MODELLING AND SIMULATION
OF A
SOLAR ENERGY SYSTEM

Markus Weiner

Advisor:
Dr. François E. Cellier

A research report submitted to the
Department of Electrical and Computer Engineering
University of Arizona
and the
Institut für Thermodynamik und Wärmetechnik
Universität Stuttgart

in partial fulfillment of the requirements
for the Degree of
Diplomingenieur
at the
Universität Stuttgart

1 9 9 2

ACKNOWLEDGMENTS

I would like to thank my advisor Dr. François E. Cellier, Dept. of Electrical and Computer Engineering, for guidance and assistance during my studies at the University of Arizona.

Furthermore, I wish to acknowledge the co-operation of Dr. George Mignon and Dr. Nader V. Chalfoun of the Environmental Research Lab for introducing me to CALPAS 3 and DOE-2 and assisting me with the building description. I also would like to thank the Solar Energy Research Facility and Dr. Rocco Fazzolare for giving me access to his DOE-2 program.

Prof. Dr. Voss, Institut für Energiewirtschaft und Rationelle Energieanwendung, Universität Stuttgart, arranged and organized this student exchange program between Stuttgart and Tucson and the German Academic Exchange Service provided the financial support of the exchange.

I would like to thank Glenn Farrenkopf and Dr. François E. Cellier, who helped me prepare this report.

I am deeply indebted to my parents, whose encouragement and financial support made it possible for me to study in Germany and the USA.

Finally, I would like to express my gratitude to all my friends for their valued company and moral support.

Contents

Abstract

List of Figures

List of Tables

Abstract

A Building Analysis Program is a modelling and simulation tool that determines the thermodynamic behavior of a building structure. It helps to detect the influence of materials, architectural features and air-conditioning systems during a building's design as well as in an existing building's improvement in terms of climate control.

A structure model is assembled from provided submodels, which are representations of the basic physical components comprising the entire structure model. For an accurate model description, the user must supply sufficient data describing the thermodynamic characteristics of the materials modeled, and must define the submodel interconnections.

This thesis deals with a new methodology for modelling a system's basic components and their interconnections so that an accurate model of the entire system can be feasibly constructed and simulated. The bond graph technique is used to model the physical laws of heat transfer between the basic components, and the modelling language DYMOLA is used to provide a convenient platform for creating hierarchical and modular component descriptions in very readable code. The bond graphs can be coded directly into DYMOLA, which can generate code for a simulation language such as ACSL. ACSL is then used to obtain the numerical solution of the equations which describe the system.

The goal of this work is to prove that continuous system simulation using bond graphs coded in DYMOLA are feasible for building analysis in terms of modelling capabilities, accuracy of results, and computation time.

This new approach is compared with two state-of-the-art building analysis programs, namely CALPAS3 and DOE-2. For the comparison, the thermal performance of a test house is determined by all three programs and the results for temperature profiles and heat flow rates have been found to be very similar. DYMOLA proves to be an excellent modelling tool and very suitable for modelling systems with modular components. However, the high accuracy achieved with continuous system simulation requires considerably longer computation times.

1 Introduction

1.1 Motivation

Since 1970 there has been a tremendous surge of interest and research in solar energy systems, mainly due to the increasing cost and diminishing supply of nonrenewable energy. A re-evaluation of energy use is required and concepts of infinite and less expensive energy supplies should be considered. Long ago it was discovered that our sun's energy could be used to provide heat for comfort in our buildings. Some of the old Indian dwellings in Arizona show the perfect use of passive solar heating by the use of thick mud walls with high heat capacity. But to achieve optimal thermal performance in today's buildings, advances must be made in the technical and economical aspects of solar heating devices.

Solar heating may be applied to both *active* and *passive systems.* Active solar heating systems contain elements like collectors, storage units, and distribution equipment, while passive solar heating systems use only the building structure to collect and store the energy as well as to distribute it to the interior. Passive solar heating performance is highly dependent on architectural design. The goal is to maximize solar energy input in winter and minimize it in summer. This may be accomplished by construction elements with high heat capacity and movable insulation or shading devices. Both types of systems will still depend on auxiliary energy sources in order to provide the desired level of comfort and reliability, but the aim will always be to minimize the *loads*, that is, the cooling load and the heating load.

We also must evaluate our passive solar heated system's efficiency. The least cost method in order to achieve comfortable living conditions might be used. However, the lowest costs to achieve this demand will be always found in a combination of solar supporting construction and backup heating/cooling device.

Point of interest hereby is the thermal performance of the building in order to determine the loads and devise the necessary auxiliary system. The thermal performance is influenced by building materials, architecture, building construction and schedules of occupation. The choice of building materials and the change of constructive elements and their influence on thermal, technical, and economical performance must be determined in advance, only then the optimal value can be obtained in a less expensive way. Many parameters must be evaluated to optimize the general design, i.e., the structural envelope. Modelling and simulation in advance of construction have become indispensable in finding the right performance data and providing essential information for the engineer or architect to make his/her decision.

