25	0.016
1800	30	0.022
2250	30	0.022
2700	30	0.022
3150	30	0.022
3600	30	0.022
3600	20	0.01
4050	20	0.01
4500	20	0.01
4950	20	0.01
5400	20	0.01

Table 5.1.3.3-1. *Example 3: Boundary values for the simulation*

5.1.3.4. Running the simulation

After entering all the required data, the simulation was run. The initial pre-processing dialog that is displayed before the simulation begins is shown in the earlier examples. Plots generated from the application can be printed by selecting the Print menu option in the dialog opened by clicking the 'Show Results' button after the simulation has ended.

5.1.3.5. Analyzing the simulation results

As in the previous examples, the output generated using the new system exactly matched with the output files generated by older version.

The Figure 5.1.3.5-1 shows the plot between the cooling coil outlet air temperature and time. Similar to earlier examples, two curves are drawn in the plot. The first curve called "Old Air Temperature" was plotted using data generated by running the simulation with the old simulation (.SIM) file. The second curve called "New Air Temperature" was plotted using data generated by running the simulation with the new simulation file. The proportional and integral constants have been carefully chosen for the simulation. For the values shown in the graph, damped oscillations are produced with rapid system response. Reducing these values by 50 percent, from 9 and 0.12 to 6 and 0.08 produces a rapid and oscillation-free response. Values of 12 and 0.12 for proportional and integral gains respectively produced undamped oscillations, with a period of about 92 seconds. This is because of the prevalent hysteresis effect in the control valve, which allows errors to accumulate while the actuator crosses its dead band.

The greater degree of activity and faster rates of change in this simulation cause HVACSIM+ to take shorter time steps, increasing both the number of time steps taken and the execution time by about 75 percent. To track the fast oscillations in the simulation HVACSIM spends most of its time at the minimum time step and never allows the step size to exceed six seconds once the oscillations begin. The average time step between simulation times 3600 and 5400 in this example is under one second. Users should be aware that modeling a system that oscillates relatively rapidly could be extremely expensive computationally.

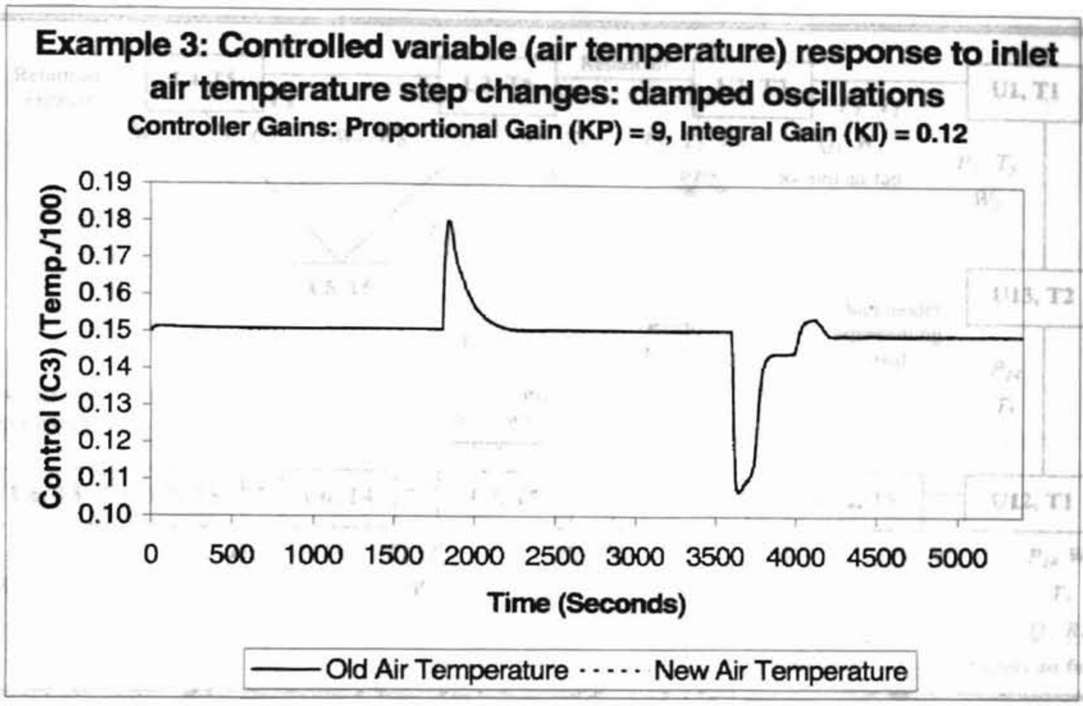


Figure 5.1.3.5-1. *Example 3: Controlled variable (air temperature) response*

5.1.4. Example 4: Air handling unit without heat exchangers

This example represents an air handler model. The primary purpose of this example is to check the pressure and flow relationships in the model. Dampers are used to represent the flow resistance of heat exchangers. Heat transfer effects are not considered in this example. A duct model has been used to represent the load. The system is represented schematically in the Figure 5.1.4-1.

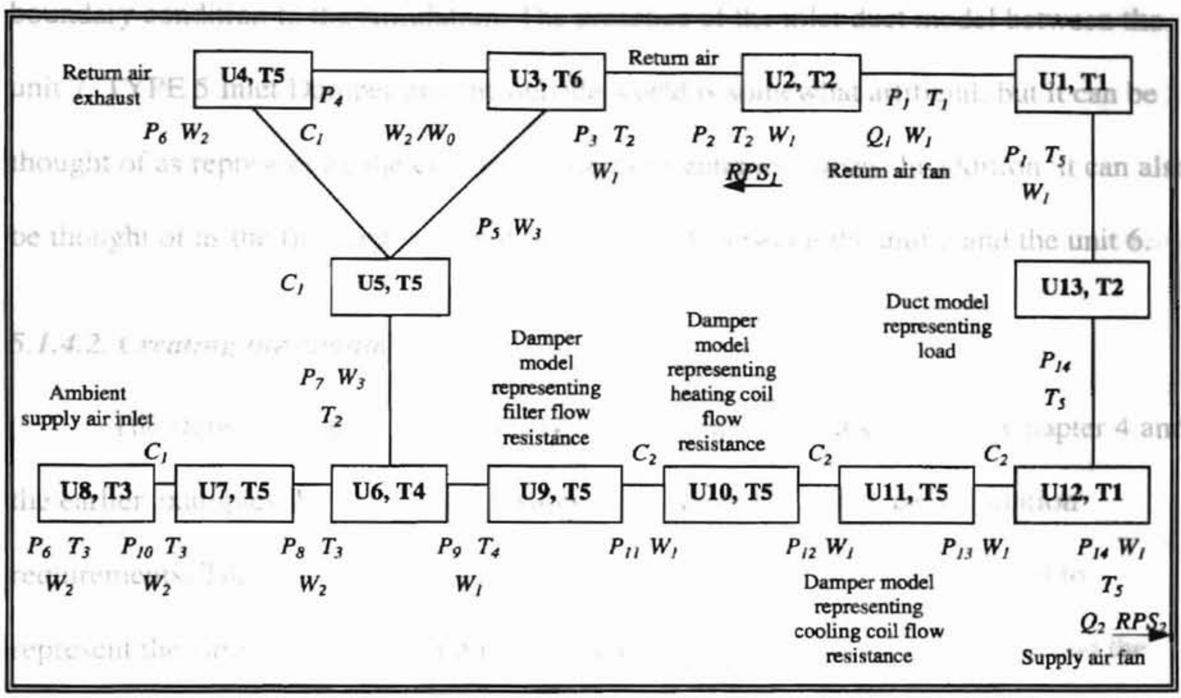


Figure 5.1.4-1. *Example 4: System Schematic*

5.1.4.1. System Description

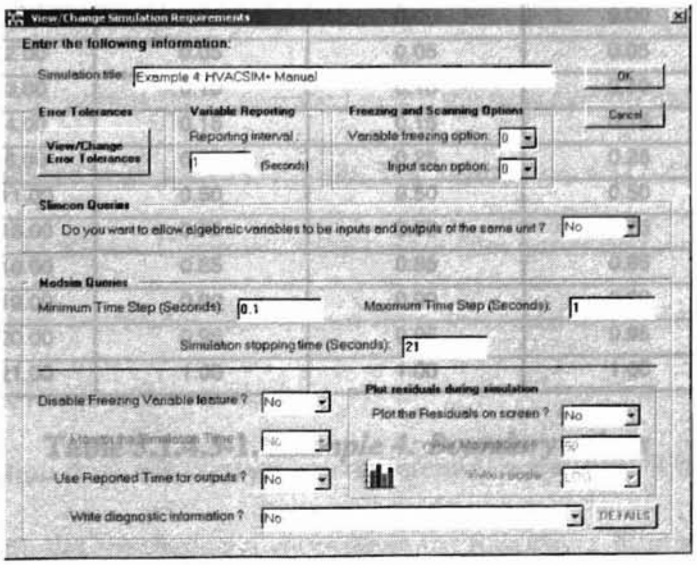
This example consists of thirteen units- TYPE 1 Fan Or Pump, TYPE 2 Conduit (Duct Or Pipe), TYPE 6 Flow Split, TYPE 5 Damper Or Valve, TYPE 4 Flow Merge and TYPE 3 Inlet Conduit (Duct Or Pipe). There are two each of TYPE 1 and TYPE 2 components. There are six TYPE 5 components. The categorization of TYPE 1, TYPE 2 and TYPE 3 components is given in the earlier examples. The inlet pressure for TYPE 4, TYPE 5 and TYPE 6 components is calculated from flow rate and outlet pressure. Descriptions of the units involved can be found in the Appendix A.

Forty state variables are identified in the figure: 15 pressures, 4 flow variables, 11 temperatures, 6 control signals, 2 power variables and 2 rotational speed variables. The ambient supply airflow rate is labeled W_2 in the figure. The unit 8, TYPE 3 Inlet Duct serves to calculate this flow rate and to make the outdoor air pressure, labeled P_6, a boundary condition to the simulation. The presence of the inlet duct model between the unit 7, TYPE 5 Inlet Damper and the outside world is somewhat artificial, but it can be thought of as representing the combined effects of entrance losses. In addition, it can also be thought of as the flow resistance of any ductwork between the unit 7 and the unit 6.

5.1.4.2. Creating the simulation

The steps involved in creating a typical simulation are discussed in Chapter 4 and the earlier examples. Figure 5.1.4.2-1 shows the dialog displaying the simulation requirements. The workspace diagram included in the Appendix C can be used to represent the simulation in the application. It shows the various units involved and the connections between them along with the reported variables in the simulation.

Double clicking on a unit displays a dialog where the user enters the initial values for the input and output variables and parameters. The Appendix C provides a listing of the initial values for each unit involved in the simulation. The parameters for the cooling coil model were determined for a specific piece of equipment in a different simulation.


View/Change Simulation Requirements
OK
Cancel
View/Change Error Tolerances
DETAILS

Enter the following information:

Simulation title: Example 4 HVACSIM+ Manual

Error Tolerances

Variable Reporting
Reporting interval: 1 (Seconds)

Freezing and Scanning Options
Variable freezing option: 0
Input scan option: 0

Slimcon Queries
Do you want to allow algebraic variables to be inputs and outputs of the same unit ? No

Modsim Queries
Minimum Time Step (Seconds): 0.1
Maximum Time Step (Seconds): 1
Simulation stopping time (Seconds): 21

Disable Freezing Variable feature ? No
Monitor the Simulation Time ? No
Use Reported Time for outputs ? No
Write diagnostic information ? No

Plot residuals during simulation
Plot the Residuals on screen ? No
Plot Magnification ? 50

Figure 5.1.4.2-1. *Example 4: Simulation requirements dialog*

The reported variables involved in the system are:

1. TYPE 1 unit 12: Second input (outlet pressure).
2. TYPE 1 unit 1: First input (mass flow rate).
3. TYPE 5 unit 4: First (mass flow rate) and last (relative position of damper) inputs.
4. TYPE 4 unit 6: Last output (outlet temperature).

The reported variables are suitably named so that they can be easily identified for plotting. If the variables are not named by the user, the application uses the description names provided in the Types Component Library (TYPAR.DAT) file as variable names.

The boundary conditions in the simulation represent the same data in the boundary value file. They are all given the same name as 'Control Signal'.

5.1.4.3. Preparing the Boundary Value File

A complete listing of the boundary values is shown in Table 5.1.4.3-1. The first column represents time. The second, third and fourth columns represent control signals.

Time	Control Signal	Control Signal	Control Signal
0.00	0.00	0.00	0.00
1.00	0.00	0.00	0.00
2.00	0.05	0.05	0.05
3.00	0.10	0.10	0.10
4.00	0.15	0.15	0.15
6.00	0.25	0.25	0.25
11.00	0.50	0.50	0.50
16.00	0.75	0.75	0.75
18.00	0.85	0.85	0.85
19.00	0.90	0.90	0.90
20.00	0.95	0.95	0.95
21.00	1.00	1.00	1.00

Table 5.1.4.3-1. *Example 4: Boundary values*

5.1.4.4. Running the simulation

After entering all the required data, the simulation was run. Figure 5.1.4.4-1 shows the plot of a reported variable (mass flow rate of fluid) with time.

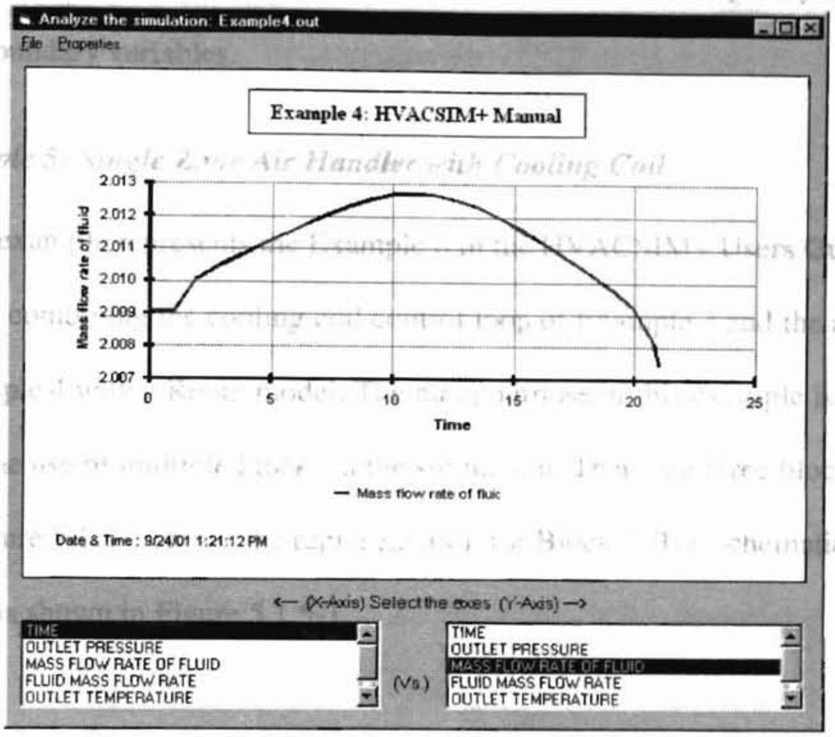


Figure 5.1.4.4-1. *Example 4: Plot of reported variables*

5.1.4.5. Analyzing the simulation results

Once again, the output files generated using the new system exactly matched with the output files generated'with the old version. The simulation was recreated using the older version of HVACSIM+. Comparing the output files generated, we can conclude that the simulation was successful. The output files generated using the new system matched exactly with their older versions.

This simulation does not contain any differential equations. When only algebraic equations are present, the simulation always takes the maximum time step, with two exceptions. The two exceptions are the first time step which is always the minimum time step and the last time step which is chosen to exactly reach the simulation stopping time. In the absence of differential equations, there is no start-up transient, making initialization unnecessary, and changes in the simulation are driven purely by the time-dependent boundary variables.

5.1.5. Example 5: Single Zone Air Handler with Cooling Coil

This example represents the Example 6 in the HVACSIM+ Users Guide. It was generated by combining the cooling coil control loop of Example 3 and the air handling unit of Example 4 with a Room model. The main purpose of this example is to demostrate the use of multiple blocks in the simulation. There are three blocks in this example. Figure 5.1.3-1 shows the representation for Block 1. The schematic of Block 2 and Block 3 is shown in Figure 5.1.5-1.

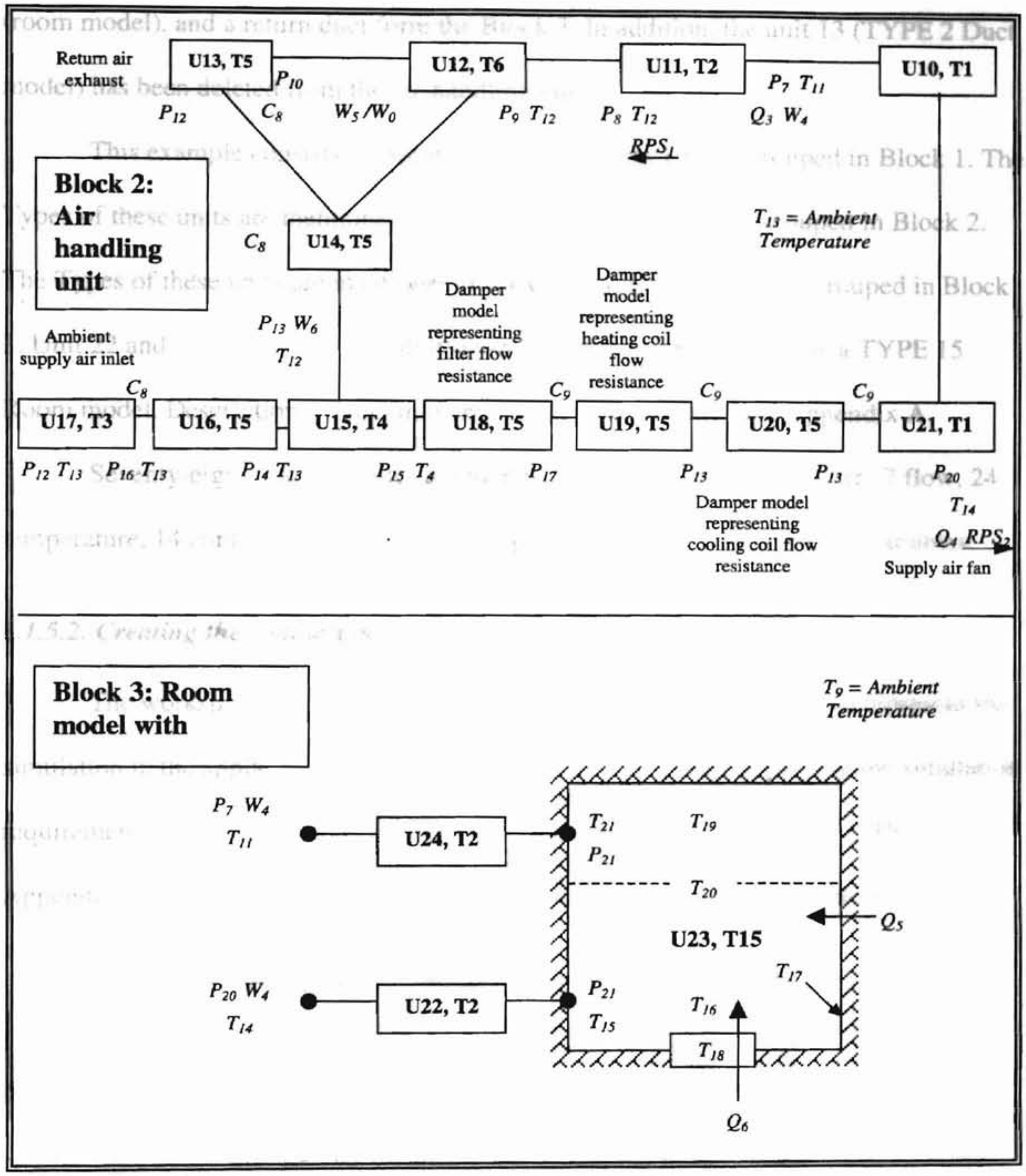


Figure 5.1.5-1. *Example 5: System schematic diagram*

5.1.5.1. Model Description

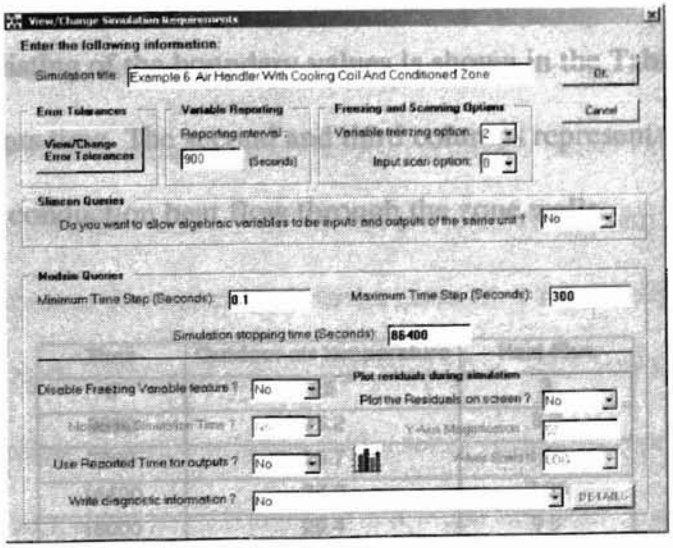
Block 1 consists of the cooling coil control loop from Example 3. Block 2 consists of the air-handling unit from Example 4. A supply duct, a conditioned zone (room model), and a return duct form the Block 3. In addition, the unit 13 (TYPE 2 Duct model) has been deleted from the air-handling unit.

This example consists of twenty-four units. Units 1-9 are grouped in Block 1. The Types of these units are mentioned in Example 3. Units 10-21 are grouped in Block 2. The Types of these units are mentioned in Example 4. Units 22-24 are grouped in Block 3. Unit 22 and 24 are TYPE 2 Conduit (Duct or Pipe) models. Unit 23 is a TYPE 15 Room model. Descriptions of the units involved can be found in the Appendix A.

Seventy-eight state variables are identified in the figure: 23 pressure, 7 flow, 24 temperature, 14 control-signal, 2 rotational speed, 6 power, and 2 humidity variables.

5.1.5.2. Creating the simulation

The workspace diagram included in the Appendix C can be used to represent the simulation in the application. Figure 5.1.5.2-1 shows the dialog displaying the simulation requirements. This simulation uses the same error tolerances as Example 1. The Appendix C lists the initial values for all the units involved in the simulation.


View/Change Simulation Requirements
OK
Cancel
View/Change Error Tolerances
DETAILS

Enter the following information:

Simulation title: Example 6 Air Handler With Cooling Coil And Conditioned Zone

Error Tolerances

Variable Reporting
Reporting interval: 900 (Seconds)

Freezing and Scanning Options
Variable freezing option: 2
Input scan option: 0

Simcon Queries
Do you want to allow algebraic variables to be inputs and outputs of the same unit ? No

Modsim Queries
Minimum Time Step (Seconds): 0.1
Maximum Time Step (Seconds): 300
Simulation stopping time (Seconds): 86400

Disable Freezing Variable feature ? No
Modify the Simulation Time ?
Use Reported Time for outputs ? No
Write diagnostic information ? No

Plot residuals during simulation
Plot the Residuals on screen ? No
Y-Axis Magnification
Y-Axis Scale LOG

Figure 5.1.5.2-1. *Example 5: Simulation requirements dialog*

The reported variables involved in the system are:

1. TYPE 5 unit 16: First output (inlet pressure)
2. TYPE 3 unit 1: Second output (mass flow rate of the fluid)
3. TYPE 12 unit 8: Third input (mass flow rate of dry air)
4. TYPE 7 unit 6: First input (ambient temperature)
5. TYPE 12 unit 8: Fourth input (inlet dry bulb air temperature)
6. TYPE 15 unit 23: Fourth input (wall mass temperature)
7. TYPE 15 unit 23: Fifth output (average room air temperature)
8. TYPE 15 unit 23: Sixth output (exhaust air temperature)
9. TYPE 13 unit 4: Sixth input (control variable from controller)
10. TYPE 13 unit 4: Sixth output (valve stem relative position)
11. TYPE 7 unit 6: Second input (sensor output)
12. TYPE 12 unit 8: Sixth output (wet fraction of coil surface area)
13. TYPE 12 unit 8: Eighth input (outlet air humidity ratio)

5.1.5.3. Preparing the Boundary Value File

A complete listing of the boundary values is shown in the Table 5.1.5.3-1. The first column represents time. The second and third columns represent the outdoor air temperature and the conduction heat flow through the zone walls.

Time	Outdoor air temperature	Heat Flux
0	25	3
3600	25.2	3.3
7200	25.7	4.05
14400	27.9	7.35
18000	29.4	9.6

Table 5.1.5.3-1. *Example 5: Boundary values*

Time	Outdoor air temperature	Heat Flux
21600	31.1	12.15
25200	32.7	14.55
28800	33.9	16.35
32400	34.7	17.55
33120	34.77	17.66
33840	34.84	17.77
34560	34.9	17.85
35280	34.93	17.9
36000	35	18
39600	34.7	17.55
41400	34.45	17.17
41760	34.38	17.07
42120	34.3	16.95
43200	34	16.5
46800	32.9	14.85
50400	31.6	12.9
54000	30.3	10.95
57600	29.2	9.3
61200	28.2	7.8
64800	27.4	6.6
68400	26.8	5.7
72000	26.3	4.95
75600	25.8	4.2
79200	25.4	3.6
82800	25.1	3.15
86400	25	3
90000	25.2	3.3
93600	25.7	4.05
97200	26.6	5.4

Table 5.1.5.3-1(Contd.). *Example 5: Boundary values*

5.1.5.4. Running the simulation

Initially, when the simulation was run with arbitrary initial values, it failed to converge on the first time step and a FORTRAN error (floating point overflow) was displayed. Investigation of the reported variable, mass flow rate of dry air, showed a value less than zero at the first time step in the output file. This may suggest that the equation solver is unable to find a solution to the set of equations representing airflow rates and pressures. Since the set of equations is divided between blocks 2 and 3, they are more difficult to solve. The simulation was rerun with a better set of initial values obtained from Example 3 and Example 4 and it ran smoothly.

5.1.5.5. Analyzing the simulation results

Similar to the previous examples, this simulation results were compared their old counterparts. The results matched exactly. The Figures 5.1.5.5-1 and 5.1.5.5-2 show plots of the temperature response and the control signal response, respectively.

This simulation emphasizes that dividing a tightly coupled set of equations into separate blocks (or superblocks) can sometimes cause problems. For this simulation, a better set of initial guesses solved the problem. In some cases, it may be necessary to modify the block structure of a simulation.

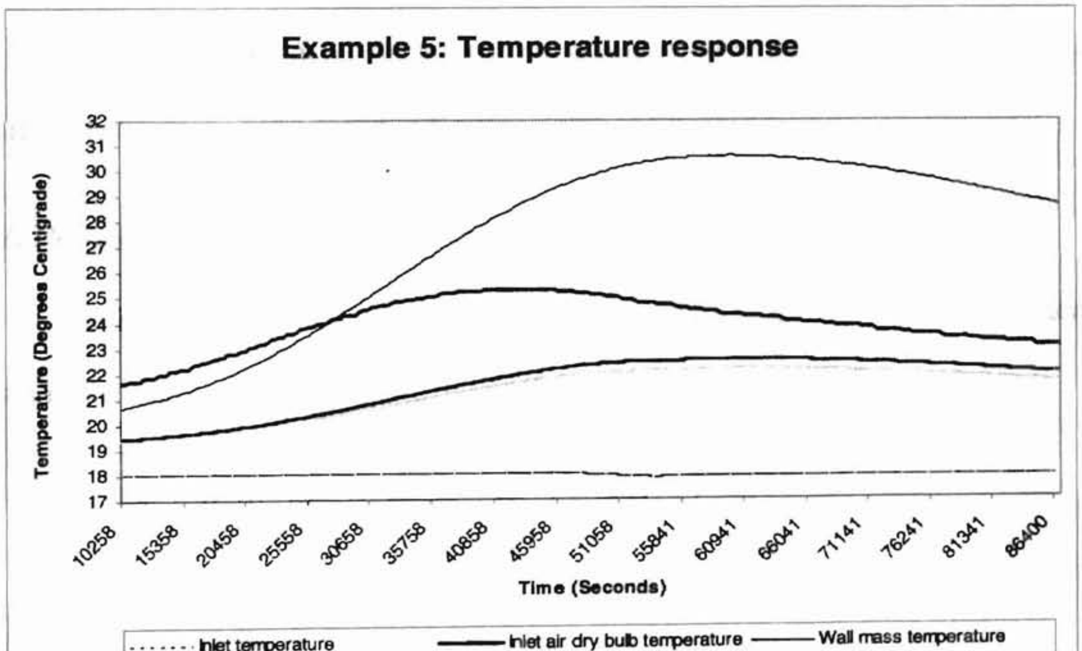


Table 5.1.5.5-1. *Example 5: Temperature response*

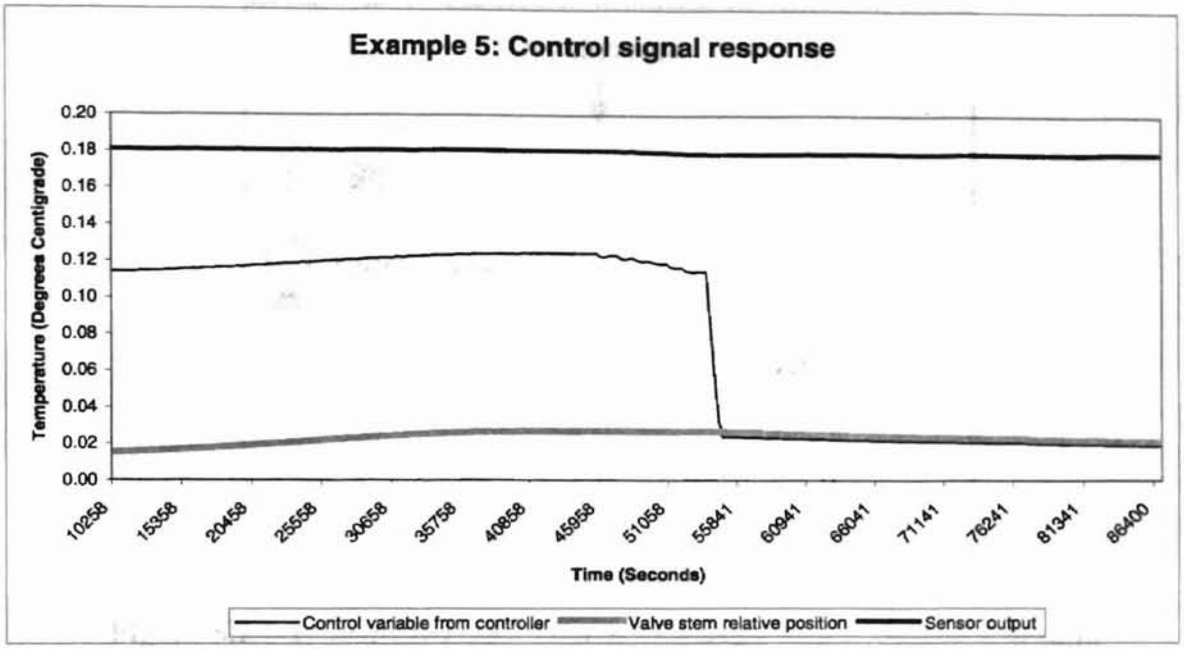


Table 5.1.5.5-2. *Example 5: Control signal response*

5.2. Simulation of Smart Bridge Systems

The following sections discuss the implementation of three configurations of a Smart Bridge system, introduced in Section 3.3.2 of Chapter 3.

5.2.1. Design Procedure Step 1

This simulation considers the load-side sub-system of the Smart Bridge system as shown in the Figure 5.2.1-1.

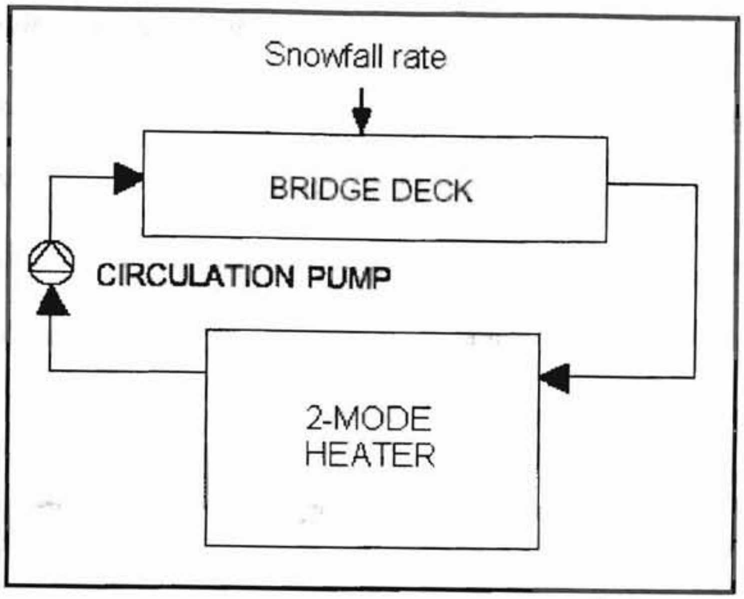


Figure 5.2.1-1. *Smart Bridge – Design procedure step 1: System schematic*

5.2.1.1. Model Description

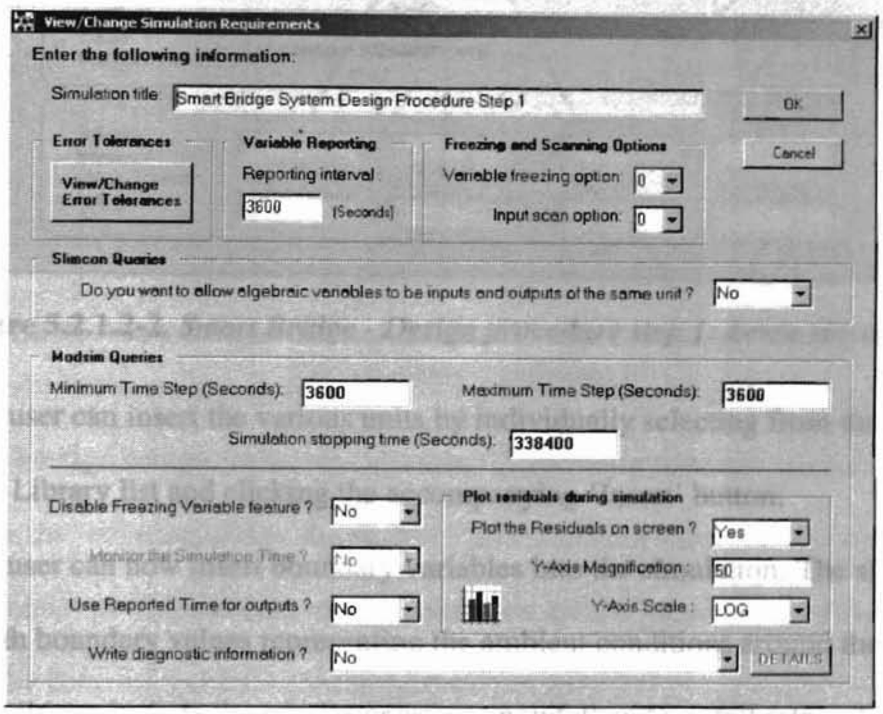
This configuration consists of three units: the TYPE 700 slab model that represents the bridge deck, the TYPE 740 2-mode heater model and the TYPE 750 pump model. Descriptions of the units involved can be found in Appendix A.

The source-side sub-system in the Smart Bridge system, consisting of a ground loop heat exchanger, is replaced with a heater model. The model incorporates the important properties of the source-side of the system.

The input variable, snowfall rate, on the slab model is used as a boundary variable to trigger a snow event. The outlet fluid temperature from the slab model is supplied to the heater. When the deck temperature reduces, the heater heats a fluid, which is either a mixture of water and propylene glycol or a mixture of water and ethylene glycol, and supplies it to the deck.

5.2.1.2. Creating the system model

The user can initially set the simulation requirements by selecting the SETTINGS->SIMULATION REQUIREMENTS menu option. The user may enter suitable data for the simulation title, the various error tolerances, reporting interval, freezing and scanning options, required diagnostics and simulation times. The Figure 5.2.1.2-1 shows the dialog displaying the selected simulation requirements.


View/Change Simulation Requirements
OK
Cancel
View/Change Error Tolerances
DETAILS

Enter the following information:

Simulation title: Smart Bridge System Design Procedure Step 1

Error Tolerances

Variable Reporting
Reporting interval: 3600 (Seconds)

Freezing and Scanning Options
Variable freezing option: 0
Input scan option: 0

Slimcon Queries
Do you want to allow algebraic variables to be inputs and outputs of the same unit ? No

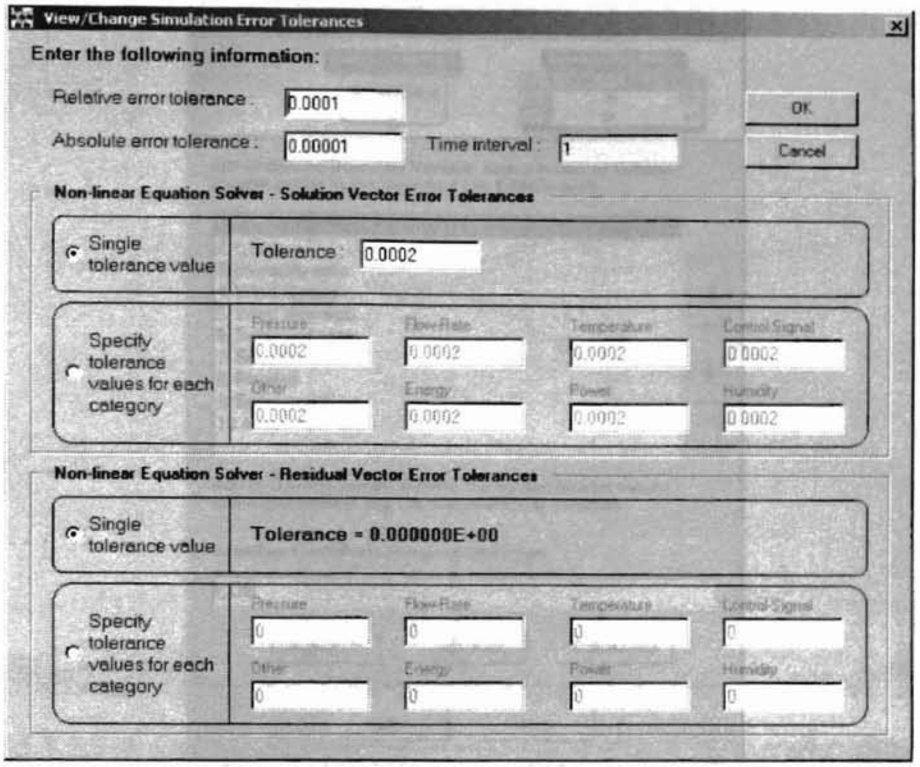
Modsim Queries
Minimum Time Step (Seconds): 3600
Maximum Time Step (Seconds): 3600
Simulation stopping time (Seconds): 338400

Disable Freezing Variable feature ? No
Monitor the Simulation Time ? No
Use Reported Time for outputs ? No
Write diagnostic information ? No

Plot residuals during simulation
Plot the Residuals on screen ? Yes
Y-Axis Magnification: 50
Y-Axis Scale: LOG

Figure 5.2.1.2-1. *Smart Bridge - Design procedure step 1: Simulation requirements*

The error tolerances selected for the simulation are shown in Figure 5.2.1.2-2. It should be noted that the time interval size is set at the minimum value of 1 indicating that the integration routine will consider more iterations if the solution of the differential equations is not converging.



Enter the following information:

Relative error tolerance : 0.0001

Absolute error tolerance : 0.00001   Time interval : 1

Non-linear Equation Solver - Solution Vector Error Tolerances

● Single tolerance value   Tolerance : 0.0002

○ Specify tolerance values for each category

Pressure	FlowRate	Temperature	Control Signal
0.0002	0.0002	0.0002	0.0002
Other	**Energy**	**Power**	**Humidity**
0.0002	0.0002	0.0002	0.0002

Non-linear Equation Solver - Residual Vector Error Tolerances

● Single tolerance value   **Tolerance = 0.000000E+00**

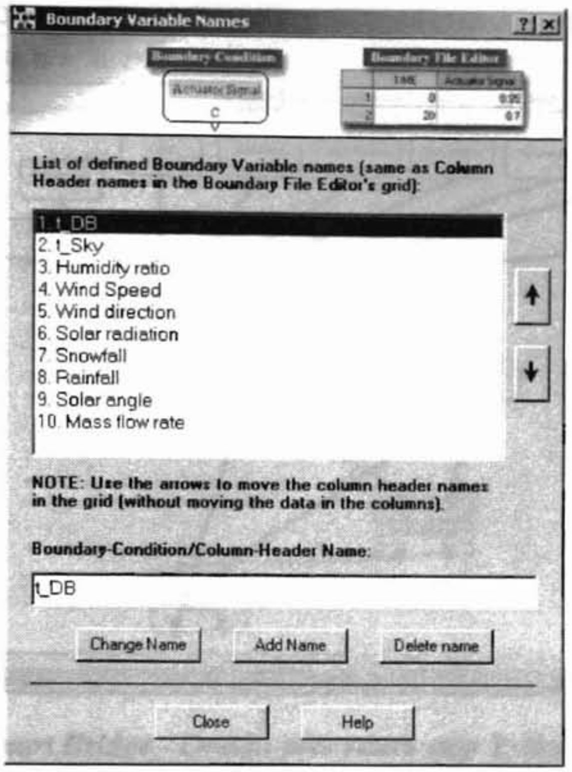
○ Specify tolerance values for each category

Pressure	FlowRate	Temperature	Control Signal
0	0	0	0
Other	**Energy**	**Power**	**Humidity**
0	0	0	0

Figure 5.2.1.2-2. *Smart Bridge - Design procedure step 1: Error tolerances*

The user can insert the various units by individually selecting from the Types Component Library list and clicking the accompanying 'Insert' button.

The user can now insert boundary variables into the simulation. The slab model is supplied with boundary values representing the ambient conditions around the deck. The ambient conditions include the wind properties, snowfall and rainfall rates, sky temperature, etc. Boundary variables are given names that may be created using the appropriate menu option in the boundary file editor. A dialog for creating and modifying time-dependent boundary condition names in the simulation using the boundary file editor is shown in Figure 5.2.1.2-2. The figure shows the names created for this simulation.



List of defined Boundary Variable names (same as Column Header names in the Boundary File Editor's grid):

1. t_DB
2. t_Sky
3. Humidity ratio
4. Wind Speed
5. Wind direction
6. Solar radiation
7. Snowfall
8. Rainfall
9. Solar angle
10. Mass flow rate

NOTE: Use the arrows to move the column header names in the grid (without moving the data in the columns).

Boundary-Condition/Column-Header Name:

t_DB

Figure 5.2.1.2-3. *Smart Bridge - Design procedure step 1: Dialog showing the names for the boundary conditions*

The average bridge surface-temperature is the major reported variable in the system since the intention of this simulation is to maintain a value for this variable. The other important reported variables in the simulation are the snow free area ratio on the bridge and the fluid temperatures in the system.

Figure 5.2.1.2-4 shows the workspace configuration of the system. The figure shows the various units, boundary conditions and reported variables in the simulation.

The user can now enter the initial values for inputs, outputs and parameters of the units involved in the simulation by double clicking on the components. The initial values selected for each unit are shown in the Figures 5.2.1.2-5 and 5.2.1.2-8. For all units, the initial values for input and output variables are set at zero.

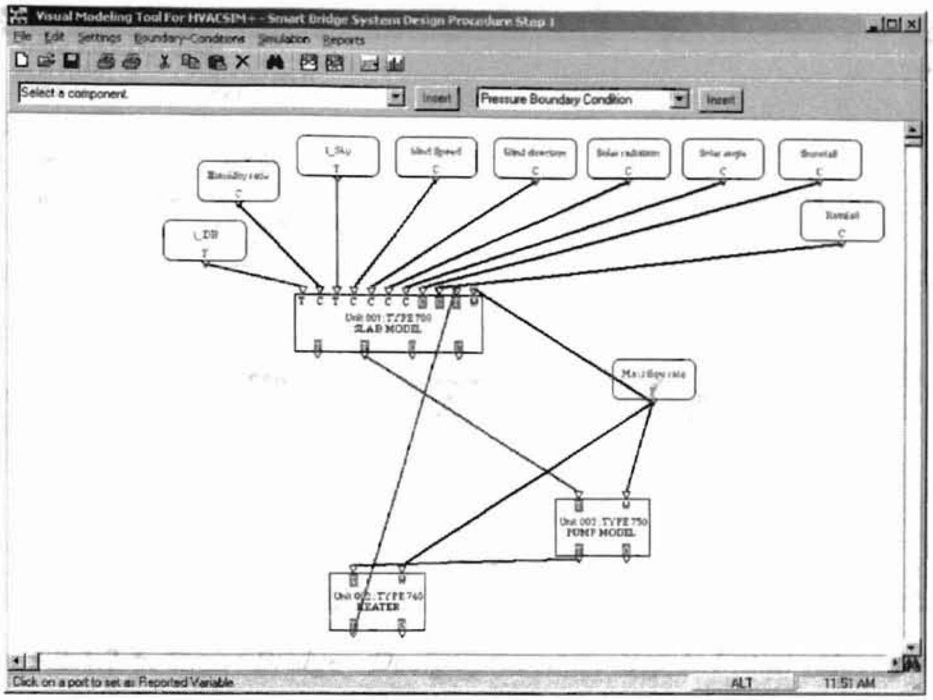
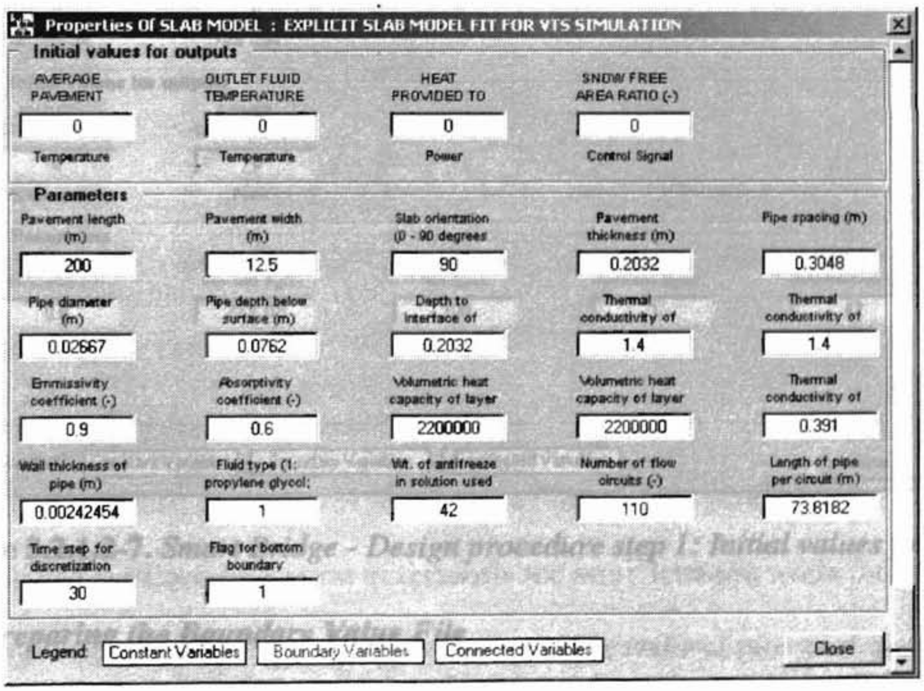


Figure 5.2.1.2-4. *Smart Bridge - Design procedure step 1: Screenshot showing the various units, boundary conditions and reported variables in the simulation*



Properties Of SLAB MODEL : EXPLICIT SLAB MODEL FIT FOR VTS SIMULATION

Initial values for outputs

AVERAGE PAVEMENT	OUTLET FLUID TEMPERATURE	HEAT PROVIDED TO	SNOW FREE AREA RATIO (-)
0	0	0	0
Temperature	Temperature	Power	Control Signal

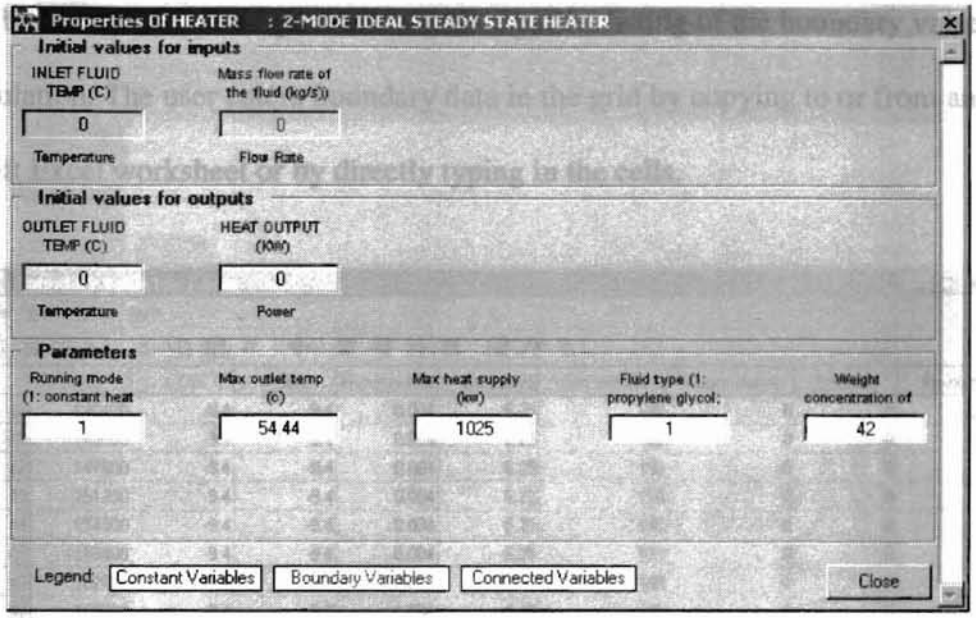
Parameters

Pavement length (m)	Pavement width (m)	Slab orientation (0 - 90 degrees	Pavement thickness (m)	Pipe spacing (m)
200	12.5	90	0.2032	0.3048
Pipe diameter (m)	Pipe depth below surface (m)	Depth to interface of	Thermal conductivity of	Thermal conductivity of
0.02667	0.0762	0.2032	1.4	1.4
Emmissivity coefficient (-)	Absorptivity coefficient (-)	Volumetric heat capacity of layer	Volumetric heat capacity of layer	Thermal conductivity of
0.9	0.6	2200000	2200000	0.391
Wall thickness of pipe (m)	Fluid type (1: propylene glycol;	Wt. of antifreeze in solution used	Number of flow circuits (-)	Length of pipe per circuit (m)
0.00242454	1	42	110	73.8182
Time step for discretization	Flag for bottom boundary			
30	1			

Legend: Constant Variables | Boundary Variables | Connected Variables

Close

Figure 5.2.1.2-5. *Smart Bridge - Design procedure step 1: Initial values for unit 1*


Properties Of HEATER : 2-MODE IDEAL STEADY STATE HEATER

Initial values for inputs

INLET FLUID TEMP (C): 0 (Temperature)

Mass flow rate of the fluid (kg/s)): 0 (Flow Rate)

Initial values for outputs

OUTLET FLUID TEMP (C): 0 (Temperature)

HEAT OUTPUT (KW): 0 (Power)

Parameters

Running mode (1: constant heat: 1

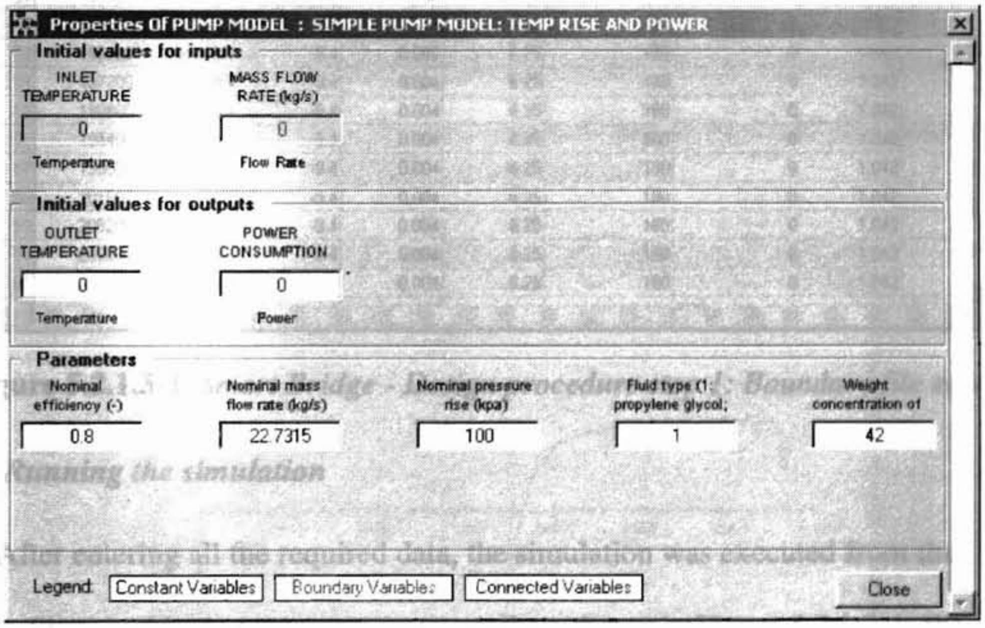
Max outlet temp (c): 54 44

Max heat supply (kw): 1025

Fluid type (1: propylene glycol;: 1

Weight concentration of: 42

Figure 5.2.1.2-6. *Smart Bridge - Design procedure step 1: Initial values for unit 2*


Properties Of PUMP MODEL : SIMPLE PUMP MODEL: TEMP RISE AND POWER

Initial values for inputs

INLET TEMPERATURE: 0 (Temperature)

MASS FLOW RATE (kg/s): 0 (Flow Rate)

Initial values for outputs

OUTLET TEMPERATURE: 0 (Temperature)

POWER CONSUMPTION: 0 (Power)

Parameters

Nominal efficiency (-): 0.8

Nominal mass flow rate (kg/s): 22.7315

Nominal pressure rise (kpa): 100

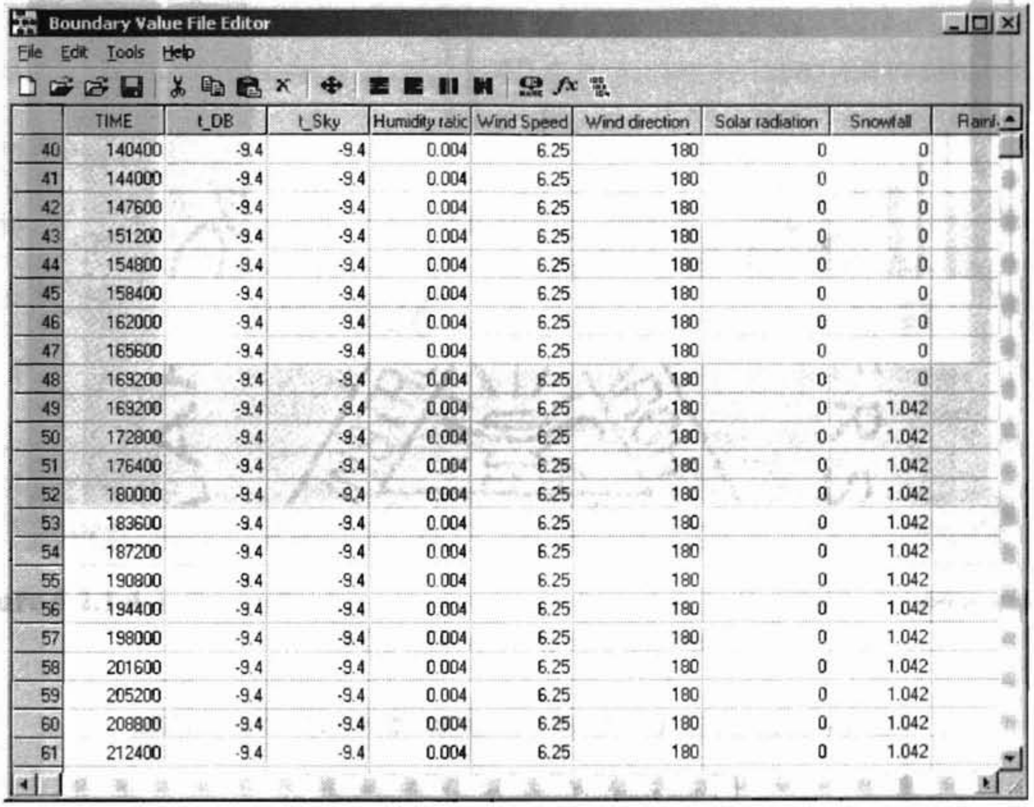
Fluid type (1: propylene glycol;: 1

Weight concentration of: 42

Figure 5.2.1.2-7. *Smart Bridge - Design procedure step 1: Initial values for unit 3*

5.2.1.3. Preparing the Boundary Value File

A complete listing of boundary values selected for the boundary variables is shown in the Appendix D. A snow event is represented by setting the snowfall rate to a value other than zero. The boundary file editor shown in Figure 5.2.1.3-1 can be used to create a boundary file. The figure also shows a partial listing of the boundary values for this simulation. The user enters boundary data in the grid by copying to or from an open Microsoft Excel worksheet or by directly typing in the cells.

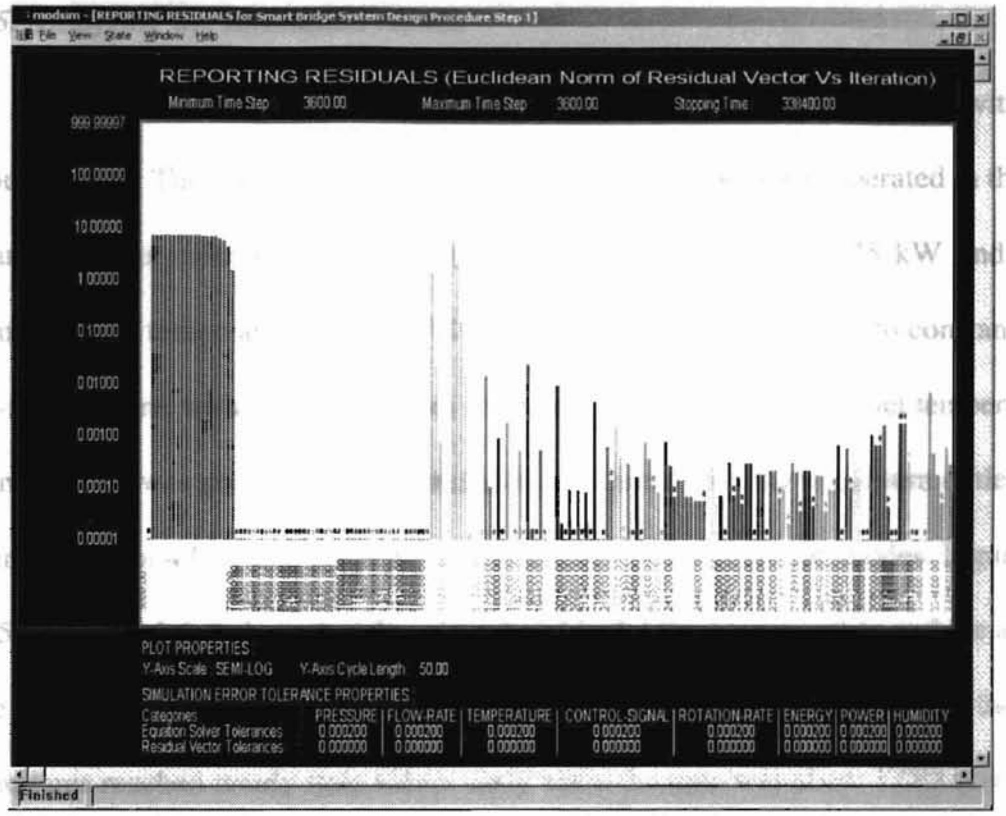


	TIME	t_DB	t_Sky	Humidity ratic	Wind Speed	Wind direction	Solar radiation	Snowfall	Rainf
40	140400	-9.4	-9.4	0.004	6.25	180	0	0	
41	144000	-9.4	-9.4	0.004	6.25	180	0	0	
42	147600	-9.4	-9.4	0.004	6.25	180	0	0	
43	151200	-9.4	-9.4	0.004	6.25	180	0	0	
44	154800	-9.4	-9.4	0.004	6.25	180	0	0	
45	158400	-9.4	-9.4	0.004	6.25	180	0	0	
46	162000	-9.4	-9.4	0.004	6.25	180	0	0	
47	165600	-9.4	-9.4	0.004	6.25	180	0	0	
48	169200	-9.4	-9.4	0.004	6.25	180	0	0	
49	169200	-9.4	-9.4	0.004	6.25	180	0	1.042	
50	172800	-9.4	-9.4	0.004	6.25	180	0	1.042	
51	176400	-9.4	-9.4	0.004	6.25	180	0	1.042	
52	180000	-9.4	-9.4	0.004	6.25	180	0	1.042	
53	183600	-9.4	-9.4	0.004	6.25	180	0	1.042	
54	187200	-9.4	-9.4	0.004	6.25	180	0	1.042	
55	190800	-9.4	-9.4	0.004	6.25	180	0	1.042	
56	194400	-9.4	-9.4	0.004	6.25	180	0	1.042	
57	198000	-9.4	-9.4	0.004	6.25	180	0	1.042	
58	201600	-9.4	-9.4	0.004	6.25	180	0	1.042	
59	205200	-9.4	-9.4	0.004	6.25	180	0	1.042	
60	208800	-9.4	-9.4	0.004	6.25	180	0	1.042	
61	212400	-9.4	-9.4	0.004	6.25	180	0	1.042	

Figure 5.2.1.3-1. *Smart Bridge - Design procedure step 1: Boundary file editor*

5.2.1.4. Running the simulation

After entering all the required data, the simulation was executed from the interface. Since residual plotting was requested, as shown in Figure 5.2.1.2-1, the interface plotted the Euclidean norm of residuals for every iteration while the simulation was running. Figure 5.2.1.4-1 shows a screen shot of the residual plot generated by MODSIM. Figure 5.2.1.4-2 shows a screen shot of the plot generated by the simulation result plotting form in the interface. The figure displays the plot of the average bridge surface-temperature against time. The simulation output file is listed in Appendix D.



modsim - [REPORTING RESIDUALS for Smart Bridge System Design Procedure Step 1]

REPORTING RESIDUALS (Euclidean Norm of Residual Vector Vs Iteration)

Minimum Time Step 3600.00 Maximum Time Step 3600.00 Stopping Time 338400.00

PLOT PROPERTIES

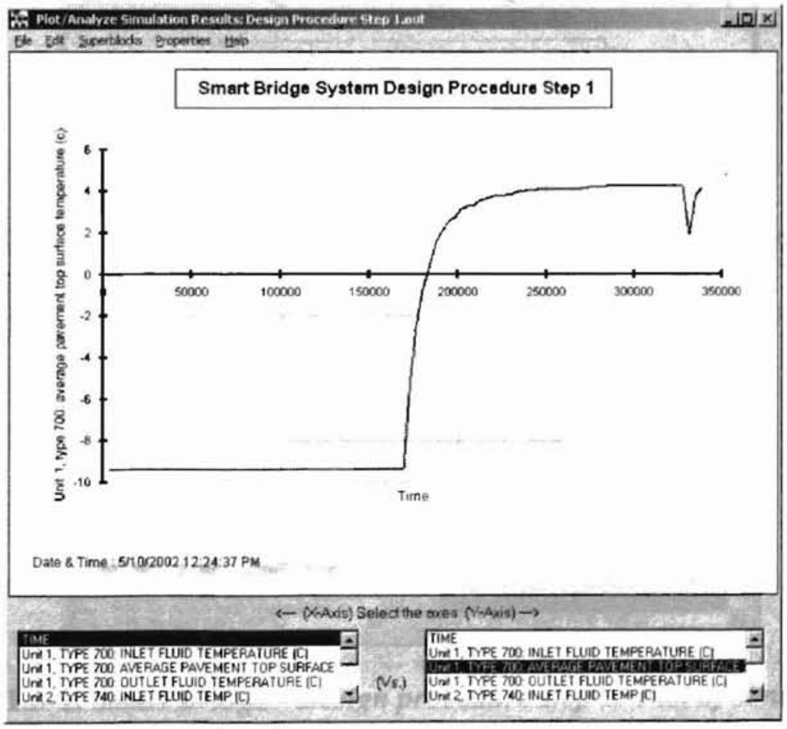
Y-Axis Scale : SEMI-LOG Y-Axis Cycle Length 50.00

SIMULATION ERROR TOLERANCE PROPERTIES

Categories	PRESSURE	FLOW-RATE	TEMPERATURE	CONTROL-SIGNAL	ROTATION-RATE	ENERGY	POWER	HUMIDITY
Equation Solver Tolerances	0.000200	0.000200	0.000200	0.000200	0.000200	0.000200	0.000200	0.000200
Residual Vector Tolerances	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000

Finished

Figure 5.2.1.4-1. *Smart Bridge - Design procedure step 1: Residual plot generated by MODSIM during simulation*



Plot/Analyze Simulation Results: Design Procedure Step 1.out

Smart Bridge System Design Procedure Step 1

Date & Time : 5/10/2002 12:24:37 PM

<— (X-Axis) Select the axes (Y-Axis) —>

TIME
Unit 1, TYPE 700: INLET FLUID TEMPERATURE (C)
Unit 1, TYPE 700: AVERAGE PAVEMENT TOP SURFACE
Unit 1, TYPE 700: OUTLET FLUID TEMPERATURE (C)
Unit 2, TYPE 740: INLET FLUID TEMP (C)

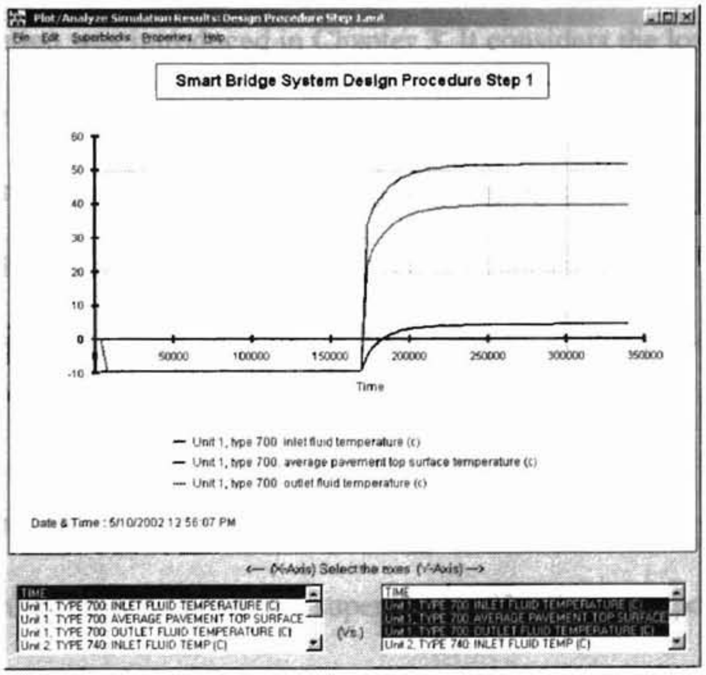
(Vs.)

TIME
Unit 1, TYPE 700: INLET FLUID TEMPERATURE (C)
Unit 1, TYPE 700: AVERAGE PAVEMENT TOP SURFACE
Unit 1, TYPE 700: OUTLET FLUID TEMPERATURE (C)
Unit 2, TYPE 740: INLET FLUID TEMP (C)

Figure 5.2.1.4-2. *Smart Bridge - Design procedure step 1: Plot of simulation results*

5.2.1.5. Analyzing the simulation results

Similar to the previous simulation, this simulation result was compared with their old counterparts. The results matched exactly. The heater model was operated in the constant-heat-flux mode for this simulation with the heat flux set at 1025 kW and maximum outlet temperature at $54.44\,°C$. When the mode was changed to constant-outlet-temperature, with the same values for heat flux and maximum outlet temperature, the same output was generated. The simulation was rerun with the maximum outlet temperature set at $47\,°C$ and the heat flux set at 1025 kW in both the modes. Figures 5.2.1.5-1 and 5.2.1.5-2 show the plot of inlet and outlet fluid temperature and the average bridge surface-temperature in the two modes, respectively. In the second mode, the fluid temperatures reached steady state values early when the time was about 175000 seconds. In the first mode, the temperatures reached steady state at about 275000 seconds.

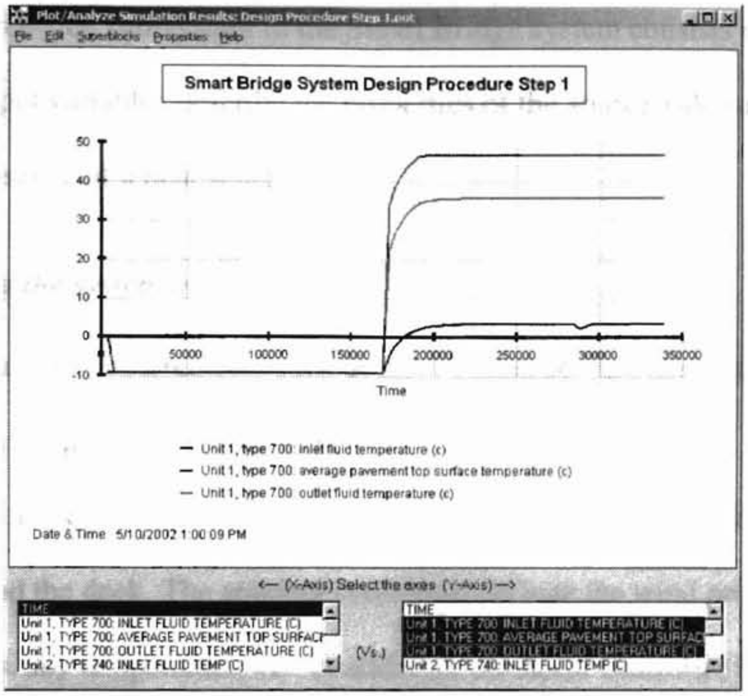

Plot/Analyze Simulation Results: Design Procedure Step 1.ent
File
Edit
Superblocks
Properties
Help
Smart Bridge System Design Procedure Step 1
60
50
40
30
20
10
0
-10
50000
100000
150000
200000
250000
300000
350000
Time
— Unit 1, type 700 inlet fluid temperature (c)
— Unit 1, type 700 average pavement top surface temperature (c)
— Unit 1, type 700 outlet fluid temperature (c)
<— (X-Axis) Select the axes (Y-Axis) —>
(Vs.)

Date & Time : 5/10/2002 12 56 07 PM

TIME
Unit 1, TYPE 700 INLET FLUID TEMPERATURE (C)
Unit 1, TYPE 700 AVERAGE PAVEMENT TOP SURFACE
Unit 1, TYPE 700 OUTLET FLUID TEMPERATURE (C)
Unit 2, TYPE 740 INLET FLUID TEMP (C)

TIME
Unit 1, TYPE 700 INLET FLUID TEMPERATURE (C)
Unit 1, TYPE 700 AVERAGE PAVEMENT TOP SURFACE
Unit 1, TYPE 700 OUTLET FLUID TEMPERATURE (C)
Unit 2, TYPE 740 INLET FLUID TEMP (C)

Figure 5.2.1.5-1. *Smart Bridge - Design procedure step 1: Plot of simulation results when heater is operating in mode 1 with the maximum outlet temperature set at 47 degrees centigrade*



Plot/Analyze Simulation Results: Design Procedure Step 1.out
File Edit Superblocks Properties Help
Smart Bridge System Design Procedure Step 1
50
40
30
20
10
0
-10
50000
100000
150000
200000
250000
300000
350000
Time
— Unit 1, type 700: inlet fluid temperature (c)
— Unit 1, type 700: average pavement top surface temperature (c)
— Unit 1, type 700: outlet fluid temperature (c)
<— (X-Axis) Select the axes (Y-Axis) —>
TIME
Unit 1, TYPE 700: INLET FLUID TEMPERATURE (C)
Unit 1, TYPE 700: AVERAGE PAVEMENT TOP SURFACE
Unit 1, TYPE 700: OUTLET FLUID TEMPERATURE (C)
Unit 2, TYPE 740: INLET FLUID TEMP (C)
(Vs.)
TIME
Unit 1, TYPE 700: INLET FLUID TEMPERATURE (C)
Unit 1, TYPE 700: AVERAGE PAVEMENT TOP SURFAC
Unit 1, TYPE 700: OUTLET FLUID TEMPERATURE (C)
Unit 2, TYPE 740: INLET FLUID TEMP (C)

Date & Time 5/10/2002 1:00:09 PM

Figure 5.2.1.5-2. *Smart Bridge - Design procedure step 1: Plot of simulation results when heater is operating in mode 2 with the maximum outlet temperature set at 47 degrees centigrade*

5.2.2. Design Procedure Step 2

This simulation was introduced in Chapter 3. It considers the load-side sub-system of the Smart Bridge system as shown in the Figure 3.3.2.1-1. The 2-mode heater model in Design Procedure Step 1 was replaced with a gang of water-to-water heat pump. A controller was also introduced that receives weather data and decides the number of heat pumps to be operated. The units involved and their Types were discussed in Section 3.3.2 of Chapter 3.

5.2.2.1. Model Description

This configuration consists of two superblocks. The linear proportional controller is the only unit in superblock 1. All the other units are grouped together in superblock 2. Descriptions of the units involved can be found in Appendix A.

The source-side sub-system in the Smart Bridge system consists of a heat pump model whose input variables describe the properties of the source-side sub-system, thus, justifying the absence of a heat source.

5.2.2.2. Creating the system model

The simulation requirements such as error tolerances and simulation times were kept the same as the previous design step.

The slab model is supplied with boundary values representing the ambient conditions around the deck. The ambient conditions include the wind properties, snowfall and rainfall rates, sky temperature, etc. In addition, the linear controller is supplied with a snow flag that, like the snowfall rate, represents a snow event. Boundary variables are given names that may be created using the appropriate menu option in the boundary file editor.

Similar to the previous procedure, the average bridge surface-temperature is the major reported variable in the system.

Figure 5.2.1.2-4 shows the workspace configuration of the system. The various units, boundary conditions and reported variables in the simulation are displayed in the figure.

The initial values selected for each unit are shown in the Figures 5.2.1.2-2 and 5.2.1.2-5. For all units, the initial values for input and output variables are set at zero.

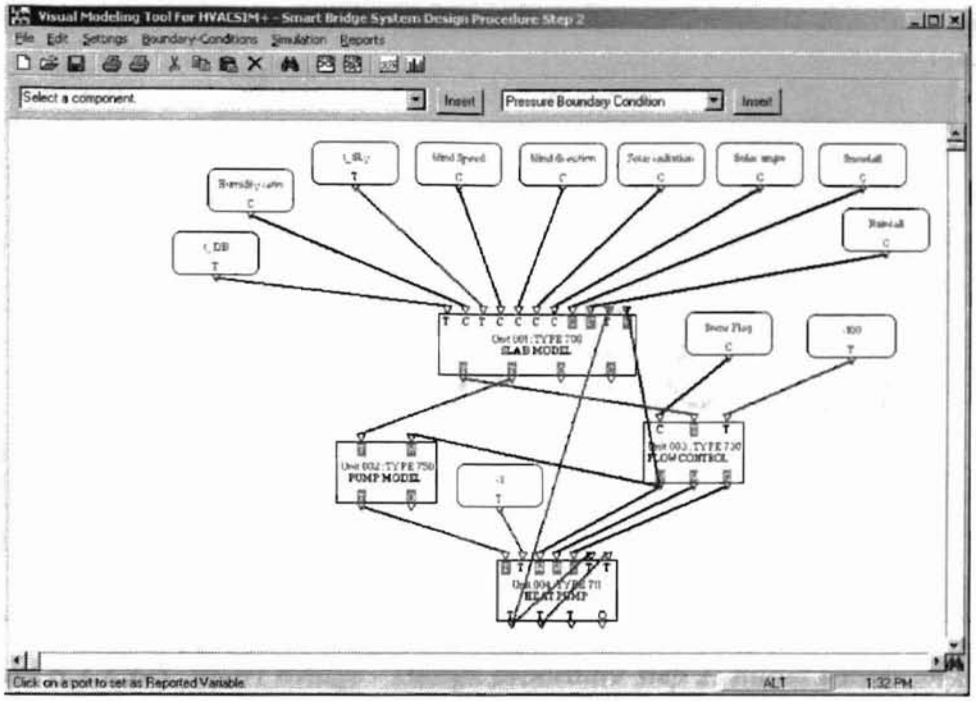
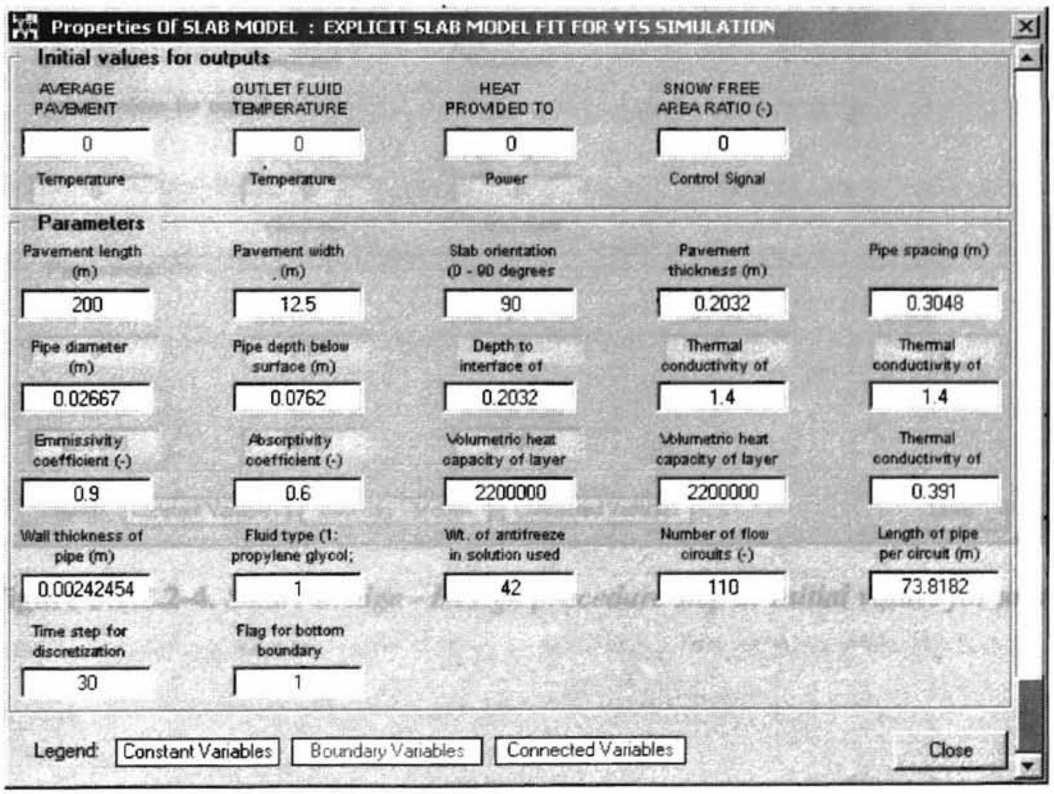


Figure 5.2.2.2-1. *Smart Bridge - Design procedure step 2: Screenshot showing the various units, boundary conditions and reported variables in the simulation*



Properties Of SLAB MODEL : EXPLICIT SLAB MODEL FIT FOR VTS SIMULATION

Initial values for outputs

AVERAGE PAVEMENT	OUTLET FLUID TEMPERATURE	HEAT PROVIDED TO	SNOW FREE AREA RATIO (-)
0	0	0	0
Temperature	Temperature	Power	Control Signal

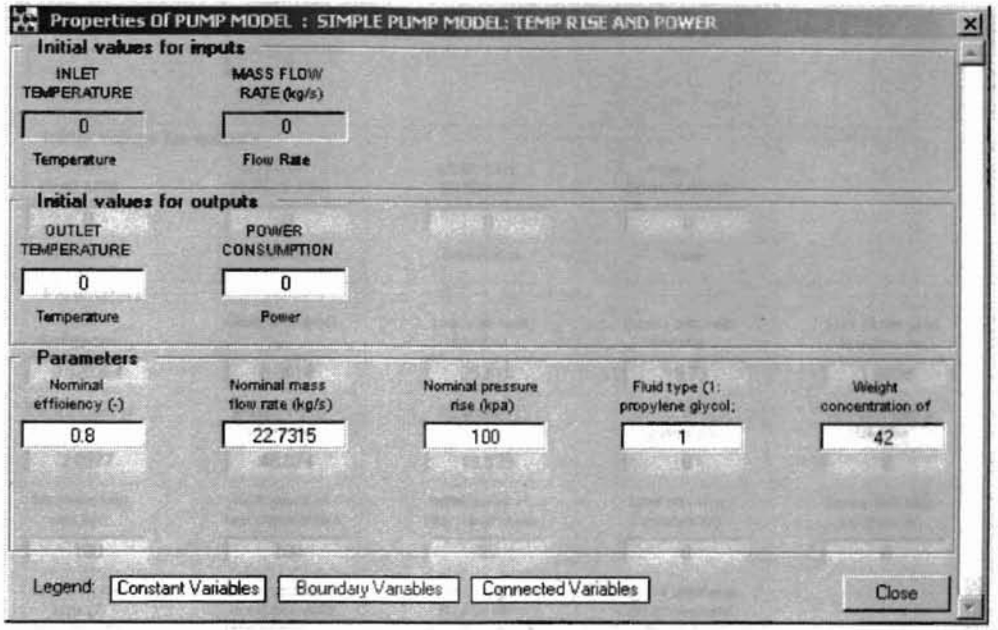
Parameters

Pavement length (m)	Pavement width (m)	Slab orientation (0 - 90 degrees	Pavement thickness (m)	Pipe spacing (m)
200	12.5	90	0.2032	0.3048
Pipe diameter (m)	Pipe depth below surface (m)	Depth to interface of	Thermal conductivity of	Thermal conductivity of
0.02667	0.0762	0.2032	1.4	1.4
Emmissivity coefficient (-)	Absorptivity coefficient (-)	Volumetric heat capacity of layer	Volumetric heat capacity of layer	Thermal conductivity of
0.9	0.6	2200000	2200000	0.391
Wall thickness of pipe (m)	Fluid type (1: propylene glycol;	Wt. of antifreeze in solution used	Number of flow circuits (-)	Length of pipe per circuit (m)
0.00242454	1	42	110	73.8182
Time step for discretization	Flag for bottom boundary			
30	1			

Legend: Constant Variables | Boundary Variables | Connected Variables

Close

Figure 5.2.2.2-2. *Smart Bridge - Design procedure step 2: Initial values for unit 1*


Properties Of PUMP MODEL : SIMPLE PUMP MODEL: TEMP RISE AND POWER

Initial values for inputs

INLET TEMPERATURE	MASS FLOW RATE (kg/s)
0	0
Temperature	Flow Rate

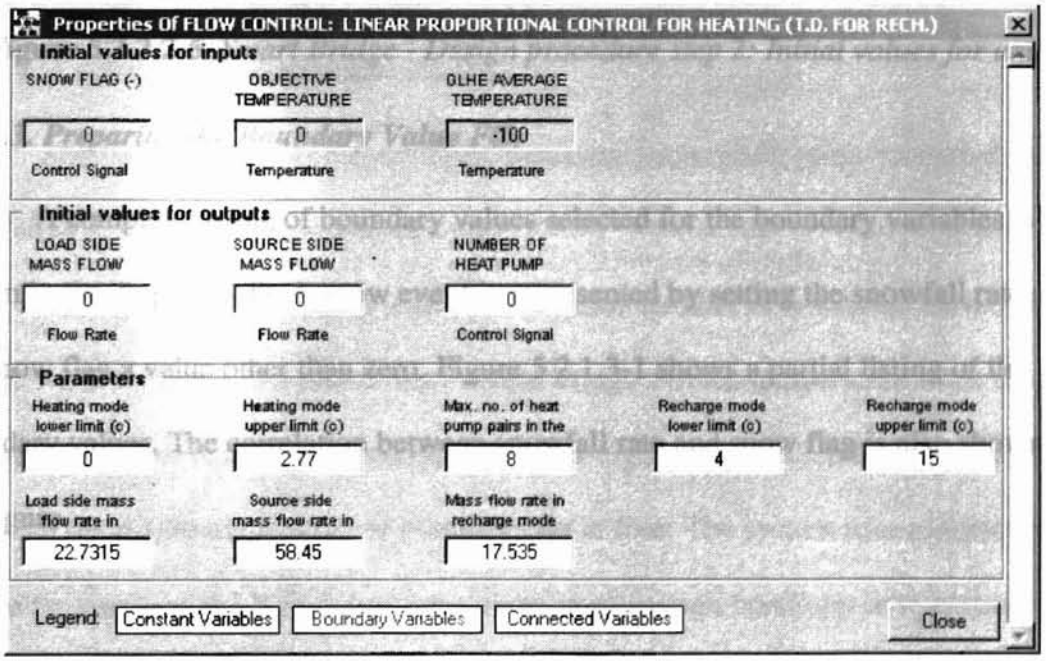
Initial values for outputs

OUTLET TEMPERATURE	POWER CONSUMPTION
0	0
Temperature	Power

Parameters

Nominal efficiency (-)	Nominal mass flow rate (kg/s)	Nominal pressure rise (kpa)	Fluid type (1: propylene glycol;	Weight concentration of
0.8	22.7315	100	1	42

Figure 5.2.2.2-3. *Smart Bridge - Design procedure step 2: Initial values for unit 2*


Properties Of FLOW CONTROL: LINEAR PROPORTIONAL CONTROL FOR HEATING (T.D. FOR RECH.)

Initial values for inputs

SNOW FLAG (-)	OBJECTIVE TEMPERATURE	GLHE AVERAGE TEMPERATURE
0	0	-100
Control Signal	Temperature	Temperature

Initial values for outputs

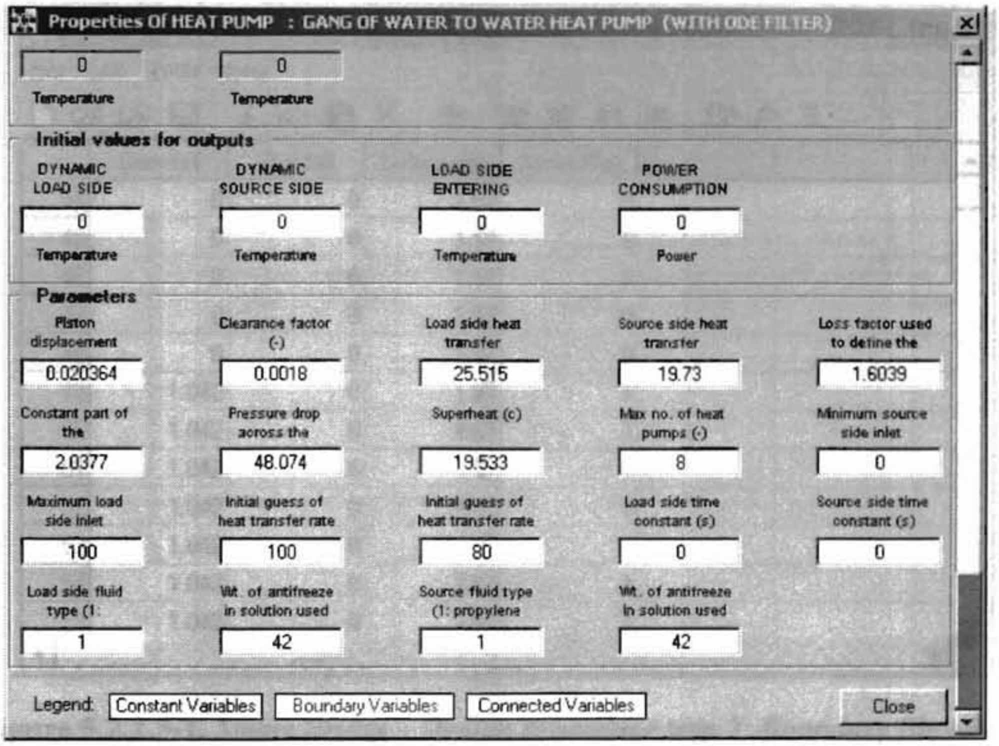
LOAD SIDE MASS FLOW	SOURCE SIDE MASS FLOW	NUMBER OF HEAT PUMP
0	0	0
Flow Rate	Flow Rate	Control Signal

Parameters

Heating mode lower limit (c)	Heating mode upper limit (c)	Max. no. of heat pump pairs in the	Recharge mode lower limit (c)	Recharge mode upper limit (c)
0	2.77	8	4	15

Load side mass flow rate in	Source side mass flow rate in	Mass flow rate in recharge mode
22.7315	58.45	17.535

Figure 5.2.2.2-4. *Smart Bridge - Design procedure step 2: Initial values for unit 3*



Properties Of HEAT PUMP : GANG OF WATER TO WATER HEAT PUMP (WITH ODE FILTER)

0	0
Temperature	Temperature

Initial values for outputs

DYNAMIC LOAD SIDE	DYNAMIC SOURCE SIDE	LOAD SIDE ENTERING	POWER CONSUMPTION
0	0	0	0
Temperature	Temperature	Temperature	Power

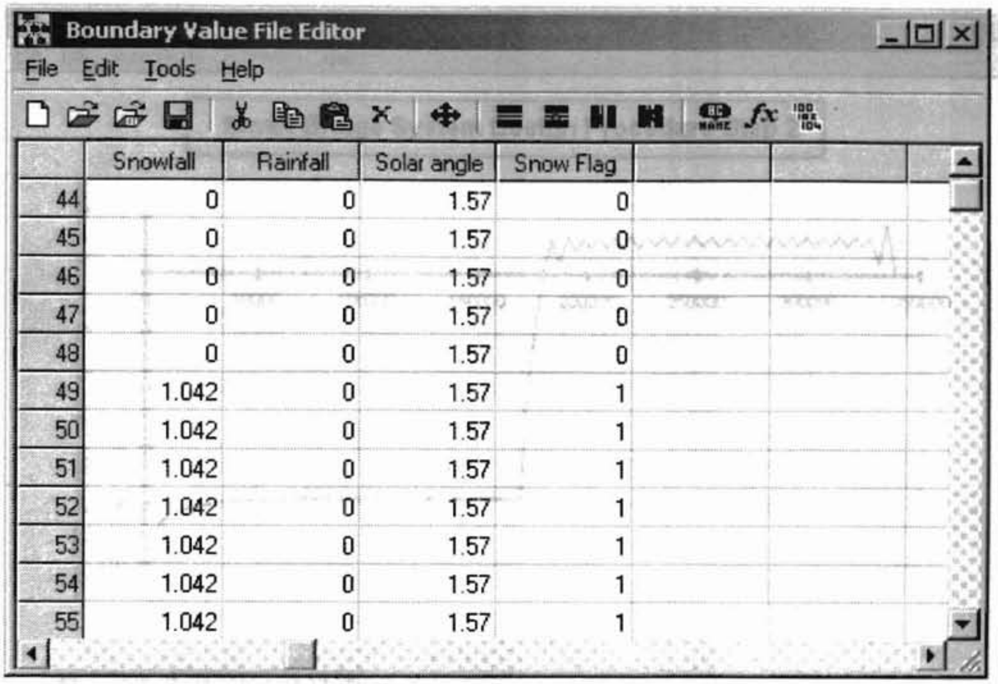
Parameters

Piston displacement	Clearance factor (-)	Load side heat transfer	Source side heat transfer	Loss factor used to define the
0.020364	0.0018	25.515	19.73	1.6039
Constant part of the	Pressure drop across the	Superheat (c)	Max no. of heat pumps (-)	Minimum source side inlet
2.0377	48.074	19.533	8	0
Maximum load side inlet	Initial guess of heat transfer rate	Initial guess of heat transfer rate	Load side time constant (s)	Source side time constant (s)
100	100	80	0	0
Load side fluid type (1:	Wt. of antifreeze in solution used	Source fluid type (1: propylene	Wt. of antifreeze in solution used	
1	42	1	42	

Figure 5.2.2.2-5. *Smart Bridge - Design procedure step 2: Initial values for unit 4*

5.2.2.3. Preparing the Boundary Value File

A complete listing of boundary values selected for the boundary variables is shown in the Appendix D. A snow event is represented by setting the snowfall rate and the snow flag a value other than zero. Figure 5.2.1.3-1 shows a partial listing of the boundary values. The correlation between snowfall rate and snow flag is also shown in the figure.



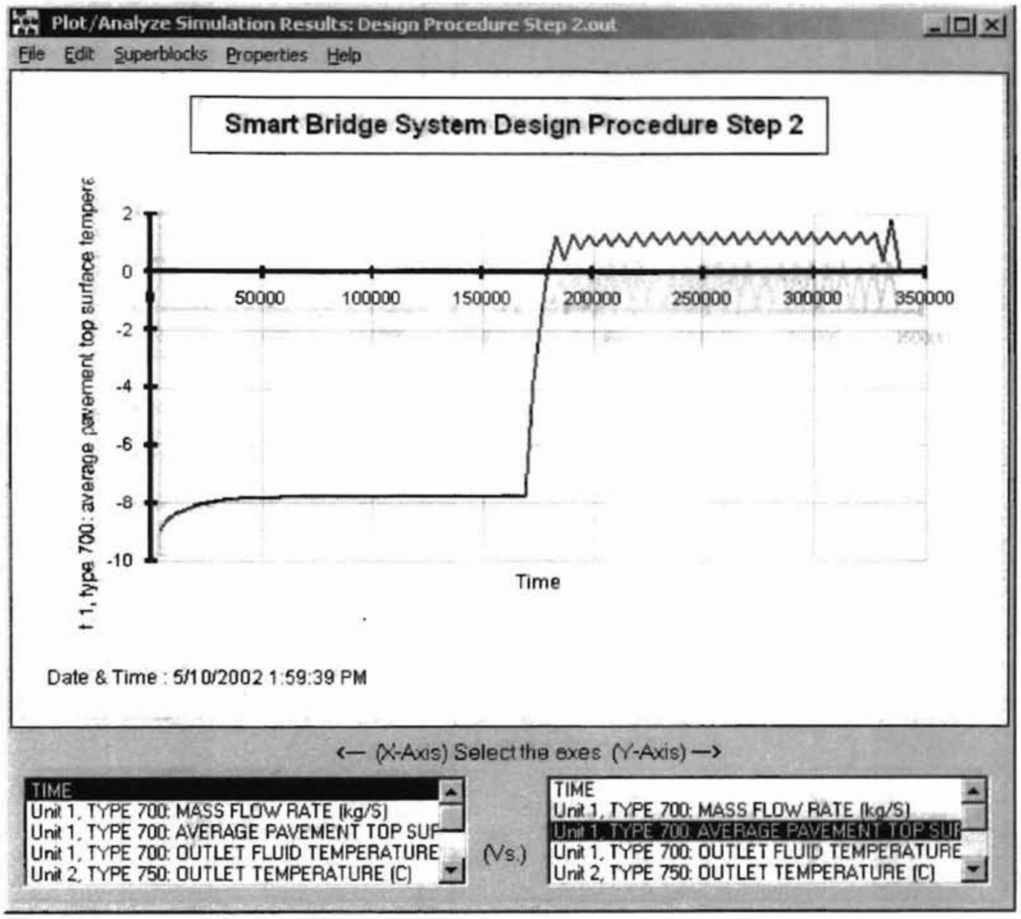
	Snowfall	Rainfall	Solar angle	Snow Flag		
44	0	0	1.57	0		
45	0	0	1.57	0		
46	0	0	1.57	0		
47	0	0	1.57	0		
48	0	0	1.57	0		
49	1.042	0	1.57	1		
50	1.042	0	1.57	1		
51	1.042	0	1.57	1		
52	1.042	0	1.57	1		
53	1.042	0	1.57	1		
54	1.042	0	1.57	1		
55	1.042	0	1.57	1		

Figure 5.2.2.3-1. *Smart Bridge - Design procedure step 2: Boundary file editor*

5.2.2.4. Analyzing the simulation results

Similar to the previous simulation, this simulation results were compared their old counterparts. The results matched exactly.

The simulation was rerun with the maximum number of boreholes halved to four. Figure 5.2.2.4-1 shows the plot of the average bridge surface-temperature when the maximum number of available boreholes is set at eight. Figure 5.2.2.4-2 shows a similar plot with the maximum number of boreholes set at four. The system tries to maintain the bridge temperature at 1 °C but due to the absence of enough boreholes in operation to perform the heat transfer between the fluid and the ground, the output shows fluctuations. Appendix D provides the complete listing of the output file generated during the simulation.



Plot/Analyze Simulation Results: Design Procedure Step 2.out
File Edit Superblocks Properties Help
Smart Bridge System Design Procedure Step 2
t 1, type 700: average pavement top surface tempera
2
0
-2
-4
-6
-8
-10
50000
100000
150000
200000
250000
300000
350000
Time

Date & Time : 5/10/2002 1:59:39 PM

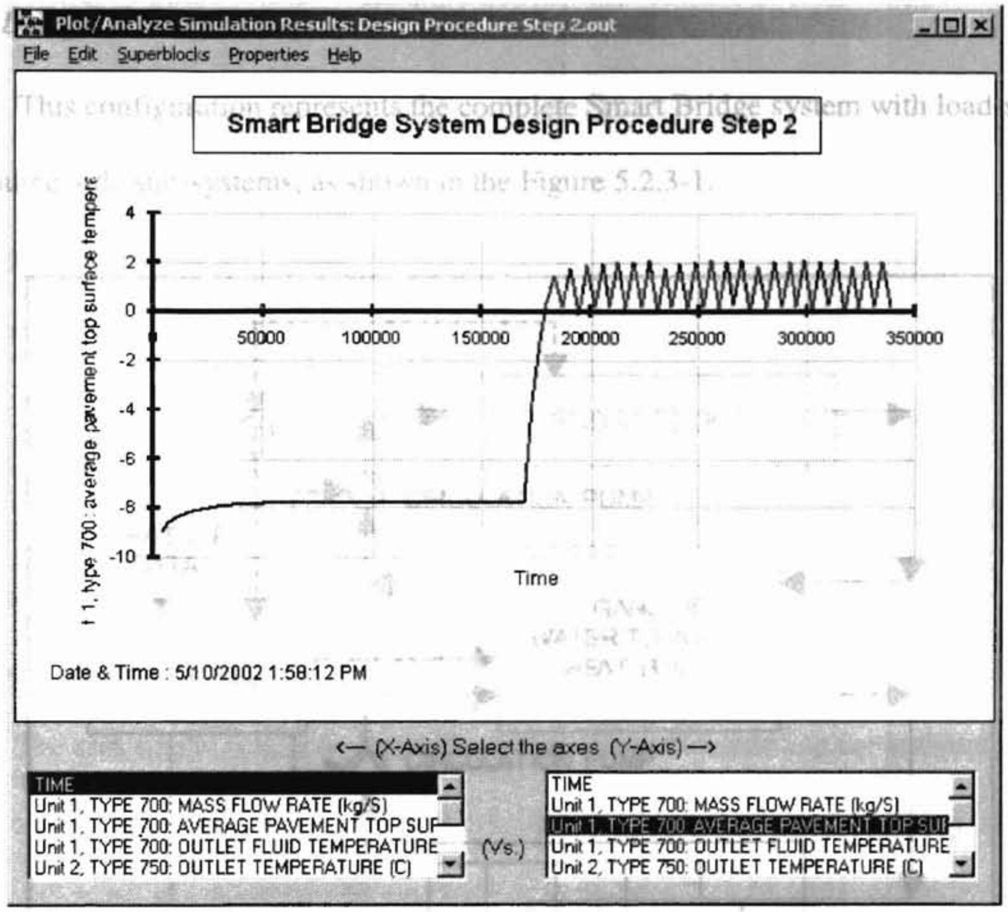
<— (X-Axis) Select the axes (Y-Axis) —>

TIME
Unit 1, TYPE 700: MASS FLOW RATE (kg/S)
Unit 1, TYPE 700: AVERAGE PAVEMENT TOP SUF
Unit 1, TYPE 700: OUTLET FLUID TEMPERATURE
Unit 2, TYPE 750: OUTLET TEMPERATURE (C)

(Vs.)