1.2 Simulation of Thermal Processes

The intention for simulating a building as a thermal system can be various. There are many aspects on which designers need to focus. For example, the room conditions, heat gains or losses, the influence of heat storage, or more generally the load profiles that influence the thermal performance. Flexibility in construction of the elements used and a high variability in elements are important to represent many different kind of buildings. The purpose of all simulations will always be to predict the thermal behavior of the modelled building, but the intention of the user might differ. An engineer will put more emphasis on load profiles in order to devise the backup system, while an architect is more concerned about high flexibility in order to express a wide range of architectural design.

different applications Considering the alternatives in architectural design in contrast to investigations related to the usage of new building materials or even the focus on different heating and cooling systems shows us different points of view. The wide range of applications for a building analysis program is reaching from the architectural over the material science to the air-conditioning engineering approach. Each approach emphasizes a different detail of thermal performance. The simulation program, however, should satisfy all these demands while taking the various professional backgrounds into account.

flexible inputs According to the different approaches the input data contains informations about building size, building design, systems and building use. The *size* of the simulated structure can range from a residential home or a small office building to a huge, multiple story commercial building. The *design* of the structure might specify different building materials, multilayer walls, architectural designs like bay-windows, or passive solar devices attached to or incorporated into the building. Diverse *systems* may be used to cool and supply heat to the living space, such as air-conditioning distribution systems or radiators. The systems might consist of conventional backup systems such as a gas furnace or be supported by active or passive solar systems. The *building use* determines schedules for lighting and internal gain and the peak hours of heating and cooling demand.

flexible outputs Various applications focus on different parameters in order to determine and optimize them. These values must be provided in an output-file which is flexible enough to satisfy the intention of the user.

accuracy To predict and optimize the thermal performance of the future building, we need a high reliability in the simulation. In order to resemble reality as closely as possible, the simulation needs to accurately model the law of physics. The modelling technique used in the simulation must therefore provide the capability to express physical laws.

computation speed and costs Simulation costs are directly related to computation times. There is great incentive to use a simulation program if practical computation times can be expected and if the effort required to get familiar with the program is not too great. This certainly has something to do with the applied methodology but also with the right documentation.

discrete vs continuous simulation One part of the methodology required to tackle the problems of building analysis simulation is, certainly, the decision of whether to use discrete or continuous simulation techniques. We must determine the frequency of the model evaluation, which has an impact on the accuracy of our calculations and on the computational efficiency we will obtain.

The conventional simulation programs for building analysis have basically used the discrete-time approach to calculate heat transfer or heat storage during a certain time Δt. In discrete-time models the time axis is discretized so that the state variable changes once from one time step t_0 to the next $t_0 + \Delta t$. Therefore, they are represented through a set of *difference equations*.

By using continuous-time simulation, step sizes will be reduced to a sufficiently small increment in order to accurately model the system's behavior. Therefore, the state variables change constantly with the time variable as expressed in the following definition [1]:

> 'Continuous-time models are characterized by the fact that
> within a finite time span the state variables change their values
> infinitely often.'

Typically, continuous-time models are described in the form of *ordinary differential equations* (ODE's) or *partial differential equations* (PDE's), which are solved numerically using a integration algorithm. This algorithm will determine the accuracy of calculations and the frequency of state variable evaluation by providing either a fixed step size or variable step sizes.

1.3 Modelling of Thermal Processes

The modelling technique is very important to achieve accurate results. It also influences the flexibility of the program during the building description. Therefore, it should provide a precise expression of the heat transfer phenomena. Along with this the modelling technique must be able to apply to a wide range of building designs.

In order to achieve this we need basic models which are hierarchically structured and model the heat transfer through constructive elements like walls, roofs, doors, etc. These models can be easily connected to each other without bothering about the heat transfer equations. Because they model basic heat transfer phenomena those model types can be used not only to express the envelope but also to determine the thermal performance of passive solar concepts. These are direct gain through exposed windows, collector-storage walls combining a massive wall and an outside glazing, or green house attachments.

Furthermore, we are able to find the different mechanisms of heat transfer like conduction, convection, radiation, infiltration, ventilation, etc., in hierarchical models so that we can use them for any desired purpose.

As mentioned above, we want to express physical laws and equations very closely to obtain proper calculations. Therefore, we will introduce a new modelling technique using bond graphs. We will use continuous time simulation and a modelling language called DYMOLA, which is very capable for bond graph modelling. This approach combines high accuracy with excellent modelling properties.

We will compare the new approach with two state-of-the-art simulation programs for building envelopes, namely CALPAS3 [2] and DOE-2 [3].

2 Modelling of a Building Structure

2.1 The Test Building

The structure we want to use for the comparison of our simulations is a building which is part of a number of test structures at the *Environmental Research Laboratory* (ERL) of the University of Arizona. In order to focus on conductive heat transfer through high mass elements, a massive adobe structure has been chosen.

This building was set up in the early eighties as a project of the U.S. Department of Housing and Urban Development to fit the requirements of a low-cost solar home design for the Navajo Indians in Northern Arizona. The structure is built of 16″-thick asphalt emulsion stabilized adobe, with large south-facing windows, a tilted metal roof, and an attached solar/screen porch on the south side. It is comprised of a large livingroom, kitchen and bathroom placed on the south, and two bed rooms on the north and north-east side. The floorplan for the whole structure can be found in Appendix C.