TIME
Unit 1, TYPE 700: MASS FLOW RATE (kg/S)
Unit 1, TYPE 700: AVERAGE PAVEMENT TOP SUF
Unit 1, TYPE 700: OUTLET FLUID TEMPERATURE
Unit 2, TYPE 750: OUTLET TEMPERATURE (C)

Figure 5.2.2.4-1. *Smart Bridge - Design procedure step 2: Plot of simulation results with maximum number of boreholes set at eight*


Plot/Analyze Simulation Results: Design Procedure Step 2.out
File Edit Superblocks Properties Help
Smart Bridge System Design Procedure Step 2
t 1, type 700: average pavement top surface tempera
4
2
0
-2
-4
-6
-8
-10
50000
100000
150000
200000
250000
300000
350000
Time
<— (X-Axis) Select the axes (Y-Axis) —>
(Vs.)

Date & Time : 5/10/2002 1:58:12 PM

X-Axis	Y-Axis
TIME	TIME
Unit 1, TYPE 700: MASS FLOW RATE (kg/S)	Unit 1, TYPE 700: MASS FLOW RATE (kg/S)
Unit 1, TYPE 700: AVERAGE PAVEMENT TOP SUF	Unit 1, TYPE 700: AVERAGE PAVEMENT TOP SUI
Unit 1, TYPE 700: OUTLET FLUID TEMPERATURE	Unit 1, TYPE 700: OUTLET FLUID TEMPERATURE
Unit 2, TYPE 750: OUTLET TEMPERATURE (C)	Unit 2, TYPE 750: OUTLET TEMPERATURE (C)

Figure 5.2.2.4-2. *Smart Bridge - Design procedure step 2: Plot of simulation results with maximum number of boreholes set at four*

5.2.3. Design Procedure Step 3

This configuration represents the complete Smart Bridge system with load-side and source-side sub-systems, as shown in the Figure 5.2.3-1.

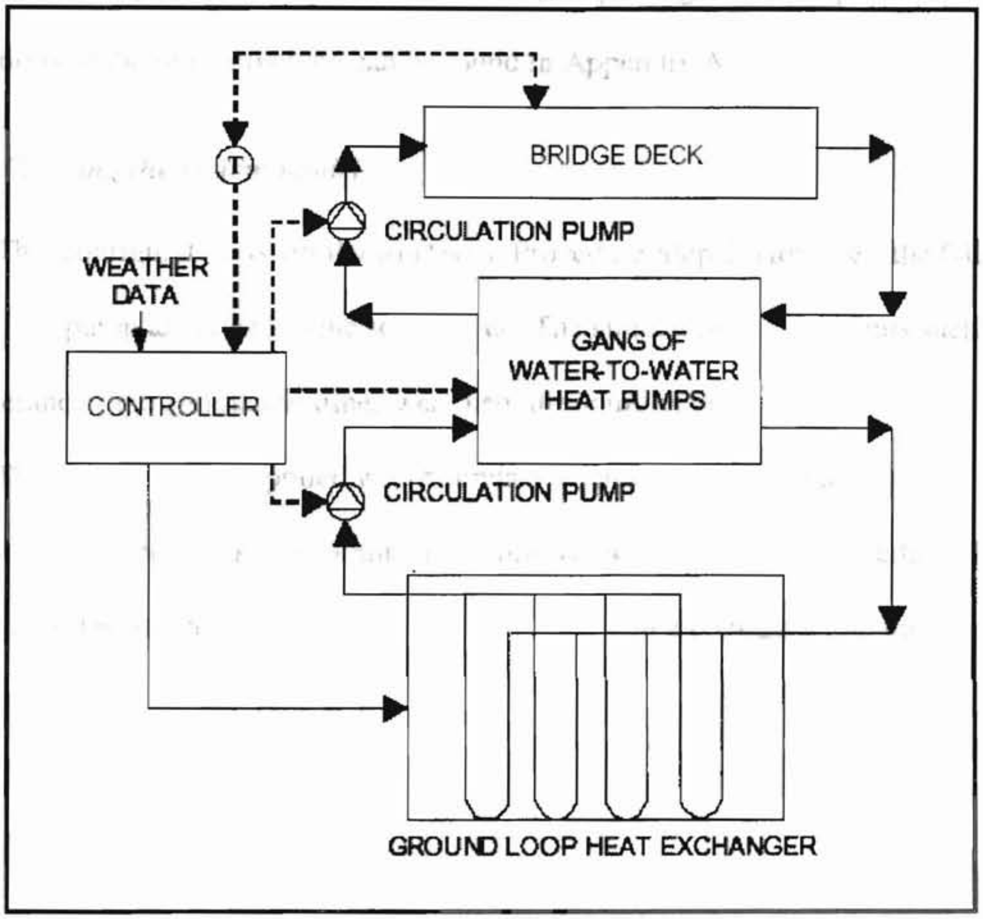


Figure 5.2.3-1. *Smart Bridge – Design procedure step 3: System schematic*

5.2.3.1. Model Description

This configuration consists of six units:

- TYPE 700: Slab model representing a bridge deck
- TYPE 750: Simple pump model
- TYPE 730: Linear flow controller
- TYPE 750: Simple pump model

- TYPE 711: Gang of water-to-water heat pump model
- TYPE 721: Ground loop heat exchanger model

The units are organized into two superblocks. The linear proportional controller is the only unit in superblock 1. All other units are grouped together in superblock 2. Descriptions of the units involved can be found in Appendix A.

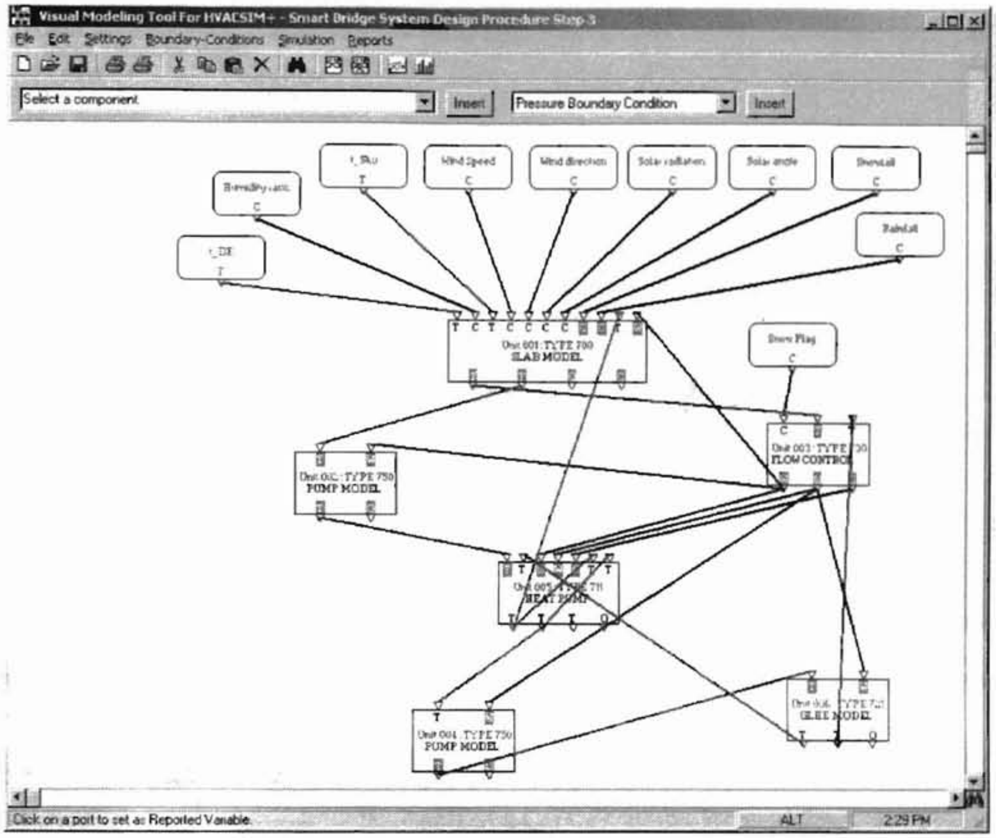
5.2.3.2. Creating the system model

This configuration is similar to Design Procedure Step 2. However, the GLHE model and a pump are added to the source-side. The simulation requirements such as error tolerances and simulation times were kept the same as the previous design steps.

The slab model is supplied with boundary values representing the ambient conditions around the deck. The ambient conditions include the wind properties, snowfall and rainfall rates, sky temperature, etc. Similar to the step 2 configuration, the linear controller is supplied with a snow flag, which like the snowfall rate represents a snow event. Boundary variables are given names that may be created using the appropriate menu option in the boundary file editor.

Figure 5.2.1.2-4 shows the workspace configuration of the system. The various units, boundary conditions and reported variables in the simulation are displayed.

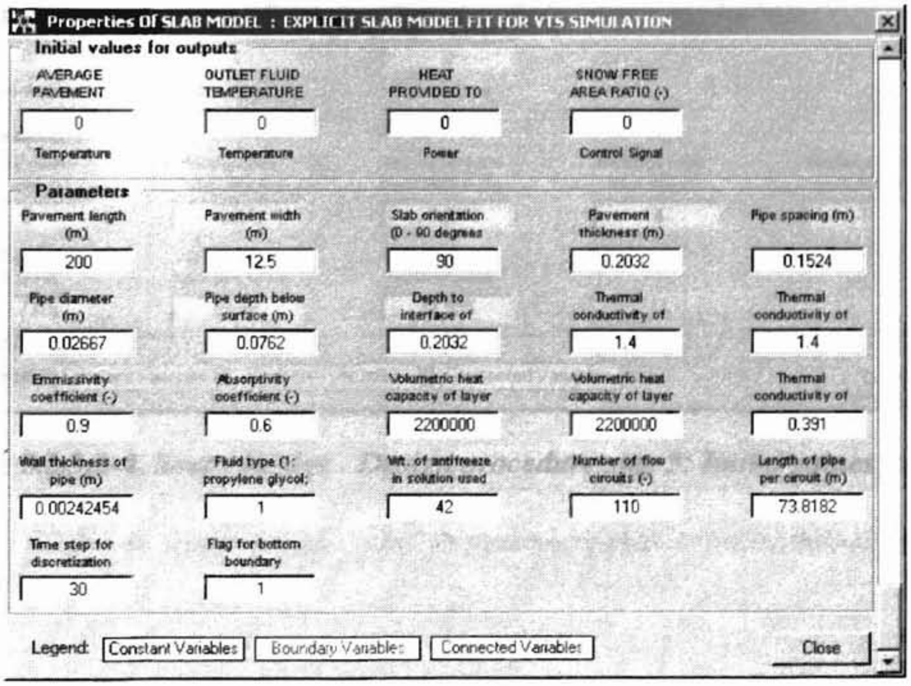
The initial values selected for each unit are shown in the Figures 5.2.3.2-2 and 5.2.3.2-5. For all units, the initial values for input and output variables are set at zero.



Visual Modeling Tool For HVACSIM+ - Smart Bridge System Design Procedure Step 3

Click on a port to set as Reported Variable.

Figure 5.2.3.2-1. *Smart Bridge - Design procedure step 3: Screenshot showing the various units, boundary conditions and reported variables in the simulation*



Properties Of SLAB MODEL : EXPLICIT SLAB MODEL FIT FOR VTS SIMULATION

Initial values for outputs

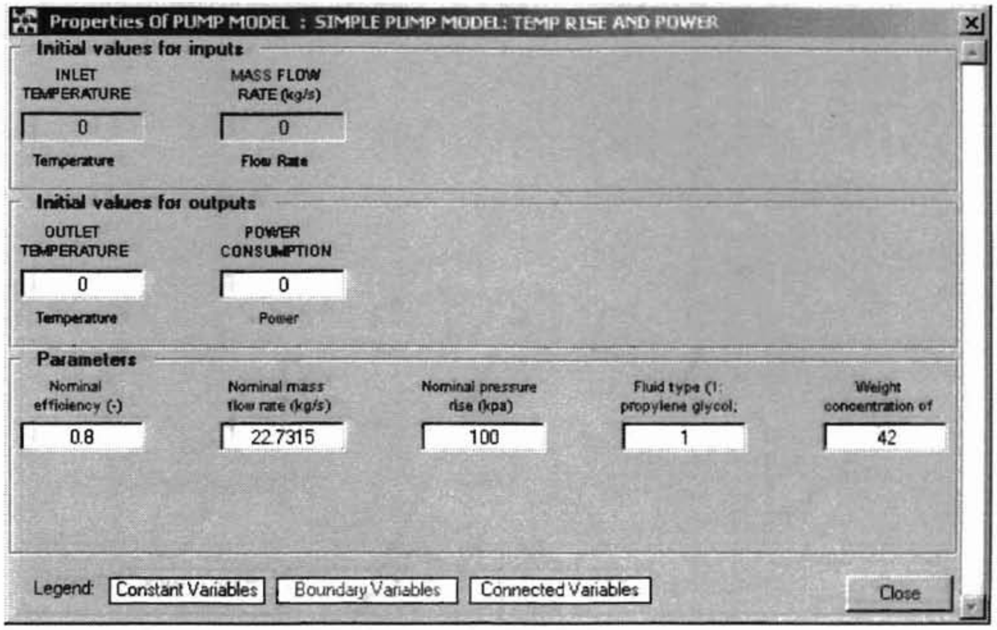
AVERAGE PAVEMENT	OUTLET FLUID TEMPERATURE	HEAT PROVIDED TO	SNOW FREE AREA RATIO (-)
0	0	0	0
Temperature	Temperature	Power	Control Signal

Parameters

Pavement length (m)	Pavement width (m)	Slab orientation (0 - 90 degrees	Pavement thickness (m)	Pipe spacing (m)
200	12.5	90	0.2032	0.1524
Pipe diameter (m)	Pipe depth below surface (m)	Depth to interface of	Thermal conductivity of	Thermal conductivity of
0.02667	0.0762	0.2032	1.4	1.4
Emmissivity coefficient (-)	Absorptivity coefficient (-)	Volumetric heat capacity of layer	Volumetric heat capacity of layer	Thermal conductivity of
0.9	0.6	2200000	2200000	0.391
Wall thickness of pipe (m)	Fluid type (1: propylene glycol;	Wt. of antifreeze in solution used	Number of flow circuits (-)	Length of pipe per circuit (m)
0.00242454	1	42	110	73.8182
Time step for discretization	Flag for bottom boundary			
30	1			

Legend: Constant Variables | Boundary Variables | Connected Variables

Figure 5.2.3.2-2. *Smart Bridge - Design procedure step 3: Initial values for unit 1*



Properties Of PUMP MODEL : SIMPLE PUMP MODEL: TEMP RISE AND POWER

Initial values for inputs

INLET TEMPERATURE	MASS FLOW RATE (kg/s)
0	0
Temperature	Flow Rate

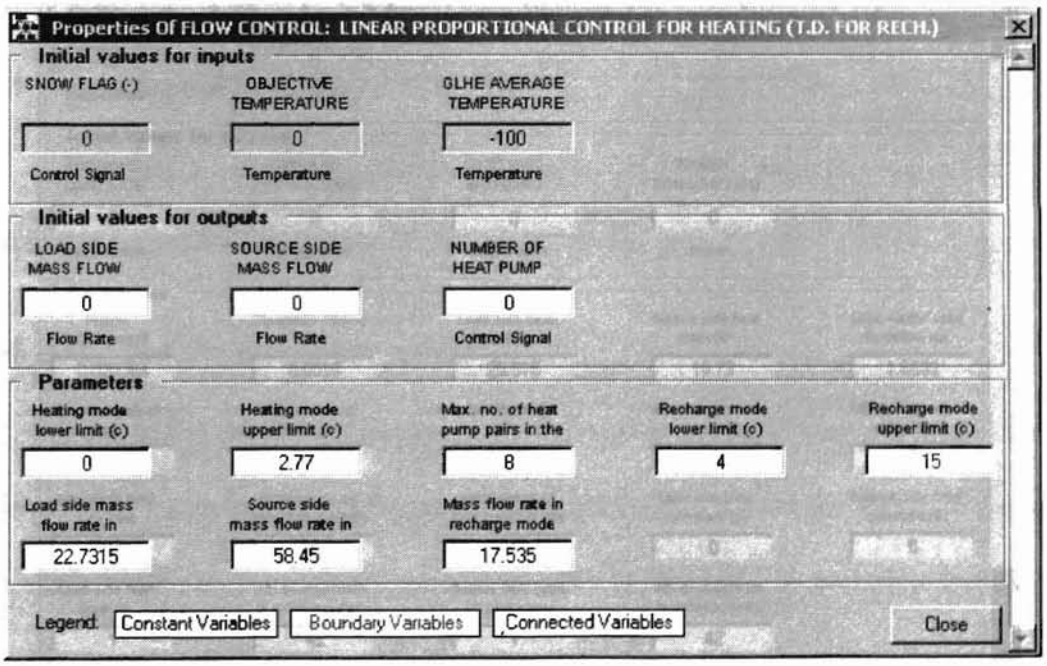
Initial values for outputs

OUTLET TEMPERATURE	POWER CONSUMPTION
0	0
Temperature	Power

Parameters

Nominal efficiency (-)	Nominal mass flow rate (kg/s)	Nominal pressure rise (kpa)	Fluid type (1: propylene glycol;	Weight concentration of
0.8	22.7315	100	1	42

Figure 5.2.3.2-3. *Smart Bridge - Design procedure step 3: Initial values for unit 2*



Properties Of FLOW CONTROL: LINEAR PROPORTIONAL CONTROL FOR HEATING (T.D. FOR RECH.)

Initial values for inputs

SNOW FLAG (-)	OBJECTIVE TEMPERATURE	GLHE AVERAGE TEMPERATURE
0	0	-100
Control Signal	Temperature	Temperature

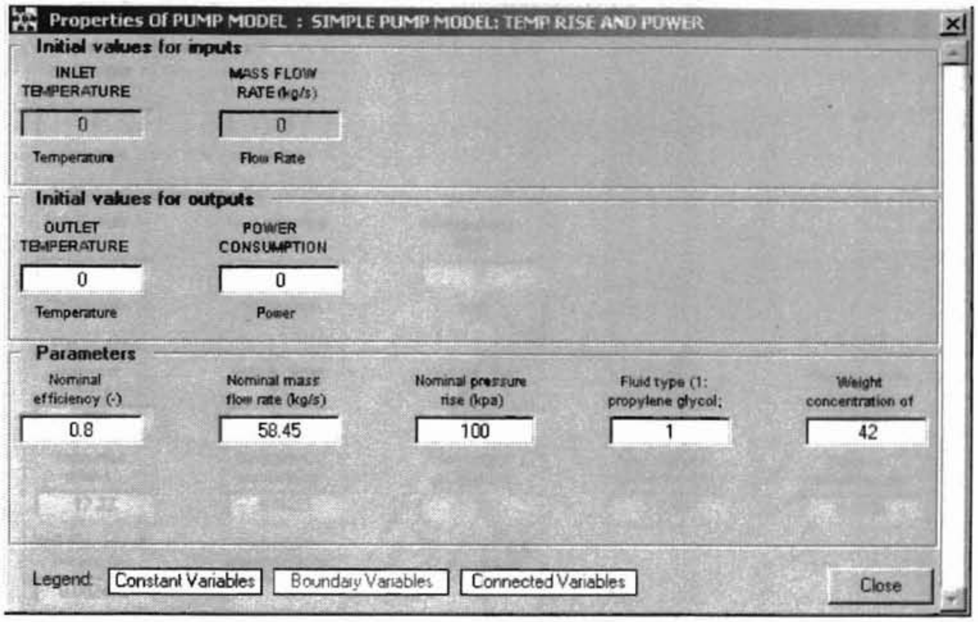
Initial values for outputs

LOAD SIDE MASS FLOW	SOURCE SIDE MASS FLOW	NUMBER OF HEAT PUMP
0	0	0
Flow Rate	Flow Rate	Control Signal

Parameters

Heating mode lower limit (c)	Heating mode upper limit (c)	Max. no. of heat pump pairs in the	Recharge mode lower limit (c)	Recharge mode upper limit (c)
0	2.77	8	4	15
Load side mass flow rate in	Source side mass flow rate in	Mass flow rate in recharge mode		
22.7315	58.45	17.535		

Figure 5.2.3.2-4. *Smart Bridge - Design procedure step 3: Initial values for unit 3*



Properties Of PUMP MODEL : SIMPLE PUMP MODEL: TEMP RISE AND POWER

Initial values for inputs

INLET TEMPERATURE	MASS FLOW RATE (kg/s)
0	0
Temperature	Flow Rate

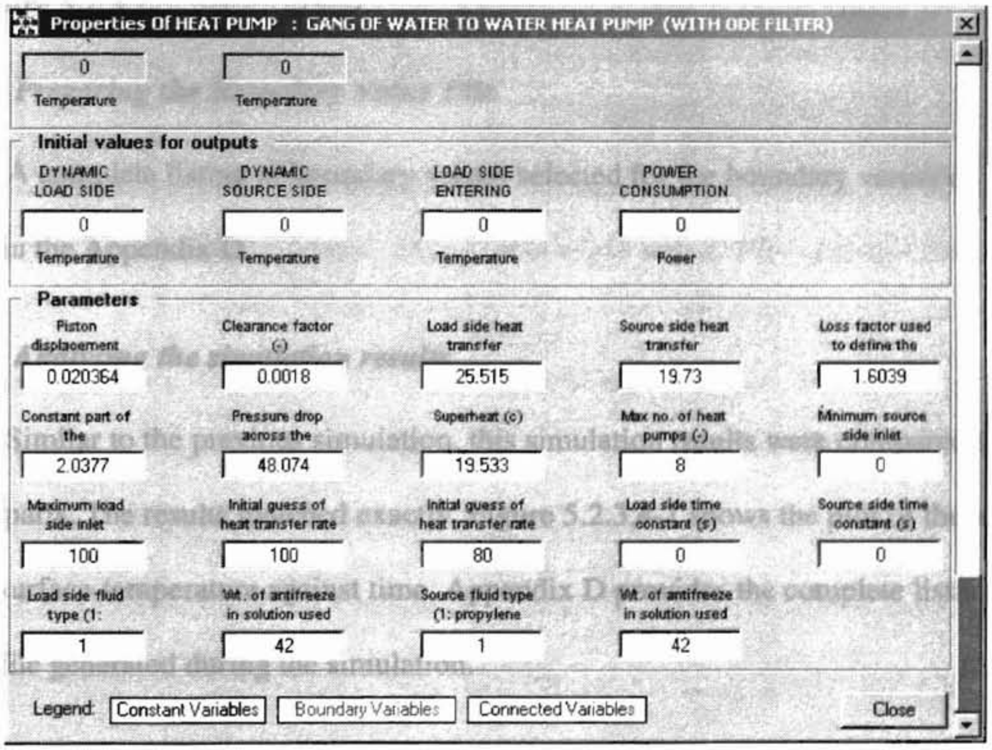
Initial values for outputs

OUTLET TEMPERATURE	POWER CONSUMPTION
0	0
Temperature	Power

Parameters

Nominal efficiency (-)	Nominal mass flow rate (kg/s)	Nominal pressure rise (kpa)	Fluid type (1: propylene glycol;	Weight concentration of
0.8	58.45	100	1	42

Figure 5.2.3.2-5. *Smart Bridge - Design procedure step 3: Initial values for unit 4*



Properties Of HEAT PUMP : GANG OF WATER TO WATER HEAT PUMP (WITH ODE FILTER)

0	0
Temperature	Temperature

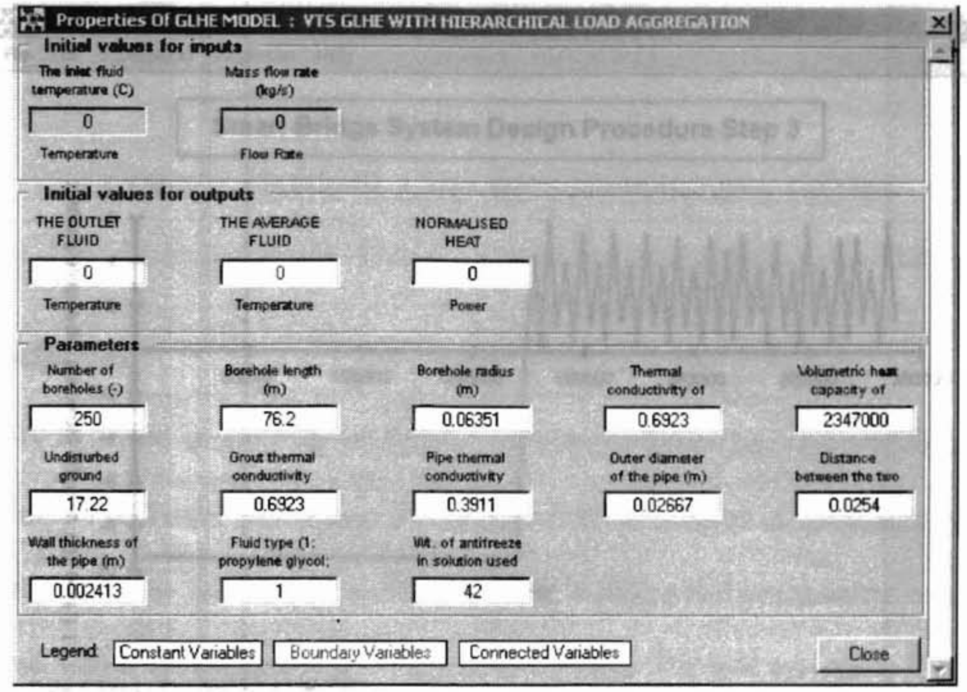
Initial values for outputs

DYNAMIC LOAD SIDE	DYNAMIC SOURCE SIDE	LOAD SIDE ENTERING	POWER CONSUMPTION
0	0	0	0
Temperature	Temperature	Temperature	Power

Parameters

Piston displacement	Clearance factor (-)	Load side heat transfer	Source side heat transfer	Loss factor used to define the
0.020364	0.0018	25.515	19.73	1.6039
Constant part of the	Pressure drop across the	Superheat (c)	Max no. of heat pumps (-)	Minimum source side inlet
2.0377	48.074	19.533	8	0
Maximum load side inlet	Initial guess of heat transfer rate	Initial guess of heat transfer rate	Load side time constant (s)	Source side time constant (s)
100	100	80	0	0
Load side fluid type (1:	Wt. of antifreeze in solution used	Source fluid type (1: propylene	Wt. of antifreeze in solution used	
1	42	1	42	

Figure 5.2.3.2-6. *Smart Bridge - Design procedure step 3: Initial values for unit 5*



Properties Of GLHE MODEL : VTS GLHE WITH HIERARCHICAL LOAD AGGREGATION

Initial values for inputs

The inlet fluid temperature (C): 0 (Temperature)

Mass flow rate (kg/s): 0 (Flow Rate)

Initial values for outputs

THE OUTLET FLUID: 0 (Temperature)

THE AVERAGE FLUID: 0 (Temperature)

NORMALISED HEAT: 0 (Power)

Parameters

Number of boreholes (-): 250

Borehole length (m): 76.2

Borehole radius (m): 0.06351

Thermal conductivity of: 0.6923

Volumetric heat capacity of: 2347000

Undisturbed ground: 17.22

Grout thermal conductivity: 0.6923

Pipe thermal conductivity: 0.3911

Outer diameter of the pipe (m): 0.02667

Distance between the two: 0.0254

Wall thickness of the pipe (m): 0.002413

Fluid type (1: propylene glycol;: 1

Wt. of antifreeze in solution used: 42

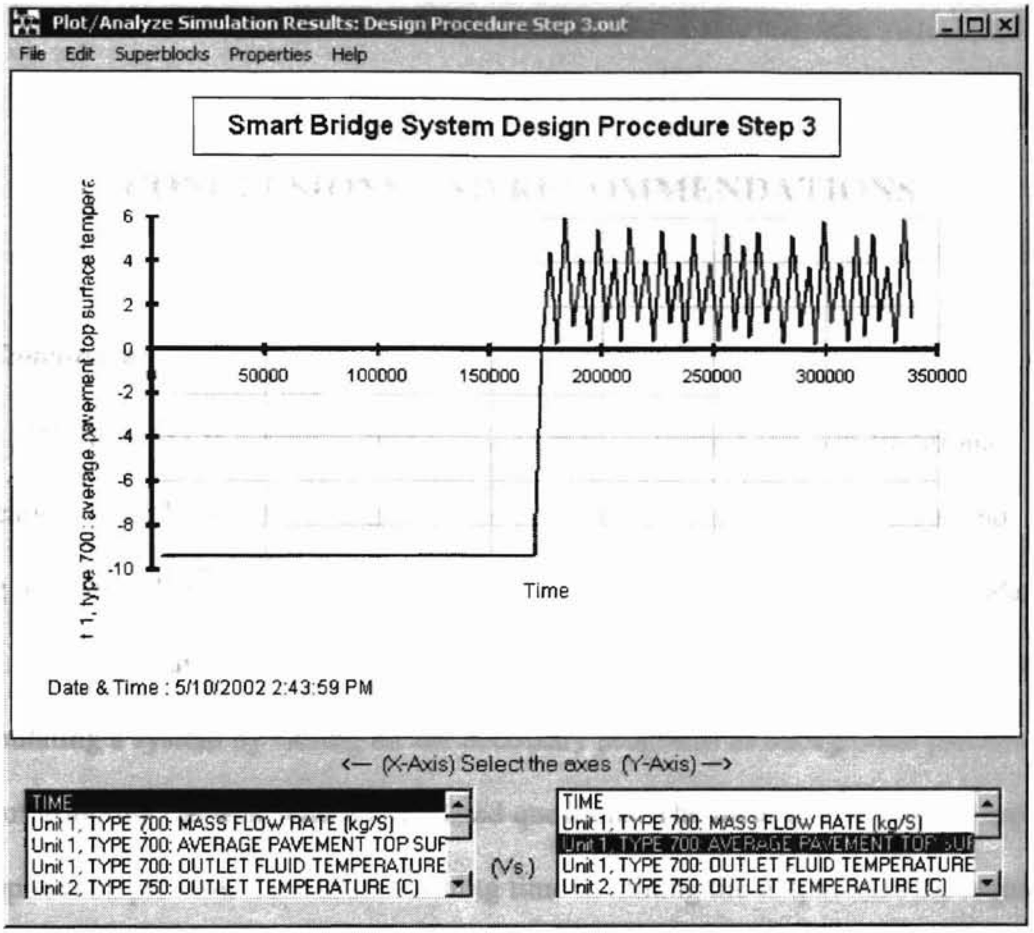
Figure 5.2.3.2-7. *Smart Bridge - Design procedure step 3: Initial values for unit 6*

5.2.3.3. Preparing the Boundary Value File

A complete listing of boundary values selected for the boundary variables is shown in the Appendix D.

5.2.3.4. Analyzing the simulation results

Similar to the previous simulation, this simulation results were compared their old counterparts. The results matched exactly. Figure 5.2.3.4-1 shows the plot of the average bridge surface-temperature against time. Appendix D provides the complete listing of the output file generated during the simulation.


Plot/Analyze Simulation Results: Design Procedure Step 3.out
File
Edit
Superblocks
Properties
Help
Smart Bridge System Design Procedure Step 3
t 1, type 700: average pavement top surface tempera
6
4
2
0
-2
-4
-6
-8
-10
50000
100000
150000
200000
250000
300000
350000
Time
<— (X-Axis) Select the axes (Y-Axis) —>
(Vs.)

Date & Time : 5/10/2002 2:43:59 PM

TIME
Unit 1, TYPE 700: MASS FLOW RATE (kg/S)
Unit 1, TYPE 700: AVERAGE PAVEMENT TOP SUF
Unit 1, TYPE 700: OUTLET FLUID TEMPERATURE
Unit 2, TYPE 750: OUTLET TEMPERATURE (C)

TIME
Unit 1, TYPE 700: MASS FLOW RATE (kg/S)
Unit 1, TYPE 700: AVERAGE PAVEMENT TOP SUF
Unit 1, TYPE 700: OUTLET FLUID TEMPERATURE
Unit 2, TYPE 750: OUTLET TEMPERATURE (C)

Figure 5.2.3.4-1. *Smart Bridge - Design procedure step 3: Plot of simulation results*

CHAPTER 6

CONCLUSIONS AND RECOMMENDATIONS

6.1. Conclusions

HVACSIM+ is currently being used in both traditional and non-traditional applications. The Visual Modeling Tool reduces both simulation development and debugging time. The boundary file editor helps in building and maintaining boundary value files, which are cumbersome to create and handle. The tool automates the process of simulating a system by calling all the necessary programs as background processes without troubling the user with complicated questions. The inclusion of plotting tools in the application prevents users from wasting time in mining the output files for required information. The users can now quickly and clearly understand the structure and characteristics of the system being simulated. In addition, the implementation of new convergence criteria for solving equation sets provides the users with flexibility in specifying the desired simulation performance.

Chapter 3 discusses the problems in HVACSIM+, reviewed in Chapter 2. It also discusses the performance evaluation of the new convergence metrics. In this chapter, it is shown through experimental validation that the parameters in the convergence criteria can be varied to target the simulation performance. The following conclusions were reached:

- The new convergence criteria do not guarantee convergence instead; they allow the user to specify the constraints that manipulate the rigidity of the criteria.
- Since the convergence criteria work in tandem with the methods for solving sets of nonlinear algebraic equations, the criteria also do not guarantee that the solver will converge.

Chapter 5 discusses the implementation of example simulations using the visual tool, described in Chapter 4. The examples demonstrate that the visual tool generates valid input for a wide range of system configurations. In addition, it was shown that:

- Modeling complex systems is quick and easy with the tool.
- Graphically represented information provides better understanding of the system.
- Better analysis can be performed on the simulation output with plots generated by the tool.

Finally, Chapter 5 demonstrated the suitability of the visual tool for non-traditional applications. A heated bridge deck system was successfully modeled in the visual environment.

6.2. Recommendations

The work presented in this thesis is a step towards finding solutions for executing simulations using HVACSIM+. However, its scope is limited. The author suggests research in the following areas to further enhance the work.

The features in the visual tool may prove insufficient as the complexity of the simulation increases and new simulation analysis techniques are proposed in the field of

HVAC system simulation. Although the visual tool contains many features, regular maintenance of the software is required to facilitate implementation of new algorithms. For example, a feature that can show the performance of selected variables in the simulation might prove useful. A plotting routine that can graphically represent the performance of a variable in the simulation might also prove to be useful. This feature might help in understanding the influence of a variable in the simulation.

The numerical methods used in HVACSIM+ should also be updated when techniques proven to improve the performance of the simulation are available. Considering artificial intelligence procedures or genetic algorithms may be an effective choice.

There is a need to update MODSIM to include the capability of simulating new HVAC component models as they are developed. Research involving the study of new and upcoming tools might prove beneficial. Research sources such as U.S. Department of Energy (Washington, DC), National Institute of Standards and Technology (Gaithersburg, Maryland), International Building Performance Simulation Association (College Station, Texas), and the Building Environmental Performance Analysis Club may be useful.

The visual tool does not allow users to add components to its library. The visual tool may be distributed with a FORTRAN compiler that can rebuild MODSIM on the fly. The MODSIM program contains code to simulate components recognized by HVACSIM+. The user supplies details about the component and provides an algorithm to simulate the component. The visual tool may validate the algorithm, add the code to the Fortran project, rebuild MODSIM and associate it with the tool.

REFERENCES

BESA. Building Energy Systems Analysis (BESA) Design User Manual, (BESA Design version 2.0), Scanada Consultants Limited, Canada. 1993.

Birdsall, B., Buhl, W. F., Ellington, K. L., Erdem, A. E. and Winkelmann, F. C. Overview of the DOE-2 Building Energy Analysis Program Version 2.1D, LBL-19735, Rev. 1, Lawrence Berkeley Laboratory, February. 1990.

BLAST User Reference, Volume 1 & 2, BLAST Support Office, Department of Mechanical and Industrial Engineering, University of Illinois at Urbana-Champaign. 1991.

Brayton, R. K., Gustavson, F. G., and Hachtel, G. D. A New Efficient Algorithm for Solving Differential – Algebraic Systems Using Implicit Backward Differential Formulas. Proc. IEEE. Vol. 60, No. 1. January, 1972. Pages 98-108.

Broyden, C. G. A class of Methods for Solving Nonlinear Simultaneous Equations. Math. Comp. Vol. 19. 1965. Pages 577-593.

Buhl, W. F., Erdem, A. E., Winkelmann, F. C. and Sowell, E. F. Recent Improvements In SPARK; Strong Component Decomposition, Multivalued Objects, And Graphical Interface. Proc. Building Simulation '93, The International Building Performance Simulation Association, Adelaide, Australia. August 16-18, 1993.

Buzzi-Ferraris, G., and Tronconi, E. 1993. An Improved Convergence Criterion in the Solution of Nonlinear Algebraic Equations. Computer & Chemical Engineering. Vol. 17, No. 100, pp.1053-1056.

Chiasson, A. D., and Spitler, J. D. Modeling Approach to Design of a Ground-Source Heat Pump Bridge Deck Heating System. Transportation Research Record 1741. Paper No. S00-0054. Pages 207-215. 2001.

Clark, D. R. HVACSIM+ Building Systems and Equipment Simulation Program Reference Manual. NBSIR 84-2996. National Bureau of Standards, January, 1985.

Clark, D. R. and May, W. R, Jr. HVACSIM+ Building Systems and Equipment Simulation Program Users Guide. NBSIR 85-3243. National Bureau of Standards, September, 1985.

Crawley, Drury B, Linda K Lawrie, Curtis O Pedersen, and Frederick C Winkelmann. EnergyPlus: Energy Simulation Program, in ASHRAE Journal, Vol. 42, No. 4 (April), pp. 49-56. 2000.

Dennis, J. E. and, Schnabel, R. B. 1983. *Numerical Methods for Unconstrained Optimization and Nonlinear Equations*. New Jersey: Prentice-Hall, Inc.

ESRU, The ESP-r System for Building Energy Simulations: User Guide Version 9 Series, University of Strathclyde, Glasgow UK. 2000

Gear, C. W. 1971. The Automatic Integration of Ordinary Differential Equations. Comm. *ACM* 14:176-179

Haves, P. and L. K. Norford. 1995. ASHRAE 825- RP Final Report. *A Standard Simulation Test Bed for the Evaluation of Control Algorithms and Strategies.*

Hensen, J.L.M. 1995. On System Simulation for Building Performance Evaluation. Proc. 4th IBPSA World Congress on Building Simulation '95. Int. Building Performance Simulation Association. Madison, WI. Madison. August, 1995. Pages 259-267

Hiebert, K. L. An Evaluation of Mathematical Software that Solves Systems of Nonlinear Equations. ACM Transactions. Mathematical Software. Vol. 8, No. 1. March, 1982. Pages 5-20.

Kusuda, T. NBSLD, The Computer Program for Heating and Cooling Loads in Building. BSS 69. National Bureau of Standards. July, 1976.

MacMillan, D. 1999. Relaxing convergence conditions to improve the convergence rate.. Ph.D. Thesis. University of Colorado at Denver.

NBS. 1985. *HVACSIM+ Building Systems and Equipment Simulation Program Reference*. Gaithersburg: National Bureau of Standards.

Moller, S. K. and Wooldridge, M. J. User Guide for the Computer Program BUNYIP: Building Energy Investigation Package (Version 2.0), Technical Report - TR6, Division of Energy Technology, CSIRO, Victoria, Australia. 1985.

Nicholas I. M. Gould, Dominique Orban, Annick Sartenaer and Philippe L. Toint. Componentwise fast convergence in the solution of full-rank systems of nonlinear equations. Cerfacs Technical Report No. TR/PA/00/56, revised August 2001.

Ohadi, M. M., Meyer, J. R. and Pollington, K. M. The applicability and limitations of a simplified building energy analysis model (ASEAM2.1) — a comparison study, ASHRAE Transactions, 95 (Part 1): 1163-1173. 1989.

Park, C., Clark D. R., Kelly G. E. 1985. An Overview of HVACSIM+, a Dynamic Building/HVAC/Control Systems Simulation Program. *Building Energy Simulation Conference,* Seattle, Washington. August 21-22, 1985.

Per Sahlin. Modelling and Simulation Methods for Modular Continuous Systems in Buildings. Department of Building Sciences, Royal Institute of Technology, Stockholm, Sweden. May, 1996.

Paloschi, Jorge R. A Hybrid Continuation Algorithm to Solve Algebraic Nonlinear Equations. Computer & Chemical Engineering. Vol. 18, No. SUPPL July 5-7, 1993 1994. Pages S201-S209 0098-1354.

Powell, M. J. D. 1970. *A Hybrid Method for Nonlinear Equations in Numerical Methods*. London: P. Rabinowitz, Ed., Gordon and Breach.

Ramamoorthy, Mahadevan. 2001. Applications Of Hybrid Ground Source Heat Pump Systems To Buildings And Bridge Decks. M.S. Thesis. Oklahoma State University.

Schonwalder, P. P., and Sans, J.M. 1995. A Discrete-Time Approach to the Steady State Analysis and Optimization of Non-linear Autonomous Circuits. International Journal of Circuit Theory and Applications. Vol. 23, pp. 297-310.

Shacham, M. 1990. A Variable Order Method for Solution Of A Nonlinear Algebraic Equations. Computers & Chemical Engineering. Vol. 14, No. 6, pp. 621-629.

Sowell, Edward. F., and Philip Haves. *Efficient Solution Strategies for Building Energy System Simulation*. March 30, 2000.

Spedicato, E. Numerical Experience with Newton-like methods for Nonlinear Algebraic systems. Computing (Vienna/New York). Vol. 58, No. 1. 1997 Springer-Verlag Wein Pages 69-89.

Spitler, Jeff. Using the Earth for Energy Storage. Heat transfer engineering. An International Journal. Vol. 22, No. 6. November-December 2001.

Trane Company. TRACE 600 Engineering Manual, TRCE-UM-602, January. 1992a.

Trane Company. TRACE 600 User Manual, TRCE-UM-601, January. 1992b.

TRNSYS: a Transient System Simulation Program, (version 12.2), Engineering Experiment Station Report 38-12, Solar Energy Laboratory, University of Wisconsin-Madison. 1988.

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

Wickham, F. (Editor-in-Chief). User Guide for the APEC Energy Simulation, Mechanical Engineering Services Design Aids No. DA12, Australian Government Publishing Service, Canberra. 1985.

Yosuke Nishi, Xiangyang Chen , and Nobuo Nakahara. Proc. of Building Simulation '99 Conference. Kyoto, Japan. Vol. 2. September 13-15, 1999, pp. 669-674.

APPENDIX A

DESCRIPTION OF HVACSIM+ COMPONENT MODELS

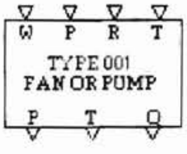
The following pages provide a summary of the component models supplied with HVACSIM+. The descriptions of the models and their configurations are provided. The HVACSIM+ Reference manual discusses the Types 1-26 in detail and the HVACSIM+ Building Loads Calculation manual provides detailed description of the Types 50-53.

A.1. TYPE 1: FAN OR PUMP

General Description

The pump or fan model calculates a pressure rise and efficiency as functions of a mass flow rate, using dimensionless performance curves. The performance curves are represented by polynomials, with empirically determined coefficients that must be supplied by the user. The efficiency is used to compute a temperature rise across the fan or pump, as well as power consumption. The model can work with water or air.

Component Configuration



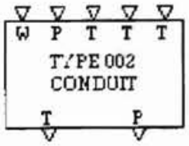
	Description	Units
Inputs		
1	Mass flow rate of the fluid	kg/s
2	Outlet pressure	kPa
3	Rotational speed	rps
4	Inlet temperature	°C
Outputs		
1	Inlet pressure	kPa
2	Outlet temperature	°C
3	Power consumption	kW
Parameters		
1-5	Coefficients for head vs. flow curve	No units
6-10	Coefficient for efficiency curve	No units
11	Diameter of blades or impeller	m
12	Mode: Air = 1, Water = 2	No units

A.2. TYPE 2: CONDUIT (DUCT OR PIPE)

General Description

The conduit model is designed to account for three effects: thermal losses, transport delays, and dynamics due to thermal capacitance. The last parameter, *MODE*, determines whether the Type behaves as a pipe or duct. If the value of *MODE* is 1 or –1, the model represents a duct. If the value of *MODE* is 2 or –2, the model represents a pipe.

Component Configuration



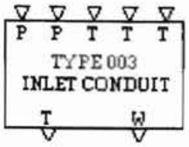
	Description	Units
Inputs		
1	Mass flow rate of the fluid	kg/s
2	Outlet pressure	kPa
3	Inlet temperature	°C
4	Ambient temperature	°C
5	Outlet temperature	°C
Outputs		
1	Outlet temperature	°C
2	Inlet pressure	kPa
Parameters		
1	Inside surface heat transfer coefficient * area	W/°C
2	Outside surface heat transfer coefficient * area	W/°C
3	Thermal capacitance of conduit material	kJ/°C
4	Volume of the conduit	m^3
5	Flow resistance coefficient	No units
6	Height of outlet above inlet	m
7	Mode: • 1 -> Air (Simple dynamics with no differential equations • 2 -> Water (Simple dynamics with no differential equations • -1 -> Air (Detailed dynamics with one differential equation) • -2 -> Water (Detailed dynamics with one differential equation)	No units

A.3. TYPE 3: INLET CONDUIT (PIPE OR DUCT)

General Description

In the Type 2 Conduit model, the inlet pressure is calculated from the outlet pressure and the mass flow rate. In the Type 3 Inlet Conduit model, the mass flow rate is calculated from the inlet and outlet pressures. In all other respects, the two models are identical.

Component Configuration



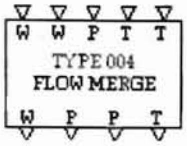
	Description	Units
Inputs		
1	Inlet pressure	kPa
2	Outlet pressure	kPa
3	Inlet temperature	°C
4	Ambient temperature	°C
5	Outlet temperature	°C
Outputs		
1	Outlet temperature	°C
2	Mass flow rate	kg/s
Parameters		
1	Inside surface heat transfer coefficient * area	W/°C
2	Outside surface heat transfer coefficient * area	W/°C
3	Thermal capacitance of conduit material	kJ/°C
4	Volume of the conduit	m^3
5	Flow resistance coefficient	No units
6	Height of outlet above inlet	m
7	Mode: • 1 -> Air (Simple dynamics with no differential equations • 2 -> Water (Simple dynamics with no differential equations • -1 -> Air (Detailed dynamics with one differential equation) • -2 -> Water (Detailed dynamics with one differential equation)	No units

A.4. TYPE 4: FLOW MERGE

General Description

This component models the merging of two flow streams. It calculates the mass flow rate and temperature of the single outlet flow-stream, and the pressures at the two inlets. The model assumes that the flow resistance parameter is the same for all three branches.

Component Configuration



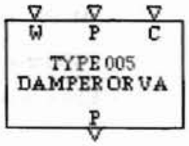
	Description	Units
Inputs		
1	Flow rate at first inlet	kg/s
2	Flow rate at second inlet	kg/s
3	Pressure at outlet	kPa
4	Temperature at first inlet	°C
5	Temperature at second inlet	°C
Outputs		
1	Flow rate at outlet	kg/s
2	Pressure at first inlet	kPa
3	Pressure at second inlet	kPa
4	Temperature at outlet	°C
Parameters		
1	Flow resistance coefficient	No units

A.5. TYPE 5: DAMPER OR VALVE

General Description

This model represents dampers or having inherent characteristics that are linear, exponential, or intermediate between linear and exponential. The characteristic is determined by the third parameter, the weighting factor. When this factor is zero, the model represents an exponential valve or damper. When the factor is one, a linear valve or damper is modeled. Intermediate values of the weighting factor produce intermediate characteristics.

Component Configuration



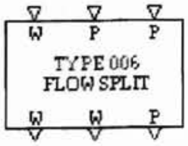
	Description	Units
Inputs		
1	Mass flow rate	kg/s
2	Outlet pressure	kPa
3	Relative position of damper or valve	No units
Outputs		
1	Inlet pressure	kPa
Parameters		
1	Full open resistance	0.001/(kg.m)
2	Leakage parameter	No units
3	Weighting factor for linear term of flow resistance	No units
4	Mode: • 0 -> Damper closed when relative position is 0 • 1 -> Damper closed when relative position is 1	No units

A.6. TYPE 6: FLOW SPLIT

General Description

This component models the division of a flow stream into two flow streams. It calculates the mass flow rates of the two outlets, and the pressure at the single inlet. The model assumes that the flow resistance parameter is the same for all three branches.

Component Configuration



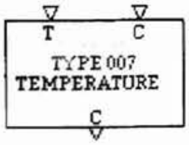
	Description	Units
Inputs		
1	Mass flow rate at inlet	kg/s
2	Pressure at first outlet	kPa
3	Pressure at second outlet	kPa
Outputs		
1	Flow rate at first outlet	kg/s
2	Flow rate at second outlet	kg/s
3	Pressure at inlet	kPa
Parameters		
1	Flow resistance coefficient	No units

A.7. TYPE 7: TEMPERATURE SENSOR

General Description

Temperature sensors are modeled by a simple-first-order differential equation with a single time constant. The input to the component is modified using a gain and an offset supplied as parameters, allowing the output to be treated as a voltage level, a control signal, or a temperature in either Fahrenheit or Celsius. If a Celsius output is desired, an offset of zero and a gain of one are used. If a control variable between zero and one is required for use as a controller input, the offset is the minimum allowable temperature and the gain is the maximum allowable temperature minus the minimum temperature.

Component Configuration



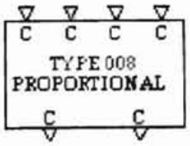
	Description	Units
Inputs		
1	Temperature input	°C
2	Temperature sensor output	°C
Outputs		
1	Temperature sensor output	°C
Parameters		
1	Time constant of the sensor	No units
2	Offset value	No units
3	Gain value	No units

A.8. TYPE 8: PROPORTIONAL-INTEGRAL CONTROLLER

General Description

This component models an analog proportional-integral controller. Proportional controller alone can be modified by setting the integral gain to zero. A time constant is used to model the response time of the controller.

Component Configuration



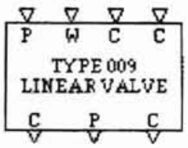
	Description	Units
Inputs		
1	Controlled variable	No units
2	Set point for controlled variable	No units
3	Integral portion of control signal	No units
4	Output control signal	No units
Outputs		
1	Integral portion of control signal (feedback connection to Input 3)	No units
2	Output control signal (feedback connection to Input 4)	No units
Parameters		
1	Proportional gain	No units
2	Integral gain	No units
3	Controller time constant	No units

A.9. TYPE 9: LINEAR VALVE WITH PNEUMATIC ACTUATOR

General Description

The model represents a linear valve and includes a pneumatic actuator model. When used in a system with other component models, this model accurately models the effects of valve authority. The valve authority is a factor used to distinguish between the inherent and installed characteristics of a valve. The dynamic response of the actuator is modeled by a first order differential equation. The actuator can be effectively removed by setting the time constant and the hysteresis parameter to zero.

Component Configuration



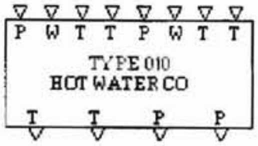
	Description	Units
Inputs		
1	Pressure at outlet	kPa
2	Mass flow rate	No units
3	Input control signal	No units
4	Actuator relative position	No units
Outputs		
1	Actuator relative position (feedback connection to Input 4)	No units
2	Pressure at inlet	No units
Parameters		
1	Flow resistance when valve is open	0.001/(kg.m)
2	Actuator time constant	No units
3	Leakage parameter	No units
4	Hysteresis parameter	No units

A.10. TYPE 10: HOT WATER TO AIR HEATING COIL (SIMPLE)

General Description

The model represents the simplest water-to-air heat exchanger. A constant overall heat transfer coefficient is assumed.

Component Configuration



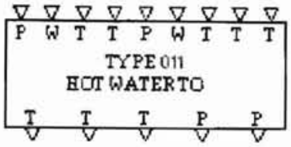
	Description	Units
Inputs		
1	Outlet air pressure	kPa
2	Air mass flow rate	kg/s
3	Inlet air temperature	°C
4	Outlet air temperature	°C
5	Outlet water pressure	kPa
6	Water mass flow rate	kg/s
7	Inlet water temperature	°C
8	Outlet water temperature	°C
Outputs		
1	Outlet air temperature	°C
2	Outlet water temperature	°C
3	Inlet air pressure	kPa
4	Inlet water pressure	kPa
Parameters		
1	Overall heat transfer coefficient * area	W/°C
2	Flow resistance parameter, air side	No units
3	Flow resistance parameter, water side	No units
4	Water side volume of coil	m^3
5	Capacitive term of coil time constant	No units

A.11. TYPE 11: HOT WATER TO AIR HEATING COIL (DETAILED)

General Description

The model represents a detailed water-to-air heat exchanger. A constant overall heat transfer coefficient is not assumed. Separate inside and outside heat-transfer-coefficients are calculated as functions of the water and air flow-rates.

Component Configuration



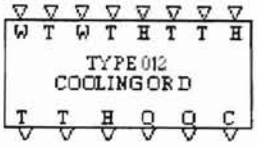
	Description	Units
Inputs		
1	Outlet air pressure	kPa
2	Air mass flow rate	kg/s
3	Inlet air temperature	°C
4	Outlet air temperature	°C
5	Outlet water pressure	kPa
6	Water mass flow rate	kg/s
7	Inlet water temperature	°C
8	Outlet water temperature	°C
9	Dummy temperature used to calculate air temperature dynamics	°C
Outputs		
1	Outlet air temperature	°C
2	Outlet water temperature	°C
3	Dummy temperature (same as Input 9)	°C
4	Inlet air pressure	kPa
5	Inlet water pressure	kPa
Parameters		
1	Overall heat transfer coefficient * area (air)	W/°C
2	Overall heat transfer coefficient * area (air)	W/°C
3	Overall heat transfer coefficient * area (water)	W/°C
4	Overall heat transfer coefficient * area (water)	W/°C
5	Flow resistance parameter, air side	No units
6	Flow resistance parameter, water side	No units
7	Water side volume of coil	m^3
8	Mass time specific heat of coil material	kJ/°C

A.12. TYPE 12: COOLING OR DEHUMIDIFYING COIL

General Description

This model represents a circular finned or continuous finned serpentine heat exchanger with four or more rows in counterflow crossflow configuration. The model accounts for condensation on the outside surface of the coil. The information provided in the parameters is used to calculate the fin efficiencies, heat transfer coefficients, and the time constants.

Component Configuration



	Description	Units
Inputs		
1	Water mass flow rate	kg/s
2	Inlet water temperature	°C
3	Dry air mass flow rate	kg/s
4	Inlet air dry-bulb temperature	°C
5	Inlet air humidity ratio	No units
6	Dynamic outlet water temperature	°C
7	Dynamic outlet air temperature	°C
8	Dynamic outlet air humidity ratio	No units
Outputs		
1	Dynamic outlet water temperature	°C
2	Dynamic outlet air temperature	°C
3	Dynamic outlet air humidity ratio	No units
4	Total cooling load at steady state	kW
5	Sensible cooling load at steady state	kW
6	Wet fraction of total surface area	No units
Parameters		
1	Coil type: 0 = flat fins, 1 = circular fins	No units
2	Exterior tube surface area	m^2
3	Fin surface area	m^2
4	Internal surface area	m^2
5	Ratio of minimum flow area to face area	No units

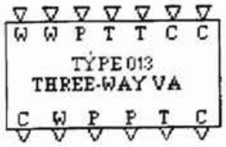
	Description	Units
Parameters		
6	Thermal conductivity of fin material	kW/(m.°C)
7	Coil face area	m^2
8	Number of fins per centimeter	No units
9	Number of tubes per row	No units
10	Number of rows	No units
11	Tube outside diameter	m
12	Tube inside diameter	m
13	Fin thickness	m
14	Mass times specific heat of coil material	kJ/°C
15	Tube longitudinal spacing	m
16	Fin diameter	m
17	Coil depth in air flow direction	m
18	Thermal conductivity of tube material	kW/(m.°C)

A.13. TYPE 13: THREE-WAY VALVE WITH ACTUATOR

General Description

This component represents a valve with two inlet ports and one outlet port. The position of the valve is determined by an input control signal. Port 1 of the valve is closed when the control-signal holds 0 and open when the signal holds 1. The sixth parameter, *MODE*, determines the inherent characteristics of the two inlet ports. The valve model includes a pneumatic actuator model.

Component Configuration



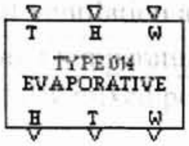
	Description	Units
Inputs		
1	Mass flow rate, inlet port 1	kg/s
2	Mass flow rate, inlet port 2	kg/s
3	Pressure at outlet	kPa
4	Temperature, inlet port 1	°C
5	Temperature, inlet port 2	°C
6	Input control signal	No units
7	Actuator relative position	No units
Outputs		
1	Actuator relative position	No units
2	Outlet mass flow rate	kg/s
3	Pressure, inlet port 1	kPa
4	Pressure, inlet port 2	kPa
5	Temperature at outlet	°C
Parameters		
1	Port 1 - Flow resistance when open	0.001/(kg.m)
2	Port 2 - Flow resistance when open	0.001/(kg.m)
3	Leakage parameter	No units
4	Actuator time constant	No units
5	Hysteresis parameter	No units
6	Mode: Determines characteristics of ports 1 and 2 0 => Linear/Linear 1 => Exponential/Linear 2 => Exponential/Exponential	No units

A.14. TYPE 14: EVAPORATIVE HUMIDIFIER

General Description

This component represents an air stream passing over a water pan or through a porous pad, increasing the humidity and decreasing the temperature of the air. The model assumes that the process is adiabatic, which means that the water temperature is assumed equal to the entering air wet bulb temperature. In addition, the process is assumed to occur at constant pressure. Any pressure drop through the humidifier may be modeled by increasing the flow resistance of an adjacent component.

Component Configuration



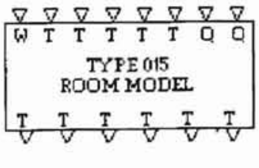
	Description	Units
Inputs		
1	Inlet air temperature	°C
2	Inlet air humidity ratio	No units
3	Inlet moist air mass flow rate	kg/s
Outputs		
1	Outlet air humidity ratio	No units
2	Outlet air temperature	°C
3	Outlet moist air mass flow rate	kg/s
Parameters		
1	Water to air heat transfer coefficient times area	kW/°C

A.15. TYPE 15: ROOM MODEL

General Description

The room model considers the air mass, an interior mass, and a wall mass to represent a room. No modeling of the building shell is attempted. The heat flows are treated as inputs to the model. The air mass is artificially divided into two parts: a fully mixed part near the ventilation supply, and a piston flow part near the ventilation exhaust. The fraction of air mass in these parts is adjustable. The temperatures in the model are assumed uniform.

Component Configuration



	Description	Units
Inputs		
1	Mass flow rate of ventilation air	kg/s
2	Ventilation air inlet temperature	°C
3	Temperature of fully mixed portion of air mass	°C
4	Wall mass temperature	°C
5	Interior mass temperature	°C
6	Spatial average temperature of the piston flow portion of the air mass	°C
7	Conduction heat flow into wall mass	kJ/(kg.°C)
8	Heat flow due to internal gains	kJ/(kg.°C)
Outputs		
1	Temperature of fully mixed portion of air mass	°C
2	Wall mass temperature	°C
3	Interior mass temperature	°C
4	Spatial average temperature of the piston flow portion of the air mass	°C
5	Average room air temperature	°C
6	Exhaust air temperature	°C
Parameters		
1	Volume of room air mass	m^3
2	Thermal capacitance of walls	kJ/°C
3	Thermal capacitance of interior mass	kJ/°C
4	Heat transfer coefficient times area for the wall mass	W/°C
5	Heat transfer coefficient times area for the interior mass	W/°C
6	Fraction of the air mass which is fully mixed	No units

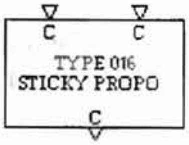
A.16. TYPE 16: STICKY PROPORTIONAL CONTROLLER

General Description

This component represents the proportional controller with upper and lower limits on the rate of change of the output control signal. The limits are determined by the second parameter, error band. If the error signal changes by an amount less that the error band, the change is ignored and the output signal remains constant until the cumulative changes in error equal or exceed the error band.

NOTE: this component is to be used only to pass control signals between superblocks. The variable defined by its output should not be solved simultaneously.

Component Configuration



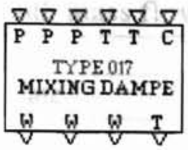
	Description	**Units**
Inputs		
1	Measured temperature input control signal	No units
2	Temperature set point control signal	No units
Outputs		
1	Controller output	No units
Parameters		
1	Proportional gain	No units
2	Error band	No units
3	Offset	No units

A.17. TYPE 17: MIXING DAMPERS AND MERGE

General Description

This model represents a pair of mixing dampers, mechanically linked so that one damper opens as the other closes. The first inlet is closed when the input control signal is zero, and open when it is one. The reverse is true for the second inlet. Hysteresis effects and an actuator time constant are not included in the model.

Component Configuration



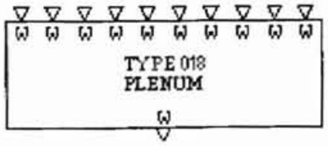
	Description	Units
Inputs		
1	Pressure at first inlet	kPa
2	Pressure at second inlet	kPa
3	Pressure at outlet	kPa
4	Temperature at first inlet	°C
5	Temperature at second inlet	°C
6	Input control signal	No units
Outputs		
1	Flow rate at first inlet	kg/s
2	Flow rate at second inlet	kg/s
3	Flow rate at outlet	kg/s
4	Temperature at outlet	°C
Parameters		
1	Full open resistance	0.001/(kg.m)
2	Leakage parameter	No units
3	Weighting factor for linear term of flow resistance	No units

A.18. TYPE 18: PLENUM

General Description

The purpose of this model is to output the sum of its inputs (mass flow rates). A maximum of ten inputs can be specified.

Component Configuration



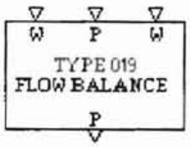
	Description	Units
Inputs		
1-10	First, second,…tenth mass flow rate	kg/s
Outputs		
1	Outlet mass flow rate	kg/s
Parameters		
1	Number of input mass flow rates	No units

A.19. TYPE 19: FLOW BALANCE CONTROL

General Description

This model does not represent any physical component. It serves instead to minimize mathematical problems that arise when a closed loop flow stream is divided between two superblocks. It functions by providing one superblock with an estimate of the net flow resistance encountered by the flow stream in another superblock.

Component Configuration



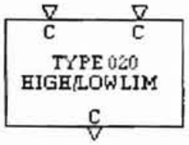
	Description	Units
Inputs		
1	Mass flow rate determined in this superblock	kg/s
2	Downstream pressure, where flow stream enters this superblock	kPa
3	Mass flow rate determined in another superblock	kg/s
Outputs		
1	Upstream pressure, where flow stream exits this superblock	kPa
Parameters		
1	Flow resistance parameter at time = 0	No units

A.20. TYPE 20: HIGH LIMIT OR LOW LIMIT CONTROLLER

General Description

This component represents a proportional controller that exerts a control action only when the measured input signal is on one side of a fixed set point. For a high limit controller, no control action is exerted unless the measured input signal exceeds the set point. For a low limit controller, control action is exerted only when the controlled input falls below the set point. The component represents a high limit controller when the second parameter, set point time, is positive, and a low limit when the set point time is negative.

Component Configuration



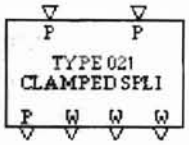
	Description	Units
Inputs		
1	Measured temperature input control signal	No units
2	Controller output	No units
Outputs		
1	Controller output	No units
Parameters		
1	Proportional gain	No units
2	Set point times	No units

A.21. TYPE 21: 'GROUNDED' WATER OR AIR SPLIT

General Description

This component models the division of a flow stream into two flow streams. It is referred to as 'grounded' by analogy to a grounded electrical junction: the pressure at the center of the split is a constant, entered as a parameter. The model assumes that the flow resistance parameter is the same for all three branches.

Component Configuration



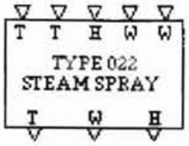
	Description	Units
Inputs		
1	Pressure at first outlet	kPa
2	Pressure at second outlet	kPa
Outputs		
1	Pressure at inlet	kPa
2	Flow rate at inlet	No units
3	Flow rate at first outlet	kg/s
4	Flow rate at second outlet	kg/s
Parameters		
1	Pressure at center of split	kPa
2	Flow resistance coefficient	No units

A.22. TYPE 22: STEAM SPRAY HUMIDIFER

General Description

The model calculates an outlet air temperature, flow rate, and humidity ratio for a constant pressure process in which steam is injected into an airflow stream to increase the humidity of the air. The outlet air humidity ratio is limited either by the steam flow rate or by the saturation humidity ratio (at atmospheric pressure) times a saturation efficiency. The mass flow rate of the air increases slightly due to the addition of moisture.

Component Configuration



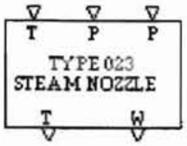
	Description	Units
Inputs		
1	Steam temperature	No units
2	Inlet air temperature	°C
3	Inlet air humidity ratio	No units
4	Inlet air mass flow rate	kg/s
5	Steam mass flow rate	kg/s
Outputs		
1	Outlet air temperature	°C
2	Outlet air mass flow rate	kg/s
3	Outlet air humidity ratio	No units
Parameters		
1	Saturation efficiency	No units

A.23. TYPE 23: STEAM NOZZLE

General Description

The steam nozzle calculates a mass flow rate and exit steam temperature, given the downstream pressure and the upstream steam stagnation temperature and pressure. The model assumes an adiabatic, isentropic process, considers condensation and uses real gas correlations to determine the steam properties.

Component Configuration



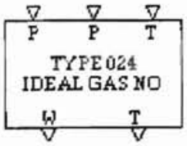
	Description	Units
Inputs		
1	Upstream stagnation temperature	°C
2	Upstream stagnation pressure	kPa
3	Exit pressure	kPa
Outputs		
1	Exit temperature	°C
2	Steam mass flow rate	kg/s
Parameters		
1	Nozzle exit cross sectional area	No units
2	Absolute pressure corresponding to zero gage pressure	kPa

A.24. TYPE 24: IDEAL GAS NOZZLE

General Description

This component models a convergent nozzle for any ideal gas and assumes the nozzle to be isentropic, adiabatic and without condensation.

Component Configuration



	Description	Units
Inputs		
1	Upstream stagnation pressure	kPa
2	Downstream pressure	kPa
3	Stagnation temperature	°C

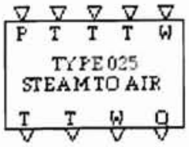
	Description	Units
Outputs		
1	Mass flow rate	kg/s
2	Exit temperature	°C
Parameters		
1	Nozzle exit cross sectional area	m^2
2	Absolute pressure corresponding to zero gage pressure	kPa
3	Ratio of specific heats	No units
4	Gas constant per unit mass	No units

A.25. TYPE 25: STEAM TO AIR HEAT EXCHANGER

General Description

This component represents a cross-flow heat exchanger consisting of horizontal tubes with circular fins. Air flowing across the finned tubes is heated, while steam inside the tubes is condensed. The steam inside is assumed to be equipped with a trap that allows condensate but not vapor to leave the tubes. The condensate outlet temperature is assumed a fixed number of degrees below the saturation temperature. In addition, the average temperature of steam and condensate is assumed equal to the saturation temperature, and condensation is assumed to occur at constant pressure.

Component Configuration



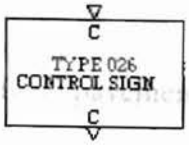
	Description	Units
Inputs		
1	Steam pressure	kPa
2	Inlet steam temperature	°C
3	Inlet air temperature	°C
4	Outlet air temperature	°C
5	Mass flow rate of air	kg/s
Outputs		
1	Outlet air temperature	°C
2	Outlet temperature of condensate	°C
3	Mass flow rate of steam	kg/s
4	Rate of heat transfer to air	kW
Parameters		
1	Sub-cooling of condensate below saturation	°C
2	Minimum air flow area	m^2
3	Inside heat transfer area	m^2
4	Primary (tube) outside surface area	m^2
5	Secondary (fin) outside surface area	m^2
6	Tube inside diameter	m
7	Tube outside diameter	m
8	Tube thermal conductivity	kJ/(kg.K)
9	Fin diameter	m
10	Fin thermal conductivity	kJ/(kg.K)
11	Fin thickness	m
12	Number of fins per meter	No units
13	Thermal capacitance of metal (fins and tubes)	kJ/K

A.26. TYPE 26: CONTROL SIGNAL INVERTER

General Description

This component is a very simple representation of an inverting relay. Its single input is a control signal between zero and one, and its output is one minus the input. In a typical simulation, a single control variable might represent both the output of a controller and the relative position of a valve or damper.

Component Configuration



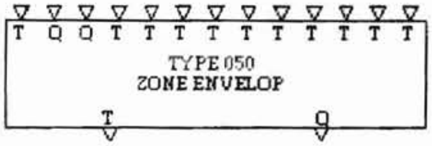
	Description	Units
Inputs		
1	Input control signal	No units
Outputs		
1	Output control signal	No units
Parameters		
NONE		

A.27. TYPE 50: ZONE ENVELOPE

General Description

This model is part of the building shell model and it computes the convective heat gain (from building surface) and mean radiant temperature. This model must be in a superblock that takes a user-selected, fixed time step.

Component Configuration



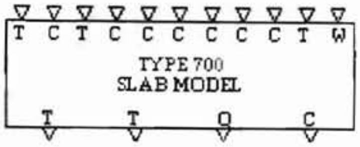
	Description	Units
Inputs		
1	Input control signal	No units
Outputs		
1	Output control signal	No units
Parameters		
NONE		

A.28. TYPE 700: Slab Model: Explicit Slab Model

General Description

This models represents an artificial pavement heating system, which consists of a system of hot fluid-carrying pipes in the slab. The temperature distribution in the slab is determined by an explicit transient finite-difference approach.

Component Configuration



	Description	Units
Inputs		
1	Ambient Air Temperature	°C
2	Humidity Ratio Of Air	Kg Water/kg Dry-air
3	Sky Temperature	°C
4	Wind Speed	m/s
5	Wind Direction (Degrees From North 0-90)	Degrees
6	Solar Radiation	W/m^2
7	Solar Angle Of Incidence	Radians
8	Snowfall Rate In Water Equivalent Per Hr	mm/hr
9	Rainfall Rate In Water Equivalent Per Hr	mm/hr
10	Inlet Fluid Temperature	°C
11	Mass Flow Rate	kg/s
Outputs		
1	Average Pavement Top Surface Temp.	°C
2	Outlet Fluid Temperature	°C
3	Heat Provided To Pavement Surface	kW
4	Snow Free Area Ratio	No units
Parameters		
1	Pavement Length	m
2	Pavement Width	m
3	Slab Orientation (0 - 90 Degrees From North)	Degrees
4	Pavement Thickness	m
5	Pipe Spacing	m
6	Pipe Diameter	m
7	Pipe Depth Below Surface	m

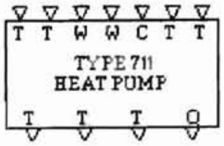
	Description	Units
Parameters		
8	Depth To Interface Of Material 1-2	m
9	Thermal Conductivity Of Layer 1	W/(m.°C)
10	Thermal Conductivity Of Layer 2	W/(m.°C)
11	Emmissivity Coefficient	No units
12	Absorptivity Coefficient	No units
13	Volumetric Heat Capacity Of Layer 1	J/(m^3.°C)
14	Volumetric Heat Capacity Of Layer 2	J/(m^3.°C)
15	Thermal Conductivity Of Pipe Material	W/(m.°C)
16	Wall Thickness Of Pipe	m
17	Fluid Type (1: Propylene Glycol; 2: Ethylene Glycol)	No units
18	Wt. Of Antifreeze In Solution Used (%)	No units
19	Number Of Flow Circuits	No units
20	Length Of Pipe Per Circuit	m
21	Time Step For Discretization Equation	s
22	Flag For Bottom Boundary Condition (0=Insulated; 1=Convection Type)	No units

A.29. TYPE 711: Heat Pump: Gang of Water to Water Heat pump

General Description

This model is a Fractional Heat Pump System model. It presents a series of water to water heat pumps.

Component Configuration



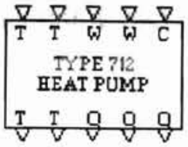
	Description	**Units**
Inputs		
1	Load Side (Or Condenser) Entering Fluid Temp.	°C
2	Source Side (Or Evaporator) Entering Fluid Temp.	°C
3	Load Side Mass Flow Rate	kg/s
4	Source Side Mass Flow Rate	kg/s
5	Number Of Heat Pumps In Operation	No units
6	Dynamic Load Side (Or Condenser) Leaving Fluid Temp.	°C
7	Dynamic Source Side (Or Evaporator) Leaving Fluid Temp.	°C
Outputs		
1	Dynamic Load Side (Or Condenser) Leaving Fluid Temp.	°C
2	Dynamic Source Side (Or Evaporator) Leaving Fluid Temp.	°C
3	Load Side Entering Fluid Temperature To The Second Heat Pump	°C
4	Power Consumption Of The Heat Pump	kW
Parameters		
1	Piston Displacement	m^3 / s
2	Clearance Factor	No units
3	Load Side Heat Transfer Coefficient	kW/K
4	Source Side Heat Transfer Coefficient	kW/K
5	Loss Factor Used To Define The Electromechanical Loss For Compressor	No units
6	Constant Part Of The Electromechanical Loss	kW
7	Pressure Drop Across The Suction Valve	kPa
8	Superheat	°C
9	Max No. Of Heat Pumps	No units
10	Minimum Source Side Inlet Temperature	°C
11	Maximum Load Side Inlet Temperature	°C
12	Initial Guess Of Heat Transfer Rate For Load Side	kW
13	Initial Guess Of Heat Transfer Rate For Source Side	kW
14	Load Side Time Constant	s
15	Source Side Time Constant	s
16	Load Side Fluid Type (1: Propylene Glycol; 2: Ethylene Glycol)	No units
17	Wt. Of Antifreeze In Solution Used In Load Side (%)	No units
18	Source Fluid Type (1: Propylene Glycol; 2: Ethylene Glycol)	No units
19	Wt. Of Antifreeze In Solution Used In Source Side (%)	No units

A.30. TYPE 712: Heat Pump: Single Water to Water Heat pump Model

General Description

This model simulates a simple heat pump. The model is used only for heating.

Component Configuration



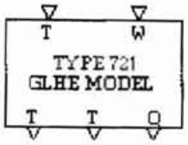
	Description	Units
Inputs		
1	Load Side (Or Condenser) Entering Fluid Temp.	°C
2	Source Side (Or Evaporator) Entering Fluid Temp.	°C
3	Load Side Mass Flow Rate	kg/s
4	Source Side Mass Flow Rate	kg/s
5	Control Signal (1: On; 0: Off)	No units
Outputs		
1	Load Side (Or Condenser) Leaving Fluid Temp	°C
2	Source Side (Or Evaporator) Leaving Fluid Temp	°C
3	Power Consumption Of The Heat Pump	kW
4	Load Side (Or Condenser) Heat Transfer Rate	kW
5	Source Side (Or Evaporator) Heat Transfer Rate	kW
Parameters		
1	VS	No units
2	VI	No units
3	Load Side Heat Transfer Coefficient	kW/K
4	Source Side Heat Transfer Coefficient	kW/K
5	Electromechanical Loss Factor For Compressor	No units
6	Constant Part Of The Electromechanical Loss	kW
7	Superheat	°C
8	Minimum Source Side Inlet Temperature	°C
9	Maximum Load Side Inlet Temperature	°C
10	Initial Guess Of Heat Transfer Rate For Load Side	kW
11	Initial Guess Of Heat Transfer Rate For Source Side	kW
12	Load Side Fluid Type (1: Propylene Glycol; 2: Ethylene Glycol)	No units
13	Wt. Of Antifreeze In Solution Used In Load Side (%)	No units
14	Source Fluid Type (1: Propylene Glycol; 2: Ethylene Glycol)	No units
15	Wt. Of Antifreeze In Solution Used In Source Side (%)	No units

A.31. TYPE 721: GLHE Model: Hierarchical Load Aggregation Model

General Description

This model is used to simulate the heat transfer in Vertical U-tube Ground Heat exchangers. This model computes the average fluid temperature and the outlet fluid temperature for a given inlet fluid temperature and mass flow rate based on a given borehole geometry.

Component Configuration



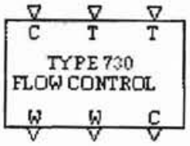
	Description	Units
Inputs		
1	Inlet fluid temperature	°C
2	Mass flow rate	kg/s
Outputs		
1	Outlet fluid temperature	°C
2	Average fluid temperature	°C
3	Normalized heat extraction rate for current time step	W/m
Parameters		
1	Number of boreholes	No units
2	Borehole length	m
3	Borehole radius	m
4	Thermal conductivity of the ground	W/(mK)
5	Volumetric heat capacity of ground	No units
6	Undisturbed ground temperature	°C
7	Grout thermal conductivity	W/(mK)
8	Pipe thermal conductivity	W/(mK)
9	Outer diameter of the pipe	m
10	Distance between the two legs of the U Tube	m
11	Wall thickness of the pipe	m
12	Fluid Type (1: Propylene Glycol; 2: Ethylene Glycol)	No units
13	Wt. Of Antifreeze In Solution Used (%)	No units

A.32. TYPE 730: Flow Control: Linear Proportional Control (Heating)

General Description

This model computes the number of heat pump pairs in operation depending on a simple linear proportional control algorithm. Maximum number of heat pump pairs are used if the bridge deck surface temperature is less than a user specified lower limit; and 1 pair of heat pump is used if the temperature is above a certain upper limit. It also sends designed value of mass flow rate to each component model of the system according to operation mode of heating or recharge. To avoid the numerical problem this controller model needs to be included in a separated SUPERBLOCK.

Component Configuration



	Description	Units
Inputs		
1	Snow Flag	No units
2	Objective Temperature For Control	°C
3	GLHE Average Temperature	°C
Outputs		
1	Load side mass flow rate	No units
2	Source side mass flow rate	kg/s
3	Number of heat pump pairs in operation	No units
Parameters		
1	Heating Mode Lower Limit	°C
2	Heating Mode Upper Limit	°C
3	Max. No. Of Heat Pump Pairs In The System	No units
4	Recharge Mode Lower Limit	°C
5	Recharge Mode Upper Limit	°C
6	Load side mass flow rate in heating mode	kg/s
7	Source side mass flow rate in heating mode	kg/s
8	Mass flow rate in recharge mode	kg/s

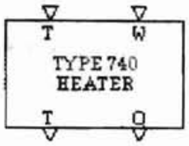
A.33. TYPE 740: Heater: 2-Mode Ideal Steady State Heater

General Description

This model is an ideal electrical heater model, which can operate at two modes:

- Mode 1: Constant heat flux supply
- Mode 2: Constant fluid-outlet temperature

Component Configuration



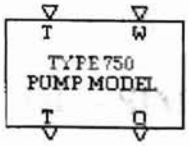
	Description	**Units**
Inputs		
1	Inlet Fluid Temperature	°C
2	Mass flow rate of the fluid	kg/s
Outputs		
1	Outlet Fluid Temperature	°C
2	Heat Output	kW
Parameters		
1	Running Mode (1: Constant heat flux; 2: Constant outlet temperature)	No units
2	Maximum Outlet Temp	°C
3	Maximum Heat Supply	kW
4	Fluid Type (1: Propylene Glycol, 2: Ethylene Glycol)	No units
5	Weight Concentration Of Antifreeze In Percentage (%)	No units

A.34. TYPE 750: Pump Model: Simple Pump Model

General Description

This model represents a simple pump model, which determines the outlet flow depending on the control signal. It calculates the temperature rise and power consumption of the heat pump with given efficiency, maximum flow rate and the pressure drop across the pump.

Component Configuration



	Description	Units
Inputs		
1	Inlet Fluid Temperature	°C
2	Mass flow rate of the fluid	kg/s
Outputs		
1	Outlet Fluid Temperature	°C
2	Power Consumption	kW
Parameters		
1	Nominal Efficiency	No units
2	Nominal mass flow rate	kg/s
3	Nominal Pressure rise	kW
4	Fluid Type (1: Propylene Glycol, 2: Ethylene Glycol)	No units
5	Weight Concentration Of Antifreeze In Percentage (%)	No units

APPENDIX B

VISUAL MODELING TOOL – SIMULATION TASK TREE

Build a simulation
Get pre-processing information
Identify models needed for the simulation
Identify the initial values for input/output ports for the selected models
Identify the parameter values
Identify variable that will read input from a boundary file
Identify the ports that will be connected
Identify variables whose state needs to be reported in the output files
Identify the boundary values to be written into the boundary file
Set simulation requirements [Simulation Requirements option in the Settings menu]
Enter simulation title
Enter error tolerances [Click on the Error Tolerances button to show a form for entry]
Enter simulation times
Enter reporting, freezing and scanning details
Answer questions related to diagnostics that need to be performed
Add units to the workspace
Select types from the Types Component Library list box
Insert the components
Position the components in the workspace [Drag and drop the units in the workspace]
Enter initial values to output ports and parameters [Double click the unit icon]
Select the reported variables [Left mouse button click on a port while holding the ALT key]
Add boundary conditions to the workspace (they connect to only input ports on units)
Select category from the Boundary Conditions Library list box
Insert the boundary conditions.
Position the boundary conditions in the workspace [Drag and drop the icons in the workspace]
Find candidates for constant boundary conditions [Double click on the icon]
Change the type from time-dependent to constant [Use the radio buttons]
Enter a constant value
Find candidate for time-dependent boundary conditions [Double click on the icon]
Enter unique names
Add connections [Hold left mouse button on an output port, drag the mouse and

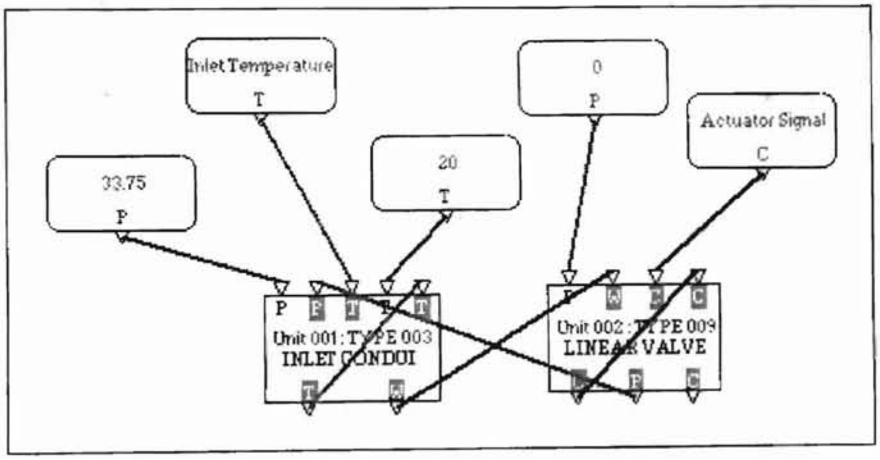
leave on input port]
Add boundary values to the workspace
Open the Boundary File Editor [Boundary File Editor option in the Boundary-Conditions menu]
If new file/data-set
Fill the Time values
Fill the values for each boundary variable (identified by its unique name) at every time
If boundary file already exists
If all the data from the existing file is needed [Read-data-from-file option in the File menu]
Select the file
Extract the contents of the file to the grid.
If only specific column of data is needed [Read-column-from-file option in the File menu]
Select the file
Extract the column from the file to the grid.
If file is open in a spreadsheet belonging to another application (like Microsoft Excel)
Copy and paste the data into the grid
Close the editor
Save the simulation workspace [Save or Save-As options from the File menu]
Execute the simulation [Run option from the Simulation menu]
If any errors
Identify the errors
Correct the errors
If no errors
Plot the simulation results/residuals [Click on the supplied button when simulation ends or use Reports menu from the main window]

APPENDIX C

HVACSIM+ EXAMPLE SIMULATIONS

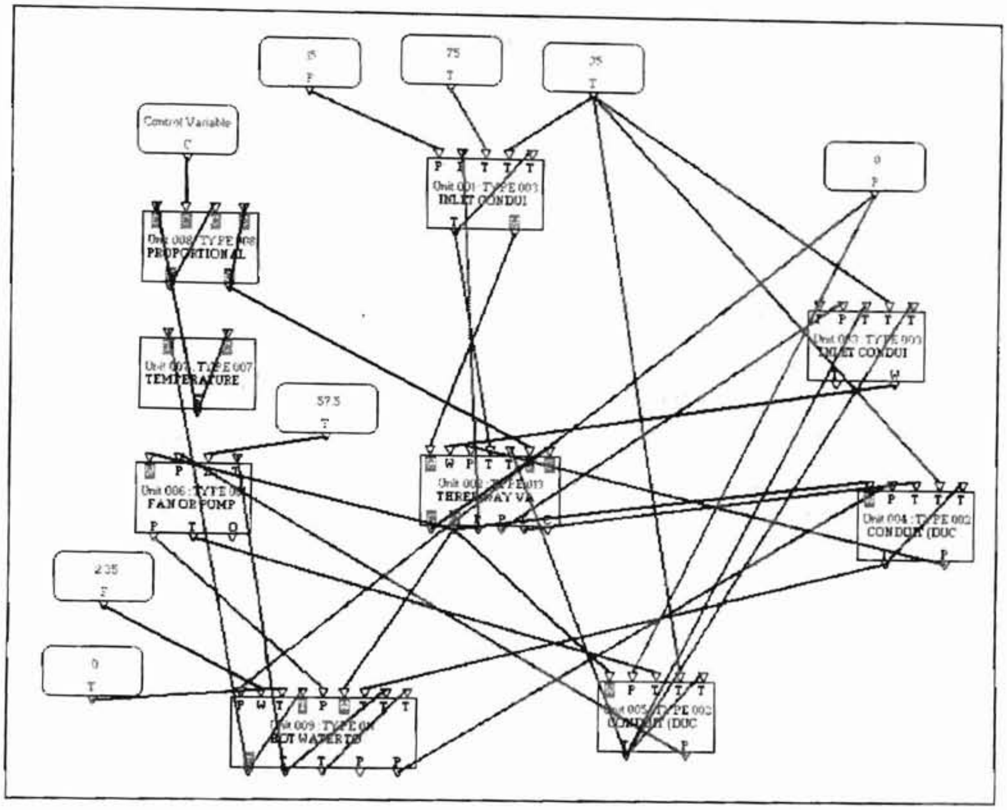
C.1. Example 1: Inlet pipe and linear valve system

C.1.1. Workspace Diagram



C.2. Example 2: Heating Coil Control Loop, Variable Inlet Water Temperature Control

C.2.1. Workspace Diagram



C.2.2. Initial values for various units

Properties of Inlet Conduit (Duct or Pipe)		
Variables	**Description**	**Value**
Input Variables	Inlet fluid pressure	15
	Output fluid pressure	10
	Inlet fluid temperature	75
	Ambient air temperature	35
	Outlet fluid temperature	70
Output Variables	Outlet fluid temperature	70
	Mass flow rate of fluid	0.04
Parameters	Inside heat transfer coefficient by area (KW/C)	16.4
	Outside heat transfer coefficient by area (KW/C)	0.17
	Thermal capacitance of conduit material (KJ/C)	25.2
	Volume (m^3)	0.015
	Flow resistance [0.001/(Kg.m)]	13.5
	Height of outlet above inlet (m)	0.0
	Mode: 2-Water, 1-Air	2

Figure C.2.2-1. *Example 2: Initial values for TYPE 3 unit 1*

Properties of Three Way Valve Model		
Variables	**Description**	**Value**
Input Variables	Inlet mass flow rate (used when control signals are 1)	0.04
	Inlet mass flow rate (used when control signals are 0)	1
	Outlet pressure	-15
	Inlet temperature (used when control signals are 1)	70
	Inlet temperature (used when control signals are 0)	24
	Control variable from controller	0.4
	Actuator relative position	0.4
Output Variables	Actuator relative position	0.4
	Outlet mass flow rate	1.04
	Inlet pressure (used when control signals are 1)	10
	Inlet pressure (used when control signals are 0)	-15
	Outlet temperature	26.5
	Value stem relative position	0
Parameters	Flow resistance (used when control signals are 1)	0.3
	Flow resistance (used when control signals are 0)	1.5
	Leakage parameter	0.004
	Actuator time constant (Seconds)	10
	Hysteresis parameter	0.1
	Control signals Mode: 0->Lin/Lin, 1->Exp/Lin, else->Exp/Exp	1

Figure C.2.2-2. *Example 2: Initial values for TYPE 13 unit 2*

Properties of Inlet Conduit (Duct or Pipe)		
Variables	**Description**	**Value**
Input Variables	Inlet fluid pressure	0
	Output fluid pressure	-15
	Inlet fluid temperature	24
	Ambient air temperature	35
	Outlet fluid temperature	24
Output Variables	Outlet fluid temperature	24
	Mass flow rate of fluid	1.0
Parameters	Inside heat transfer coefficient by area (KW/C)	0
	Outside heat transfer coefficient by area (KW/C)	0
	Thermal capacitance of conduit material (KJ/C)	0
	Volume (m^3)	0
	Flow resistance [0.001/(Kg.m)]	14.8
	Height of outlet above inlet (m)	0.5588
	Mode: 2-Water, 1-Air	2

Figure C.2.2-3. *Example 2: Initial values for TYPE 3 unit 3*

Properties of Conduit (Duct or Pipe)		
Variables	**Description**	**Value**
Input Variables	Mass flow rate of fluid	1.04
	Outlet pressure	-32
	Inlet temperature of fluid	26.5
	Ambient temperature	35
	Outlet fluid temperature	26.5
Output Variables	Outlet fluid temperature	26.5
	Inlet pressure	-15
Parameters	Inside heat transfer coefficient by area	2.97
	Outside heat transfer coefficient by area	0.0031
	Thermal capacitance of conduit material	2.277
	Volume	0.00251
	Flow resistance	14.36
	Height of outlet above inlet	-0.5588
	Mode: 2-Water, 1-Air	2

Figure C.2.2-4. *Example 2: Initial value for TYPE 2 unit 4*

Properties of Conduit (Duct or Pipe)		
Variables	**Description**	**Value**
Input Variables	Mass flow rate of fluid	1.04
	Outlet pressure	0
	Inlet temperature of fluid	24
	Ambient temperature	35
	Outlet fluid temperature	24
Output Variables	Outlet fluid temperature	24
	Inlet pressure	24
Parameters	Inside heat transfer coefficient by area	2.7
	Outside heat transfer coefficient by area	0.00281
	Thermal capacitance of conduit material	2.07
	Volume	0.002278
	Flow resistance	11.9
	Height of outlet above inlet	0
	Mode: 2-Water, 1-Air	2

Figure C.2.2-5. *Example 2: Initial value for TYPE 2 unit 5*

Properties of Hot Water To Air Heating Coil		
Variables	**Description**	**Value**
Input Variables	Outlet air pressure	0
	Air mass flow rate	2.35
	Inlet air temperature	0
	Outlet air temperature	18
	Outlet water pressure	-40
	Water mass flow rate	1.04
	Inlet water temperature	26.5
	Outlet water temperature	24
	Dummy temperature	20
Output Variables	Outlet air temperature	18
	Outlet water temperature	24
	Dummy temperature	20
	Inlet air pressure	0.7
	Inlet water pressure	-32
Parameters	Coefficient for air side heat transfer coefficient	1.7631
	Exponent for air side heat transfer correlation	0.448735
	Coefficient for water side heat transfer correlation	14.1111
	Exponent for water side heat transfer correlation	1.13024
	Air flow resistance parameter	0.12
	Water flow resistance parameter	7
	Water side volume of coil	0.0073
	Capacitance of coil material	26.2

Figure C.2.2-6. *Example 2: Initial value for TYPE 11 unit 6*

Properties of Temperature Sensor		
Variables	**Description**	**Value**
Input Variables	Inlet temperature	18
	Sensor output	0.18
Output Variables	Sensor output	0.18
Parameters	Sensor time constant (Seconds)	20
	Temperature offset (C)	0
	Temperature gain (C)	100

Figure C.2.2-7. *Example 2: Initial value for TYPE 7 unit 8*

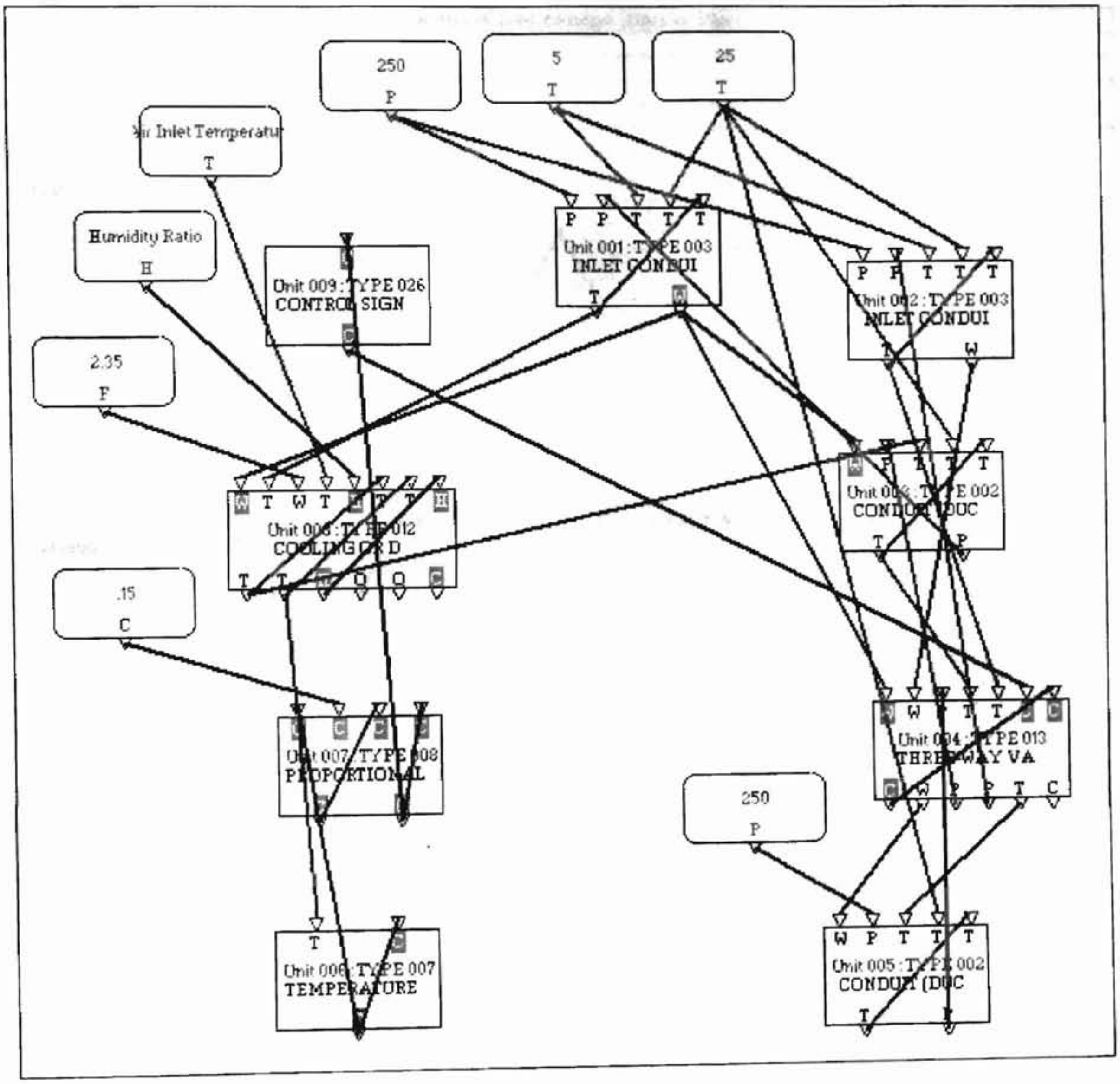
Properties of Fan Or Pump		
Variables	**Description**	**Value**
Input Variables	Mass flow rate of fluid	1.04
	Outlet pressure	24
	Fan or pump rotational speed	57.5
	Inlet fluid temperature	24
Output Variables	Inlet pressure	-40
	Outlet fluid temperature	24
	Power consumption	0.1
Parameters	1st pressure coefficient	1.65
	2nd pressure coefficient	-70
	3rd pressure coefficient	5
	4th pressure coefficient	0
	5th pressure coefficient	30
	1st efficiency coefficient	0.95
	2nd efficiency coefficient	0
	3rd efficiency coefficient	0
	4th efficiency coefficient	0
	5th efficiency coefficient	0
	Diameter (m)	0.127
	Mode: Air=1, Water=2	2

Figure C.2.2-8. *Example 2: Initial values for TYPE 1 unit 7*

Properties of Proportional-Integral Controller		
Variables	**Description**	**Value**
Input Variables	Controlled variable	0.18
	Set point for controlled variable	0.2
	Integral portion of control signal	0
	Output control signal	0.4
Output Variables	Integral portion of control signal	0
	Output control signal	0.4
Parameters	Proportional gain	2
	Integral gain	0.02
	Controller time constant	2

Figure C.2.2-9. *Example 2: Initial values for TYPE 8 unit 9*

C.3. Example 3: Cooling Coil Control Loop with Variable Water Flow Rate



C.3.1. Initial values for various units

Properties of Inlet Conduit (Duct or Pipe)

Variables	**Description**	**Value**
Input Variables	Inlet fluid pressure	250
	Output fluid pressure	227.754
	Inlet fluid temperature	5
	Ambient air temperature	25
	Outlet fluid temperature	5.013
Output Variables	Outlet fluid temperature	5.013
	Mass flow rate of fluid	1.1393
Parameters	Inside heat transfer coefficient by area (KW/C)	2.97
	Outside heat transfer coefficient by area (KW/C)	0.0031
	Thermal capacitance of conduit material (KJ/C)	2.277
	Volume (m^3)	0.00251
	Flow resistance [0.001/(Kg.m)]	21.36
	Height of outlet above inlet (m)	-0.5588
	Mode: 2-Water, 1-Air	2