Adobe is the traditional building material in the Southwest and appropriate for use as a thermal storage building material in passively cooled and heated homes. It provides a long time lag for temperature change and retards the flow of heat into the home in the summer and out of the home in the winter. Furthermore, adobe embodies low energy and material costs because it basically contains sand, fine gravel and clay mixed with water and asphalt emulsion as a stabilizer.

The solar porch or sunspace is used as a sun-collector. Hereby, we take advantage of the effect that low-mass structures follow the outside temperature distribution better than the high-mass building. Allowing irradiation through the large glazing areas, the sunspace can supply the building with heat during the winter. The heat exchange between both spaces can be provided by forced convection with a fan or conduction through the common wall.

Limitations and Assumptions The three programs which we will compare are not expected to perform identically. The differences in the ability to describe the building and determine its thermal performance can be found in the physical and geometrical description of the structure and its formal specification.

Additionally, our test structure is not inhabited and only rarely equipped. This leads us to some assumptions we made, in order to describe this test building:

1. The absence of inhabitants changes the behavior of the structure by means of *internal gain*, *additional mass*, heat exchange through occasionally *open doors and windows* and there is also no *schedule.*
    - Internal Gain is the heat source occurring due to lighting, people and appliances. The rates of heat given off depend very much on season, environmental conditions and activity and follow different schedules. Due to the absence of occupants and equipment, internal gain has been omitted.
    - Additional mass, mainly furniture, rags, carpet, and tapestry, has also been neglected, since none of the spaces are furnished. That means only the air capacity inside of the space is relevant to heat storage.
    - It has been assumed that all doors are closed and heat transfer only occurs by conduction and convection, whereas doors and window frames are considered low mass elements.
2. In order to control the radiation heat gain the test building is equipped with Venetian Blinds at the windows. Depending on their position, radiation will be totally reflected, absorbed, or partially emitted to the space. This procedure relies on a schedule and none of the programs were capable of determining the solar gain dependent on the incline of the blinds. Therefore, the building was modelled without the blinds.
3. The heat transfer between sunspace and house was only modelled by convection and conduction through high and low mass elements of the common wall. Although a duct and a fan is installed in the house to provide forced convection we will assume that nobody will operate it.
4. Ventilation and Infiltration are both effected by different conditions inside and outside the building. While ventilation is the intentional air exchange by using a fan or opening a window, infiltration is the random flow of air through openings. Both phenomena result in a closer relationship between inside and outside air temperatures.

Although the modelling is fairly simple, the exchange rate is tricky to determine and the effect will cut short all other heat exchange mechanisms. For this reason and expecting a result which is a little more distinct, we decided to suppress Ventilation and Infiltration.

2.2 Building Interpretation of CALPAS3 and DOE-2

Besides providing the implementation of an alternative bond graph model for expressing the thermal performance of a building, another main purpose of this work is to provide the comparison of this approach with state-of-the-art building analysis programs. For the realization of this part two software packages have been chosen that are fairly well-known and very competitive on the market. One of them is the CALPAS3 [2] program, developed by the Berkeley Solar Group, the other is a software called DOE-2, supported by the U.S. Department of Energy. The latter was written at the Lawrence Berkeley Laboratory, California, Los Alamos National Laboratory, New Mexico, and Argonne National Laboratory, Illinois. Both programs have been used in a PC-version where the DOE-2 implementation was a commercial product called MICRO-DOE-2 Version 2.1 C [4].

All programs were run to obtain the thermal performance of the test building. Then, the results of the different approaches were compared for accuracy, modelling capability and flexibility. However, the main purpose of this thesis is to introduce the new approach. Therefore, the competitor products will not be explained in depth, for questions regarding details, but their user manuals may be refered to.

2.2.1 CALPAS3

CALPAS3 analyzes the energy performance of passive solar and conventional residences. It is easier to use than the larger DOE-2 program and according to the distributor, is still very reliable. The program is mainly used as a design and evaluation tool by architects, energy consultants and educators throughout the United States.

CALPAS3 provides a flexible input format in the form of an input file. This file can be created with any text editor following a certain syntax. CALPAS3 is capable of modelling the building envelope with an unlimited number of walls and windows. It also models mass elements including floor slabs, interior walls, exterior walls, mass wall with its own gazing, and the floor slab over a rockbed. Windows or other glazings are classified according to the glass properties, solar gain distribution, foreground reflectivity, or shading. Backup heating and cooling systems are automatically constructed by the program to establish a given thermostat set point for cooling and heating. Hourly internal gains from occupants, lights, cooking, etc, can be modelled using a residential schedule for the occurrence of these gains. Special passive solar features such as an attached sunspace, water walls, trombe walls and underslab rockbeds are also provided.

The CALPAS3 input file of the test structure is given in Appendix B. The code is fairly self-explanatory and contains all the information CALPAS3 needs to determine the thermal analysis of the test building. This input file will be submitted to CALPAS3 which checks for errors, simulates the building performance and produces an output file which is printed or viewed on the screen. The output file contains the reports chosen in the input description Those reports can provide monthly, daily or hourly data. Energy balance reports for each space, a solar gain report for sunlit construction elements or temperatures for the spaces can be obtained. For accuracy reasons hourly reports were chosen to determine energy balance and temperatures. In order to compare the solar gain calculations the solar gain report for several windows have been selected.