Figure C.3.1-1. *Example 3: Initial values for unit 1*

Properties of Inlet Conduit (Duct or Pipe)

Variables	**Description**	**Value**
Input Variables	Inlet fluid pressure	250
	Output fluid pressure	222.467
	Inlet fluid temperature	5
	Ambient air temperature	25
	Outlet fluid temperature	5.00093
Output Variables	Outlet fluid temperature	5.00093
	Mass flow rate of fluid	1.49359
Parameters	Inside heat transfer coefficient by area (KW/C)	0.276
	Outside heat transfer coefficient by area (KW/C)	0.00029
	Thermal capacitance of conduit material (KJ/C)	0.2116
	Volume (m^3)	0.00023
	Flow resistance [0.001/(Kg.m)]	14.8
	Height of outlet above inlet (m)	-0.5588
	Mode: 2-Water, 1-Air	2

Figure C.3.1-2. *Example 3: Initial values for unit 2*

Properties of Conduit (Duct or Pipe)

Variables	**Description**	**Value**
Input Variables	Mass flow rate of fluid	1.1393
	Outlet pressure	212.307
	Inlet temperature of fluid	16.8008
	Ambient temperature	25
	Outlet fluid temperature	16.0806
Output Variables	Outlet fluid temperature	16.0806
	Inlet pressure	227.754
Parameters	Inside heat transfer coefficient by area	2.7
	Outside heat transfer coefficient by area	0.00281
	Thermal capacitance of conduit material	2.07
	Volume	0.002278
	Flow resistance	11.9
	Height of outlet above inlet	0
	Mode: 2-Water, 1-Air	2

Figure C.3.1-3. *Example 3: Initial values for unit 3*

Properties of Three Way Valve Model		
Variables	**Description**	**Value**
Input Variables	Inlet mass flow rate (used when control signals are 1)	1.1393
	Inlet mass flow rate (used when control signals are 0)	1.49359
	Outlet pressure	210.571
	Inlet temperature (used when control signals are 1)	16.0806
	Inlet temperature (used when control signals are 0)	5.0093
	Control variable from controller	0.424409
	Actuator relative position	0.424409
Output Variables	Actuator relative position	0.424409
	Outlet mass flow rate	2.63289
	Inlet pressure (used when control signals are 1)	212.307
	Inlet pressure (used when control signals are 0)	222.467
	Outlet temperature	10.109
	Value stem relative position	0
Parameters	Flow resistance (used when control signals are 1)	0.3
	Flow resistance (used when control signals are 0)	1.5
	Leakage parameter	0.004
	Actuator time constant (Seconds)	10
	Hysteresis parameter	0.1
	Control signals Mode: 0->Lin/Lin, 1->Exp/Lin, else->Exp/Exp	0

Figure C.3.1-4. *Example 3: Initial values for unit 4*

Properties of Conduit (Duct or Pipe)		
Variables	**Description**	**Value**
Input Variables	Mass flow rate of fluid	2.63289
	Outlet pressure	-250
	Inlet temperature of fluid	10.109
	Ambient temperature	25
	Outlet fluid temperature	10.1684
Output Variables	Outlet fluid temperature	10.1684
	Inlet pressure	210.571
Parameters	Inside heat transfer coefficient by area	38
	Outside heat transfer coefficient by area	0.044
	Thermal capacitance of conduit material	32.2
	Volume	0.0355
	Flow resistance	66.44
	Height of outlet above inlet	0
	Mode: 2-Water, 1-Air	2

Figure C.3.1-5. *Example 3: Initial values for unit 5*

Properties of Temperature Sensor		
Variables	**Description**	**Value**
Input Variables	Inlet temperature	15.0079
	Sensor output	0.150079
Output Variables	Sensor output	0.150079
Parameters	Sensor time constant (Seconds)	20
	Temperature offset (C)	0
	Temperature gain (C)	100

Figure C.3.1-6. *Example 3: Initial values for unit 6*

	Properties of Proportional-Integral Controller	
Variables	**Description**	**Value**
Input Variables	Controlled variable	0.150079
	Set point for controlled variable	0.15
	Integral portion of control signal	0.576066
	Output control signal	0.575589
Output Variables	Integral portion of control signal	0.576066
	Output control signal	0.575589
Parameters	Proportional gain	9
	Integral gain	0.12
	Controller time constant	2

Figure C.3.1-7. *Example 3: Initial values for unit 7*

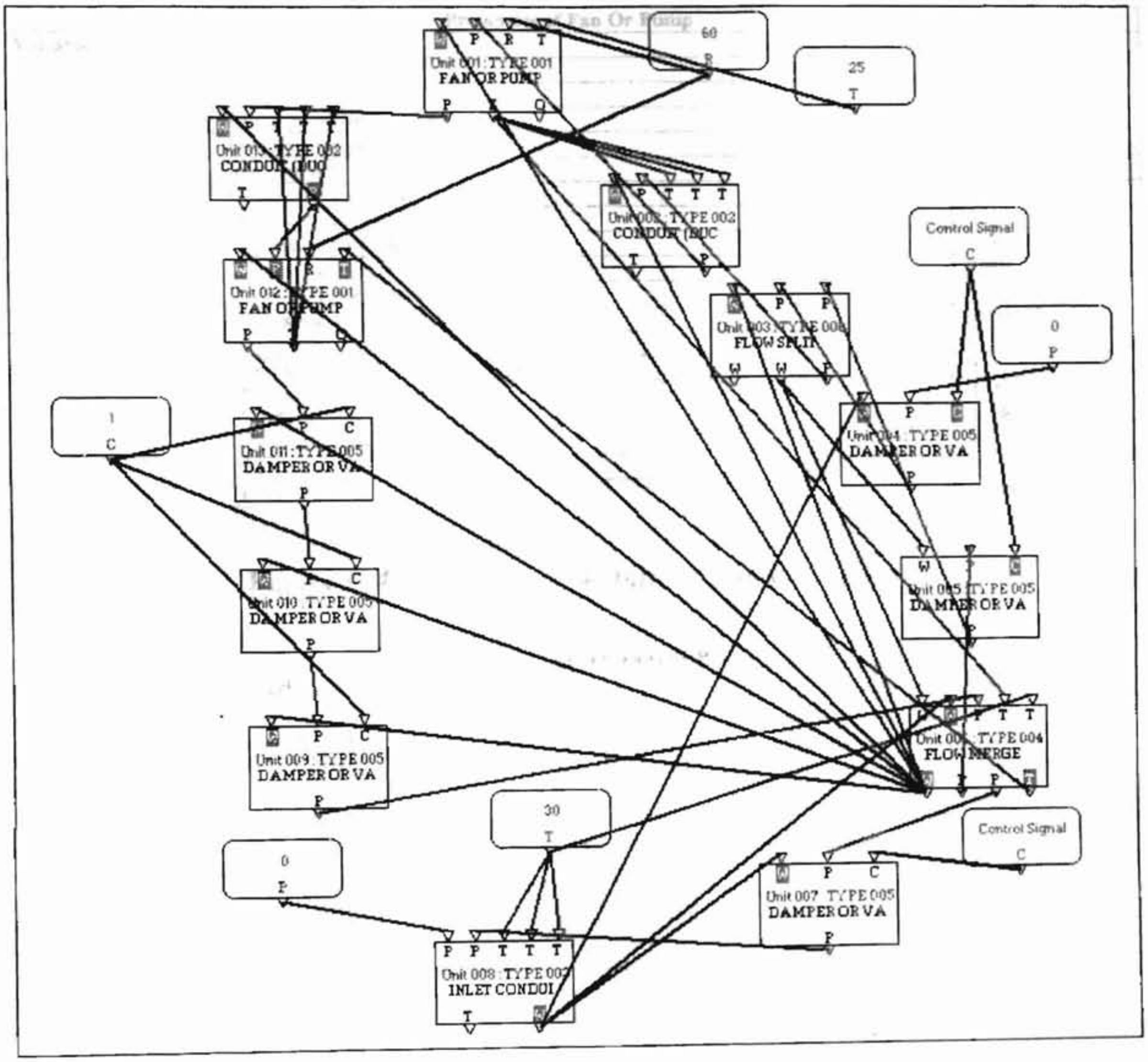
	Properties of Cooling Or Dehumidifying Coil	
Variables	**Description**	**Value**
Input Variables	Water mass flow rate	1.1393
	Inlet water temperature	5.013
	Dry air mass flow rate	2.35
	Inlet dry bulb air temperature	25
	Inlet air humidity	0.016
	Outlet water temperature	16.8008
	Outlet air temperature	15.0079
	Outlet air humidity	0.0106262
Output Variables	Outlet water temperature	16.8008
	Outlet air dry bulb temperature	15.0079
	Outlet air humidity	0.0106262
	Total cooling load	56.2285
	Sensible cooling load	23.8925
	Wet fraction of coil surface area	1
Parameters	Coil type: 0=flat continuous fins, 1=circular fins	0
	Primary (tube exterior) surface area (m2)	6.2381
	Secondary (fin) surface area (m2)	101.715
	Internal surface area (m2)	6.2
	Ratio of minimum air flow area to face area	0.416
	Fin material thermal conductivity [KW/(m k)]	0.204
	Coil face area (m2)	0.7226
	Number of fins per centimeter	5.5118
	Number of circuits	16
	Number of tubes per circuit	6
	Outside tube diameter (m)	0.01588
	Inside tube diameter (m)	0.01445
	Fin thickness (m)	0.00015
	Thermal capacitance (mass x specific heat) of coil (kj/k)	34.669
	Tube row spacing in air flow direction (m)	0.0259
	Fin diameter [if par(1)=1] or length [if par(1)=0] (m)	0.508
	Coil depth in air flow direction (m)	0.1651
	Tube thermal conductivity [kw/(m k)]	0.386

Figure C.3.1-8. *Example 3: Initial values for unit 8*

	Properties of Control Signal Inverter	
Variables	**Description**	**Value**
Input Variables	Input control	0.575589
Output Variables	Output control	0.424409
Parameters	Multiplier	1

Figure C.3.1-9. *Example 3: Initial values for unit 9*

C.4. Example 4: Air handling unit without heat exchangers



C.4.1. Initial values for various units

Properties of Fan Or Pump		
Variables	**Description**	**Value**
Input Variables	Mass flow rate of fluid	2
	Outlet pressure	0.25
	Fan or pump rotational speed	60
	Inlet fluid temperature	25
Output Variables	Inlet pressure	-1.75
	Outlet fluid temperature	25
	Power consumption	0
Parameters	1st pressure coefficient	3.64
	2nd pressure coefficient	0.801
	3rd pressure coefficient	-0.19
	4th pressure coefficient	-0.00445
	5th pressure coefficient	0
	1st efficiency coefficient	0
	2nd efficiency coefficient	0.564
	3rd efficiency coefficient	-0.0862
	4th efficiency coefficient	0
	5th efficiency coefficient	0
	Diameter (m)	0.3365
	Mode: Air=1, Water=2	1

Figure C.4.1-1. *Example 4: Initial values for unit 1*

Properties of Conduit (Duct or Pipe)		
Variables	**Description**	**Value**
Input Variables	Mass flow rate of fluid	2
	Outlet pressure	0.2
	Inlet temperature of fluid	25
	Ambient temperature	25
	Outlet fluid temperature	25
Output Variables	Outlet fluid temperature	0
	Inlet pressure	0.25
Parameters	Inside heat transfer coefficient by area	0
	Outside heat transfer coefficient by area	0
	Thermal capacitance of conduit material	0
	Volume	0
	Flow resistance	0.01
	Height of outlet above inlet	0
	Mode: 2-Water, 1-Air	1

Figure C.4.1-2. *Example 4: Initial values for unit 2*

Properties of Flow Split		
Variables	**Description**	**Value**
Input Variables	Inlet mass flow rate	2
	Output pressure 1	0.19
	Outlet pressure 2	0.16
Output Variables	Outlet mass flow rate 1	0
	Outlet mass flow rate 2	1.9
	Inlet pressure	0.2
Parameters	Flow resistance	0.011

Figure C.4.1-3. *Example 4: Initial values for unit 3*

	Properties of Damper	
Variables	**Description**	**Value**
Input Variables	Fluid mass flow rate	0.01
	Output pressure	0
	Control: relative position of damper or valve	0
Output Variables	Inlet pressure	0.19
Parameters	Flow resistance, damper or valve open	0.001
	Leakage parameter	0.012
	Characteristic: 0=>exp., 1=>lin., Intermediate => intermed.	0.87
	Mode: 0=>closed when control=0; 1=>closed when control=1	0

Figure C.4.1-4. *Example 4: Initial values for unit 4*

	Properties of Damper	
Variables	**Description**	**Value**
Input Variables	Fluid mass flow rate	1.9
	Output pressure	0.15
	Control: relative position of damper or valve	0
Output Variables	Inlet pressure	0.16
Parameters	Flow resistance, damper or valve open	0.001
	Leakage parameter	0.012
	Characteristic: 0=>exp., 1=>lin., Intermediate => intermed.	0.87
	Mode: 0=>closed when control=0; 1=>closed when control=1	1

Figure C.4.1-5. *Example 4: Initial values for unit 5*

	Properties of Flow Merge	
Variables	**Description**	**Value**
Input Variables	Inlet mass flow rate 1	1.9
	Inlet mass flow rate 2	0.01
	Outlet pressure	-0.04
	Inlet temperature	25
	Inlet temperature	30
Output Variables	Outlet mass flow rate	2
	Inlet pressure 1	0.15
	Inlet pressure 2	0.15
	Outlet pressure	26
Parameters	Flow resistance	0.011

Figure C.4.1-6. *Example 4: Initial values for unit 6*

	Properties of Damper	
Variables	**Description**	**Value**
Input Variables	Fluid mass flow rate	0.01
	Output pressure	0.15
	Control: relative position of damper or valve	0
Output Variables	Inlet pressure	-0.000001
Parameters	Flow resistance, damper or valve open	0.001
	Leakage parameter	0.012
	Characteristic: 0=>exp., 1=>lin., Intermediate => intermed.	0.87
	Mode: 0=>closed when control=0; 1=>closed when control=1	0

Figure C.4.1-7. *Example 4: Initial values for unit 7*

Properties of Inlet Conduit (Duct or Pipe)		
Variables	**Description**	**Value**
Input Variables	Inlet fluid pressure	0
	Output fluid pressure	-0.000001
	Inlet fluid temperature	30
	Ambient air temperature	30
	Outlet fluid temperature	30
Output Variables	Outlet fluid temperature	0
	Mass flow rate of fluid	0.01
Parameters	Inside heat transfer coefficient by area (KW/C)	0
	Outside heat transfer coefficient by area (KW/C)	0
	Thermal capacitance of conduit material (KJ/C)	0
	Volume (m^3)	0
	Flow resistance [0.001/(Kg.m)]	0.001
	Height of outlet above inlet (m)	0
	Mode: 2-Water, 1-Air	1

Figure C.4.1-8. *Example 4: Initial values for unit 8*

Properties of Damper		
Variables	**Description**	**Value**
Input Variables	Fluid mass flow rate	2
	Output pressure	-0.08
	Control: relative position of damper or valve	1
Output Variables	Inlet pressure	-0.04
Parameters	Flow resistance, damper or valve open	0.04
	Leakage parameter	0.00004
	Characteristic: 0=>exp., 1=>lin., Intermediate => intermed.	1
	Mode: 0=>closed when control=0; 1=>closed when control=1	0

Figure C.4.1-9. *Example 4: Initial values for unit 9*

Properties of Damper		
Variables	**Description**	**Value**
Input Variables	Fluid mass flow rate	2
	Output pressure	-0.12
	Control: relative position of damper or valve	1
Output Variables	Inlet pressure	-0.08
Parameters	Flow resistance, damper or valve open	0.012
	Leakage parameter	0.000012
	Characteristic: 0=>exp., 1=>lin., Intermediate => intermed.	1
	Mode: 0=>closed when control=0; 1=>closed when control=1	0

Figure C.4.1-10. *Example 4: Initial values for unit 10*

Properties of Damper		
Variables	**Description**	**Value**
Input Variables	Fluid mass flow rate	2
	Output pressure	-0.25
	Control: relative position of damper or valve	1
Output Variables	Inlet pressure	-0.12
Parameters	Flow resistance, damper or valve open	0.04
	Leakage parameter	0.00004
	Characteristic: 0=>exp., 1=>lin., Intermediate => intermed.	1
	Mode: 0=>closed when control=0; 1=>closed when control=1	0

Figure C.4.1-11. *Example 4: Initial values for unit 11*

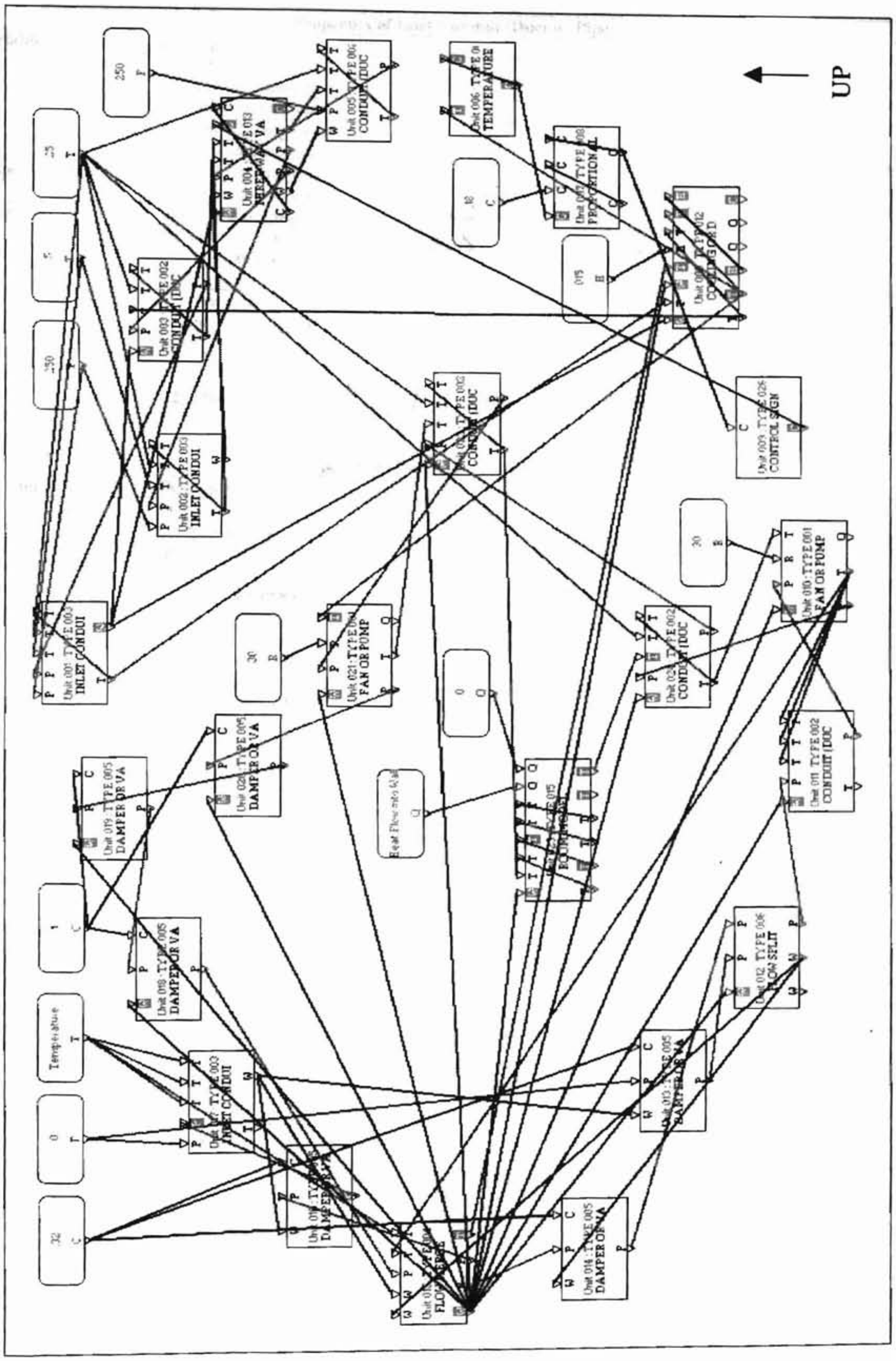
Properties of Fan Or Pump		
Variables	**Description**	**Value**
Input Variables	Mass flow rate of fluid	2
	Outlet pressure	1.75
	Fan or pump rotational speed	60
	Inlet fluid temperature	26
Output Variables	Inlet pressure	-0.25
	Outlet fluid temperature	26
	Power consumption	0
Parameters	1st pressure coefficient	3.64
	2nd pressure coefficient	0.801
	3rd pressure coefficient	-0.19
	4th pressure coefficient	-0.00445
	5th pressure coefficient	0
	1st efficiency coefficient	0
	2nd efficiency coefficient	0.564
	3rd efficiency coefficient	-0.0862
	4th efficiency coefficient	0
	5th efficiency coefficient	0
	Diameter (m)	0.3365
	Mode: Air=1, Water=2	1

Figure C.4.1-12. *Example 4: Initial values for unit 12*

Properties of Conduit (Duct or Pipe)		
Variables	**Description**	**Value**
Input Variables	Mass flow rate of fluid	2
	Outlet pressure	-1.75
	Inlet temperature of fluid	26
	Ambient temperature	26
	Outlet fluid temperature	26
Output Variables	Outlet fluid temperature	0
	Inlet pressure	1.75
Parameters	Inside heat transfer coefficient by area	0
	Outside heat transfer coefficient by area	0
	Thermal capacitance of conduit material	0
	Volume	500
	Flow resistance	0.875
	Height of outlet above inlet	0
	Mode: 2-Water, 1-Air	1

Figure C.4.1-13. *Example 4: Initial values for unit 13*

C.5. Example 5: Single Zone Air Handler with Cooling Coil



Unit 001 : TYPE 003 INLET CONDUI
Unit 002 : TYPE 003 INLET CONDUI
Unit 003 : TYPE 002 CONDUIT (DUC
Unit 004 : TYPE 013 THREE WAY VA
Unit 005 : TYPE 002 CONDUIT (DUC
Unit 006 : TYPE 0 TEMPERATURE
Unit 007 : TYPE 008 PROPORTIONAL
Unit 008 : TYPE 012 COOLING CRD
Unit 009 : TYPE 026 CONTROL SIGN
Unit 010 : TYPE 001 FAN OR PUMP
Unit 011 : TYPE 002 CONDUIT (DUC
Unit 012 : TYPE 006 FLOW SPLIT
Unit 013 : TYPE 005 DAMPER OR VA
Unit 014 : TYPE 005 DAMPER OR VA
Unit 015 : TYPE 004 FLOW MERGE
Unit 016 : TYPE 005 DAMPER OR VA
Unit 017 : TYPE 003 INLET CONDUI
Unit 018 : TYPE 005 DAMPER OR VA
Unit 019 : TYPE 005 DAMPER OR VA
Unit 020 : TYPE 005 DAMPER OR VA
Unit 021 : TYPE 001 FAN OR PUMP
Unit 022 : TYPE 002 CONDUIT (DUC
Unit 023 : TYPE 002 CONDUIT (DUC
Unit 024 : TYPE 015 ROOM MODEL
.32 C
0 T
Temperature T
1 C
250 T
.5 T
.25 T
250 P
30 R
30 R
0 Q
Heat Flow into Wall Q
.18 C
.015 H
UP

210

C.5.1. Initial values for various units

	Properties of Inlet Conduit (Duct or Pipe)	
Variables	**Description**	**Value**
Input Variables	Inlet fluid pressure	250
	Output fluid pressure	227.6
	Inlet fluid temperature	5
	Ambient air temperature	25
	Outlet fluid temperature	5.013
Output Variables	Outlet fluid temperature	5.013
	Mass flow rate of fluid	1.1142
Parameters	Inside heat transfer coefficient by area (KW/C)	2.97
	Outside heat transfer coefficient by area (KW/C)	0.0031
	Thermal capacitance of conduit material (KJ/C)	2.277
	Volume (m^3)	0.00251
	Flow resistance [0.001/(Kg.m)]	21.36
	Height of outlet above inlet (m)	-0.5588
	Mode: 2-Water, 1-Air	2

Figure C.5.1-1. *Example 5: Initial values for unit 1*

	Properties of Inlet Conduit (Duct or Pipe)	
Variables	**Description**	**Value**
Input Variables	Inlet fluid pressure	250
	Output fluid pressure	227.6
	Inlet fluid temperature	5
	Ambient air temperature	25
	Outlet fluid temperature	5
Output Variables	Outlet fluid temperature	5
	Mass flow rate of fluid	1.491
Parameters	Inside heat transfer coefficient by area (KW/C)	0.276
	Outside heat transfer coefficient by area (KW/C)	0.00029
	Thermal capacitance of conduit material (KJ/C)	0.2116
	Volume (m^3)	0.00023
	Flow resistance [0.001/(Kg.m)]	14.8
	Height of outlet above inlet (m)	-0.5588
	Mode: 2-Water, 1-Air	2

Figure C.5.1-2. *Example 5: Initial values for unit 2*

	Properties of Conduit (Duct or Pipe)	
Variables	**Description**	**Value**
Input Variables	Mass flow rate of fluid	1.142
	Outlet pressure	212.1
	Inlet temperature of fluid	16.79
	Ambient temperature	25
	Outlet fluid temperature	16.8
Output Variables	Outlet fluid temperature	16.8
	Inlet pressure	227.6
Parameters	Inside heat transfer coefficient by area	2.7
	Outside heat transfer coefficient by area	0.00281
	Thermal capacitance of conduit material	2.07
	Volume	0.002278
	Flow resistance	11.9
	Height of outlet above inlet	0
	Mode: 2-Water, 1-Air	2

Figure C.5.1-3. *Example 5: Initial values for unit 3*

Properties of Three Way Valve Model

Variables	Description	Value
Input Variables	Inlet mass flow rate (used when control signals are 1)	1.142
	Inlet mass flow rate (used when control signals are 0)	1.491
	Outlet pressure	210.4
	Inlet temperature (used when control signals are 1)	16.8
	Inlet temperature (used when control signals are 0)	5
	Control variable from controller	0.428
	Actuator relative position	0.428
Output Variables	Actuator relative position	0.428
	Outlet mass flow rate	2.633
	Inlet pressure (used when control signals are 1)	212.1
	Inlet pressure (used when control signals are 0)	222.6
	Outlet temperature	10.12
	Value stem relative position	0
Parameters	Flow resistance (used when control signals are 1)	0.3
	Flow resistance (used when control signals are 0)	1.5
	Leakage parameter	0.004
	Actuator time constant (Seconds)	10
	Hysteresis parameter	0.1
	Control signals Mode: 0->Lin/Lin, 1->Exp/Lin, else->Exp/Exp	0

Figure C.5.1-4. *Example 5: Initial values for unit 4*

Properties of Conduit (Duct or Pipe)

Variables	Description	Value
Input Variables	Mass flow rate of fluid	2.633
	Outlet pressure	-250
	· Inlet temperature of fluid	10.12
	Ambient temperature	25
	Outlet fluid temperature	10.18
Output Variables	Outlet fluid temperature	10.18
	Inlet pressure	210.4
Parameters	Inside heat transfer coefficient by area	38
	Outside heat transfer coefficient by area	0.044
	Thermal capacitance of conduit material	32.2
	Volume	0.0355
	Flow resistance	66.44
	Height of outlet above inlet	0
	Mode: 2-Water, 1-Air	2

Figure C.5.1-5. *Example 5: Initial values for unit 5*

Properties of Temperature Sensor

Variables	Description	Value
Input Variables	Inlet temperature	20
	Sensor output	0.22
Output Variables	Sensor output	0.22
Parameters	Sensor time constant (Seconds)	20
	Temperature offset (C)	0
	Temperature gain (C)	100

Figure C.5.1-6. *Example 5: Initial values for unit 6*

Properties of Proportional-Integral Controller		
Variables	**Description**	**Value**
Input Variables	Controlled variable	0.22
	Set point for controlled variable	0.18
	Integral portion of control signal	0.572
	Output control signal	0.572
Output Variables	Integral portion of control signal	0.572
	Output control signal	0.572
Parameters	Proportional gain	6
	Integral gain	0.08
	Controller time constant	2

Figure C.5.1-7. *Example 5: Initial values for unit 7*

Properties of Cooling Or Dehumidifying Coil		
Variables	**Description**	**Value**
Input Variables	Water mass flow rate	1.142
	Inlet water temperature	5.013
	Dry air mass flow rate	2.14
	Inlet dry bulb air temperature	32
	Inlet air humidity	0.015
	Outlet water temperature	16.79
	Outlet air temperature	20
	Outlet air humidity	0.005
Output Variables	Outlet water temperature	16.79
	Outlet air dry bulb temperature	20
	Outlet air humidity	0.005
	Total cooling load	56.3
	Sensible cooling load	24
	Wet fraction of coil surface area	1
Parameters	Coil type: 0=flat continuous fins, 1=circular fins	0
	Primary (tube exterior) surface area (m2)	6.2381
	Secondary (fin) surface area (m2)	101.715
	Internal surface area (m2)	6.2
	Ratio of minimum air flow area to face area	0.416
	Fin material thermal conductivity [KW/(m k)]	0.204
	Coil face area (m2)	0.7226
	Number of fins per centimeter	5.5118
	Number of circuits	16
	Number of tubes per circuit	6
	Outside tube diameter (m)	0.01588
	Inside tube diameter (m)	0.01445
	Fin thickness (m)	0.00015
	Thermal capacitance (mass x specific heat) of coil (kj/k)	34.669
	Tube row spacing in air flow direction (m)	0.0259
	Fin diameter [if par(1)=1] or length [if par(1)=0] (m)	0.508
	Coil depth in air flow direction (m)	0.1651
	Tube thermal conductivity [kw/(m k)]	0.386

Figure C.5.1-8. *Example 5: Initial values for unit 8*

Properties of Control Signal Inverter		
Variables	**Description**	**Value**
Input Variables	Input control	0.572
Output Variables	Output control	0.428
Parameters	Multiplier	1

Figure C.5.1-9. *Example 5: Initial values for unit 9*

Properties of Fan Or Pump		
Variables	**Description**	**Value**
Input Variables	Mass flow rate of fluid	2.14
	Outlet pressure	0.0902
	Fan or pump rotational speed	30
	Inlet fluid temperature	27
Output Variables	Inlet pressure	0.399
	Outlet fluid temperature	28
	Power consumption	9.2
Parameters	1st pressure coefficient	3.64
	2nd pressure coefficient	0.801
	3rd pressure coefficient	-0.19
	4th pressure coefficient	-0.00445
	5th pressure coefficient	0
	1st efficiency coefficient	0
	2nd efficiency coefficient	0.564
	3rd efficiency coefficient	-0.0862
	4th efficiency coefficient	0
	5th efficiency coefficient	0
	Diameter (m)	0.3192
	Mode: Air=1, Water=2	1

Figure C.5.1-10. *Example 5: Initial values for TYPE 1 unit 10*

Properties of Conduit (Duct or Pipe)		
Variables	**Description**	**Value**
Input Variables	Mass flow rate of fluid	2.14
	Outlet pressure	0.0444
	Inlet temperature of fluid	28
	Ambient temperature	28
	Outlet fluid temperature	28
Output Variables	Outlet fluid temperature	0
	Inlet pressure	0.0902
Parameters	Inside heat transfer coefficient by area	0
	Outside heat transfer coefficient by area	0
	Thermal capacitance of conduit material	0
	Volume	0
	Flow resistance	0.01
	Height of outlet above inlet	0
	Mode: 2-Water, 1-Air	1

Figure C.5.1-11. *Example 5: Initial values for unit 11*

Properties of Flow Split		
Variables	**Description**	**Value**
Input Variables	Inlet mass flow rate	2.14
	Output pressure 1	0.0176
	Outlet pressure 2	0.00508
Output Variables	Outlet mass flow rate 1	0
	Outlet mass flow rate 2	1.6
	Inlet pressure	0.0444
Parameters	Flow resistance	0.011

Figure C.5.1-12. *Example 5: Initial values for unit 12*

Properties of Damper

Variables	**Description**	**Value**
Input Variables	Fluid mass flow rate	0.536
	Output pressure	0
	Control: relative position of damper or valve	0.32
Output Variables	Inlet pressure	0.0176
Parameters	Flow resistance, damper or valve open	0.001
	Leakage parameter	0.012
	Characteristic: 0=>exp., 1=>lin., Intermediate => intermed.	0.87
	Mode: 0=>closed when control=0; 1=>closed when control=1	0

Figure C.5.1-13. *Example 5: Initial values for unit 13*

Properties of Damper

Variables	**Description**	**Value**
Input Variables	Fluid mass flow rate	1.6
	Output pressure	-0.00537
	Control: relative position of damper or valve	0.32
Output Variables	Inlet pressure	0.00508
Parameters	Flow resistance, damper or valve open	0.001
	Leakage parameter	0.012
	Characteristic: 0=>exp., 1=>lin., Intermediate => intermed.	0.87
	Mode: 0=>closed when control=0; 1=>closed when control=1	1

Figure C.5.1-14. *Example 5: Initial values for unit 14*

Properties of Flow Merge

Variables	**Description**	**Value**
Input Variables	Inlet mass flow rate 1	1.6
	Inlet mass flow rate 2	0.536
	Outlet pressure	-0.0447
	Inlet temperature	28
	Inlet temperature	0
Output Variables	Outlet mass flow rate	2.14
	Inlet pressure 1	-0.00537
	Inlet pressure 2	-0.0179
	Outlet pressure	32
Parameters	Flow resistance	0.011

Figure C.5.1-15. *Example 5: Initial values for unit 15*

Properties of Damper

Variables	**Description**	**Value**
Input Variables	Fluid mass flow rate	0.536
	Output pressure	-0.0179
	Control: relative position of damper or valve	0.32
Output Variables	Inlet pressure	-0.000288
Parameters	Flow resistance, damper or valve open	0.001
	Leakage parameter	0.012
	Characteristic: 0=>exp., 1=>lin., Intermediate => intermed.	0.87
	Mode: 0=>closed when control=0; 1=>closed when control=1	0

Figure C.5.1-16. *Example 5: Initial values for unit 16*

	Properties of Inlet Conduit (Duct or Pipe)	
Variables	**Description**	**Value**
Input Variables	Inlet fluid pressure	0
	Output fluid pressure	-0.000288
	Inlet fluid temperature	0
	Ambient air temperature	0
	Outlet fluid temperature	0
Output Variables	Outlet fluid temperature	0
	Mass flow rate of fluid	0.536
Parameters	Inside heat transfer coefficient by area (KW/C)	0
	Outside heat transfer coefficient by area (KW/C)	0
	Thermal capacitance of conduit material (KJ/C)	0
	Volume (m^3)	0
	Flow resistance [0.001/(Kg.m)]	0.001
	Height of outlet above inlet (m)	0
	Mode: 2-Water, 1-Air	1

Figure C.5.1-17. *Example 5: Initial values for unit 17*

	Properties of Damper	
Variables	**Description**	**Value**
Input Variables	Fluid mass flow rate	2.14
	Output pressure	-0.228
	Control: relative position of damper or valve	1
Output Variables	Inlet pressure	-0.0447
Parameters	Flow resistance, damper or valve open	0.04
	Leakage parameter	0.00004
	Characteristic: 0=>exp., 1=>lin., Intermediate => intermed.	1
	Mode: 0=>closed when control=0; 1=>closed when control=1	0

Figure C.5.1-18. *Example 5: Initial values for unit 18*

	Properties of Damper	
Variables	**Description**	**Value**
Input Variables	Fluid mass flow rate	2.14
	Output pressure	-0.283
	Control: relative position of damper or valve	1
Output Variables	Inlet pressure	-0.228
Parameters	Flow resistance, damper or valve open	0.012
	Leakage parameter	0.000012
	Characteristic: 0=>exp., 1=>lin., Intermediate => intermed.	1
	Mode: 0=>closed when control=0; 1=>closed when control=1	0

Figure C.5.1-19. *Example 5: Initial values for unit 19*

	Properties of Damper	
Variables	**Description**	**Value**
Input Variables	Fluid mass flow rate	2.14
	Output pressure	-0.338
	Control: relative position of damper or valve	1
Output Variables	Inlet pressure	-0.283
Parameters	Flow resistance, damper or valve open	0.012
	Leakage parameter	0.000012
	Characteristic: 0=>exp., 1=>lin., Intermediate => intermed.	1
	Mode: 0=>closed when control=0; 1=>closed when control=1	0

Figure C.5.1-20. *Example 5: Initial values for unit 20*

Properties of Fan Or Pump		
Variables	**Description**	**Value**
Input Variables	Mass flow rate of fluid	2.14
	Outlet pressure	0.151
	Fan or pump rotational speed	30
	Inlet fluid temperature	20
Output Variables	Inlet pressure	-0.338
	Outlet fluid temperature	22
	Power consumption	9.2
Parameters	1st pressure coefficient	3.64
	2nd pressure coefficient	0.801
	3rd pressure coefficient	-0.19
	4th pressure coefficient	-0.00445
	5th pressure coefficient	0
	1st efficiency coefficient	0
	2nd efficiency coefficient	0.564
	3rd efficiency coefficient	-0.0862
	4th efficiency coefficient	0
	5th efficiency coefficient	0
	Diameter (m)	0.3192
	Mode: Air=1, Water=2	1

Figure C.5.1-21. *Example 5: Initial values for TYPE 1 unit 21*

Properties of Conduit (Duct or Pipe)		
Variables	**Description**	**Value**
Input Variables	Mass flow rate of fluid	2.14
	Outlet pressure	-0.124
	Inlet temperature of fluid	22
	Ambient temperature	25
	Outlet fluid temperature	20
Output Variables	Outlet fluid temperature	20
	Inlet pressure	0.151
Parameters	Inside heat transfer coefficient by area	1
	Outside heat transfer coefficient by area	0.25
	Thermal capacitance of conduit material	220
	Volume	7.75
	Flow resistance	0.06
	Height of outlet above inlet	0
	Mode: 2-Water, 1-Air	1

Figure C.5.1-22. *Example 5: Initial values for unit 22*

Properties of Room Model		
Variables	**Description**	**Value**
Input Variables	Mass flow rate of ventilation air	2.14
	Ventilation air inlet temperature	20
	Temperature of fully mixed portion of air mass	20
	Wall mass temperature	20
	Interior mass temperature	20
	Spatial average temperature of the piston flow portion of the air mass	20
	Conduction heat flow into wall mass	0
	Heat flow due to internal gains	0
Output Variables	Temperature of fully mixed portion of air mass	20
	Wall mass temperature	20
	Interior mass temperature	20
	Spatial average temperature of the piston flow portion of the air mass	20
	Average room air temperature	20
	Exhaust air temperature	20
Parameters	Volume of room air mass	1440
	Thermal capacitance of walls	40000
	Thermal capacitance of interior mass	4000
	Heat transfer coefficient times area for the wall mass	1
	Heat transfer coefficient times area for the interior mass	0.2
	Fraction of the air mass which is fully mixed	0.9

Figure C.5.1-23. *Example 5: Initial values for unit 23*

Properties of Conduit (Duct or Pipe)		
Variables	**Description**	**Value**
Input Variables	Mass flow rate of fluid	2.14
	Outlet pressure	0.399
	Inlet temperature of fluid	20
	Ambient temperature	25
	Outlet fluid temperature	27
Output Variables	Outlet fluid temperature	27
	Inlet pressure	-0.124
Parameters	Inside heat transfer coefficient by area	1
	Outside heat transfer coefficient by area	0.25
	Thermal capacitance of conduit material	220
	Volume	7.75
	Flow resistance	0.06
	Height of outlet above inlet	0
	Mode: 2-Water, 1-Air	1

Figure C.5.1-24. *Example 5: Initial values for unit 24*

APPENDIX D

SMART BRIDGE SIMULATIONS

D.1. Smart Bridge System – Design Procedure 1

D.1.1. Boundary values

Time	t_DB	t_Sky	Humidity ratio	Wind Speed	Wind direction	Solar radiation	Snowfall	Rainfall	Solar angle	Mass flow rate
0	-9.4	-9.4	0.004	6.25	180	0	0	0	1.6	0
3600	-9.4	-9.4	0.004	6.25	180	0	0	0	1.6	0
7200	-9.4	-9.4	0.004	6.25	180	0	0	0	1.6	0
10800	-9.4	-9.4	0.004	6.25	180	0	0	0	1.6	0
14400	-9.4	-9.4	0.004	6.25	180	0	0	0	1.6	0
18000	-9.4	-9.4	0.004	6.25	180	0	0	0	1.6	0
21600	-9.4	-9.4	0.004	6.25	180	0	0	0	1.6	0
25200	-9.4	-9.4	0.004	6.25	180	0	0	0	1.6	0
28800	-9.4	-9.4	0.004	6.25	180	0	0	0	1.6	0
32400	-9.4	-9.4	0.004	6.25	180	0	0	0	1.6	0
36000	-9.4	-9.4	0.004	6.25	180	0	0	0	1.6	0
39600	-9.4	-9.4	0.004	6.25	180	0	0	0	1.6	0
43200	-9.4	-9.4	0.004	6.25	180	0	0	0	1.6	0
46800	-9.4	-9.4	0.004	6.25	180	0	0	0	1.6	0
50400	-9.4	-9.4	0.004	6.25	180	0	0	0	1.6	0
54000	-9.4	-9.4	0.004	6.25	180	0	0	0	1.6	0
57600	-9.4	-9.4	0.004	6.25	180	0	0	0	1.6	0
61200	-9.4	-9.4	0.004	6.25	180	0	0	0	1.6	0
64800	-9.4	-9.4	0.004	6.25	180	0	0	0	1.6	0
68400	-9.4	-9.4	0.004	6.25	180	0	0	0	1.6	0
72000	-9.4	-9.4	0.004	6.25	180	0	0	0	1.6	0
75600	-9.4	-9.4	0.004	6.25	180	0	0	0	1.6	0
79200	-9.4	-9.4	0.004	6.25	180	0	0	0	1.6	0
82800	-9.4	-9.4	0.004	6.25	180	0	0	0	1.6	0
86400	-9.4	-9.4	0.004	6.25	180	0	0	0	1.6	0
90000	-9.4	-9.4	0.004	6.25	180	0	0	0	1.6	0
93600	-9.4	-9.4	0.004	6.25	180	0	0	0	1.6	0
97200	-9.4	-9.4	0.004	6.25	180	0	0	0	1.6	0
100800	-9.4	-9.4	0.004	6.25	180	0	0	0	1.6	0
104400	-9.4	-9.4	0.004	6.25	180	0	0	0	1.6	0
108000	-9.4	-9.4	0.004	6.25	180	0	0	0	1.6	0
111600	-9.4	-9.4	0.004	6.25	180	0	0	0	1.6	0

115200	-9.4	-9.4	0.004	6.25	180	0	0	0	1.6	0
118800	-9.4	-9.4	0.004	6.25	180	0	0	0	1.6	0
122400	-9.4	-9.4	0.004	6.25	180	0	0	0	1.6	0
126000	-9.4	-9.4	0.004	6.25	180	0	0	0	1.6	0
129600	-9.4	-9.4	0.004	6.25	180	0	0	0	1.6	0
133200	-9.4	-9.4	0.004	6.25	180	0	0	0	1.6	0
136800	-9.4	-9.4	0.004	6.25	180	0	0	0	1.6	0
140400	-9.4	-9.4	0.004	6.25	180	0	0	0	1.6	0
144000	-9.4	-9.4	0.004	6.25	180	0	0	0	1.6	0
147600	-9.4	-9.4	0.004	6.25	180	0	0	0	1.6	0
151200	-9.4	-9.4	0.004	6.25	180	0	0	0	1.6	0
154800	-9.4	-9.4	0.004	6.25	180	0	0	0	1.6	0
158400	-9.4	-9.4	0.004	6.25	180	0	0	0	1.6	0
162000	-9.4	-9.4	0.004	6.25	180	0	0	0	1.6	0
165600	-9.4	-9.4	0.004	6.25	180	0	0	0	1.6	0
169200	-9.4	-9.4	0.004	6.25	180	0	0	0	1.6	0
169200	-9.4	-9.4	0.004	6.25	180	0	1.04	0	1.6	22.7315
172800	-9.4	-9.4	0.004	6.25	180	0	1.04	0	1.6	22.7315
176400	-9.4	-9.4	0.004	6.25	180	0	1.04	0	1.6	22.7315
180000	-9.4	-9.4	0.004	6.25	180	0	1.04	0	1.6	22.7315
183600	-9.4	-9.4	0.004	6.25	180	0	1.04	0	1.6	22.7315
187200	-9.4	-9.4	0.004	6.25	180	0	1.04	0	1.6	22.7315
190800	-9.4	-9.4	0.004	6.25	180	0	1.04	0	1.6	22.7315
194400	-9.4	-9.4	0.004	6.25	180	0	1.04	0	1.6	22.7315
198000	-9.4	-9.4	0.004	6.25	180	0	1.04	0	1.6	22.7315
201600	-9.4	-9.4	0.004	6.25	180	0	1.04	0	1.6	22.7315
205200	-9.4	-9.4	0.004	6.25	180	0	1.04	0	1.6	22.7315
208800	-9.4	-9.4	0.004	6.25	180	0	1.04	0	1.6	22.7315
212400	-9.4	-9.4	0.004	6.25	180	0	1.04	0	1.6	22.7315
216000	-9.4	-9.4	0.004	6.25	180	0	1.04	0	1.6	22.7315
219600	-9.4	-9.4	0.004	6.25	180	0	1.04	0	1.6	22.7315
223200	-9.4	-9.4	0.004	6.25	180	0	1.04	0	1.6	22.7315
226800	-9.4	-9.4	0.004	6.25	180	0	1.04	0	1.6	22.7315
230400	-9.4	-9.4	0.004	6.25	180	0	1.04	0	1.6	22.7315
234000	-9.4	-9.4	0.004	6.25	180	0	1.04	0	1.6	22.7315
237600	-9.4	-9.4	0.004	6.25	180	0	1.04	0	1.6	22.7315
241200	-9.4	-9.4	0.004	6.25	180	0	1.04	0	1.6	22.7315
244800	-9.4	-9.4	0.004	6.25	180	0	1.04	0	1.6	22.7315
248400	-9.4	-9.4	0.004	6.25	180	0	1.04	0	1.6	22.7315
252000	-9.4	-9.4	0.004	6.25	180	0	1.04	0	1.6	22.7315
255600	-9.4	-9.4	0.004	6.25	180	0	1.04	0	1.6	22.7315
259200	-9.4	-9.4	0.004	6.25	180	0	1.04	0	1.6	22.7315
262800	-9.4	-9.4	0.004	6.25	180	0	1.04	0	1.6	22.7315
266400	-9.4	-9.4	0.004	6.25	180	0	1.04	0	1.6	22.7315
270000	-9.4	-9.4	0.004	6.25	180	0	1.04	0	1.6	22.7315
273600	-9.4	-9.4	0.004	6.25	180	0	1.04	0	1.6	22.7315
277200	-9.4	-9.4	0.004	6.25	180	0	1.04	0	1.6	22.7315
280800	-9.4	-9.4	0.004	6.25	180	0	1.04	0	1.6	22.7315
284400	-9.4	-9.4	0.004	6.25	180	0	1.04	0	1.6	22.7315
288000	-9.4	-9.4	0.004	6.25	180	0	1.04	0	1.6	22.7315

291600	-9.4	-9.4	0.004	6.25	180	0	1.04	0	1.6	22.7315
295200	-9.4	-9.4	0.004	6.25	180	0	1.04	0	1.6	22.7315
298800	-9.4	-9.4	0.004	6.25	180	0	1.04	0	1.6	22.7315
302400	-9.4	-9.4	0.004	6.25	180	0	1.04	0	1.6	22.7315
306000	-9.4	-9.4	0.004	6.25	180	0	1.04	0	1.6	22.7315
309600	-9.4	-9.4	0.004	6.25	180	0	1.04	0	1.6	22.7315
313200	-9.4	-9.4	0.004	6.25	180	0	1.04	0	1.6	22.7315
316800	-9.4	-9.4	0.004	6.25	180	0	1.04	0	1.6	22.7315
320400	-9.4	-9.4	0.004	6.25	180	0	1.04	0	1.6	22.7315
324000	-9.4	-9.4	0.004	6.25	180	0	1.04	0	1.6	22.7315
327600	-9.4	-9.4	0.004	6.25	180	0	1.04	0	1.6	22.7315
331200	-9.4	-9.4	0.004	6.25	180	0	1.04	0	1.6	22.7315
334800	-9.4	-9.4	0.004	6.25	180	0	1.04	0	1.6	22.7315
338400	-9.4	-9.4	0.004	6.25	180	0	1.04	0	1.6	22.7315