CALPAS3 is an hour-by-hour thermal network simulation program which basically uses the heat transfer methods suggested by ASHRAE [5]. For the calculation of transient conductive heat transfer in mass elements, it applies backward difference equations; this means CALPAS3 does not simultaneously calculate air and mass temperatures. Thus, the air temperature is first obtained by using an estimated mass temperature. In the next step, the mass temperatures are updated one by one according to the adequate surrounding air temperature. Besides that, CALPAS3 makes a number of simplifying assumptions in order to allow execution rapidly. These include the frequency of calculations of solar geometry and transmission factors for the glazing elements, which by default are calculated only on a monthly basis, as well as combined radiant/convective coefficients and constant film coefficients.

Solar gain calculations are made by deriving the transmission values for every window considering their orientation, declination, glass type, foreground reflectance and shading.

The transmission value is then applied to solar data taken from the hourly weather file. The weather file also supplies hourly dry-bulb and wet-bulb temperatures as well as wind speed and direction.

2.2.2 DOE-2

DOE-2 was developed as a building analysis program to explore the energy behavior of proposed and existing buildings and their associated heating, ventilation and air conditioning systems (HVAC). The result is a program that not only determines the thermal performance of a building but also reflects the interaction with heating and cooling systems. DOE-2 allows us to focus on distributing systems and devices for heating and cooling separately. It also provides for an economic analysis of the building design. The program consists of five sub-sequences:

- The *Building Design Language* translator. This is the first step of the modelling procedure. The program reads the user supplied data out of the input file and incorporates it into the models. At the end, response factors are calculated which are later used for time dependent heat flow through multilayer walls.

- The *Loads Simulation Sub-program.* This section calculates the hourly heating and cooling load for each user-designed space. These loads are derived using design-conditions, e.g., design-temperature set point at 70° F. The space is kept at design-conditions and loads are determined considering weather conditions and solar data. Other factors that influence the load calculation are time delay of heat transfer from massive walls and roofs as well as internal heat gain or shading.

- The *System Simulation Sub-program.* This is the so-called secondary HVAC simulation. It determines the loads of the HVAC system which are needed to obtain the required space loads. While the space loads are approximated for the design-day and the habitable space demand, system loads take into account the user specified comfort range, the operating schedule of the HVAC equipment and the outside air requirement in order to satisfy the temperature and humidity set points.

- The *Plant Simulation Sub-program*, the primary HVAC simulation subroutine. This part simulates the behavior of boilers, turbines, chiller, solar collectors, etc. These devices are predetermined by the required heat extraction and addition to the secondary system. This sub-program also calculates fuel and electrical demand as well as the costs of these sources.
- The *Economic Analysis Sub-program.* This section makes a life cycle cost analysis of different building and device design alternatives.

The input file is written in the syntax of the Building Design Language. This language is very well structured and allows the user to provide general data before actually describing the geometry of the building. Building materials and multilayer constructions can be predefined and assigned to a keyword. This keyword can then be used whenever such a construction occurs. Additionally, it can be referred to predefined schedules during the building description. Besides that, the input file contains data for the different sub-programs in separate sections. The commands at the beginning and end of those sections invoke the different sub-programs, e.g., `input loads .. end .. compute loads ...` The building description is provided in the loads section which also serves as the basis for the other program parts.

Every sub-program is able to provide the user with a variety of reports on either a monthly or an hourly basis. By putting together a variable list for the hourly report, once calculated, every desired value can be recorded.

DOE-2 is designed for the analysis of the relationship and interaction of the structures of, and systems in a building. A large number of options are related to space-conditioning and system design. The derivation of values and parameters for those keywords requires a lot of effort at the creation of the input file. Overall, DOE-2 is a very complex and non-hassle-free program. The user needs more experience to work successfully with DOE-2 than with CALPAS3. This is also reflected in the DOE-2 input file which can be found in Appendix B.

Without going into depth, two interesting approaches of DOE-2 will be mentioned next.

- By looking at the input file it can be seen that DOE-2 uses different coordinate systems to locate the structure elements. Those coordinate systems are hierarchically structured whereas the main system is the building coordinate system. In order to locate a space in the building, the origin of that space's coordinates will be expressed in building coordinates. The position of a space element like a bounding wall, however, is determined by space coordinates, and elements integrated in a wall, like a window, are located in wall coordinates. In this way the location of every element can be expressed arbitrarily in any coordinate system by matrix operations including translation and rotation. It should be mentioned that in the terminology of DOE-2, a space is described as a unique building block with a certain capacity and construction elements at its boundaries, like walls, floors, roofs, etc.

- Another interesting approach is the way in which DOE-2 treats transient heat conduction for the hour-by-hour heat transfer calculation. A thermal response factor method [6] is used to evaluate heat conduction through multi-layer walls and roofs at a selected time. This method utilizes the superposition principle so that the overall thermal response of the building structure is the sum of the responses caused by many temperature pulses during preceding significant times. The heat conduction equation is then solved by employing a matrix equation of Laplace transforms. Thus, by simulating the transient boundary temperatures by a train of pulse, and by summing up the heat flux caused by each pulse, the total heat flux at a given time can be derived.

3 Description of the DYMOLA Approach

Before getting started with the actual model description, some aspects should be discussed concerning the modelling methodology and its terminology. In order to understand the DYMOLA source code and the application of this tool, some remarks referring to the special properties of the software are also provided.

3.1 Modelling with Bond Graphs

The bond graph, introduced by Henry Paynter, 1961 [7], incorporates the ability to describe simultaneously the formal expression of a process and its topological structure. That makes bond graphs perfectly suitable for describing the transportation phenomena of energy and power. In this way the bond graph combines good modelling capability due to the visual structure with the description of the physical laws related to this process. Furthermore, bond graphs provide a common model for a wide range of systems ranging from electrical, mechanical, hydraulic, pneumatic and thermal systems to applications such as economics. [1, 8]

3.1.1 The Power Bond

Energy is a basic commodity in a system. It flows over the boundaries into or out of the system, is stored or dissipated inside, but can always be balanced over the entire system. Power is the rate of energy flow and can be expressed in two variables associated with the system, the cause and the effect. This is very convenient because power is not easily measured and engineers prefer to split power into particular components which can be measured easily and can be given physical interpretations.

The bond as a symbol is shaped like a harpoon that connects an across variable (cause), the 'effort' e, written on the side of the harpoon's hook, with a through variable (effect), called 'flow' f, indicated on the side away of the hook, as shown in Fig.(1).

The formal relation between effort and flow in a power bond indicates that the product of the variables results in the power through the bond at any time T.

$$P = ef \tag{1}$$

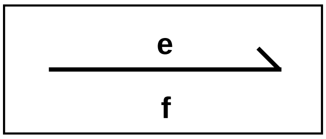


Figure 1: Bond graph

System	Effort e	Flow f
Electrical	Voltage u [V]	Current i [A]
Mechanical		
Translation	Force F [N]	Velocity v [$\mathrm{m\,sec^{-1}}$]
Rotation	Torque T [N m]	Angular velocity ω [$\mathrm{rad\,sec^{-1}}$]
Hydraulic	Pressure p [$\mathrm{N\,m^{-2}}$]	Volume flow q [$\mathrm{m^3\,sec^{-1}}$]
Chemical	Chemical potential μ [$\mathrm{J\,mole^{-1}}$]	Molar flow ν [$\mathrm{mole\,sec^{-1}}$]
Thermodynamical	Temperature T [K]	Entropy flow $\frac{dS}{dt}$ [$\mathrm{W\,K^{-1}}$]

Table 1: Effort and flow variables

Some common physical variables used to express effort and flow for different system application are shown in Table 1 [1].

3.1.2 Basic Features

Some basic features of bond graph applications will be shown below. In this way the construction of bond graph models will become more evident.

Junctions In order to represent topological structure as well as the energy conservation law, two kinds of junctions have been introduced by Paynter, a 0-junction and a 1-junction, as shown in Fig.(2).

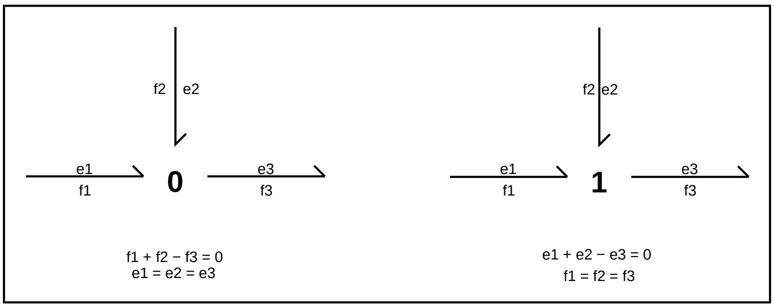


Figure 2: 0-junction and 1-junction of the bond graph terminology

At the 0-junction the flow adds up to zero while all efforts are equal, and at the 1-junction all effort variables add up to zero while all flows are equal.

1-Port Elements Bond graphs represent the energy transport into or out of a system, as well as the connection of elements inside the system. These elements will influence the desired response of the system and could be of dissipative or capacitive nature, like a resistor or a capacity in an electrical circuit, as well as a source, sink of effort or flow.

Typically, such elements are described by 1-ports because they interact with their environment at one single port, e.g., a capacity where we either insert or regain power. Some basic 1-port elements are given in Table 2 and Fig.(3) [9].

Element	Relation
Resistance	$e = R\,f$
Capacitance	$C\,\frac{d\,e}{dt} = f$
Inductance	$e = L\,\frac{d\,f}{dt}$
Effort source	$e = E(t)$, f arbitrary
Flow source	$f = F(t)$, e arbitrary

Table 2: Basic 1-port elements

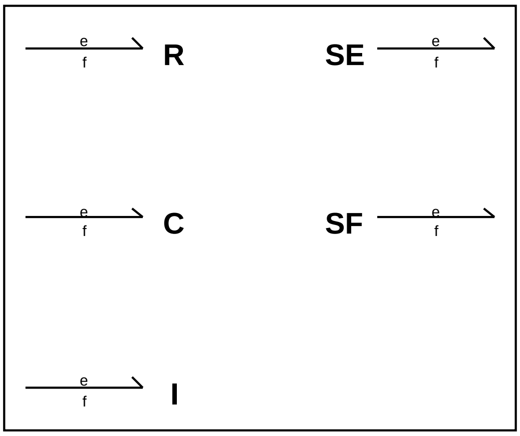


Figure 3: Single port elements

Multi Port Elements In order to represent the conversion of energy from one system to another, a 2-port element has been introduced with power interactions on both ports. The transfer of power from one form of energy to another is certainly an advantage of the bond graph approach because, as seen before, each system provides its set of effort and flow variables.