D.1.2. Listing of simulation output file

```
SUPERBLOCK 1  50400.00
 -9.40000   -9.40000   -9.40000   -9.40000    0.00000
  0.00000    1.00000    0.00000    0.00000    0.00000
SUPERBLOCK 1  54000.00
 -9.40000   -9.40000   -9.40000   -9.40000    0.00000
  0.00000    1.00000    0.00000    0.00000    0.00000
SUPERBLOCK 1  57600.00
 -9.40000   -9.40000   -9.40000   -9.40000    0.00000
  0.00000    1.00000    0.00000    0.00000    0.00000
SUPERBLOCK 1  61200.00
 -9.40000   -9.40000   -9.40000   -9.40000    0.00000
  0.00000    1.00000    0.00000    0.00000    0.00000
SUPERBLOCK 1  64800.00
 -9.40000   -9.40000   -9.40000   -9.40000    0.00000
  0.00000    1.00000    0.00000    0.00000    0.00000
SUPERBLOCK 1  68400.00
 -9.40000   -9.40000   -9.40000   -9.40000    0.00000
  0.00000    1.00000    0.00000    0.00000    0.00000
SUPERBLOCK 1  72000.00
 -9.40000   -9.40000   -9.40000   -9.40000    0.00000
  0.00000    1.00000    0.00000    0.00000    0.00000
SUPERBLOCK 1  75600.00
 -9.40000   -9.40000   -9.40000   -9.40000    0.00000
  0.00000    1.00000    0.00000    0.00000    0.00000
SUPERBLOCK 1  79200.00
 -9.40000   -9.40000   -9.40000   -9.40000    0.00000
  0.00000    1.00000    0.00000    0.00000    0.00000
SUPERBLOCK 1  82800.00
 -9.40000   -9.40000   -9.40000   -9.40000    0.00000
  0.00000    1.00000    0.00000    0.00000    0.00000
SUPERBLOCK 1  86400.00
 -9.40000   -9.40000   -9.40000   -9.40000    0.00000
  0.00000    1.00000    0.00000    0.00000    0.00000
SUPERBLOCK 1  90000.00
 -9.40000   -9.40000   -9.40000   -9.40000    0.00000
  0.00000    1.00000    0.00000    0.00000    0.00000
SUPERBLOCK 1  93600.00
 -9.40000   -9.40000   -9.40000   -9.40000    0.00000
  0.00000    1.00000    0.00000    0.00000    0.00000
SUPERBLOCK 1  97200.00
 -9.40000   -9.40000   -9.40000   -9.40000    0.00000
  0.00000    1.00000    0.00000    0.00000    0.00000
SUPERBLOCK 1 100800.00
 -9.40000   -9.40000   -9.40000   -9.40000    0.00000
  0.00000    1.00000    0.00000    0.00000    0.00000
SUPERBLOCK 1 104400.00
 -9.40000   -9.40000   -9.40000   -9.40000    0.00000
  0.00000    1.00000    0.00000    0.00000    0.00000
SUPERBLOCK 1 108000.00
 -9.40000   -9.40000   -9.40000   -9.40000    0.00000
  0.00000    1.00000    0.00000    0.00000    0.00000
SUPERBLOCK 1 111600.00
 -9.40000   -9.40000   -9.40000   -9.40000    0.00000
  0.00000    1.00000    0.00000    0.00000    0.00000
SUPERBLOCK 1 115200.00
 -9.40000   -9.40000   -9.40000   -9.40000    0.00000
  0.00000    1.00000    0.00000    0.00000    0.00000
SUPERBLOCK 1 118800.00
 -9.40000   -9.40000   -9.40000   -9.40000    0.00000
  0.00000    1.00000    0.00000    0.00000    0.00000
SUPERBLOCK 1 122400.00
 -9.40000   -9.40000   -9.40000   -9.40000    0.00000
  0.00000    1.00000    0.00000    0.00000    0.00000
SUPERBLOCK 1 126000.00
 -9.40000   -9.40000   -9.40000   -9.40000    0.00000
  0.00000    1.00000    0.00000    0.00000    0.00000
SUPERBLOCK 1 129600.00
 -9.40000   -9.40000   -9.40000   -9.40000    0.00000
  0.00000    1.00000    0.00000    0.00000    0.00000
SUPERBLOCK 1 133200.00
 -9.40000   -9.40000   -9.40000   -9.40000    0.00000
  0.00000    1.00000    0.00000    0.00000    0.00000
SUPERBLOCK 1 136800.00
 -9.40000   -9.40000   -9.40000   -9.40000    0.00000
  0.00000    1.00000    0.00000    0.00000    0.00000
SUPERBLOCK 1 140400.00
 -9.40000   -9.40000   -9.40000   -9.40000    0.00000
  0.00000    1.00000    0.00000    0.00000    0.00000
SUPERBLOCK 1 144000.00
 -9.40000   -9.40000   -9.40000   -9.40000    0.00000
  0.00000    1.00000    0.00000    0.00000    0.00000
SUPERBLOCK 1 147600.00
 -9.40000   -9.40000   -9.40000   -9.40000    0.00000
  0.00000    1.00000    0.00000    0.00000    0.00000
SUPERBLOCK 1 151200.00
 -9.40000   -9.40000   -9.40000   -9.40000    0.00000
  0.00000    1.00000    0.00000    0.00000    0.00000
SUPERBLOCK 1 154800.00
```

```
  -9.40000    -9.40000    -9.40000    -9.40000    0.00000
   0.00000     1.00000     0.00000     0.00000    0.00000
SUPERBLOCK 1 158400.00
  -9.40000    -9.40000    -9.40000    -9.40000    0.00000
   0.00000     1.00000     0.00000     0.00000    0.00000
SUPERBLOCK 1 162000.00
  -9.40000    -9.40000    -9.40000    -9.40000    0.00000
   0.00000     1.00000     0.00000     0.00000    0.00000
SUPERBLOCK 1 165600.00
  -9.40000    -9.40000    -9.40000    -9.40000    0.00000
   0.00000     1.00000     0.00000     0.00000    0.00000
SUPERBLOCK 1 169200.00
  -9.40000    -9.40000    -9.40000    -9.40000    0.00000
   0.00000     1.00000     0.00000     0.00000    0.00000
SUPERBLOCK 1 172800.00
   33.7774    -4.92820     21.8612     21.8678    1.04200
   0.00000    8.333334E-02   1001.25    1025.00    2.74093
SUPERBLOCK 1 176400.00
   38.3746    -2.49688     26.4584     26.4649    1.04200
   0.00000    0.250000     1005.44     1025.00    2.74733
SUPERBLOCK 1 180000.00
   41.2089   -0.739332     29.2927     29.2992    1.04200
   0.00000    0.333333     1008.01     1025.00    2.75142
SUPERBLOCK 1 183600.00
   43.2967    0.469255     31.3806     31.3871    1.04200
   0.00000    0.416667     1009.91     1025.00    2.75452
SUPERBLOCK 1 187200.00
   44.9724     1.42089     33.0562     33.0627    1.04200
   0.00000    0.416667     1011.44     1025.00    2.75705
SUPERBLOCK 1 190800.00
   46.2578     2.04544     34.3416     34.3481    1.04200
   0.00000    0.500000     1012.61     1025.00    2.75901
SUPERBLOCK 1 194400.00
   47.3057     2.54896     35.3895     35.3960    1.04200
   0.00000    0.500000     1013.57     1025.00    2.76064
SUPERBLOCK 1 198000.00
   48.0894     2.77771     36.1733     36.1798    1.04200
   0.00000    0.583333     1014.28     1025.00    2.76186
SUPERBLOCK 1 201600.00
   48.7527     3.12853     36.8366     36.8431    1.04200
   0.00000    0.583333     1014.89     1025.00    2.76291
SUPERBLOCK 1 205200.00
   49.2510     3.26023     37.3348     37.3414    1.04200
   0.00000    0.583333     1015.34     1025.00    2.76370
SUPERBLOCK 1 208800.00
   49.6160     3.35327     37.6998     37.7063    1.04200
   0.00000    0.583333     1015.67     1025.00    2.76428
SUPERBLOCK 1 212400.00
   49.9200     3.53820     38.0038     38.0103    1.04200
   0.00000    0.666667     1015.95     1025.00    2.76476
SUPERBLOCK 1 216000.00
   50.1753     3.67846     38.2591     38.2657    1.04200
   0.00000    0.666667     1016.19     1025.00    2.76517
SUPERBLOCK 1 219600.00
   50.3685     3.73825     38.4522     38.4589    1.04200
   0.00000    0.666667     1016.37     1025.00    2.76548
SUPERBLOCK 1 223200.00
   50.5120     3.78119     38.5959     38.6024    1.04200
   0.00000    0.666667     1016.49     1025.00    2.76571
SUPERBLOCK 1 226800.00
   50.6199     3.81328     38.7037     38.7102    1.04200
   0.00000    0.666667     1016.59     1025.00    2.76588
SUPERBLOCK 1 230400.00
   50.7060     3.85561     38.7899     38.7964    1.04200
   0.00000    0.750000     1016.67     1025.00    2.76588
SUPERBLOCK 1 234000.00
   50.7947     3.99408     38.8784     38.8850    1.04200
   0.00000    0.750000     1016.76     1025.00    2.76616
SUPERBLOCK 1 237600.00
   50.8727     4.02488     38.9565     38.9630    1.04200
   0.00000    0.750000     1016.82     1025.00    2.76616
SUPERBLOCK 1 241200.00
   50.9281     4.04499     39.0118     39.0184    1.04200
   0.00000    0.750000     1016.88     1025.00    2.76638
SUPERBLOCK 1 244800.00
   50.9697     4.05954     39.0535     39.0600    1.04200
   0.00000    0.750000     1016.91     1025.00    2.76638
SUPERBLOCK 1 248400.00
   51.0017     4.07053     39.0855     39.0920    1.04200
   0.00000    0.750000     1016.94     1025.00    2.76638
SUPERBLOCK 1 252000.00
   51.0268     4.07898     39.1106     39.1171    1.04200
   0.00000    0.750000     1016.96     1025.00    2.76654
SUPERBLOCK 1 255600.00
   51.0463     4.08551     39.1302     39.1367    1.04200
   0.00000    0.750000     1016.98     1025.00    2.76654
SUPERBLOCK 1 259200.00
   51.0601     4.09046     39.1439     39.1504    1.04200
   0.00000    0.750000     1016.99     1025.00    2.76654
SUPERBLOCK 1 262800.00
   51.0719     4.09434     39.1557     39.1623    1.04200
   0.00000    0.750000     1017.01     1025.00    2.76654
SUPERBLOCK 1 266400.00
   51.0812     4.09737     39.1650     39.1715    1.04200
   0.00000    0.750000     1017.01     1025.00    2.76654
SUPERBLOCK 1 270000.00
   51.0941     4.15340     39.1780     39.1845    1.04200
   0.00000    0.833333     1017.02     1025.00    2.76654
SUPERBLOCK 1 273600.00
   51.1149     4.19431     39.1987     39.2052    1.04200
   0.00000    0.833333     1017.05     1025.00    2.76654
SUPERBLOCK 1 277200.00
   51.1319     4.20348     39.2157     39.2223    1.04200
   0.00000    0.833333     1017.06     1025.00    2.76671
SUPERBLOCK 1 280800.00
   51.1453     4.20918     39.2291     39.2356    1.04200
   0.00000    0.833333     1017.08     1025.00    2.76671
SUPERBLOCK 1 284400.00
   51.1553     4.21320     39.2391     39.2457    1.04200
   0.00000    0.833333     1017.08     1025.00    2.76671
SUPERBLOCK 1 288000.00
   51.1631     4.21622     39.2468     39.2534    1.04200
   0.00000    0.833333     1017.10     1025.00    2.76671
SUPERBLOCK 1 291600.00
   51.1684     4.21849     39.2522     39.2587    1.04200
   0.00000    0.833333     1017.09     1025.00    2.76671
SUPERBLOCK 1 295200.00
   51.1740     4.22035     39.2577     39.2644    1.04200
   0.00000    0.833333     1017.11     1025.00    2.76671
SUPERBLOCK 1 298800.00
   51.1778     4.22181     39.2615     39.2682    1.04200
   0.00000    0.833333     1017.12     1025.00    2.76671
SUPERBLOCK 1 302400.00
   51.1778     4.22275     39.2635     39.2682    1.04200
   0.00000    0.833333     1016.95     1025.00    2.76671
SUPERBLOCK 1 306000.00
   51.1839     4.22380     39.2671     39.2743    1.04200
   0.00000    0.833333     1017.16     1025.00    2.76671
SUPERBLOCK 1 309600.00
   51.1861     4.22464     39.2699     39.2764    1.04200
   0.00000    0.833333     1017.11     1025.00    2.76671
SUPERBLOCK 1 313200.00
   51.1861     4.22516     39.2715     39.2764    1.04200
   0.00000    0.833333     1016.97     1025.00    2.76671
SUPERBLOCK 1 316800.00
   51.1861     4.22550     39.2722     39.2780    1.04200
   0.00000    0.833333     1016.92     1025.00    2.76671
SUPERBLOCK 1 320400.00
   51.1934     4.22625     39.2755     39.2837    1.04200
   0.00000    0.833333     1017.26     1025.00    2.76671
```

```
SUPERBLOCK 1 324000.00
  51.1944     4.22679    39.2779    39.2847    1.04200
  0.00000     0.833333   1017.14    1025.00    2.76671
SUPERBLOCK 1 327600.00
  51.1944     4.22710    39.2789    39.2847    1.04200
  0.00000     0.833333   1017.06    1025.00    2.76671
SUPERBLOCK 1 331200.00
  51.1944     1.89321    39.2793    39.2847    1.04200
  0.00000     0.583333   1017.02    1025.00    2.76671
SUPERBLOCK 1 334800.00
  50.8271     3.81992    38.9109    38.9175    1.04200
  0.00000     0.833333   1016.79    1025.00    2.76622
SUPERBLOCK 1 338400.00
  51.0369     4.14963    39.1209    39.1273    1.04200
  0.00000     0.833333   1016.97    1025.00    2.76655
**************************************************
Smart Bridge System Design Procedure Step 1
5/12/2002 7:25:02 PM
```

```
G:\Smart Bridge\Design Procedure Step 1.dat
1
10
Unit 1, TYPE 700: INLET FLUID TEMPERATURE (C)
Unit 1, TYPE 700: AVERAGE PAVEMENT TOP SURFACE
TEMPERATURE (C)
Unit 1, TYPE 700: OUTLET FLUID TEMPERATURE (C)
Unit 2, TYPE 740: INLET FLUID TEMP (C)
Unit 1, TYPE 700: SNOWFALL RATE IN WATER
EQUIVALENT PER HR (mm/hr)
Unit 1, TYPE 700: RAINFALL RATE IN WATER
EQUIVALENT PER HR (mm/hr)
Unit 1, TYPE 700: SNOW FREE AREA RATIO (-)
Unit 1, TYPE 700: HEAT PROVIDED TO PAVEMENT
SURFACE (kW)
Unit 2, TYPE 740: HEAT OUTPUT (kW)
Unit 3, TYPE 750: POWER CONSUMPTION (kW)
```

D.2. Smart Bridge System – Design Procedure 2

D.2.1. Boundary values

Time	t_DB	t_Sky	Humidity ratio	Wind Speed	Wind direction	Solar radiation	Snowfall	Rainfall	Solar angle	Snow Flag
0	-9.4	-9.4	0.004	6.25	180	0	0	0	1.57	0
3600	-9.4	-9.4	0.004	6.25	180	0	0	0	1.57	0
7200	-9.4	-9.4	0.004	6.25	180	0	0	0	1.57	0
10800	-9.4	-9.4	0.004	6.25	180	0	0	0	1.57	0
14400	-9.4	-9.4	0.004	6.25	180	0	0	0	1.57	0
18000	-9.4	-9.4	0.004	6.25	180	0	0	0	1.57	0
21600	-9.4	-9.4	0.004	6.25	180	0	0	0	1.57	0
25200	-9.4	-9.4	0.004	6.25	180	0	0	0	1.57	0
28800	-9.4	-9.4	0.004	6.25	180	0	0	0	1.57	0
32400	-9.4	-9.4	0.004	6.25	180	0	0	0	1.57	0
36000	-9.4	-9.4	0.004	6.25	180	0	0	0	1.57	0
39600	-9.4	-9.4	0.004	6.25	180	0	0	0	1.57	0
43200	-9.4	-9.4	0.004	6.25	180	0	0	0	1.57	0
46800	-9.4	-9.4	0.004	6.25	180	0	0	0	1.57	0
50400	-9.4	-9.4	0.004	6.25	180	0	0	0	1.57	0
54000	-9.4	-9.4	0.004	6.25	180	0	0	0	1.57	0
57600	-9.4	-9.4	0.004	6.25	180	0	0	0	1.57	0
61200	-9.4	-9.4	0.004	6.25	180	0	0	0	1.57	0
64800	-9.4	-9.4	0.004	6.25	180	0	0	0	1.57	0
68400	-9.4	-9.4	0.004	6.25	180	0	0	0	1.57	0
72000	-9.4	-9.4	0.004	6.25	180	0	0	0	1.57	0
75600	-9.4	-9.4	0.004	6.25	180	0	0	0	1.57	0
79200	-9.4	-9.4	0.004	6.25	180	0	0	0	1.57	0
82800	-9.4	-9.4	0.004	6.25	180	0	0	0	1.57	0
86400	-9.4	-9.4	0.004	6.25	180	0	0	0	1.57	0
90000	-9.4	-9.4	0.004	6.25	180	0	0	0	1.57	0
93600	-9.4	-9.4	0.004	6.25	180	0	0	0	1.57	0

97200	-9.4	-9.4	0.004	6.25	180	0	0	0	1.57	0
100800	-9.4	-9.4	0.004	6.25	180	0	0	0	1.57	0
104400	-9.4	-9.4	0.004	6.25	180	0	0	0	1.57	0
108000	-9.4	-9.4	0.004	6.25	180	0	0	0	1.57	0
111600	-9.4	-9.4	0.004	6.25	180	0	0	0	1.57	0
115200	-9.4	-9.4	0.004	6.25	180	0	0	0	1.57	0
118800	-9.4	-9.4	0.004	6.25	180	0	0	0	1.57	0
122400	-9.4	-9.4	0.004	6.25	180	0	0	0	1.57	0
126000	-9.4	-9.4	0.004	6.25	180	0	0	0	1.57	0
129600	-9.4	-9.4	0.004	6.25	180	0	0	0	1.57	0
133200	-9.4	-9.4	0.004	6.25	180	0	0	0	1.57	0
136800	-9.4	-9.4	0.004	6.25	180	0	0	0	1.57	0
140400	-9.4	-9.4	0.004	6.25	180	0	0	0	1.57	0
144000	-9.4	-9.4	0.004	6.25	180	0	0	0	1.57	0
147600	-9.4	-9.4	0.004	6.25	180	0	0	0	1.57	0
151200	-9.4	-9.4	0.004	6.25	180	0	0	0	1.57	0
154800	-9.4	-9.4	0.004	6.25	180	0	0	0	1.57	0
158400	-9.4	-9.4	0.004	6.25	180	0	0	0	1.57	0
162000	-9.4	-9.4	0.004	6.25	180	0	0	0	1.57	0
165600	-9.4	-9.4	0.004	6.25	180	0	0	0	1.57	0
169200	-9.4	-9.4	0.004	6.25	180	0	0	0	1.57	0
169200	-9.4	-9.4	0.004	6.25	180	0	1.042	0	1.57	1
172800	-9.4	-9.4	0.004	6.25	180	0	1.042	0	1.57	1
176400	-9.4	-9.4	0.004	6.25	180	0	1.042	0	1.57	1
180000	-9.4	-9.4	0.004	6.25	180	0	1.042	0	1.57	1
183600	-9.4	-9.4	0.004	6.25	180	0	1.042	0	1.57	1
187200	-9.4	-9.4	0.004	6.25	180	0	1.042	0	1.57	1
190800	-9.4	-9.4	0.004	6.25	180	0	1.042	0	1.57	1
194400	-9.4	-9.4	0.004	6.25	180	0	1.042	0	1.57	1
198000	-9.4	-9.4	0.004	6.25	180	0	1.042	0	1.57	1
201600	-9.4	-9.4	0.004	6.25	180	0	1.042	0	1.57	1
205200	-9.4	-9.4	0.004	6.25	180	0	1.042	0	1.57	1
208800	-9.4	-9.4	0.004	6.25	180	0	1.042	0	1.57	1
212400	-9.4	-9.4	0.004	6.25	180	0	1.042	0	1.57	1
216000	-9.4	-9.4	0.004	6.25	180	0	1.042	0	1.57	1
219600	-9.4	-9.4	0.004	6.25	180	0	1.042	0	1.57	1
223200	-9.4	-9.4	0.004	6.25	180	0	1.042	0	1.57	1
226800	-9.4	-9.4	0.004	6.25	180	0	1.042	0	1.57	1
230400	-9.4	-9.4	0.004	6.25	180	0	1.042	0	1.57	1
234000	-9.4	-9.4	0.004	6.25	180	0	1.042	0	1.57	1
237600	-9.4	-9.4	0.004	6.25	180	0	1.042	0	1.57	1
241200	-9.4	-9.4	0.004	6.25	180	0	1.042	0	1.57	1
244800	-9.4	-9.4	0.004	6.25	180	0	1.042	0	1.57	1
248400	-9.4	-9.4	0.004	6.25	180	0	1.042	0	1.57	1
252000	-9.4	-9.4	0.004	6.25	180	0	1.042	0	1.57	1
255600	-9.4	-9.4	0.004	6.25	180	0	1.042	0	1.57	1
259200	-9.4	-9.4	0.004	6.25	180	0	1.042	0	1.57	1
262800	-9.4	-9.4	0.004	6.25	180	0	1.042	0	1.57	1
266400	-9.4	-9.4	0.004	6.25	180	0	1.042	0	1.57	1
270000	-9.4	-9.4	0.004	6.25	180	0	1.042	0	1.57	1

273600	-9.4	-9.4	0.004	6.25	180	0	1.042	0	1.57	1
277200	-9.4	-9.4	0.004	6.25	180	0	1.042	0	1.57	1
280800	-9.4	-9.4	0.004	6.25	180	0	1.042	0	1.57	1
284400	-9.4	-9.4	0.004	6.25	180	0	1.042	0	1.57	1
288000	-9.4	-9.4	0.004	6.25	180	0	1.042	0	1.57	1
291600	-9.4	-9.4	0.004	6.25	180	0	1.042	0	1.57	1
295200	-9.4	-9.4	0.004	6.25	180	0	1.042	0	1.57	1
298800	-9.4	-9.4	0.004	6.25	180	0	1.042	0	1.57	1
302400	-9.4	-9.4	0.004	6.25	180	0	1.042	0	1.57	1
306000	-9.4	-9.4	0.004	6.25	180	0	1.042	0	1.57	1
309600	-9.4	-9.4	0.004	6.25	180	0	1.042	0	1.57	1
313200	-9.4	-9.4	0.004	6.25	180	0	1.042	0	1.57	1
316800	-9.4	-9.4	0.004	6.25	180	0	1.042	0	1.57	1
320400	-9.4	-9.4	0.004	6.25	180	0	1.042	0	1.57	1
324000	-9.4	-9.4	0.004	6.25	180	0	1.042	0	1.57	1
327600	-9.4	-9.4	0.004	6.25	180	0	1.042	0	1.57	1
331200	-9.4	-9.4	0.004	6.25	180	0	1.042	0	1.57	1
334800	-9.4	-9.4	0.004	6.25	180	0	1.042	0	1.57	1
338400	-9.4	-9.4	0.004	6.25	180	0	1.042	0	1.57	1

D.2.2. Listing of the simulation output file

```
SUPERBLOCK 1  3600.00
  17.5350    0.00000
SUPERBLOCK 2  3600.00
  17.5350   -8.92532   -4.71572   -4.71176    0.00000
  0.00000    1.00000   107.968    1.24363
SUPERBLOCK 1  7200.00
  17.5350    0.00000
SUPERBLOCK 2  7200.00
  17.5350   -8.59046   -4.55412   -4.55017    0.00000
  0.00000    1.00000   97.8069    1.24370
SUPERBLOCK 1  10800.00
  17.5350    0.00000
SUPERBLOCK 2  10800.00
  17.5350   -8.38302   -4.46754   -4.46359    0.00000
  0.00000    1.00000   92.3618    1.24370
SUPERBLOCK 1  14400.00
  17.5350    0.00000
SUPERBLOCK 2  14400.00
  17.5350   -8.23793   -4.40788   -4.40392    0.00000
  0.00000    1.00000   88.6092    1.24377
SUPERBLOCK 1  18000.00
  17.5350    0.00000
SUPERBLOCK 2  18000.00
  17.5350   -8.13017   -4.36360   -4.35965    0.00000
  0.00000    1.00000   85.8244    1.24377
SUPERBLOCK 1  21600.00
  17.5350    0.00000
SUPERBLOCK 2  21600.00
  17.5350   -8.04796   -4.32949   -4.32554    0.00000
  0.00000    1.00000   83.6788    1.24377
SUPERBLOCK 1  25200.00
  17.5350    0.00000
SUPERBLOCK 2  25200.00
  17.5350   -7.98451   -4.30285   -4.29889    0.00000
  0.00000    1.00000   82.0030    1.24377
SUPERBLOCK 1  28800.00
  17.5350    0.00000
SUPERBLOCK 2  28800.00
  17.5350   -7.93526   -4.28216   -4.27820    0.00000
  0.00000    1.00000   80.7016    1.24377
SUPERBLOCK 1  32400.00
  17.5350    0.00000
SUPERBLOCK 2  32400.00
  17.5350   -7.89694   -4.26640   -4.26244    0.00000
  0.00000    1.00000   79.7100    1.24384
SUPERBLOCK 1  36000.00
  17.5350    0.00000
SUPERBLOCK 2  36000.00
  17.5350   -7.86710   -4.25362   -4.24966    0.00000
  0.00000    1.00000   78.9062    1.24384
SUPERBLOCK 1  39600.00
  17.5350    0.00000
SUPERBLOCK 2  39600.00
  17.5350   -7.84383   -4.24429   -4.24033    0.00000
  0.00000    1.00000   78.3192    1.24384
SUPERBLOCK 1  43200.00
  17.5350    0.00000
SUPERBLOCK 2  43200.00
  17.5350   -7.82570   -4.23669   -4.23274    0.00000
  0.00000    1.00000   77.8415    1.24384
SUPERBLOCK 1  46800.00
  17.5350    0.00000
SUPERBLOCK 2  46800.00
  17.5350   -7.81156   -4.23065   -4.22670    0.00000
  0.00000    1.00000   77.4615    1.24384
SUPERBLOCK 1  50400.00
  17.5350    0.00000
SUPERBLOCK 2  50400.00
  17.5350   -7.80052   -4.22576   -4.22180    0.00000
  0.00000    1.00000   77.1536    1.24384
SUPERBLOCK 1  54000.00
  17.5350    0.00000
SUPERBLOCK 2  54000.00
  17.5350   -7.79191   -4.22214   -4.21819    0.00000
  0.00000    1.00000    76.9262    1.24384
SUPERBLOCK 1  57600.00
  17.5350    0.00000
SUPERBLOCK 2  57600.00
  17.5350    -7.78519   -4.21954   -4.21559   0.00000
  0.00000    1.00000    76.7627    1.24384
SUPERBLOCK 1  61200.00
  17.5350    0.00000
SUPERBLOCK 2  61200.00
  17.5350    -7.77994   -4.21673   -4.21277   0.00000
  0.00000    1.00000    76.5855    1.24384
SUPERBLOCK 1  64800.00
  17.5350    0.00000
SUPERBLOCK 2  64800.00
  17.5350    -7.77583   -4.21447   -4.21052   0.00000
  0.00000    1.00000    76.4437    1.24384
SUPERBLOCK 1  68400.00
  17.5350    0.00000
SUPERBLOCK 2  68400.00
  17.5350    -7.77262   -4.21271   -4.20875   0.00000
  0.00000    1.00000    76.3325    1.24384
SUPERBLOCK 1  72000.00
  17.5350    0.00000
SUPERBLOCK 2  72000.00
  17.5350    -7.77012   -4.21132   -4.20737   0.00000
  0.00000    1.00000    76.2455    1.24384
SUPERBLOCK 1  75600.00
  17.5350    0.00000
SUPERBLOCK 2  75600.00
  17.5350    -7.76816   -4.21024   -4.20629   0.00000
  0.00000    1.00000    76.1776    1.24384
SUPERBLOCK 1  79200.00
  17.5350    0.00000
SUPERBLOCK 2  79200.00
  17.5350    -7.76663   -4.20940   -4.20544   0.00000
  0.00000    1.00000    76.1245    1.24384
SUPERBLOCK 1  82800.00
  17.5350    0.00000
SUPERBLOCK 2  82800.00
  17.5350    -7.76543   -4.20874   -4.20479   0.00000
  0.00000    1.00000    76.0830    1.24384
SUPERBLOCK 1  86400.00
  17.5350    0.00000
SUPERBLOCK 2  86400.00
  17.5350    -7.76450   -4.20822   -4.20427   0.00000
  0.00000    1.00000    76.0506    1.24384
SUPERBLOCK 1  90000.00
  17.5350    0.00000
SUPERBLOCK 2  90000.00
  17.5350    -7.76377   -4.20782   -4.20387   0.00000
  0.00000    1.00000    76.0253    1.24384
SUPERBLOCK 1  93600.00
  17.5350    0.00000
SUPERBLOCK 2  93600.00
  17.5350    -7.76320   -4.20751   -4.20355   0.00000
  0.00000    1.00000    76.0055    1.24384
SUPERBLOCK 1  97200.00
  17.5350    0.00000
SUPERBLOCK 2  97200.00
  17.5350    -7.76275   -4.20726   -4.20331   0.00000
  0.00000    1.00000    75.9901    1.24384
SUPERBLOCK 1 100800.00
  17.5350    0.00000
SUPERBLOCK 2 100800.00
  17.5350    -7.76241   -4.20707   -4.20331   0.00000
  0.00000    1.00000    75.9780    1.24384
SUPERBLOCK 1 104400.00
  17.5350    0.00000
SUPERBLOCK 2 104400.00
  17.5350    -7.76213   -4.20692   -4.20297   0.00000
  0.00000    1.00000    75.9685    1.24384
SUPERBLOCK 1 108000.00
  17.5350    0.00000
SUPERBLOCK 2 108000.00
  17.5350    -7.76192   -4.20680   -4.20285   0.00000
  0.00000    1.00000    75.9612    1.24384
SUPERBLOCK 1 111600.00
  17.5350    0.00000
SUPERBLOCK 2 111600.00
  17.5350    -7.76175   -4.20671   -4.20285   0.00000
  0.00000    1.00000    75.9554    1.24384
SUPERBLOCK 1 115200.00
  17.5350    0.00000
SUPERBLOCK 2 115200.00
  17.5350    -7.76162   -4.20664   -4.20285   0.00000
  0.00000    1.00000    75.9509    1.24384
SUPERBLOCK 1 118800.00
  17.5350    0.00000
SUPERBLOCK 2 118800.00
  17.5350    -7.76152   -4.20658   -4.20263   0.00000
  0.00000    1.00000    75.9474    1.24384
SUPERBLOCK 1 122400.00
  17.5350    0.00000
SUPERBLOCK 2 122400.00
  17.5350    -7.76144   -4.20654   -4.20258   0.00000
  0.00000    1.00000    75.9446    1.24384
SUPERBLOCK 1 126000.00
  17.5350    0.00000
SUPERBLOCK 2 126000.00
  17.5350    -7.76138   -4.20651   -4.20258   0.00000
  0.00000    1.00000    75.9425    1.24384
SUPERBLOCK 1 129600.00
  17.5350    0.00000
SUPERBLOCK 2 129600.00
  17.5350    -7.76133   -4.20648   -4.20258   0.00000
  0.00000    1.00000    75.9408    1.24384
SUPERBLOCK 1 133200.00
  17.5350    0.00000
SUPERBLOCK 2 133200.00
  17.5350    -7.76129   -4.20646   -4.20258   0.00000
  0.00000    1.00000    75.9394    1.24384
SUPERBLOCK 1 136800.00
  17.5350    0.00000
SUPERBLOCK 2 136800.00
  17.5350    -7.76126   -4.20644   -4.20258   0.00000
  0.00000    1.00000    75.9384    1.24384
SUPERBLOCK 1 140400.00
  17.5350    0.00000
SUPERBLOCK 2 140400.00
  17.5350    -7.76124   -4.20643   -4.20258   0.00000
  0.00000    1.00000    75.9376    1.24384
SUPERBLOCK 1 144000.00
  17.5350    0.00000
SUPERBLOCK 2 144000.00
  17.5350    -7.76122   -4.20642   -4.20258   0.00000
  0.00000    1.00000    75.9369    1.24384
SUPERBLOCK 1 147600.00
  17.5350    0.00000
SUPERBLOCK 2 147600.00
  17.5350    -7.76121   -4.20641   -4.20258   0.00000
  0.00000    1.00000    75.9364    1.24384
SUPERBLOCK 1 151200.00
  17.5350    0.00000
SUPERBLOCK 2 151200.00
  17.5350    -7.76121   -4.20641   -4.20258   0.00000
  0.00000    1.00000    75.9362    1.24384
SUPERBLOCK 1 154800.00
  17.5350    0.00000
SUPERBLOCK 2 154800.00
  17.5350    -7.76121   -4.20640   -4.20258   0.00000
  0.00000     1.00000    75.9361     1.24384
SUPERBLOCK 1 158400.00
  17.5350     0.00000
SUPERBLOCK 2 158400.00
  17.5350    -7.76121    -4.20640    -4.20258    0.00000
  0.00000     1.00000    75.9361     1.24384
SUPERBLOCK 1 162000.00
  17.5350     0.00000
SUPERBLOCK 2 162000.00
  17.5350    -7.76121    -4.20640    -4.20258    0.00000
  0.00000     1.00000    75.9361     1.24384
SUPERBLOCK 1 165600.00
  17.5350     0.00000
SUPERBLOCK 2 165600.00
  17.5350    -7.76121    -4.20640    -4.20258    0.00000
  0.00000     1.00000    75.9361     1.24384
SUPERBLOCK 1 169200.00
  17.5350     0.00000
SUPERBLOCK 2 169200.00
  17.5350    -7.76121    -4.20640    -4.20258    0.00000
  0.00000     1.00000    75.9361     1.24384
SUPERBLOCK 1 172800.00
  58.4500     8.00000
SUPERBLOCK 2 172800.00
  22.7315    -3.75847    24.3288     24.3354     1.04200
  0.00000    0.166667    1027.96     2.74432
SUPERBLOCK 1 176400.00
  58.4500     8.00000
SUPERBLOCK 2 176400.00
  22.7315    -1.44163    28.6451     28.6516     1.04200
  0.00000    0.333333    1032.01     2.75048
SUPERBLOCK 1 180000.00
  58.4500     8.00000
SUPERBLOCK 2 180000.00
  22.7315    6.833338E-02  31.2745    31.2810
1.04200
  0.00000    0.416667    1034.33     2.75436
SUPERBLOCK 1 183600.00
  58.4500     8.00000
SUPERBLOCK 2 183600.00
  22.7315     1.24014    33.2426.    33.2492     1.04200
  0.00000    0.416667    1035.99     2.75733
SUPERBLOCK 1 187200.00
  58.4500     5.00000
SUPERBLOCK 2 187200.00
  22.7315    0.444254    22.6158     22.6223     1.04200
  0.00000    0.416667    640.284     2.74196
SUPERBLOCK 1 190800.00
  58.4500     7.00000
SUPERBLOCK 2 190800.00
  22.7315     1.32203    30.1037     30.1102     1.04200
  0.00000    0.500000    903.596     2.75262
SUPERBLOCK 1 194400.00
  58.4500     5.00000
SUPERBLOCK 2 194400.00
  22.7315    0.817246    23.1491     23.1556     1.04200
  0.00000    0.500000    640.618     2.74269
SUPERBLOCK 1 198000.00
  58.4500     6.00000
SUPERBLOCK 2 198000.00
  22.7315     1.26366    26.7924     26.7989     1.04200
  0.00000    0.500000    771.806     2.74780
SUPERBLOCK 1 201600.00
  58.4500     5.00000
SUPERBLOCK 2 201600.00
  22.7315    0.914991    23.1862     23.1927     1.04200
  0.00000    0.500000    640.641     2.74274
SUPERBLOCK 1 205200.00
  58.4500     6.00000
SUPERBLOCK 2 205200.00
```

```
  22.7315     1.31543    26.9072     26.9138     1.04200
  0.00000    0.500000    771.886     2.74797
SUPERBLOCK 1 208800.00
  58.4500     5.00000
SUPERBLOCK 2 208800.00
  22.7315    0.952808    23.2851     23.2917     1.04200
  0.00000    0.500000    640.702     2.74288
SUPERBLOCK 1 212400.00
  58.4500     6.00000
SUPERBLOCK 2 212400.00
  22.7315     1.34731    26.9900     26.9966     1.04200
  0.00000    0.500000    771.944     2.74809
SUPERBLOCK 1 216000.00
  58.4500     5.00000
SUPERBLOCK 2 216000.00
  22.7315    0.945884    23.3138     23.3256     1.04200
  0.00000    0.500000    641.167     2.74292
SUPERBLOCK 1 219600.00
  58.4500     6.00000
SUPERBLOCK 2 219600.00
  22.7315     1.36076    27.0276     27.0341     1.04200
  0.00000    0.500000    771.971     2.74814
SUPERBLOCK 1 223200.00
  58.4500     5.00000
SUPERBLOCK 2 223200.00
  22.7315    0.956088    23.3318     23.3384     1.04200
  0.00000    0.500000    640.734     2.74294
SUPERBLOCK 1 226800.00
  58.4500     6.00000
SUPERBLOCK 2 226800.00
  22.7315     1.36985    27.0515     27.0581     1.04200
  0.00000    0.500000    771.986     2.74817
SUPERBLOCK 1 230400.00
  58.4500     5.00000
SUPERBLOCK 2 230400.00
  22.7315    0.962922    23.3501     23.3567     1.04200
  0.00000    0.500000    640.744     2.74297
SUPERBLOCK 1 234000.00
  58.4500     6.00000
SUPERBLOCK 2 234000.00
  22.7315     1.37548    27.0663     27.0729     1.04200
  0.00000    0.500000    771.997     2.74820
SUPERBLOCK 1 237600.00
  58.4500     5.00000
SUPERBLOCK 2 237600.00
  22.7315    0.967027    23.3611     23.3676     1.04200
  0.00000    0.500000    640.751     2.74298
SUPERBLOCK 1 241200.00
  58.4500     6.00000
SUPERBLOCK 2 241200.00
  22.7315     1.37884    27.0751     27.0817     1.04200
  0.00000    0.500000    772.004     2.74821
SUPERBLOCK 1 244800.00
  58.4500     5.00000
SUPERBLOCK 2 244800.00
  22.7315    0.969467    23.3676     23.3742     1.04200
  0.00000    0.500000    640.755     2.74299
SUPERBLOCK 1 248400.00
  58.4500     6.00000
SUPERBLOCK 2 248400.00
  22.7315     1.38082    27.0803     27.0869     1.04200
  0.00000    0.500000    772.007     2.74822
SUPERBLOCK 1 252000.00
  58.4500     5.00000
SUPERBLOCK 2 252000.00
  22.7315    0.970910    23.3715     23.3781     1.04200
  0.00000    0.500000    640.757     2.74300
SUPERBLOCK 1 255600.00
  58.4500     6.00000
SUPERBLOCK 2 255600.00
```

```
  22.7315    1.38200    27.0834    27.0899    1.04200
  0.00000    0.500000   772.009    2.74822
SUPERBLOCK 1 259200.00
  58.4500    5.00000
SUPERBLOCK 2 259200.00
  22.7315    0.971765   23.3738    23.3803    1.04200
  0.00000    0.500000   640.759    2.74300
SUPERBLOCK 1 262800.00
  58.4500    6.00000
SUPERBLOCK 2 262800.00
  22.7315    1.38270    27.0852    27.0918    1.04200
  0.00000    0.500000   772.011    2.74822
SUPERBLOCK 1 266400.00
  58.4500    5.00000
SUPERBLOCK 2 266400.00
  22.7315    0.972276   23.3752    23.3817    1.04200
  0.00000    0.500000   640.761    2.74300
SUPERBLOCK 1 270000.00
  58.4500    6.00000
SUPERBLOCK 2 270000.00
  22.7315    1.38311    27.0863    27.0929    1.04200
  0.00000    0.500000   772.012    2.74822
SUPERBLOCK 1 273600.00
  58.4500    5.00000
SUPERBLOCK 2 273600.00
  22.7315    0.972576   23.3760    23.3825    1.04200
  0.00000    0.500000   640.761    2.74300
SUPERBLOCK 1 277200.00
  58.4500    6.00000
SUPERBLOCK 2 277200.00
  22.7315    1.38336    27.0870    27.0935    1.04200
  0.00000    0.500000   772.013    2.74823
SUPERBLOCK 1 280800.00
  58.4500    5.00000
SUPERBLOCK 2 280800.00
  22.7315    0.972753   23.3764    23.3830    1.04200
  0.00000    0.500000   640.760    2.74300
SUPERBLOCK 1 284400.00
  58.4500    6.00000
SUPERBLOCK 2 284400.00
  22.7315    1.38351    27.0874    27.0939    1.04200
  0.00000    0.500000   772.013    2.74823
SUPERBLOCK 1 288000.00
  58.4500    5.00000
SUPERBLOCK 2 288000.00
  22.7315    0.972862   23.3767    23.3833    1.04200
  0.00000    0.500000   640.762    2.74300
SUPERBLOCK 1 291600.00
  58.4500    6.00000
SUPERBLOCK 2 291600.00
  22.7315    1.38359    27.0876    27.0941    1.04200
  0.00000    0.500000   772.012    2.74823
SUPERBLOCK 1 295200.00
  58.4500    5.00000
SUPERBLOCK 2 295200.00
  22.7315    0.972924   23.3769    23.3835    1.04200
  0.00000    0.500000   640.762    2.74300
SUPERBLOCK 1 298800.00
  58.4500    6.00000
SUPERBLOCK 2 298800.00
  22.7315    1.38364    27.0877    27.0942    1.04200
  0.00000    0.500000   772.012    2.74823
SUPERBLOCK 1 302400.00
  58.4500    5.00000
SUPERBLOCK 2 302400.00
  22.7315    0.972959   23.3770    23.3836    1.04200
  0.00000    0.500000   640.761    2.74300
SUPERBLOCK 1 306000.00
  58.4500    6.00000
SUPERBLOCK 2 306000.00
  22.7315    1.38367    27.0878    27.0943    1.04200
  0.00000    0.500000   772.013    2.74823
SUPERBLOCK 1 309600.00
  58.4500    5.00000
SUPERBLOCK 2 309600.00
  22.7315    0.972982   23.3771    23.3836    1.04200
  0.00000    0.500000   640.762    2.74300
SUPERBLOCK 1 313200.00
  58.4500    6.00000
SUPERBLOCK 2 313200.00
  22.7315    1.38369    27.0878    27.0944    1.04200
  0.00000    0.500000   772.012    2.74823
SUPERBLOCK 1 316800.00
  58.4500    5.00000
SUPERBLOCK 2 316800.00
  22.7315    0.972993   23.3771    23.3836    1.04200
  0.00000    0.500000   640.761    2.74300
SUPERBLOCK 1 320400.00
  58.4500    6.00000
SUPERBLOCK 2 320400.00
  22.7315    1.38370    27.0879    27.0944    1.04200
  0.00000    0.500000   772.013    2.74823
SUPERBLOCK 1 324000.00
  58.4500    5.00000
SUPERBLOCK 2 324000.00
  22.7315    0.973086   23.3772    23.3838    1.04200
  0.00000    0.500000   640.762    2.74300
SUPERBLOCK 1 327600.00
  58.4500    6.00000
SUPERBLOCK 2 327600.00
  22.7315    1.38379    27.0879    27.0945    1.04200
  0.00000    0.583333   772.012    2.74823
SUPERBLOCK 1 331200.00
  58.4500    5.00000
SUPERBLOCK 2 331200.00
  22.7315    0.351700   23.4343    23.4409    1.04200
  0.00000    0.500000   640.795    2.74308
SUPERBLOCK 1 334800.00
  58.4500    7.00000
SUPERBLOCK 2 334800.00
  22.7315    1.84117    31.0072    31.0137    1.04200
  0.00000    0.583333   904.283    2.75396
SUPERBLOCK 1 338400.00
  58.4500    3.00000
SUPERBLOCK 2 338400.00
  22.7315    9.425353E-02  16.0791    16.0857    1.04200
  0.00000    0.333333   381.245    2.73330
**************************************************
Smart Bridge System Design Procedure Step 2
5/12/2002 7:26:06 PM
G:\AlphaTesting\Smart Bridge\Design Procedure Step 2.dat
2
2
Unit 3, TYPE 730: Source side mass flow rate (kg/s)
Unit 3, TYPE 730: Number of heat pump pairs in operation (-)
9
Unit 1, TYPE 700: MASS FLOW RATE (kg/S)
Unit 1, TYPE 700: AVERAGE PAVEMENT TOP SURFACE
TEMPERATURE (C)
Unit 1, TYPE 700: OUTLET FLUID TEMPERATURE (C)
Unit 2, TYPE 750: OUTLET TEMPERATURE (C)
Unit 1, TYPE 700: SNOWFALL RATE IN WATER
EQUIVALENT PER HR (mm/hr)
Unit 1, TYPE 700: RAINFALL RATE IN WATER
EQUIVALENT PER HR (mm/hr)
Unit 1, TYPE 700: SNOW FREE AREA RATIO (-)
Unit 1, TYPE 700: HEAT PROVIDED TO PAVEMENT
SURFACE (kW)
Unit 2, TYPE 750: POWER CONSUMPTION (kW)
```

D.3. Smart Bridge System – Design Procedure 3

D.3.1. Boundary values

Time	t_DB	t_Sky	Humidity ratio	Wind Speed	Wind direction	Solar radiation	Snowfall	Rainfall	Solar angle	Snow Flag
0	-9.4	-9.4	0.004	6.25	180	0	0	0	1.57	0
3600	-9.4	-9.4	0.004	6.25	180	0	0	0	1.57	0
7200	-9.4	-9.4	0.004	6.25	180	0	0	0	1.57	0
10800	-9.4	-9.4	0.004	6.25	180	0	0	0	1.57	0
14400	-9.4	-9.4	0.004	6.25	180	0	0	0	1.57	0
18000	-9.4	-9.4	0.004	6.25	180	0	0	0	1.57	0
21600	-9.4	-9.4	0.004	6.25	180	0	0	0	1.57	0
25200	-9.4	-9.4	0.004	6.25	180	0	0	0	1.57	0
28800	-9.4	-9.4	0.004	6.25	180	0	0	0	1.57	0
32400	-9.4	-9.4	0.004	6.25	180	0	0	0	1.57	0
36000	-9.4	-9.4	0.004	6.25	180	0	0	0	1.57	0
39600	-9.4	-9.4	0.004	6.25	180	0	0	0	1.57	0
43200	-9.4	-9.4	0.004	6.25	180	0	0	0	1.57	0
46800	-9.4	-9.4	0.004	6.25	180	0	0	0	1.57	0
50400	-9.4	-9.4	0.004	6.25	180	0	0	0	1.57	0
54000	-9.4	-9.4	0.004	6.25	180	0	0	0	1.57	0
57600	-9.4	-9.4	0.004	6.25	180	0	0	0	1.57	0
61200	-9.4	-9.4	0.004	6.25	180	0	0	0	1.57	0
64800	-9.4	-9.4	0.004	6.25	180	0	0	0	1.57	0
68400	-9.4	-9.4	0.004	6.25	180	0	0	0	1.57	0
72000	-9.4	-9.4	0.004	6.25	180	0	0	0	1.57	0
75600	-9.4	-9.4	0.004	6.25	180	0	0	0	1.57	0
79200	-9.4	-9.4	0.004	6.25	180	0	0	0	1.57	0
82800	-9.4	-9.4	0.004	6.25	180	0	0	0	1.57	0
86400	-9.4	-9.4	0.004	6.25	180	0	0	0	1.57	0
90000	-9.4	-9.4	0.004	6.25	180	0	0	0	1.57	0
93600	-9.4	-9.4	0.004	6.25	180	0	0	0	1.57	0
97200	-9.4	-9.4	0.004	6.25	180	0	0	0	1.57	0
100800	-9.4	-9.4	0.004	6.25	180	0	0	0	1.57	0
104400	-9.4	-9.4	0.004	6.25	180	0	0	0	1.57	0
108000	-9.4	-9.4	0.004	6.25	180	0	0	0	1.57	0
111600	-9.4	-9.4	0.004	6.25	180	0	0	0	1.57	0
115200	-9.4	-9.4	0.004	6.25	180	0	0	0	1.57	0
118800	-9.4	-9.4	0.004	6.25	180	0	0	0	1.57	0
122400	-9.4	-9.4	0.004	6.25	180	0	0	0	1.57	0
126000	-9.4	-9.4	0.004	6.25	180	0	0	0	1.57	0
129600	-9.4	-9.4	0.004	6.25	180	0	0	0	1.57	0
133200	-9.4	-9.4	0.004	6.25	180	0	0	0	1.57	0
136800	-9.4	-9.4	0.004	6.25	180	0	0	0	1.57	0
140400	-9.4	-9.4	0.004	6.25	180	0	0	0	1.57	0