Two transition elements have been introduced, both *ideal* in the sense of conservation of energy. Obviously the power entering one side of the ideal (lossless) element must be the same as the power leaving the system. The two basic 2-port elements and their relations are shown in Table 3 and Fig.(4) [9].

In a way, even the previously mentioned junctions are multi-port elements, typically at least 3-ports. They also fulfill the energy conservation law, but do not provide energy conversion.

Element	Relation
Transformer	$e_1 = m\, e_2,\; m\, f_1 = f_2$
Gyrator	$e_1 = r\, f_2,\; r\, f_1 = e_2$

Table 3: Basic 2-port elements

Figure 4: 2-port elements

3.1.3 Bond Graph Causality

Another advantage of bond graph modelling is the easy detection of structural singularities and algebraic loops. Both occurrences are connected to the number of variables and number of non-trivial equations provided to determine those variables. In the case of an algebraic loop, there are not enough equations to evaluate all the variables. A structural singularity occurs, if there is more then one equations which could evaluate the same variable. This is not desired either.

Because no evaluation takes place at the bond graph, the two variables, effort and flow, connected to the bond, must be determined on either end of the bond. By introducing a short stroke perpendicular to the bond, which is placed on one of its two ends, we decide at which side the two variables are determined. The convention says that the equation involved in the side with the stroke provides the evaluation of the flow variable, while the other side does the same with the effort variable. Therefore, we have a nice way to prove whether we provide the right equations to evaluate both variables of the bond. If we now obey the same rules referred to in the construction of bond graphs, we can easily detect singularities and algebraic loops as well. Fig.(5) shows some of those rules.

At 0-junctions, where flows add up to zero, only one flow variable can be determined. At the same junction the effort needs to be computed away from the junction but only at one bond, because efforts of all bonds attached to the same 0-junction have the same values. At the 1-junction we find the same, vice versa. Only one effort variable can be determined at the junction, while the flow is computed at one bond away from the junction. If we attach a source to a bond, the stroke will indicate whether flow or effort is provided.

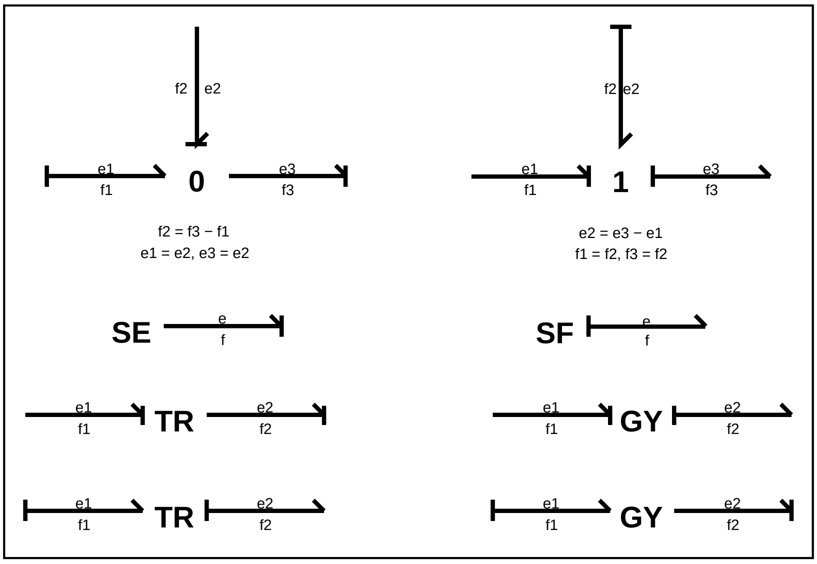


Figure 5: Rules for the bond graph modelling concerning singularities and algebraic loops

3.2 DYMOLA, the Tool

After having discussed the construction of bond graph models we need to describe the tool which is used to code the bond graphs and run the simulation. DYMOLA (DYnamic MOdeling LAnguage), a software invented by Hilding Elmquist (1979) in his PhD. Dissertation [10], serves as a front end to several simulation languages. It works like a preprocessor for the simulation languages DESIRE (Direct Executing Simulation In REal time)[11] and ACSL (Advanced Continuous Simulation Language) [12]. For the present approach the recently developed ACSL interface has been used to compile DYMOLA source code into ACSL because it is supposed to be the better simulation engine [1]. ACSL produces an executable file, which can be used to obtain the results under the control of a program called CTRL-C [13]. This software is very capable for matrix calculations and has very useful graphic routines which were used to produce graphic interpretations of the results.

DYMOLA is currently available in a version enhanced by Qingsu Wang (1989) [14] for the PC and on VAX/VMS machines coded in PASCAL.

The following DYMOLA description will focus on the main features of this modelling language, advantages and unsolved problems as well as the application in bond graphs. Since much documentation [1, 10, 14] about DYMOLA already exist, some of the explanations are closely related to them.