144000	-9.4	-9.4	0.004	6.25	180	0	0	0	1.57	0
147600	-9.4	-9.4	0.004	6.25	180	0	0	0	1.57	0
151200	-9.4	-9.4	0.004	6.25	180	0	0	0	1.57	0
154800	-9.4	-9.4	0.004	6.25	180	0	0	0	1.57	0
158400	-9.4	-9.4	0.004	6.25	180	0	0	0	1.57	0
162000	-9.4	-9.4	0.004	6.25	180	0	0	0	1.57	0
165600	-9.4	-9.4	0.004	6.25	180	0	0	0	1.57	0
169200	-9.4	-9.4	0.004	6.25	180	0	0	0	1.57	0
169200	-9.4	-9.4	0.004	6.25	180	0	1.042	0	1.57	1
172800	-9.4	-9.4	0.004	6.25	180	0	1.042	0	1.57	1
176400	-9.4	-9.4	0.004	6.25	180	0	1.042	0	1.57	1
180000	-9.4	-9.4	0.004	6.25	180	0	1.042	0	1.57	1
183600	-9.4	-9.4	0.004	6.25	180	0	1.042	0	1.57	1
187200	-9.4	-9.4	0.004	6.25	180	0	1.042	0	1.57	1
190800	-9.4	-9.4	0.004	6.25	180	0	1.042	0	1.57	1
194400	-9.4	-9.4	0.004	6.25	180	0	1.042	0	1.57	1
198000	-9.4	-9.4	0.004	6.25	180	0	1.042	0	1.57	1
201600	-9.4	-9.4	0.004	6.25	180	0	1.042	0	1.57	1
205200	-9.4	-9.4	0.004	6.25	180	0	1.042	0	1.57	1
208800	-9.4	-9.4	0.004	6.25	180	0	1.042	0	1.57	1
212400	-9.4	-9.4	0.004	6.25	180	0	1.042	0	1.57	1
216000	-9.4	-9.4	0.004	6.25	180	0	1.042	0	1.57	1
219600	-9.4	-9.4	0.004	6.25	180	0	1.042	0	1.57	1
223200	-9.4	-9.4	0.004	6.25	180	0	1.042	0	1.57	1
226800	-9.4	-9.4	0.004	6.25	180	0	1.042	0	1.57	1
230400	-9.4	-9.4	0.004	6.25	180	0	1.042	0	1.57	1
234000	-9.4	-9.4	0.004	6.25	180	0	1.042	0	1.57	1
237600	-9.4	-9.4	0.004	6.25	180	0	1.042	0	1.57	1
241200	-9.4	-9.4	0.004	6.25	180	0	1.042	0	1.57	1
244800	-9.4	-9.4	0.004	6.25	180	0	1.042	0	1.57	1
248400	-9.4	-9.4	0.004	6.25	180	0	1.042	0	1.57	1
252000	-9.4	-9.4	0.004	6.25	180	0	1.042	0	1.57	1
255600	-9.4	-9.4	0.004	6.25	180	0	1.042	0	1.57	1
259200	-9.4	-9.4	0.004	6.25	180	0	1.042	0	1.57	1
262800	-9.4	-9.4	0.004	6.25	180	0	1.042	0	1.57	1
266400	-9.4	-9.4	0.004	6.25	180	0	1.042	0	1.57	1
270000	-9.4	-9.4	0.004	6.25	180	0	1.042	0	1.57	1
273600	-9.4	-9.4	0.004	6.25	180	0	1.042	0	1.57	1
277200	-9.4	-9.4	0.004	6.25	180	0	1.042	0	1.57	1
280800	-9.4	-9.4	0.004	6.25	180	0	1.042	0	1.57	1
284400	-9.4	-9.4	0.004	6.25	180	0	1.042	0	1.57	1
288000	-9.4	-9.4	0.004	6.25	180	0	1.042	0	1.57	1
291600	-9.4	-9.4	0.004	6.25	180	0	1.042	0	1.57	1
295200	-9.4	-9.4	0.004	6.25	180	0	1.042	0	1.57	1
298800	-9.4	-9.4	0.004	6.25	180	0	1.042	0	1.57	1
302400	-9.4	-9.4	0.004	6.25	180	0	1.042	0	1.57	1
306000	-9.4	-9.4	0.004	6.25	180	0	1.042	0	1.57	1
309600	-9.4	-9.4	0.004	6.25	180	0	1.042	0	1.57	1
313200	-9.4	-9.4	0.004	6.25	180	0	1.042	0	1.57	1
316800	-9.4	-9.4	0.004	6.25	180	0	1.042	0	1.57	1

320400	-9.4	-9.4	0.004	6.25	180	0	1.042	0	1.57	1
324000	-9.4	-9.4	0.004	6.25	180	0	1.042	0	1.57	1
327600	-9.4	-9.4	0.004	6.25	180	0	1.042	0	1.57	1
331200	-9.4	-9.4	0.004	6.25	180	0	1.042	0	1.57	1
334800	-9.4	-9.4	0.004	6.25	180	0	1.042	0	1.57	1
338400	-9.4	-9.4	0.004	6.25	180	0	1.042	0	1.57	1

D.3.2. Listing of simulation output file

```
SUPERBLOCK 1  3600.00
  0.00000    0.00000
SUPERBLOCK 2  3600.00
  0.00000   -9.40000   -9.40000    0.00000    0.00000
  0.00000    0.00000    1.00000    0.00000    0.00000
  0.00000
SUPERBLOCK 1  7200.00
  0.00000    0.00000
SUPERBLOCK 2  7200.00
  0.00000   -9.40000   -9.40000   -9.40000   17.2200
  0.00000    0.00000    1.00000    0.00000    0.00000
  0.00000
SUPERBLOCK 1  10800.00
  0.00000    0.00000
SUPERBLOCK 2  10800.00
  0.00000   -9.40000   -9.40000   -9.40000   17.2200
  0.00000    0.00000    1.00000    0.00000    0.00000
  0.00000
SUPERBLOCK 1  14400.00
  0.00000    0.00000
SUPERBLOCK 2  14400.00
  0.00000   -9.40000   -9.40000   -9.40000   17.2200
  0.00000    0.00000    1.00000    0.00000    0.00000
  0.00000
SUPERBLOCK 1  18000.00
  0.00000    0.00000
SUPERBLOCK 2  18000.00
  0.00000   -9.40000   -9.40000   -9.40000   17.2200
  0.00000    0.00000    1.00000    0.00000    0.00000
  0.00000
SUPERBLOCK 1  21600.00
  0.00000    0.00000
SUPERBLOCK 2  21600.00
  0.00000   -9.40000   -9.40000   -9.40000   17.2200
  0.00000    0.00000    1.00000    0.00000    0.00000
  0.00000
SUPERBLOCK 1  25200.00
  0.00000    0.00000
SUPERBLOCK 2  25200.00
  0.00000   -9.40000   -9.40000   -9.40000   17.2200
  0.00000    0.00000    1.00000    0.00000    0.00000
  0.00000
SUPERBLOCK 1  28800.00
  0.00000    0.00000
SUPERBLOCK 2  28800.00
  0.00000   -9.40000   -9.40000   -9.40000   17.2200
  0.00000    0.00000    1.00000    0.00000    0.00000
  0.00000
SUPERBLOCK 1  32400.00
  0.00000    0.00000
SUPERBLOCK 2  32400.00
  0.00000   -9.40000   -9.40000   -9.40000   17.2200
  0.00000    0.00000    1.00000    0.00000    0.00000
  0.00000
SUPERBLOCK 1  36000.00
  0.00000    0.00000
SUPERBLOCK 2  36000.00
  0.00000   -9.40000   -9.40000   -9.40000   17.2200
  0.00000    0.00000    1.00000    0.00000    0.00000
  0.00000
SUPERBLOCK 1  39600.00
  0.00000    0.00000
SUPERBLOCK 2  39600.00
  0.00000   -9.40000   -9.40000   -9.40000   17.2200
  0.00000    0.00000    1.00000    0.00000    0.00000
  0.00000
SUPERBLOCK 1  43200.00
  0.00000    0.00000
SUPERBLOCK 2  43200.00
  0.00000   -9.40000   -9.40000   -9.40000   17.2200
  0.00000    0.00000    1.00000    0.00000    0.00000
  0.00000
SUPERBLOCK 1  46800.00
  0.00000    0.00000
SUPERBLOCK 2  46800.00
  0.00000   -9.40000   -9.40000   -9.40000   17.2200
  0.00000    0.00000    1.00000    0.00000    0.00000
  0.00000
SUPERBLOCK 1  50400.00
  0.00000    0.00000
SUPERBLOCK 2  50400.00
  0.00000   -9.40000   -9.40000   -9.40000   17.2200
  0.00000    0.00000    1.00000    0.00000    0.00000
  0.00000
SUPERBLOCK 1  54000.00
  0.00000    0.00000
SUPERBLOCK 2  54000.00
  0.00000   -9.40000   -9.40000   -9.40000   17.2200
  0.00000    0.00000    1.00000    0.00000    0.00000
  0.00000
SUPERBLOCK 1  57600.00
  0.00000    0.00000
SUPERBLOCK 2  57600.00
  0.00000   -9.40000   -9.40000   -9.40000   17.2200
  0.00000    0.00000    1.00000    0.00000    0.00000
  0.00000
SUPERBLOCK 1  61200.00
  0.00000    0.00000
SUPERBLOCK 2  61200.00
  0.00000   -9.40000   -9.40000   -9.40000   17.2200
  0.00000    0.00000    1.00000    0.00000    0.00000
  0.00000
SUPERBLOCK 1  64800.00
  0.00000    0.00000
SUPERBLOCK 2  64800.00
  0.00000   -9.40000   -9.40000   -9.40000   17.2200
  0.00000    0.00000    1.00000    0.00000    0.00000
  0.00000
SUPERBLOCK 1  68400.00
  0.00000    0.00000
```

```
SUPERBLOCK 2  68400.00
  0.00000   -9.40000   -9.40000   -9.40000    17.2200
  0.00000    0.00000    1.00000    0.00000    0.00000
  0.00000
SUPERBLOCK 1  72000.00
  0.00000    0.00000
SUPERBLOCK 2  72000.00
  0.00000   -9.40000   -9.40000   -9.40000    17.2200
  0.00000    0.00000    1.00000    0.00000    0.00000
  0.00000
SUPERBLOCK 1  75600.00
  0.00000    0.00000
SUPERBLOCK 2  75600.00
  0.00000   -9.40000   -9.40000   -9.40000    17.2200
  0.00000    0.00000    1.00000    0.00000    0.00000
  0.00000
SUPERBLOCK 1  79200.00
  0.00000    0.00000
SUPERBLOCK 2  79200.00
  0.00000   -9.40000   -9.40000   -9.40000    17.2200
  0.00000    0.00000    1.00000    0.00000    0.00000
  0.00000
SUPERBLOCK 1  82800.00
  0.00000    0.00000
SUPERBLOCK 2  82800.00
  0.00000   -9.40000   -9.40000   -9.40000    17.2200
  0.00000    0.00000    1.00000    0.00000    0.00000
  0.00000
SUPERBLOCK 1  86400.00
  0.00000    0.00000
SUPERBLOCK 2  86400.00
  0.00000   -9.40000   -9.40000   -9.40000    17.2200
  0.00000    0.00000    1.00000    0.00000    0.00000
  0.00000
SUPERBLOCK 1  90000.00
  0.00000    0.00000
SUPERBLOCK 2  90000.00
  0.00000   -9.40000   -9.40000   -9.40000    17.2200
  0.00000    0.00000    1.00000    0.00000    0.00000
  0.00000
SUPERBLOCK 1  93600.00
  0.00000    0.00000
SUPERBLOCK 2  93600.00
  0.00000   -9.40000   -9.40000   -9.40000    17.2200
  0.00000    0.00000    1.00000    0.00000    0.00000
  0.00000
SUPERBLOCK 1  97200.00
  0.00000    0.00000
SUPERBLOCK 2  97200.00
  0.00000   -9.40000   -9.40000   -9.40000    17.2200
  0.00000    0.00000    1.00000    0.00000    0.00000
  0.00000
SUPERBLOCK 1 100800.00
  0.00000    0.00000
SUPERBLOCK 2 100800.00
  0.00000   -9.40000   -9.40000   -9.40000    17.2200
  0.00000    0.00000    1.00000    0.00000    0.00000
  0.00000
SUPERBLOCK 1 104400.00
  0.00000    0.00000
SUPERBLOCK 2 104400.00
  0.00000   -9.40000   -9.40000   -9.40000    17.2200
  0.00000    0.00000    1.00000    0.00000    0.00000
  0.00000
SUPERBLOCK 1 108000.00
  0.00000    0.00000
SUPERBLOCK 2 108000.00
  0.00000   -9.40000   -9.40000   -9.40000    17.2200
  0.00000    0.00000    1.00000    0.00000    0.00000
  0.00000
```

```
SUPERBLOCK 1 111600.00
  0.00000    0.00000
SUPERBLOCK 2 111600.00
  0.00000   -9.40000   -9.40000   -9.40000    17.2200
  0.00000    0.00000    1.00000    0.00000    0.00000
  0.00000
SUPERBLOCK 1 115200.00
  0.00000    0.00000
SUPERBLOCK 2 115200.00
  0.00000   -9.40000   -9.40000   -9.40000    17.2200
  0.00000    0.00000    1.00000    0.00000    0.00000
  0.00000
SUPERBLOCK 1 118800.00
  0.00000    0.00000
SUPERBLOCK 2 118800.00
  0.00000   -9.40000   -9.40000   -9.40000    17.2200
  0.00000    0.00000    1.00000    0.00000    0.00000
  0.00000
SUPERBLOCK 1 122400.00
  0.00000    0.00000
SUPERBLOCK 2 122400.00
  0.00000   -9.40000   -9.40000   -9.40000    17.2200
  0.00000    0.00000    1.00000    0.00000    0.00000
  0.00000
SUPERBLOCK 1 126000.00
  0.00000    0.00000
SUPERBLOCK 2 126000.00
  0.00000   -9.40000   -9.40000   -9.40000    17.2200
  0.00000    0.00000    1.00000    0.00000    0.00000
  0.00000
SUPERBLOCK 1 129600.00
  0.00000    0.00000
SUPERBLOCK 2 129600.00
  0.00000   -9.40000   -9.40000   -9.40000    17.2200
  0.00000    0.00000    1.00000    0.00000    0.00000
  0.00000
SUPERBLOCK 1 133200.00
  0.00000    0.00000
SUPERBLOCK 2 133200.00
  0.00000   -9.40000   -9.40000   -9.40000    17.2200
  0.00000    0.00000    1.00000    0.00000    0.00000
  0.00000
SUPERBLOCK 1 136800.00
  0.00000    0.00000
SUPERBLOCK 2 136800.00
  0.00000   -9.40000   -9.40000   -9.40000    17.2200
  0.00000    0.00000    1.00000    0.00000    0.00000
  0.00000
SUPERBLOCK 1 140400.00
  0.00000    0.00000
SUPERBLOCK 2 140400.00
  0.00000   -9.40000   -9.40000   -9.40000    17.2200
  0.00000    0.00000    1.00000    0.00000    0.00000
  0.00000
SUPERBLOCK 1 144000.00
  0.00000    0.00000
SUPERBLOCK 2 144000.00
  0.00000   -9.40000   -9.40000   -9.40000    17.2200
  0.00000    0.00000    1.00000    0.00000    0.00000
  0.00000
SUPERBLOCK 1 147600.00
  0.00000    0.00000
SUPERBLOCK 2 147600.00
  0.00000   -9.40000   -9.40000   -9.40000    17.2200
  0.00000    0.00000    1.00000    0.00000    0.00000
  0.00000
SUPERBLOCK 1 151200.00
  0.00000    0.00000
SUPERBLOCK 2 151200.00
  0.00000   -9.40000   -9.40000   -9.40000    17.2200
```

```
 0.00000    0.00000    1.00000    0.00000    0.00000
 0.00000
SUPERBLOCK 1 154800.00
 0.00000    0.00000
SUPERBLOCK 2 154800.00
 0.00000   -9.40000   -9.40000   -9.40000   17.2200
 0.00000    0.00000    1.00000    0.00000    0.00000
 0.00000
SUPERBLOCK 1 158400.00
 0.00000    0.00000
SUPERBLOCK 2 158400.00
 0.00000   -9.40000   -9.40000   -9.40000   17.2200
 0.00000    0.00000    1.00000    0.00000    0.00000
 0.00000
SUPERBLOCK 1 162000.00
 0.00000    0.00000
SUPERBLOCK 2 162000.00
 0.00000   -9.40000   -9.40000   -9.40000   17.2200
 0.00000    0.00000    1.00000    0.00000    0.00000
 0.00000
SUPERBLOCK 1 165600.00
 0.00000    0.00000
SUPERBLOCK 2 165600.00
 0.00000   -9.40000   -9.40000   -9.40000   17.2200
 0.00000    0.00000    1.00000    0.00000    0.00000
 0.00000
SUPERBLOCK 1 169200.00
 0.00000    0.00000
SUPERBLOCK 2 169200.00
 0.00000   -9.40000   -9.40000   -9.40000   17.2200
 0.00000    0.00000    1.00000    0.00000    0.00000
 0.00000
SUPERBLOCK 1 172800.00
 58.4500    8.00000
SUPERBLOCK 2 172800.00
 22.7315    0.248773   30.1714    30.1779    9.494500E-
03
 1.04200    0.00000    0.571429   1238.74    2.75272
 6.97941
SUPERBLOCK 1 176400.00
 58.4500    8.00000
SUPERBLOCK 2 176400.00
 22.7315    4.44951    37.3809    37.3874   -1.56147
 1.04200    0.00000    1.00000    1187.47    2.76377
 6.97504
SUPERBLOCK 1 180000.00
 58.4500    1.00000
SUPERBLOCK 2 180000.00
 22.7315    0.247011   11.1732    11.1798    9.80232
 1.04200    0.00000    0.857143   175.162    2.72717
 7.00816
SUPERBLOCK 1 183600.00
 58.4500    8.00000
SUPERBLOCK 2 183600.00
 22.7315    5.97035    39.9308    39.9373   -1.59885
 1.04200    0.00000    1.00000    1186.53    2.76787
 6.97494
SUPERBLOCK 1 187200.00
 58.4500    1.00000
SUPERBLOCK 2 187200.00
 22.7315    1.06978    13.3898    13.3969    8.82377
 1.04200    0.00000    1.00000    171.755    2.72990
 7.00516
SUPERBLOCK 1 190800.00
 58.4500    5.00000
SUPERBLOCK 2 190800.00
 22.7315    4.12071    29.2559    29.2624    1.86243
 1.04200    0.00000    1.00000    778.226    2.75137
 6.98464
SUPERBLOCK 1 194400.00
```

```
 58.4500    1.00000
SUPERBLOCK 2 194400.00
 22.7315    0.393477   12.3172    12.3237    8.86191
 1.04200    0.00000    0.857143   171.679    2.72857
 7.00528
SUPERBLOCK 1 198000.00
 58.4500    7.00000
SUPERBLOCK 2 198000.00
 22.7315    5.49328    36.7016    36.7082   -1.08318
 1.04200    0.00000    1.00000    1038.42    2.76269
 6.97637
SUPERBLOCK 1 201600.00
 58.4500    1.00000
SUPERBLOCK 2 201600.00
 22.7315    1.31101    13.4857    13.4922    8.23414
 1.04200    0.00000    1.00000    169.445    2.73002
 7.00336
SUPERBLOCK 1 205200.00
 58.4500    5.00000
SUPERBLOCK 2 205200.00
 22.7315    4.13067    29.1696    29.1761    1.33892
 1.04200    0.00000    1.00000    767.629    2.75124
 6.98315
SUPERBLOCK 1 208800.00
 58.4500    1.00000
SUPERBLOCK 2 208800.00
 22.7315    0.432652   12.3933    12.3999    8.26746
 1.04200    0.00000    1.00000    169.369    2.72867
 7.00347
SUPERBLOCK 1 212400.00
 58.4500    7.00000
SUPERBLOCK 2 212400.00
 22.7315    5.54247    36.4537    36.4602   -1.52900
 1.04200    0.00000    1.00000    1025.66    2.76230
 6.97513
SUPERBLOCK 1 216000.00
 58.4500    1.00000
SUPERBLOCK 2 216000.00
 22.7315    1.31461    13.4527    13.4593    7.70116
 1.04200    0.00000    1.00000    167.373    2.72998
 7.00175
SUPERBLOCK 1 219600.00
 58.4500    5.00000
SUPERBLOCK 2 219600.00
 22.7315    4.07943    28.9266    28.9331    0.916015
 1.04200    0.00000    1.00000    759.058    2.75089
 6.98196
SUPERBLOCK 1 223200.00
 58.4500    1.00000
SUPERBLOCK 2 223200.00
 22.7315    0.402855   12.3168    12.3234    7.91742
 1.04200    0.00000    0.857143   167.992    2.72857
 7.00241
SUPERBLOCK 1 226800.00
 58.4500    7.00000
SUPERBLOCK 2 226800.00
 22.7315    5.42129    36.1658    36.1723   -1.80070
 1.04200    0.00000    1.00000    1017.89    2.76185
 6.97439
SUPERBLOCK 1 230400.00
 58.4500    1.00000
SUPERBLOCK 2 230400.00
 22.7315    1.24619    13.3357    13.3423    7.38001
 1.04200    0.00000    1.00000    166.105    2.72983
 7.00078
SUPERBLOCK 1 234000.00
 58.4500    5.00000
SUPERBLOCK 2 234000.00
 22.7315    3.99801    28.6926    28.6992    0.652786
 1.04200    0.00000    1.00000    753.715    2.75055
```

```
  6.98121
SUPERBLOCK 1 237600.00
  58.4500     1.00000
SUPERBLOCK 2 237600.00
  22.7315     0.376592    12.1845     12.1911     7.50092
  1.04200     0.00000     0.857143    166.353     2.72841
  7.00115
SUPERBLOCK 1 241200.00
  58.4500     7.00000
SUPERBLOCK 2 241200.00
  22.7315     5.27374     35.8113     35.8178     -2.11511
  1.04200     0.00000     1.00000     1008.94     2.76130
  6.97352
SUPERBLOCK 1 244800.00
  58.4500     1.00000
SUPERBLOCK 2 244800.00
  22.7315     1.14921     13.1648     13.1720     7.01545
  1.04200     0.00000     1.00000     164.722     2.72962
  6.99969
SUPERBLOCK 1 248400.00
  58.4500     5.00000
SUPERBLOCK 2 248400.00
  22.7315     3.90087     28.4129     28.4194     0.357679
  1.04200     0.00000     1.00000     747.743     2.75014
  6.98038
SUPERBLOCK 1 252000.00
  58.4500     1.00000
SUPERBLOCK 2 252000.00
  22.7315     0.378792    12.0342     12.0407     7.17323
  1.04200     0.00000     1.00000     165.060     2.72823
  7.00016
SUPERBLOCK 1 255600.00
  58.4500     7.00000
SUPERBLOCK 2 255600.00
  22.7315     5.27751     35.5361     35.5427     -2.36613
  1.04200     0.00000     1.00000     1001.82     2.76087
  6.97283
SUPERBLOCK 1 259200.00
  58.4500     1.00000
SUPERBLOCK 2 259200.00
  22.7315     0.884531    13.0621.    13.0690     6.72022
  1.04200     0.00000     1.00000     163.544     2.72949
  6.99881
SUPERBLOCK 1 262800.00
  58.4500     6.00000
SUPERBLOCK 2 262800.00
  22.7315     4.75867     32.1929     32.1995     -1.29376
  1.04200     0.00000     1.00000     871.748     2.75574
  6.97578
SUPERBLOCK 1 266400.00
  58.4500     1.00000
SUPERBLOCK 2 266400.00
  22.7315     0.559063    12.6988     12.7054     6.66502
  1.04200     0.00000     0.857143    163.238     2.72904
  6.99864
SUPERBLOCK 1 270000.00
  58.4500     7.00000
SUPERBLOCK 2 270000.00
  22.7315     5.30439     35.6975     35.7040     -2.65813
  1.04200     0.00000     1.00000     993.697     2.76112
  6.97203
SUPERBLOCK 1 273600.00
  58.4500     1.00000
SUPERBLOCK 2 273600.00
  22.7315     1.20921     13.2748     13.2813     6.36108
  1.04200     0.00000     1.00000     162.171     2.72976
  6.99774
SUPERBLOCK 1 277200.00
  58.4500     5.00000
SUPERBLOCK 2 277200.00
  22.7315     3.89281     28.2289     28.2354     -0.149101
  1.04200     0.00000     1.00000     737.662     2.74987
  6.97897
SUPERBLOCK 1 280800.00
  58.4500     1.00000
SUPERBLOCK 2 280800.00
  22.7315     0.376739    12.0010     12.0076     6.60793
  1.04200     0.00000     1.00000     162.889     2.72818
  6.99847
SUPERBLOCK 1 284400.00
  58.4500     7.00000
SUPERBLOCK 2 284400.00
  22.7315     5.19971     35.1864     35.1929     -2.79218
  1.04200     0.00000     1.00000     989.832     2.76032
  6.97167
SUPERBLOCK 1 288000.00
  58.4500     1.00000
SUPERBLOCK 2 288000.00
  22.7315     1.06089     12.9309     12.9376     6.21443
  1.04200     0.00000     1.00000     161.568     2.72933
  6.99730
SUPERBLOCK 1 291600.00
  58.4500     5.00000
SUPERBLOCK 2 291600.00
  22.7315     3.72899     27.8778     27.8844     -0.296802
  1.04200     0.00000     1.00000     734.620     2.74936
  6.97855
SUPERBLOCK 1 295200.00
  58.4500     1.00000
SUPERBLOCK 2 295200.00
  22.7315     0.270493    11.7852     11.7918     6.43455
  1.04200     0.00000     0.857143    162.186     2.72792
  6.99795
SUPERBLOCK 1 298800.00
  58.4500     8.00000
SUPERBLOCK 2 298800.00
  22.7315     5.82900     38.5697     38.5763     -4.08020
  1.04200     0.00000     1.00000     1105.59     2.76567
  6.96817
SUPERBLOCK 1 302400.00
  58.4500     1.00000
SUPERBLOCK 2 302400.00
  22.7315     1.26316     13.4620     13.4685     5.84985
  1.04200     0.00000     1.00000     160.263     2.72999
  6.99622
SUPERBLOCK 1 306000.00
  58.4500     5.00000
SUPERBLOCK 2 306000.00
  22.7315     3.92017     28.1257     28.1322     -0.516797
  1.04200     0.00000     1.00000     730.406     2.74972
  6.97794
SUPERBLOCK 1 309600.00
  58.4500     1.00000
SUPERBLOCK 2 309600.00
  22.7315     0.377741    11.9753     11.9819     6.15153
  1.04200     0.00000     1.00000     161.142     2.72815
  6.99711
SUPERBLOCK 1 313200.00
  58.4500     7.00000
SUPERBLOCK 2 313200.00
  22.7315     5.13706     34.9037     34.9102     -3.13355
  1.04200     0.00000     1.00000     980.280     2.75988
  6.97074
SUPERBLOCK 1 316800.00
  58.4500     1.00000
SUPERBLOCK 2 316800.00
  22.7315     0.653458    12.8486     12.8551     5.81046
  1.04200     0.00000     0.857143    160.008     2.72923
  6.99610
SUPERBLOCK 1 320400.00
```

```
  58.4500     7.00000
SUPERBLOCK 2 320400.00
  22.7315     5.24998     35.2890     35.2955     -3.26511
  1.04200     0.00000     1.00000     976.739     2.76048
  6.97038
SUPERBLOCK 1 324000.00
  58.4500     1.00000
SUPERBLOCK 2 324000.00
  22.7315     1.28236     13.1471     13.1537     5.61825
  1.04200     0.00000     1.00000     159.334     2.72960
  6.99553
SUPERBLOCK 1 327600.00
  58.4500     5.00000
SUPERBLOCK 2 327600.00
  22.7315     3.79183     27.8289     27.8355     -0.749759
  1.04200     0.00000     1.00000     725.745     2.74929
  6.97729
SUPERBLOCK 1 331200.00
  58.4500     1.00000
SUPERBLOCK 2 331200.00
  22.7315     0.331302    11.8289     11.8355     5.92759
  1.04200     0.00000     1.00000     160.263     2.72797
  6.99645
SUPERBLOCK 1 334800.00
  58.4500     8.00000
SUPERBLOCK 2 334800.00
  22.7315     5.89071     38.3304     38.3370     -4.44776
  1.04200     0.00000     1.00000     1093.86     2.76528
  6.96718
SUPERBLOCK 1 338400.00
```

```
  58.4500     1.00000
SUPERBLOCK 2 338400.00
  22.7315     1.42411     13.4331     13.4397     5.39724
  1.04200     0.00000     1.00000     158.550     2.72996
  6.99488
**************************************************
Smart Bridge System Design Procedure Step 3
5/12/2002 7:27:15 PM
G:\AlphaTesting\Smart Bridge\Design Procedure Step 3.dat
 2
 2
Unit 3, TYPE 730: Source side mass flow rate (kg/s)
Unit 3, TYPE 730: Number of heat pump pairs in operation (-)
 11
Unit 1, TYPE 700: MASS FLOW RATE (kg/S)
Unit 1, TYPE 700: AVERAGE PAVEMENT TOP SURFACE
TEMPERATURE (C)
Unit 1, TYPE 700: OUTLET FLUID TEMPERATURE (C)
Unit 2, TYPE 750: OUTLET TEMPERATURE (C)
Unit 4, TYPE 750: OUTLET TEMPERATURE (C)
Unit 1, TYPE 700: SNOWFALL RATE IN WATER
EQUIVALENT PER HR (mm/hr)
Unit 1, TYPE 700: RAINFALL RATE IN WATER
EQUIVALENT PER HR (mm/hr)
Unit 1, TYPE 700: SNOW FREE AREA RATIO (-)
Unit 1, TYPE 700: HEAT PROVIDED TO PAVEMENT
SURFACE (kW)
Unit 2, TYPE 750: POWER CONSUMPTION (kW)
Unit 4, TYPE 750: POWER CONSUMPTION (kW)
```

APPENDIX E

MODIFIED MODSIM CODE

The following pages provide a listing of the code changes in the MODSIM application. The Fortran project used to build MODSIM consists of many subroutines. Only those subroutines that were modified are listed here. The files MODSIM.FOR, MODINO.FOR and SNSQA.FOR have been modified.

File: MODSIM.FOR

```
C***********************************************************************
C   This application is designed to run with Visual Modeling Tool For
C   HVACSIM+.
C   Last change: 07 May 2002
C***********************************************************************
C
C   MODSIM  : A MODular SIMulation program
C             Main program of HVACSIM+ package
C
C                  Version 12.0
C
C           National Institute of Standards and Technology
C           Building and Fire Research Laboratory
C           Building Environment Division
C           Gaithersburg, Maryland 20899-8631
C           U.S.A.
C
C -------------------------------------------------------------------
C
C   Original MODSIM program written by  C. Ray Hill, NBS, June 1983
C
C   Modification into FORTRAN77 by Cheol Park, NBS, September 11, 1984
C
C   Modification for reading boundary values and initialization
C   file by  Dan Clark, NBS, Oct. 18, 1983 and May 1, 1984
C
C   Modification for building shell by Cheol Park, NBS, Dec. 27, 1984
C
C   Updated by Cheol Park & Dan Clark, May 6, 1985 & May 31, 1986
C
C ----- Version 5.0.2 February 5, 1987    Cheol Park
```

```
C
C     (1) MODSIM5, TYPE51, REPORT, INDATA, BAKDIF, and UPDATE
C         were modified.
C     (2) Parameters of MAXDEQ and MAXPAR were changed into
C         MAXDEQ=90, and MAXPAR=1200 to handle a larger simulation.
C
C ---- Version 5.2.3  August 3, 1988    Cheol Park
C
C     Modified for IBM PC or alike.
C
C ---- Version 6.2.0  April 25, 1989
C
C     (1) Parameter values were changed into:
C          MAXSBK=5, MDEQIS=30, MINOIB=70.
C     (2) I/O information code was slightly changed.
C     (3) All DO loops are indented.
C
C ---- Version 6.2.2  September 18, 1989
C
C     Parameter MAXSBK=10.
C
C ---- Version 6.2.6  February 14, 1992
C
C     The label "RVPS" is changed into "OTHR", and "AHUM" into "HUMT".
C
C ---- Version 7.0  October 7, 1992
C
C     Use of INCLUDE statement, non-ANSI standard, in place of PARAMETERs
C
C ---- Version 8.0  March 7, 1994
C
C     TYPES.FOR was updated according to Y. Zhu's suggestions.
C
C ---- Version 8.1  February 10, 1995
C
C     MODINO.FOR was modified to read new format of model definition file,
C     which is the output of SLIMCON.  The data file has descriptions of
C     each data.
C
C ---- Version 10.0  July 11, 1996 P Haves, Loughborough Univ., UK
C     Supports the "simulation test-bed" produced by ASHRAE 825-RP
C     (1) COMMON BLOCKS /INFOCB/ and /CHRON2/ added. /INFOCB/
C         facilitates warning messages
C     (2) MAIN PROGRAM: write results to output file only if reporting
C         interval for current superblock is greater than zero. Avoids
C         frequent writes to output files when reporting interval is
C         inadvertantly set to zero
C     (3) SUBROUTINE OPNFIL: Long file names supported (46 chars + '.' + 3
C         char extension = 50 chars)
C     (4) SUBROUTINE OPNFIL: unique extensions used for each file. Lower case
C         is used for convenience when file names are case sensitive.
```

```
C     (5) SUBROUTINE REPORT: avoid interpolation across a reset
C     (6) SUBROUTINE SUMMARY: NUMBER and LABEL dimensioned using
PARAMETER
C        MSUMMARY included in modsim.par
C     (7) SUBROUTINE VLAB: ID dimensioned using PARAMETER
C        MSUMMARY included in modsim.par
C     (8) SUBROUTINE (SUPER)BLOCK: list active variables in five columns
C        rather than eight to match the format of the intermediate solution
C        and the Jacobian
C     (9) SUBROUTINE (SUPER)FNC: report intermediate residuals in addition
C        to intermediate variable values
C
C     Jan. 16, 1997   Cheol Park, NIST
C     Added FRZ for selecting either freezing or unfreezing variables.
C
C    Modified by Aditya Varanasi, Oklahoma State University, to include
C       residual reporting. This version is intended to be used with the
C       VISUAL MODELING TOOL FOR HVACSIM+ created to replace HVACGEN and
C       automate the simulation process. May 7, 2002
C
C-------------------------------------------------------------------
C DISCLAIMER
C
C This program is furnished by the government and used by any
C recipient  with the express understanding that the United
C States Government makes no warranty, expressed or implied,
C concerning the accuracy, completeness, reliablilty, usability,
C or suitability for any particular purpose of the information and data
C contained in this program or furnished in connection therewith,
C and the United States shall be under no liability whatsoever
C to any person by reason of any use made thereof.  This program
C belongs to the government.  Therefore, the recipient
C further agrees not to assert any proprietary rights therein or
C to represent this program to anyone as other than a government
C program.
C
C
C -------------------------------------------------------------------
C
C  SUBPROGRAMS CALLED:
C
C
ASEMBL,BACTIV,BLOCK,BOUNDS,ECNTRL,FRZVAR,INDATA,INSCAN,INTLIZ,
C     IPERM,OPNFIL,RCONF,REPORT,RESET,RESTAT,SUMARY,SUPERB,UNFREZ,
C     UPDATE
C
C     ------ RELATED TO THE BUILDING SHELL MODEL -----
C
C     RDENV
C
C  REFERENCES:
```

```
C
C  [1] NBS, "Guide to Avaiable Mathematical Software(GAMS),"
C     Center for Applied Math., National Bureau of Standards,
C     Washington, DC, Oct. 1981.
C     Information on subrotuine SNSQ can be found here.
C
C  [2] Brayton, R.K., Gustavson, F.G., and Hachtel, G.D.,
C     "A New Efficient Algorithms for Solving Differential-Algebraic
C     Systems Using Implicit Backward Differential Formulas,"
C     Proc. IEEE, Vol. 60, No. 1, Jan. 1972.
C     Information on the method used to integrate differential
C     equations can be found here.
C
C  NOTE:
C       See MODSIM.DOC for further detailed information.
C
C***********************************************************************

      PROGRAM  MODSIM
      USE DFLIB

      INTEGER  ECNTRL,UFZFLG,UNFREZ
      LOGICAL  BLDSHL,BSCALL,INTSHL,VIEW,FRZ

      INCLUDE 'hvacsim.par'

      CHARACTER TITLE*80,LBLS(MAXLBL)*4

      DIMENSION ICALL(MAXSBK),UFZFLG(MBLKIS),ISPOS(10)

      COMMON   /BLOCKS/ IBLOCK(MAXBLK,MUNTIB),NUNITS(MAXBLK),NBLOCK
      COMMON   /BOUND/  IBOUND(MAXBND),NBOUND
      COMMON   /BSTAT/  IBSTAT(MAXBLK)
      COMMON   /CHRONO/ TIME,TSTEP,TTIME,TMIN,ITIME
      COMMON   /CHRON2/ TSTOP
      COMMON   /COEF/   AH(MAXSBK,7),GH(MAXSBK,7),DNK(MAXSBK),
     &                  FNK(MAXSBK),KORDER(MAXSBK),IP(MAXSBK,7),
     &                  IPA(MAXSBK,8)
      COMMON   /CONECT/ IN(MAXUNT,MINOIU),IOUT(MAXUNT,MINOIU),
     &                  NIN(MAXUNT),NOUT(MAXUNT)
      COMMON   /DIFEQN/ SPAST(MAXDEQ,7),SPRED(MAXDEQ),NDE(MAXUNT),
     &                  INDE(MAXUNT,MDEQIU),IDEVAR(MAXDEQ),ND,
     &                  INDESB(MAXSBK,MDEQIS),NDSB(MAXSBK)
      COMMON   /FILES/  IFILE1,IFILE2,IFILE3,IFILE4,IFILE5,IFILE6,INP
      COMMON   /FREEZE/ FRZ
      COMMON   /GA/     AM(MAXSBK,8,8),G(MAXSBK,7),A(MAXSBK,7)
      COMMON   /INFOCB/ IU,LASTIT
      COMMON   /LABELS/ TITLE
      COMMON   /MCHRON/ MTIME,MTSTEP,MSBT(MAXSBK),MSTEP(MAXSBK),
     &                  IDEFLG(MAXSBK)
      COMMON   /OPMODE/ MODE
```

```
  COMMON  /OPTION/ IFZOPT(MAXSBK),INSOPT(MAXSBK)
  COMMON  /PARAM/ PAR(MAXPAR),JPAR(MAXUNT),NPAR
  COMMON  /PROPER/ RHOA,RHOW,CPA,CPW
  COMMON  /PRTOUT/ IPRINT,TPRTON,TPRTOF
  COMMON  /RCHRON/ TNEXT(MAXSBK),TREPRT(MAXSBK)
  COMMON  /REPRT/ IREPRT(MAXSBK,MRPTIS),NREPRT(MAXSBK),
 &        IDENT(MAXSBK,MRPTIS,2)
  COMMON  /SAVE/  SAVED(MAXSAV),ISAVED(MAXUNT)
  COMMON  /SCAN/  INSB(MAXSBK,MINOIS),NINSB(MAXSBK)
  COMMON  /SCHRON/ SBTIME(MAXSBK),STEP(MAXSBK),KSTEP(MAXSBK)
  COMMON  /SIGNDE/ ISIGN(MAXSTV)
  COMMON  /SOLCOM/ STOLD(MAXSTV),JSOLVE(MAXBLK,MSEQIB),
 &        NJSOLV(MAXBLK),JSSOLV(MAXSBK,MSEQIS),
 &        NJSSLV(MAXSBK)
  COMMON  /SOLVE/ ISOLVE(MAXBLK,MSEQIB),NSOLVE(MAXBLK)
  COMMON  /SOSCOM/ RTOLX,ATOLX,XTOL
  COMMON  /SUPER/ ISUPER(MAXSBK,MBLKIS),NSUPER(MAXSBK),
 &        IOUTB(MAXBLK,MINOIB),NOUTB(MAXBLK),
 &        INB(MAXBLK,MINOIB),NINB(MAXBLK),NSBLOK
  COMMON  /SUPSOL/ ISSOLV(MAXSBK,MSEQIS),NSSOLV(MAXSBK)
  COMMON  /UNITS/ IUNITS(MAXUNT),NU
  COMMON  /VDENT/ NDENT(MAXLBL)
  COMMON  /VECTOR/ STATE(MAXSTV),TSTATE(MAXSTV),NSTATE
  COMMON  /VLBLS/ LBLS
  COMMON  /VSTAT/ ICHECK(MAXSTV),IDECHK(MAXDEQ)
  COMMON  /XINIT/ INIT,NSAVED

C   Common statements dedicated to the building shell model

  COMMON  /CTF/  U(MAXSTR),X(0:MAXNRF,MAXSTR),Y(0:MAXNRF,MAXSTR)
 &        ,Z(0:MAXNRF,MAXSTR),RJ(MAXORD,MAXSTR),
 &        NORDA(MAXSTR),NCTFA(MAXSTR)
  COMMON  /FILEB/ IFILE7,IFILE8
  COMMON  /RADFLX/ FQRAD(MAXZN)
  COMMON  /SHELL/ TSHELL,TZERO
  COMMON  /SOLDTA/ HDN,HSKY,HHOR,SSAZM,SSALT,NDN,NSKY,NHOR
  COMMON  /WALLS/ HISC(MAXNS,MAXZN),HISR(MAXNS,MAXZN),
 &        AS(MAXNS,MAXZN),FVIEW(MAXNS,MAXZN),
 &        SOLINT(MAXNS,MAXZN),TRANSM(MAXNS,MAXZN),
 &        SC(MAXNS,MAXZN),ABSIS(MAXNS,MAXZN)
  COMMON  /WEATHR/ TOA,WOA,POA,VWIND,NTOA,NWOA,NPOA,nvwind
C************************************************************************
C     Phase II - Aditya Varanasi
      COMMON   /RPLOT/  PREVX,PREVY
      COMMON   /RDIAG/
RESMON,IFILERESD,RESITER,NUMITER,YSCALE,TMAXT,
 &                  YSCALENAME
      COMMON   /FORM/   CURRENTUNIT
      COMMON   /TOLERANCE/ ONETOL,PTOL,FTOL,TTOL,CTOL,RRTOL,ETOL,
 &          PWTOL,HTOL
      COMMON   /RESTOLERANCE/ RESONETOL,PTOLRES,FTOLRES,TTOLRES,
  &            CTOLRES,RRTOLRES,ETOLRES,PWTOLRES, HTOLRES
      CHARACTER*7 YSCALENAME
      INTEGER STATUS, PLOTREPLY, EXITREPLY
      LOGICAL REPLY
      TYPE (QWINFO) WINFO
      TYPE (xycoord) XY
      TYPE (rccoord) rc
  TYPE (windowconfig) wc
      CHARACTER (LEN = 80) RRCaption
C************************************************************************

C ----------------------------------------------------------------------

C************************************************************************
C     Phase II - Aditya Varanasi
C     Default: Do not plot the residuals.
      RESMON = 0.0

C     Delete un-necessary menu items,
      REPLY = DELETEMENUQQ(1,1)
      REPLY = DELETEMENUQQ(1,1)
      REPLY = DELETEMENUQQ(2,0)
      REPLY = DELETEMENUQQ(5,1)
      REPLY = DELETEMENUQQ(5,1)

C     Set the text in the About box for the application.
      STATUS = ABOUTBOXQQ('MODular SIMulator program\nfor Visual
  &Modeling Tool For HVACSIM+\n\nBased on HVACSIM+ Version
  &12.0 developed at\nNational Institute of Standards & Technology
  &\nGaithersburg, Maryland 20899-8631  U.S.A.'C)

C   Set the name of the initial child window. Replace 'Graphic1'
  REPLY = GETWINDOWCONFIG(wc)
  wc%title = "SIMULATION PROGRESS"C
  REPLY = SETWINDOWCONFIG(wc)
  if (.NOT.REPLY) REPLY = SETWINDOWCONFIG(wc)
C************************************************************************

C   Enter data interactively as requested.

  CALL INDATA(BLDSHL,ISPOS,ISSHEL,ISVIEW,NSTVEC,TMAX,
  &         TSTOP,VIEW,INTSHL,MOUT)

  PRINT *
  PRINT *,'   ************************************************'
  PRINT *,'   *                                              *'
  PRINT *,'   *   MODSIM : A MODular SIMulation program      *'
  PRINT *,'   *   for Visual Modeling Tool for HVACSIM+      *'
  PRINT *,'   *                                              *'
  PRINT *,'   *  Based on HVACSIM+ Version 12.0 developed    *'
```

```
      PRINT *,'     *                    at                        *'
      PRINT *,'     * National Institute of Standards & Technology *'
      PRINT *,'     *   Gaithersburg, Maryland 20899-8631  U.S.A.  *'
      PRINT *,'     *                                              *'
      PRINT *,'     ************************************************'
      PRINT *

C     Read model definition data.

      CALL RCONF(TREPRT)

C***********************************************************************
C         Phase II - Aditya Varanasi

C     Store the unit number of the default child window.
       CURRENTUNIT = GETACTIVEQQ()

C         Check if the user requested for plotting the residuals.
C         If yes setup a new window.
          IF (RESMON.EQ.1.0) THEN
C           Initialize the fonts to defaults.
            PLOTREPLY = INITIALIZEFONTS()

C           Open a new child window to plot the residuals.
            OPEN (UNIT = 9999, FILE = 'USER',
     &     TITLE = 'REPORTING RESIDUALS for'//TRIM(TITLE))

C           Maximize the windows.
            WINFO.TYPE = QWIN$MAX
            STATUS = SETWSIZEQQ(9999, WINFO)

C           Draw the plot area.
            PLOTREPLY = SETCOLOR(12)
            PLOTREPLY = RECTANGLE($GBORDER, 5,5, 900,560)
            PLOTREPLY = SETCOLOR(6)
            PLOTREPLY = RECTANGLE($GBORDER, 121,71, 899,559)
            PLOTREPLY = RECTANGLE($GBORDER, 122,72, 898,558)
            PLOTREPLY = SETCOLOR(12)
            PLOTREPLY = RECTANGLE($GBORDER, 123,73, 897,557)
            PLOTREPLY = SETCOLOR(15)
            PLOTREPLY = FLOODFILL(320,480,12)

C           Set the view port coordinates.
            CALL SETVIEWPORT(124,74,896,556)
            CALL MOVETO(0,480, XY)
            PREVX = 0
            PREVY = 480
            RESITER = 1
            NUMITER = 1
          ELSE
C           Restore the MDI window.
```

```
        WINFO.TYPE = QWIN$RESTORE
        status = SETWSIZEQQ(QWIN$FRAMEWINDOW,WINFO)
C       Maximize the default child window within the restored MDI window.
        WINFO.TYPE = QWIN$MAX
      STATUS = SETWSIZEQQ(5, WINFO)
      END IF
C***************************************************************************
```

```
C    Assemble block information vectors.

     CALL ASEMBL

C    Write summary of the simulation configuration.

     CALL SUMARY(TMAX,TSTOP)
```

```
C***************************************************************************
C       Phase II - Aditya Varanasi
C       Write the header in the diagnostic file.
C       Check if the user requested for plotting the residuals.
C       If yes setup a new window.
        IF (RESMON.EQ.1.0) THEN
          WRITE(IFILERESD, *)
   &"*******************************************
   ***************************************"
          WRITE(IFILERESD, *)
   &"          VISUAL MODELING TOOL FOR
   &HVACSIM+"
          WRITE(IFILERESD, *)
   &"*******************************************
   ***************************************"
          WRITE(IFILERESD, *)
   &"       DIAGNOSTIC INFORMATION
   &(RESIDUAL PLOTTING) FILE for         "
          WRITE(IFILERESD, *) TRIM(TITLE)
          WRITE(IFILERESD, *)
   &"*******************************************
   &***************************************"
          WRITE(IFILERESD, *) " TIME        ||RESIDUAL VECTOR||
   &  CONVERGED IN (ITERATIONS)"
          WRITE(IFILERESD, *)
   &"*******************************************
   &***************************************"
        END IF
C***************************************************************************
```

```
C    Initialize simulation.

...
...
...
```