3.2.1 Special Features of DYMOLA

DYMOLA, which is a modelling language not a simulation language, is very suitable for bond graph applications and provides a code that looks readable and beautifully structured. Some of the advantages of the DYMOLA approach are certainly to assign *submodels*, *cuts*, *paths*, and easily attach elements to each other by using the *connect* statement. Contrary to most of the Continuous System Simulation Languages (CSSL), DYMOLA can sort the model equations and assigns causalities to these equations. For example, consider the equation of Ohm's law

$$u = R\,i$$

DYMOLA will also provide the solution for the current when needed.

$$i = \frac{u}{R}$$

Some Properties of DYMOLA The following list describes DYMOLA's type of currently provided constants and variables, expressions and equation handling.

1. Variables and Constants

    a) DYMOLA variables belong to either the type *terminal* or *local.* They are of type terminal if they are supposed to be connected to something outside the model. They are of type local if they are available only to the defining model. [1]

    b) Terminals can be either inputs or outputs. What they are often depends on the environment to which they are connected. However, the user can specify what he or she wants them to be by explicitly declaring them as *input* or *output* rather then as terminal. [1]

    c) Constant variables are defined either as *parameter* or as *constant.* If they need to be reassigned later in the simulation then parameter will be used. For this type a default value can be declared in case the parameter will not be assigned from outside the model. The value of a constant-type, however, will never change.

2. Derivatives and Initial Conditions

   a) In DYMOLA, first, second or higher order derivatives are expressed either by using the notation *der(.)*, *der2(.)* or using the prime *(′)*, *(′′)*. It is also legal to place derivatives anywhere in the equation, i.e., on the left and on the right side of the equal sign.

   b) In case initial conditions differing from zero need to be assigned, they have to be set from outside the model.

3. Solving Equations in DYMOLA

   a) Since in DYMOLA equations are expressed using the syntax $expression = expression$, and the equation are sorted for the appropriate variable, it is acceptable to obtain an equation of form $der(A) = B + C * A$.

   b) During the model expansion all terms will be canceled out if they are multiplied by a zero parameter. This has the disadvantage that parameters initially set to zero are eliminated even though they were intended to be changed later during the simulation. The advantage, however, is that due to this elimination an entire class of structural singularities can be avoided.

The Hierarchical Structure, Submodel Submodels are used to structure models hierarchically throughout the CSSL-languages. This is demonstrated in Fig.(6).

The disadvantage, however, in this description is the replication which occurs if two of the subsystems happen to be identical. In DYMOLA the user can avoid this by declaring a *model type*. Inside of the model one will call for such a model type using the expression *submodel*.

Let us assume the models 1 and α and the models 2 and β are the same, namely model type `a` and model type `b`. The model type `A` for model `I` and `II` will then become:

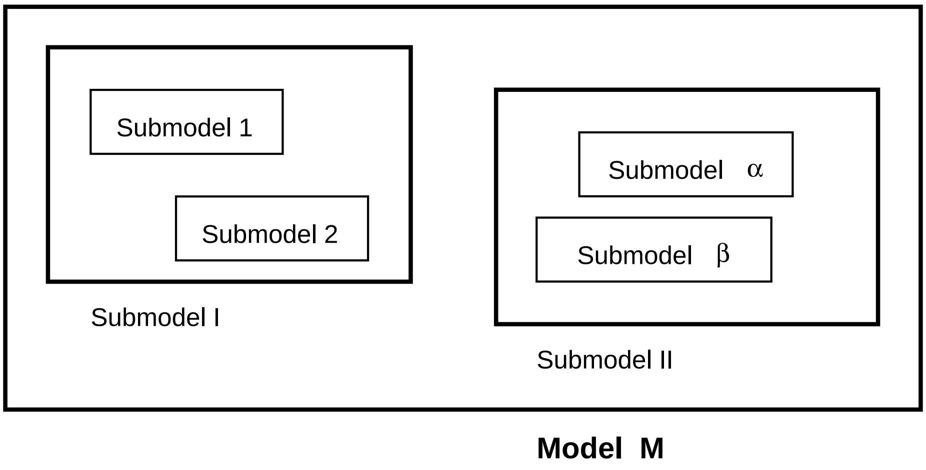


Figure 6: DYMOLA submodel configuration

```
 model type A
  ...
  submodel (a)  name (parameter_list) (ic initial_list)
  submodel (b)  name (parameter_list) (ic initial_list)
  ...
end
```

and the final expression for our model `M`

```
 model M
  ...
  submodel (A)  name (parameter_list) (ic initial_list), ->
                name (parameter_list) (ic initial_list)
  ...
end
```

The syntax of the submodel statement will be described using the last example:

- In parenthesis, the name of the model type `A` is given; note that DYMOLA is case sensitive.
- Then, arbitrarily many models of this model type can be assigned by providing a *name* and their specifications. The list will be separated by a comma ",".

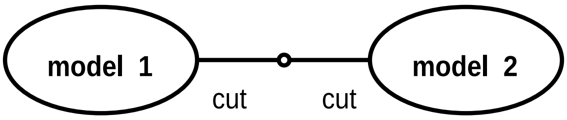


Figure 7: DYMOLA cut concept

- The optional *parameter_list* may change default values of the subsystem by assigning new ones.
- The *initial_list*, similar to the parameter_list, assigns new non zero initial conditions.