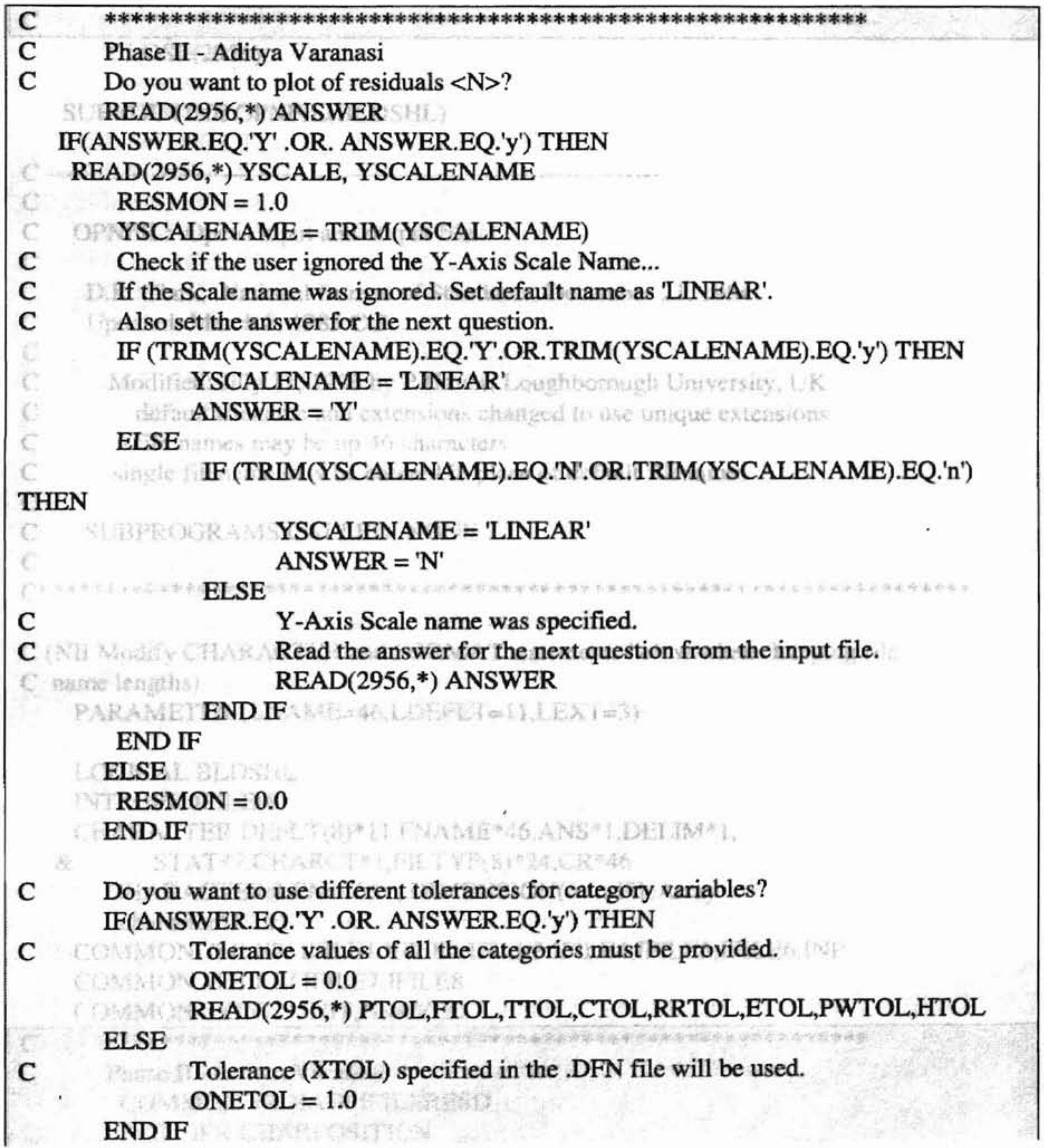
```
C     Save simulation results as an initialization file for the next
C      run.

      WRITE(IFILE6,2000) TMIN,TMAX,TSTOP,TIME,NSTATE,NSAVED
      WRITE(IFILE6,3000) (STATE(I),I=1,NSTATE)
      WRITE(IFILE6,4000) (ISIGN(I),I=1,NSTATE)
      IF(NSAVED.GT.0) THEN
       WRITE(IFILE6,3000) (SAVED(I),I=1,NSAVED)
      ENDIF
      PRINT *,' -----  INITIALIZATION FILE HAS BEEN WRITTEN -----'
     ENDIF

C ------------ ---------------------------------------------------------

C    Format statements

2000  FORMAT(4F10.2,2I5)
3000  FORMAT(1P5G15.6)
4000  FORMAT(25I3)

C***********************************************************************
C      Phase II - Aditya Varanasi
       IF (RESMON.EQ.1.0) THEN
         CLOSE(IFILERESD)
         CALL SETMESSAGEQQ (
     &   'Finished Plotting the Residuals with Exit code ',
     &   QWIN$MSG_TERM)
         CALL SETMESSAGEQQ ('\nGo to the Visual Tool?'C,
     &   QWIN$MSG_EXITQ)

         STOP '  ------------END OF SIMULATION ---------'
       ELSE
         CLOSE(IFILERESD, STATUS='DELETE', ERR=4123)
4123     CALL ABORT('MODSIM HAS ENDED')
       ENDIF
C***********************************************************************
       END PROGRAM

C***********************************************************************
C      Phase II - Aditya Varanasi
       LOGICAL(4) FUNCTION INITIALSETTINGS()
             USE DFLIB
             INTEGER STATUS
             TYPE (qwinfo) qwi
             qwi.type = QWIN$MAX
             i = SetWSizeQQ(QWIN$FRAMEWINDOW, qwi)
             INITIALSETTINGS= .true.
       END FUNCTION INITIALSETTINGS
C***********************************************************************
```

File: MODINO.FOR

```
C***********************************************************************

      BLOCK DATA

C -----------------------------------------------------------------
C
C    Updated on : Feb. 14, 1992  C.P.
C
C    CPA     Specific heat of dry air  (kJ/kg-C)
C    CPW     Specific heat of water    (kJ/kg-C)
C    RHOA    Density of air   (kg/m**3)
C    RHOW    Density of water (kg/m**3)
C
C***********************************************************************

      INCLUDE 'hvacsim.par'

      CHARACTER*4 LBLS(MAXLBL)

      COMMON   /PROPER/ RHOA,RHOW,CPA,CPW
      COMMON   /SIGNDE/ ISIGN(MAXSTV)
      COMMON   /VLBLS/  LBLS
      COMMON   /FILES/  IFILE1,IFILE2,IFILE3,IFILE4,IFILE5,IFILE6,INP
      COMMON   /FILEB/  IFILE7,IFILE8
      COMMON   /INFOCB/ IU,LASTIT
C***********************************************************************
C      Phase II - Aditya Varanasi
       COMMON   /RDIAG/  IFILERESD
C***********************************************************************

      DATA  RHOA,RHOW,CPA,CPW/1.2,1000.,1.,4.180/
      DATA  LBLS/'PRES','FLOW','TEMP','CTRL','OTHR','ENRG','POWR',
     &      'HUMT'/
      DATA  INP/5/
      DATA  IFILE1,IFILE2,IFILE3,IFILE4,IFILE5,IFILE6/7,8,9,10,11,12/
      DATA  IFILE7,IFILE8/13,14/
      DATA  ISIGN/MAXSTV*1/
      DATA  LASTIT/0/
C***********************************************************************
C      Phase II - Aditya Varanasi
       DATA  IFILERESD/15/
C***********************************************************************
      END
C***********************************************************************

...
...
```

```
...
      SUBROUTINE INDATA(BLDSHL,ISPOS,ISSHEL,ISVIEW,NSTVEC,TMAX,
     & TSTOP,VIEW,INTSHL,MOUT)

C ------------------------------------------------------------------
C
C   INDATA : Enter input data interactively before a simulation.
C
C   March 15, 1985 Cheol Park & Dan Clark
C
C   Added MOUT to select time base for writing output. Jan. 16, 1986
C   Added PRINT *,'=>' to indicate user's input.    April 25, 1989
C   Added FRZ to able or disable the freezing variables Jan. 16, 1997 C.P.
C        Modified to eliminate the use of keyboard       jan 21,2000
C   SUBPROGRAMS CALLED: OPNFIL
C
C**********************************************************************

      INCLUDE 'hvacsim.par'

      LOGICAL  BLDSHL,INTSHL,VIEW,FRZ    ! changed 1/16/97
      CHARACTER ANSWER*1,LBLS(MAXLBL)*4

      DIMENSION ISPOS(10)

      COMMON   /CHRONO/ TIME,TSTEP,TTIME,TMIN,ITIME
      COMMON   /FILEB/  IFILE7,IFILE8
      COMMON   /FILES/  IFILE1,IFILE2,IFILE3,IFILE4,IFILE5,IFILE6,INP
      COMMON   /FREEZE/ FRZ
      COMMON   /PRTOUT/ IPRINT,TPRTON,TPRTOF
      COMMON   /SHELL/  TSHELL,TZERO
      COMMON   /VLBLS/  LBLS
      COMMON   /XINIT/  INIT,NSAVED
C      *********************************************************
C      Phase II - Aditya Varanasi
       COMMON   /RDIAG/
RESMON,IFILERESD,RESITER,NUMITER,YSCALE,TMAXT,
     &                    YSCALENAME
       COMMON   /TOLERANCE/ ONETOL,PTOL,FTOL,TTOL,CTOL,RRTOL,ETOL,
     &             PWTOL,HTOL
       COMMON   /RESTOLERANCE/ ONETOLRES,PTOLRES,FTOLRES,TTOLRES,
     &        CTOLRES,RRTOLRES,ETOLRES,PWTOLRES, HTOLRES
       CHARACTER*7 YSCALENAME
C      *********************************************************

C Initialize default values

      BLDSHL=.FALSE.
      INTSHL=.FALSE.
      VIEW=.FALSE.
      FRZ = .TRUE.
      INIT=0
      IDUMP=0
      MONITR=0
      IPRINT=2
      TPRTON=0.
      TPRTOF=0.

      OPEN(2956,FILE="INPUTFILE.DAT",STATUS='UNKNOWN')

      READ(2956,*) TMIN,TMAX,TSTOP

      TMAXT = TMAX

c      PRINT *,' Will the Initialization File be called? <N>  '
      READ(2956,*) ANSWER
      IF(ANSWER.EQ.'Y' .OR. ANSWER.EQ.'y') THEN
       INIT=1
      ENDIF

C         Ask questions pertaining to all simulations

C         Open input and output files
      CALL OPNFIL(BLDSHL)

c      PRINT *,' Do you want to use Reported Time for outputs <N>?  '
      READ(2956,*) ANSWER
      IF(ANSWER.EQ.'Y' .OR. ANSWER.EQ.'y') THEN
       MOUT=2
      ELSE
       MOUT=1
      ENDIF

c      PRINT *,' Do you wish to disable Freezing Variable feature <N>?  '
      READ(2956,*) ANSWER
      IF(ANSWER.EQ.'Y' .OR. ANSWER.EQ.'y') THEN
       FRZ = .FALSE.
      ELSE
       FRZ = .TRUE.
      ENDIF


c      PRINT *,' Do you want Diagnostic Information to be written <N>? '
      READ(2956,*) ANSWER
      IF(ANSWER.EQ.'Y' .OR. ANSWER.EQ.'y') THEN
       READ(2956,*) IANS
       IF(IANS.EQ.1) THEN
        IPRINT=-1
       ELSEIF(IANS.EQ.2) THEN
        IPRINT=10000
       ELSEIF(IANS.EQ.3) THEN
        IPRINT=1
```

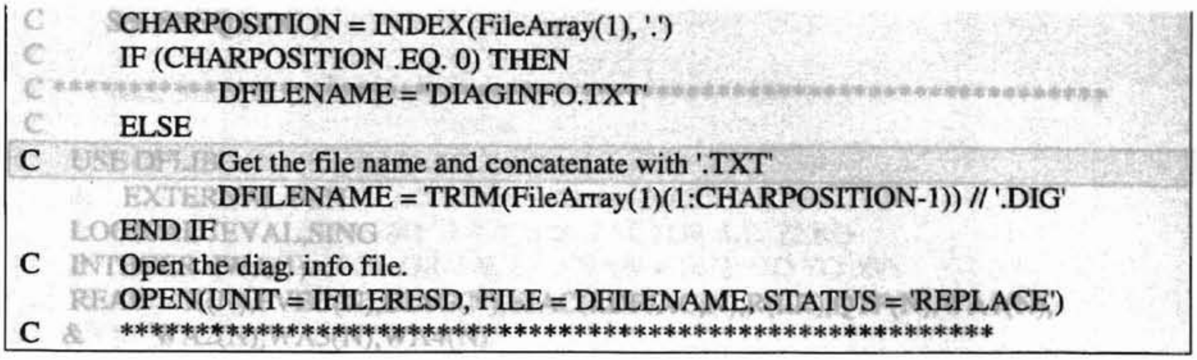
```
     ELSEIF(IANS.EQ.4) THEN
      IPRINT=2
     ENDIF
              TPRTON = TMIN
              TPRTOF = TSTOP
        end if

c    PRINT *,' Would you like to monitor simulation on Screen? <N> '
    READ(2956,*) ANSWER
    IF(ANSWER.EQ.'Y' .OR. ANSWER.EQ.'y') THEN
     VIEW=.TRUE.
    ENDIF
```



```
C        **********************************************************
C        Phase II - Aditya Varanasi
C        Do you want to plot of residuals <N>?
         READ(2956,*) ANSWER
    IF(ANSWER.EQ.'Y' .OR. ANSWER.EQ.'y') THEN
     READ(2956,*) YSCALE, YSCALENAME
          RESMON = 1.0
          YSCALENAME = TRIM(YSCALENAME)
C         Check if the user ignored the Y-Axis Scale Name...
C         If the Scale name was ignored. Set default name as 'LINEAR'.
C         Also set the answer for the next question.
          IF (TRIM(YSCALENAME).EQ.'Y'.OR.TRIM(YSCALENAME).EQ.'y') THEN
              YSCALENAME = 'LINEAR'
              ANSWER = 'Y'
          ELSE
              IF (TRIM(YSCALENAME).EQ.'N'.OR.TRIM(YSCALENAME).EQ.'n')
THEN
                    YSCALENAME = 'LINEAR'
                    ANSWER = 'N'
              ELSE
C                   Y-Axis Scale name was specified.
C                   Read the answer for the next question from the input file.
                    READ(2956,*) ANSWER
              END IF
          END IF
         ELSE
          RESMON = 0.0
         END IF

C        Do you want to use different tolerances for category variables?
         IF(ANSWER.EQ.'Y' .OR. ANSWER.EQ.'y') THEN
C             Tolerance values of all the categories must be provided.
              ONETOL = 0.0
              READ(2956,*) PTOL,FTOL,TTOL,CTOL,RRTOL,ETOL,PWTOL,HTOL
         ELSE
C             Tolerance (XTOL) specified in the .DFN file will be used.
              ONETOL = 1.0
         END IF
```

```
C       Do you want to use different tolerances for residual function?
        READ(2956,*) ANSWER
        IF(ANSWER.EQ.'Y' .OR. ANSWER.EQ.'y') THEN
C           Tolerance values of all the functions must be provided.
            ONETOLRES = 0.0
            READ(2956,*) PTOLRES,FTOLRES,TTOLRES,CTOLRES,
     &      RRTOLRES,ETOLRES,PWTOLRES,HTOLRES
        ELSE
C           No tolerance (i.e. Tolerance = 0.0) will be used.
            ONETOLRES = 1.0
        END IF
C       ********************************************************
        CLOSE(2956)

      SUBROUTINE OPNFIL(BLDSHL)

C -------------------------------------------------------------------
C
C   OPNFIL:  Opens input and output files.
C
C    D.R. Clark,  National Bureau of Standards, December 21, 1984
C    Updated: March 8, 1985 C.P.
C
C       Modified: July 11, 1996 by P Haves, Loughborough University, UK
C         - default filename and extensions changed to use unique extensions
C         - file names may be up 46 characters.
C       - single filename may be entered in place of default filenames
C
C    SUBPROGRAMS CALLED: NONE
C
C*********************************************************************

C (NB Modify CHARACTER and FORMAT statements below when changing file
C  name lengths)
      PARAMETER (LNAME=46,LDEFLT=11,LEXT=3)

      LOGICAL BLDSHL
      INTEGER IFILE(8)
      CHARACTER DEFLT(8)*11,FNAME*46,ANS*1,DELIM*1,
     &        STAT*7,CHARCT*1,FILTYP(8)*24,CR*46
        CHARACTER(LEN=15)   , DIMENSION(6) :: FileArray
         character*1  cr1
      COMMON  /FILES/ IFILE1,IFILE2,IFILE3,IFILE4,IFILE5,IFILE6,INP
      COMMON  /FILEB/ IFILE7,IFILE8
      COMMON  /XINIT/ INIT,NSAVED
C       ********************************************************
C       Phase II - Aditya Varanasi
         COMMON  /RDIAG/ IFILERESD
         INTEGER CHARPOSITION
```

```
       CHARACTER(15) DFILENAME
C      ************************************************************

      DATA FILTYP(1) /'Model Definition File,  '/
      DATA FILTYP(2) /'Boundary Variable File, '/
      DATA FILTYP(3) /'Initial State File,     '/
      DATA FILTYP(4) /'Final State File,       '/
      DATA FILTYP(5) /'Output Data File,       '/
      DATA FILTYP(6) /'Simulation Summary File,'/
      DATA FILTYP(7) /'CTF File,               '/
      DATA FILTYP(8) /'Weather Data File,      '/

      DATA CR /'                           '/
      DATA STAT /'OLD    '/
      DATA DELIM /'.'/
         data cr1/' '/
      IFILE(1)=IFILE2
      IFILE(2)=IFILE4
      IFILE(3)=IFILE5
      IFILE(4)=IFILE6
      IFILE(5)=IFILE1
      IFILE(6)=IFILE3
      IFILE(7)=IFILE7
      IFILE(8)=IFILE8

      IF(BLDSHL) THEN
       NUMFIL=8
      ELSE
       NUMFIL=6
      ENDIF

C      Open each file.
      DO 500 I=1,NUMFIL
C              Read the file name from INPUTFILE.DAT
               READ(2956,*)  FileArray(i)
       IF(I.GE.3 .AND. I.LE.6) THEN
                     OPEN(UNIT=IFILE(I),FILE=FileArray(i))
                     CLOSE(UNIT=IFILE(I),STATUS='DELETE')
                     STAT='UNKNOWN'
               ELSE
                     STAT='OLD    '
               ENDIF
               IF (I.EQ.3) goto 500
          OPEN(UNIT=IFILE(I),FILE=FileArray(i),STATUS=STAT)
500    CONTINUE

C      ************************************************************
C      Phase II - Aditya Varanasi
C      Extract the file name from the Definition file. The same file name
C              will be used for the diagnostics file.
C      Get the position of period in the file name.
```

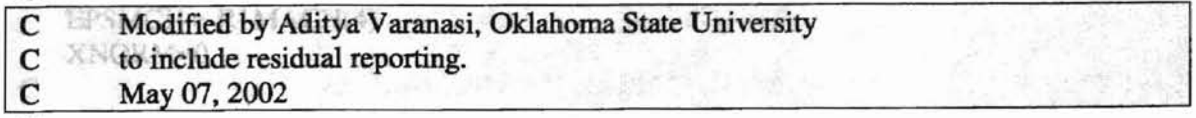


```
      CHARPOSITION = INDEX(FileArray(1), '.')
      IF (CHARPOSITION .EQ. 0) THEN
             DFILENAME = 'DIAGINFO.TXT'
      ELSE
C            Get the file name and concatenate with '.TXT'
             DFILENAME = TRIM(FileArray(1)(1:CHARPOSITION-1)) // '.DIG'
      END IF
C     Open the diag. info file.
      OPEN(UNIT = IFILERESD, FILE = DFILENAME, STATUS = 'REPLACE')
C     **********************************************************
```

```
      RETURN
      END
C***********************************************************************

...
...
...
```

File: SNSQA.FOR

```
C ***********************************************************************
C
      SUBROUTINE SNSQ1(FCN,JAC,IOPT,N,X,FVEC,FJAC,LDFJAC,XTOL,MAXFEV,
     &           ML,MU,EPSFCN,DIAG,MODE,FACTOR,NPRINT,INFO,NFEV,
     &           NJEV,R,LR,QTF,WA1,WA2,WA3,WA4,IBLK,ISBLK,LABEL)
C
C -----------------------------------------------------------------
C
C    SNSQ1 : This program is essentially same as SNSQ.
C        Called by subroutine BLOCK. Calls FDJAC2, while SNSQ calls
C        FDJAC1. Note that the argument of FCN has IBLK and ISBLK.
C
C        The original program was written by K.L. Hiebert
C
C        A minor modification was made by C. R. Hill and C. Park
C        August 13, 1984
C
C        Updated for writing diagonastic information by D.C.,
C        March 8, 1985
C
C        Modified by P. Haves, Oxford Univ., U.K. to include
C        simulation times in warning/error messages
C        April 12, 1989
C
```

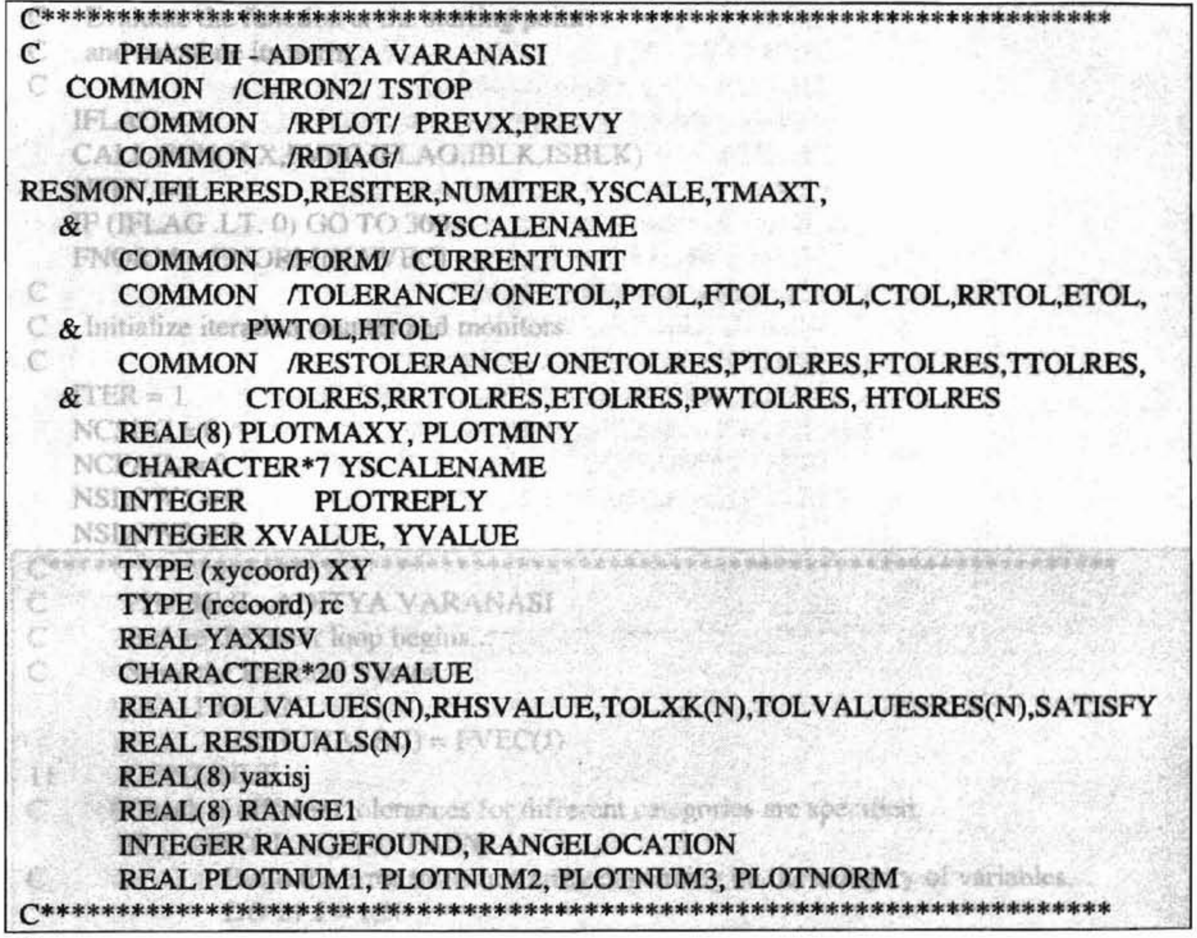


```
C        Modified by Aditya Varanasi, Oklahoma State University
C        to include residual reporting.
C        May 07, 2002
```

```
C
C    (The documention of this SNSQ routine resides elsewhere.
C     See SNSQ.DOC )
C
C ************************************************************************
C
```

```
   USE DFLIB
```

```
      EXTERNAL FCN
   LOGICAL JEVAL,SING
   INTEGER IWA(1)
   REAL    X(N),FVEC(N),DIAG(N),FJAC(LDFJAC,N),R(LR),QTF(N),WA1(N),
  &       WA2(N),WA3(N),WA4(N)
   CHARACTER*4 LABEL(N)

   COMMON  /FILES/ IFILE1,IFILE2,IFILE3,IFILE4,IFILE5,IFILE6,INP
   COMMON  /CHRONO/ TIME,TSTEP,TTIME,TMIN,ITIME
```



```
C************************************************************************
C     PHASE II - ADITYA VARANASI
   COMMON   /CHRON2/ TSTOP
      COMMON   /RPLOT/  PREVX,PREVY
      COMMON   /RDIAG/
RESMON,IFILERESD,RESITER,NUMITER,YSCALE,TMAXT,
  &                       YSCALENAME
      COMMON   /FORM/  CURRENTUNIT
      COMMON   /TOLERANCE/ ONETOL,PTOL,FTOL,TTOL,CTOL,RRTOL,ETOL,
  &           PWTOL,HTOL
      COMMON   /RESTOLERANCE/ ONETOLRES,PTOLRES,FTOLRES,TTOLRES,
  &           CTOLRES,RRTOLRES,ETOLRES,PWTOLRES, HTOLRES
      REAL(8) PLOTMAXY, PLOTMINY
      CHARACTER*7 YSCALENAME
      INTEGER      PLOTREPLY
      INTEGER XVALUE, YVALUE
      TYPE (xycoord) XY
      TYPE (rccoord) rc
      REAL YAXISV
      CHARACTER*20 SVALUE
      REAL TOLVALUES(N),RHSVALUE,TOLXK(N),TOLVALUESRES(N),SATISFY
      REAL RESIDUALS(N)
      REAL(8) yaxisj
      REAL(8) RANGE1
      INTEGER RANGEFOUND, RANGELOCATION
      REAL PLOTNUM1, PLOTNUM2, PLOTNUM3, PLOTNORM
C************************************************************************
```

```
C
   DATA    ONE,P1,P5,P001,P0001,ZERO
  &       /1.0E0,1.0E-1,5.0E-1,1.0E-3,1.0E-4,0.0E0/
C
   EPSMCH = R1MACH(4)
   XNORM=0.
C
   INFO = 0
   IFLAG = 0
   NFEV = 0
```

```
      NJEV = 0
C
C   Check the input parameters for errors.
C
     IF (IOPT .LT. 1 .OR. IOPT .GT. 2 .OR.
    &  N .LE. 0 .OR. XTOL .LT. ZERO .OR. MAXFEV .LE. 0
    &  .OR. ML .LT. 0 .OR. MU .LT. 0 .OR. FACTOR .LE. ZERO
    &  .OR. LDFJAC .LT. N .OR. LR .LT. (N*(N + 1))/2) GO TO 300
     IF (MODE .NE. 2) GO TO 20
     DO 10 J = 1, N
       IF (DIAG(J) .LE. ZERO) GO TO 300
  10   CONTINUE
  20 CONTINUE
C
C   Evaluate the function at the starting point
C   and calculate its norm.
C
     IFLAG = 1
     CALL FCN(N,X,FVEC,IFLAG,IBLK,ISBLK)
     NFEV = 1
     IF (IFLAG .LT. 0) GO TO 300
     FNORM = ENORM(N,FVEC)
C
C   Initialize iteration counter and monitors.
C
     ITER = 1
     NCSUC = 0
     NCFAIL = 0
     NSLOW1 = 0
     NSLOW2 = 0
C*************************************************************************
C       PHASE II - ADITYA VARANASI
C       Before the outer loop begins...
C       Store the Residual Vector.
        DO 11 J = 1,N
                RESIDUALS(J) = FVEC(J)
11      CONTINUE
C       Check if different tolerances for different categories are specified.
        IF (ONETOL.EQ.0.0) THEN
C               Build the error tolerance array depending on the category of variables.
                DO 21 J = 1,N
                        IF (LABEL(J).EQ.'PRES') THEN
                                TOLVALUES(J) = PTOL
                        ELSE IF (LABEL(J).EQ.'FLOW') THEN
                                TOLVALUES(J) = FTOL
                        ELSE IF (LABEL(J).EQ.'TEMP') THEN
                                TOLVALUES(J) = TTOL
                        ELSE IF (LABEL(J).EQ.'CTRL') THEN
                                TOLVALUES(J) = CTOL
                        ELSE IF (LABEL(J).EQ.'OTHR') THEN
                                TOLVALUES(J) = RRTOL
```

```
                  ELSE IF (LABEL(J).EQ.'ENRG') THEN
                        TOLVALUES(J) = ETOL
                  ELSE IF (LABEL(J).EQ.'POWR') THEN
                        TOLVALUES(J) = PWTOL
                  ELSE IF (LABEL(J).EQ.'HUMT') THEN
                        TOLVALUES(J) = HTOL
                  END IF
21          CONTINUE
      END IF
C     Check if tolerance values are specified for residual functions.
      IF (ONETOLRES.EQ.0.0) THEN
C           Build the tolerance array depending on the category of variables.
            DO 22 J = 1,N
                  IF (LABEL(J).EQ.'PRES') THEN
                        TOLVALUESRES(J) = PTOLRES
                  ELSE IF (LABEL(J).EQ.'FLOW') THEN
                        TOLVALUESRES(J) = FTOLRES
                  ELSE IF (LABEL(J).EQ.'TEMP') THEN
                        TOLVALUESRES(J) = TTOLRES
                  ELSE IF (LABEL(J).EQ.'CTRL') THEN
                        TOLVALUESRES(J) = CTOLRES
                  ELSE IF (LABEL(J).EQ.'OTHR') THEN
                        TOLVALUESRES(J) = RRTOLRES
                  ELSE IF (LABEL(J).EQ.'ENRG') THEN
                        TOLVALUESRES(J) = ETOLRES
                  ELSE IF (LABEL(J).EQ.'POWR') THEN
                        TOLVALUESRES(J) = PWTOLRES
                  ELSE IF (LABEL(J).EQ.'HUMT') THEN
                        TOLVALUESRES(J) = HTOLRES
                  END IF
22          CONTINUE
      END IF
C**********************************************************************
C
C    Beginning of the outer loop.
C
  30 CONTINUE
     JEVAL = .TRUE.
C
C     Calculate the Jacobian matrix.
C
...
...
...
C
C       Test for successful iteration.
C
       IF (RATIO .LT. P0001) GO TO 260
C
C       Successful iteration. Update X, FVEC, and their norms.
C
```

```
C*********************************************************************
C       PHASE II - ADITYA VARANASI
        DO 250 J = 1, N
          X(J) = WA2(J)
          WA2(J) = DIAG(J)*X(J)
C                       Build the arrays for the convergence test.
C                       XTOL(:)*X(:) Vector.
                        IF (ONETOL.EQ.0.0) THEN
                              TOLXK(J) = TOLVALUES(J)*WA2(J)
                        ELSE
                              TOLXK(J) = XTOL*WA2(J)
                        END IF
          FVEC(J) = WA4(J)
C                       Store the new Residuals vector.
                        RESIDUALS(J) = FVEC(J)
 250      CONTINUE
        XNORM = ENORM(N,WA2)
        FNORM = FNORM1
C                 Calculate || XTOL(:)*X(:) ||
                  RHSVALUE = ENORM(N,TOLXK)
        ITER = ITER + 1
 260    CONTINUE
C*********************************************************************
C
C        Determine the progress of the iteration.
C
        NSLOW1 = NSLOW1 + 1
        IF (ACTRED .GE. P001) NSLOW1 = 0
        IF (JEVAL) NSLOW2 = NSLOW2 + 1
        IF (ACTRED .GE. P1) NSLOW2 = 0
C
C        Test for convergence.
C
C*********************************************************************
C       PHASE II - ADITYA VARANASI
C       Old convergence criteria:
C        IF (DELTA .LE. XTOL*XNORM .OR. FNORM .EQ. ZERO) INFO = 1
C       New Convergence criteria:
C       If FNORM is zero then no further checks required.
        IF (FNORM.EQ.ZERO) THEN
              INFO = 1
        ELSE
C             Check if the Residual Tolerances are specified...
              IF (ONETOLRES.EQ.1.0) THEN
C                   Residual tolerances not specified.
C                   Check if DELTA <= || XTOL(:)*X(:) ||..
                    IF (DELTA .LE. RHSVALUE) THEN
C                         Convergence criteria met.
                          INFO = 1
                    END IF
              ELSE
```

```
C           Residual tolerances specified.
C           Check the Residual vector against the tolerance vector.
            DO J = 1,N
               IF (ABS(RESIDUALS(J)).LE.TOLVALUESRES(J)) THEN
C                 Condition satisfied for this Residual Function.
                  SATISFY = 1.0
               ELSE
C                 Condition not satisfied for this Residual Function.
                  SATISFY = 0.0
C                 Exit the loop. Convergence criteria not met.
                  EXIT
               END IF
            END DO
            IF (SATISFY.EQ.1.0) THEN
C              All conditions are satisfied. Convergence criteria met.
               INFO = 1
            END IF
         END IF
      END IF

C*****************************************************************
C*****************************************************************
C     PHASE II - ADITYA VARANASI
      IF (RESMON.EQ.1.0) THEN

      PLOTMAXY = 1000.0
      PLOTMINY = 0.00001

C      PLOT THE RESIDUALS.
C    Set the control to the plotting window.
       PLOTREPLY = SETACTIVEQQ(9999)
C      Calculate the x-axis and y-axis values.
       XVALUE = 1 + INT(MOD(RESITER*4, 779.0))
C      Identify the Y-Axis scale.
       IF (YSCALENAME.EQ.'LINEAR' .OR. YSCALENAME.EQ.'linear'
  & .OR. YSCALENAME.EQ.'Linear') THEN
C         Linear scale on Y-Axis.
          YVALUE = 400 - FNORM*YSCALE
C     Check if FNORM*SCALE is > 400, implying YSCALE is LARGE.
          IF (YVALUE .NE. ABS(YVALUE)) THEN
             YVALUE = 0
          END IF
       ELSE
C         Log scale on Y-Axis.
          RANGE1 = PLOTMAXY
          RANGELOCATION = 2
          FOUND = 0
          PLOTNORM = FNORM
          IF (FNORM .LT. PLOTMINY) THEN
             FNORM = PLOTMINY
          END IF
            DO
                IF (RANGE1.LE.PLOTMINY) THEN
                    EXIT
                END IF
                IF (FNORM.LT.RANGE1 .AND. FNORM.GE.(RANGE1/10)) THEN
                    FOUND = 1
                    EXIT
                ELSE
                    RANGE1 = RANGE1/10
                    RANGELOCATION = RANGELOCATION - 1
                END IF
            END DO
            IF (FOUND.EQ.1) THEN
                PLOTNUM1 = FNORM*(10.0**(-RANGELOCATION*1.0))
                PLOTNUM2 = LOG10(PLOTNUM1)*YSCALE
                PLOTNUM3 = PLOTNUM2 + ((RANGELOCATION+5)*YSCALE)
                YVALUE = 400 - PLOTNUM3
            ELSE
                YVALUE = 400
            END IF
            FNORM = PLOTNORM
        END IF
C       Check if the screen needs to be cleared (when the plot region is full).
        IF (XVALUE.GE.1 .AND. XVALUE.LE.4) THEN
         CALL CLEARSCREEN($GCLEARSCREEN)
         CALL SETVIEWPORT(0,0,1000,700)
         PLOTREPLY = SETCOLOR(12)
         PLOTREPLY = RECTANGLE($GBORDER, 5,5, 900,560)
         PLOTREPLY = SETCOLOR(6)
         PLOTREPLY = RECTANGLE($GBORDER, 121,71, 899,559)
         PLOTREPLY = RECTANGLE($GBORDER, 122,72, 898,558)
         PLOTREPLY = SETCOLOR(12)
         PLOTREPLY = RECTANGLE($GBORDER, 123,73, 897,557)
         PLOTREPLY = SETCOLOR(15)
         PLOTREPLY = FLOODFILL(320,480,12)
C        Set the view port coordinates.
         CALL SETVIEWPORT(124,74,896,556)
         CALL MOVETO(0,480, XY)
         PREVX = 0
         PREVY = 480
        END IF
C       Decide the plot color depending on the iteration number.
        PLOTCOLOR = 1 + INT(MOD(ITIME,14))
C       Set the color of the items.
        PLOTREPLY = SETCOLOR(PLOTCOLOR)
C       Draw the item rectangle.
        PLOTREPLY = RECTANGLE($GFILLINTERIOR, XVALUE,400, XVALUE+2,
     & YVALUE)
C       Initialize the text fonts to defaults.
        PLOTREPLY = INITIALIZEFONTS()
```

```
C          If the iterations have converged, draw a circle and show the time.
           IF (INFO .EQ. 1) THEN
C            Draw a circle.
             PLOTREPLY = ELLIPSE($GFILLINTERIOR, XVALUE, YVALUE-5,
XVALUE+2,
    &   YVALUE-10)
C            Convert real data to string.
             WRITE (SVALUE, '(F0.2)') TIME
C            Set the text font.
             PLOTREPLY = SETFONT('t"Arial"h10w5pv')
C            Provide a vertical rotation to the text.
             CALL SETGTEXTROTATION(900)
C            Move to a location where etxt has to be written.
             CALL MOVETO(XVALUE - 10, 470, XY)
C            Write the time value.
             CALL OUTGTEXT(SVALUE)
C            Rotate the text to horizontal direction.
             CALL SETGTEXTROTATION(0)
           WRITE(IFILERESD, *) TIME, FNORM, NUMITER
             NUMITER = 1
           ELSE
             NUMITER = NUMITER + 1
             WRITE(IFILERESD, *) TIME, FNORM, 0
           END IF
C          Update the necessary variables.
           PREVY = YVALUE
           PREVX = XVALUE
C          Update the plot.
C          Set the clip region.
           CALL SETVIEWPORT(0,0,1000,700)
C          Set the color of the items.
           PLOTREPLY = SETCOLOR(15)
C          Update the title.
C          Set the title text font.
           PLOTREPLY = SETFONT('t"Arial"h18w10pvb')
           CALL MOVETO (140, 20, XY)
           CALL OUTGTEXT('REPORTING RESIDUALS (Euclidean Norm of Residual
    &Vector Vs Iteration)')
C       Show the simulation times.
           PLOTREPLY = SETFONT('t"Arial"h18w5pv')
C                 Minimum time step.
           CALL MOVETO (150, 45, XY)
           CALL OUTGTEXT ('Minimum Time Step :')
           CALL MOVETO (260, 45, XY)
     WRITE (SVALUE, '(F13.2)') (TMIN)
           CALL OUTGTEXT(SVALUE)
C                 Maximum time step.
           CALL MOVETO (390, 45, XY)
           CALL OUTGTEXT ('Maximum Time Step :')
           CALL MOVETO (500, 45, XY)
     WRITE (SVALUE, '(F13.2)') (TMAXT)
```

```
      CALL OUTGTEXT(SVALUE)
C          Simulation stopping time.
      CALL MOVETO (630, 45, XY)
      CALL OUTGTEXT ('Stopping Time :')
      CALL MOVETO (720, 45, XY)
   WRITE (SVALUE, '(F13.2)') (TSTOP)
      CALL OUTGTEXT(SVALUE)
C     Update the y-axis value.
C     Set the axis text font.
      PLOTREPLY = SETFONT('t''Arial''h18w5pv')
      IF (YSCALENAME.EQ.'LINEAR' .OR. YSCALENAME.EQ.'linear'
  & .OR. YSCALENAME.EQ.'Linear') THEN
C          Linear scale on Y-Axis.
           DO I = 400, 0, -50
                WRITE (SVALUE, '(F13.3)') ((400.0 - I)/YSCALE)
                CALL MOVETO(45, 74+I-9, XY)
                CALL OUTGTEXT(SVALUE)
           END DO
      ELSE
C          Log scale on Y-Axis.
           yaxisj = PLOTMINY
           DO I = 400, 0, -YSCALE
                WRITE (SVALUE, '(F13.5)') yaxisj
                yaxisj = yaxisj * 10.0
                CALL MOVETO(45, 74+I-9, XY)
                CALL OUTGTEXT(SVALUE)
           END DO
      END IF

C     Show additional inforation about the plot and simulation below the plot.
      PLOTREPLY = SETFONT('t''Arial''h18w5pv')
C          Plot Properties.
      CALL MOVETO (125, 570, XY)
      CALL OUTGTEXT ('PLOT PROPERTIES : ')
C          Y-Axis scale.
      CALL MOVETO (125, 590, XY)
      IF (YSCALENAME.EQ.'LINEAR' .OR. YSCALENAME.EQ.'linear'
  & .OR. YSCALENAME.EQ.'Linear') THEN
           CALL OUTGTEXT ('Y-Axis Scale : LINEAR')
C          Y-Axis scaling factor.
       CALL MOVETO (275, 590, XY)
       CALL OUTGTEXT ('Y-Axis Magnification : ')
       CALL MOVETO (385, 590, XY)
    WRITE (SVALUE, '(F6.2)') (YSCALE)
       CALL OUTGTEXT (SVALUE)
      ELSE
           CALL OUTGTEXT ('Y-Axis Scale : SEMI-LOG')
C          Y-Axis scaling factor.
       CALL MOVETO (275, 590, XY)
       CALL OUTGTEXT ('Y-Axis Cycle Length :')
       CALL MOVETO (385, 590, XY)
```

```
     WRITE (SVALUE, '(F6.2)') (YSCALE)
        CALL OUTGTEXT (SVALUE)
       END IF
C        List the error tolerance involved in the simulation.
       CALL MOVETO (125, 615, XY)
       CALL OUTGTEXT ('SIMULATION ERROR TOLERANCE PROPERTIES : ')
C        Display colun headers.
C                 Since error tolerances are based on the variable category.
C                 Display a list of available categories.
       CALL MOVETO (125, 635, XY)
       CALL OUTGTEXT ('Categories')
C                 XTOL values for each category.
       CALL MOVETO (125, 650, XY)
       CALL OUTGTEXT ('Equation Solver Tolerances')
C                 Tolerances for Residual functions.
       CALL MOVETO (125, 665, XY)
       CALL OUTGTEXT ('Residual Vector Tolerances')
C        Grid seperators.
       CALL MOVETO (275, 665, XY)
       CALL OUTGTEXT (':')
       CALL MOVETO (275, 650, XY)
       CALL OUTGTEXT (':')
       CALL MOVETO (275, 635, XY)
       CALL OUTGTEXT (':')
C            Display data in the table.
C                 List of available categories.
       CALL MOVETO (290, 635, XY)
       CALL OUTGTEXT ('PRESSURE | FLOW-RATE | TEMPERATURE |
    & CONTROL-SIGNAL | ROTATION-RATE | ENERGY | POWER | HUMIDITY')
C                 Grid seperators.
C                      After PRESSURE
       CALL MOVETO (352, 650, XY)
       CALL OUTGTEXT ('|');
C                      After FLOW-RATE
       CALL MOVETO (427, 650, XY)
       CALL OUTGTEXT ('|');
C                      After TEMPERATUE
       CALL MOVETO (518, 650, XY)
       CALL OUTGTEXT ('|');
C                      After CONTROL-SIGNAL
       CALL MOVETO (629, 650, XY)
       CALL OUTGTEXT ('|');
C                      After ROTATION-RATE
       CALL MOVETO (729, 650, XY)
       CALL OUTGTEXT ('|');
C                      After ENERGY
       CALL MOVETO (784, 650, XY)
       CALL OUTGTEXT ('|');
C                      After POWER
       CALL MOVETO (835, 650, XY)
       CALL OUTGTEXT ('|');
```

```
C               Error tolerance list.
      IF (ONETOL.EQ.1.0) THEN
C                         PRESSURE - Equation solver tolerance.
          WRITE (SVALUE, '(F9.6)') (XTOL)
          CALL MOVETO(290, 650, XY)
          CALL OUTGTEXT(SVALUE)
C                         FLOW-RATE - Equation solver tolerance.
          WRITE (SVALUE, '(F9.6)') (XTOL)
          CALL MOVETO(362, 650, XY)
          CALL OUTGTEXT(SVALUE)
C                         TEMPERATURE - Equation solver tolerance.
          WRITE (SVALUE, '(F9.6)') (XTOL)
          CALL MOVETO(450, 650, XY)
          CALL OUTGTEXT(SVALUE)
C                         CONTROL-SIGNAL - Equation solver tolerance.
          WRITE (SVALUE, '(F9.6)') (XTOL)
          CALL MOVETO(550, 650, XY)
          CALL OUTGTEXT(SVALUE)
C                         ROTATION-RATE - Equation solver tolerance.
          WRITE (SVALUE, '(F9.6)') (XTOL)
          CALL MOVETO(659, 650, XY)
          CALL OUTGTEXT(SVALUE)
C                         ENERGY - Equation solver tolerance.
          WRITE (SVALUE, '(F9.6)') (XTOL)
          CALL MOVETO(732, 650, XY)
          CALL OUTGTEXT(SVALUE)
C                         POWER - Equation solver tolerance.
          WRITE (SVALUE, '(F9.6)') (XTOL)
          CALL MOVETO(786, 650, XY)
          CALL OUTGTEXT(SVALUE)
C                         HUMIDITY - Equation solver tolerance.
          WRITE (SVALUE, '(F9.6)') (XTOL)
          CALL MOVETO(840, 650, XY)
          CALL OUTGTEXT(SVALUE)
      ELSE
C                         PRESSURE - Equation solver tolerance.
          WRITE (SVALUE, '(F9.6)') (PTOL)
          CALL MOVETO(290, 650, XY)
          CALL OUTGTEXT(SVALUE)
C                         FLOW-RATE - Equation solver tolerance.
          WRITE (SVALUE, '(F9.6)') (FTOL)
          CALL MOVETO(362, 650, XY)
          CALL OUTGTEXT(SVALUE)
C                         TEMPERATURE - Equation solver tolerance.
          WRITE (SVALUE, '(F9.6)') (TTOL)
          CALL MOVETO(450, 650, XY)
          CALL OUTGTEXT(SVALUE)
C                         CONTROL-SIGNAL - Equation solver tolerance.
          WRITE (SVALUE, '(F9.6)') (CTOL)
          CALL MOVETO(550, 650, XY)
          CALL OUTGTEXT(SVALUE)
```

```
C                               ROTATION-RATE - Equation solver tolerance.
            WRITE (SVALUE, '(F9.6)') (RRTOL)
            CALL MOVETO(659, 650, XY)
            CALL OUTGTEXT(SVALUE)
C                               ENERGY - Equation solver tolerance.
            WRITE (SVALUE, '(F9.6)') (ETOL)
            CALL MOVETO(732, 650, XY)
            CALL OUTGTEXT(SVALUE)
C                               POWER - Equation solver tolerance.
            WRITE (SVALUE, '(F9.6)') (PWTOL)
            CALL MOVETO(786, 650, XY)
            CALL OUTGTEXT(SVALUE)
C                               HUMIDITY - Equation solver tolerance.
            WRITE (SVALUE, '(F9.6)') (HTOL)
            CALL MOVETO(840, 650, XY)
            CALL OUTGTEXT(SVALUE)
        ENDIF
C                     Grid seperators.
C                               After PRESSURE
        CALL MOVETO (352, 665, XY)
        CALL OUTGTEXT ('|');
C                               After FLOW-RATE
        CALL MOVETO (427, 665, XY)
        CALL OUTGTEXT ('|');
C                               After TEMPERATUE
        CALL MOVETO (518, 665, XY)
        CALL OUTGTEXT ('|');
C                               After CONTROL-SIGNAL
        CALL MOVETO (629, 665, XY)
        CALL OUTGTEXT ('|');
C                               After ROTATION-RATE
        CALL MOVETO (729, 665, XY)
        CALL OUTGTEXT ('|');
C                               After ENERGY
        CALL MOVETO (784, 665, XY)
        CALL OUTGTEXT ('|');
C                               After POWER
        CALL MOVETO (835, 665, XY)
        CALL OUTGTEXT ('|');
C                 Residual vector tolerance list.
C                               PRESSURE - Equation solver tolerance.
        WRITE (SVALUE, '(F9.6)') (PTOLRES)
        CALL MOVETO(290, 665, XY)
        CALL OUTGTEXT(SVALUE)
C                               FLOW-RATE - Equation solver tolerance.
        WRITE (SVALUE, '(F9.6)') (FTOLRES)
        CALL MOVETO(362, 665, XY)
        CALL OUTGTEXT(SVALUE)
C                               TEMPERATURE - Equation solver tolerance.
        WRITE (SVALUE, '(F9.6)') (TTOLRES)
        CALL MOVETO(450, 665, XY)
```

```
      CALL OUTGTEXT(SVALUE)
C                        CONTROL-SIGNAL - Equation solver tolerance.
      WRITE (SVALUE, '(F9.6)') (CTOLRES)
      CALL MOVETO(550, 665, XY)
      CALL OUTGTEXT(SVALUE)
C                        ROTATION-RATE - Equation solver tolerance.
      WRITE (SVALUE, '(F9.6)') (RRTOLRES)
      CALL MOVETO(659, 665, XY)
      CALL OUTGTEXT(SVALUE)
C                        ENERGY - Equation solver tolerance.
      WRITE (SVALUE, '(F9.6)') (ETOLRES)
      CALL MOVETO(732, 665, XY)
      CALL OUTGTEXT(SVALUE)
C                        POWER - Equation solver tolerance.
      WRITE (SVALUE, '(F9.6)') (PWTOLRES)
      CALL MOVETO(786, 665, XY)
      CALL OUTGTEXT(SVALUE)
C                        HUMIDITY - Equation solver tolerance.
      WRITE (SVALUE, '(F9.6)') (HTOLRES)
      CALL MOVETO(840, 665, XY)
      CALL OUTGTEXT(SVALUE)

C     Set the view port coordinates.
      CALL SETVIEWPORT(124,74,896,556)

      RESITER = RESITER + 1

C   Set the font in the default window.
      PLOTREPLY = SETACTIVEQQ(CURRENTUNIT)
      PLOTREPLY = INITIALIZEFONTS()
      PLOTREPLY = SETFONT('t"COURIER"h16w10')
     END IF
C*********************************************************************
C**DEBUG
    IF(NPRINT.EQ.2) THEN
      XTNORM=XTOL*XNORM
      WRITE(IFILE3,668) DELTA,XTNORM
 668    FORMAT(' DELTA, XTNORM AT CONVERGENCE TEST:',1P2G15.6)
    ENDIF
C**END DEBUG
      IF (INFO .NE. 0) GO TO 300
C
C      Tests for termination and stringent tolerances.
C
...
...
...
C *********************************************************************
```

2

VITA

Aditya Varanasi

Candidate for the Degree of

Master of Science

Thesis: DEVELOPMENT OF A VISUAL TOOL
FOR HVACSIM+

Major Field: Mechanical Engineering

Biographical:

Personal: Born in Elluru, Andhra Pradesh, on Mar 16, 1977, to Sundaram Varanasi and Chitra Mitra.

Education: Received Bachelor of Engineering in Mechanical Engineering from Osmania University, Hyderabad, India in June 1998. Completed the requirements for the Master of Science degree with a major in Mechanical Engineering at Oklahoma State University in May, 2002

Experience: Employed by Oklahoma State University, Department of Mechanical Engineering as a graduate teaching assistant January 2000 to May 2000 and subsequently as a research assistant from June 2000 to date.

Professional Memberships: None