The Cut and Path Concept The connection of a submodel with its environment is determined by the variables and parameters which are exchanged over the model boundary. The mechanism in DYMOLA providing the grouping of variables is called a *cut*, as shown in Fig.(7).

A cut works like an electrical plug or socket and defines an interface to the outside world. The syntax of the cut statement is given below.

`cut` *cut_name* `(`*variable_list*`)`

A more useful syntax for the bond graph approach provides across and through variables to the cut, and appears below.

`cut` *cut_name* `(`*across_variable* `/` *through_variable*`)`

Further, cuts can be grouped hierarchically together

```
cut A(x1,y1,z1), B(x2,y2,z2)
cut C[A B]
```

and one cut can be declared as the *main cut*, i.e., it will be the default cut when calling for this model.

```
main cut C[A B]
```

In order to express inherent connections from a designated source to a destination, a directed *path* can be declared. Paths determine the connection inside the model between different cuts, as shown in Fig.(8). The syntax of this statement is given below.

`path` *path_name* `<` *cut_name* – *cut_name* `>`

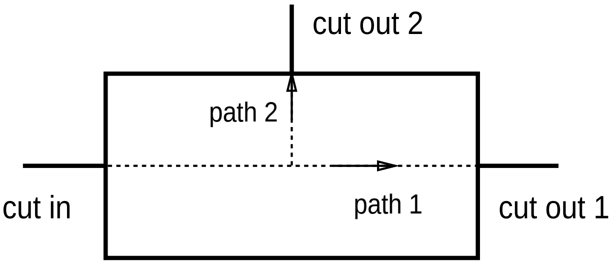


Figure 8: DYMOLA path concept

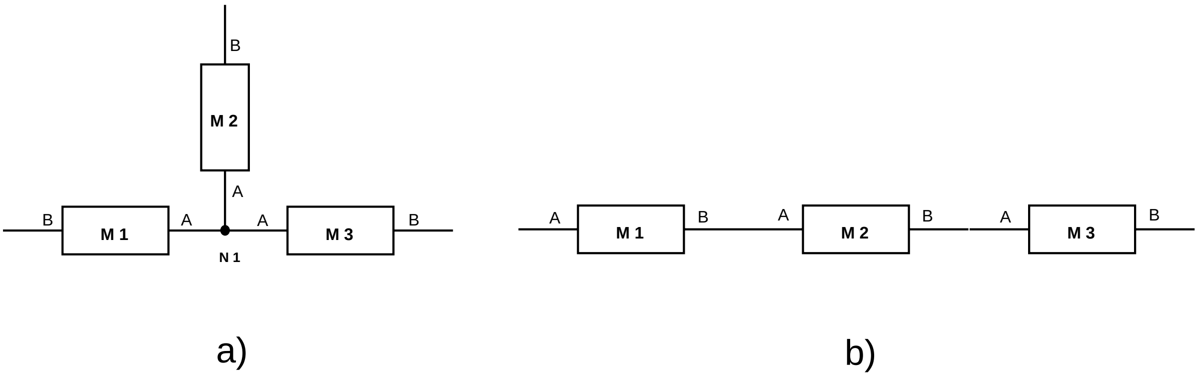


Figure 9: Examples for cut connections

The Connect Statement The *connect* statement in DYMOLA provides for the interaction of elements inside a model with each other. The entire set of variables combined in a cut will be attached to another cut or a *node* using connect:

`connect` *model_name:cut_name* `at` *model_name:cut_name*

During the expansion of the model, DYMOLA checks that the corresponding cut_variables are compatible with each other.

Let us assume three models `M1, M2, M3` exist with the same declared cuts `A` and `B`. To connect all three cuts `A` together (Fig.(9 a)) we can use a `Node N1` and write

```
connect M1:A at N1, M2:A at N1, M2:A at N1
```

or without the node

```
connect M1:A at M2:A at M3:A
```

or abbreviated

```
connect M1:A = M2:A = M3:A
```

By providing a `path P < A - B >` for every model we can express Fig.(9 b) in the following way

```
connect (P) M1 to M2 to M3            or
connect (P) M1 - M2 - M3
```

Besides `at` and `to` DYMOLA offers some additional options to connect a path in reverse, or two paths in parallel or in a loop, using `reverse` (\), `par` (//) and `loop`, respectively.

3.2.2 Modelling with Bond Graphs in DYMOLA

The strength of DYMOLA is certainly its capability to handle hierarchically structured models like bond graph models. Some details have to be mentioned, however, even though the approach is fairly straightforward.

1. The cut concept of DYMOLA allows us to declare across and through variables to the cut. These variables are dedicated to act as effort and flow in our bond graph implementation. However, we have to notice that in the declaration of cuts expressed in effort and flow, the flow variable will always be assigned in to the model. For the model `M1` in Fig.(9) this means that `A` leads in to and `B` out of the model, which can be expressed by

   ```
   cut A(e / f), B(e / -f)
   ```

2. DYMOLA provides nodes which are equivalent to the 0-junction in the bond graph terminology. This means that elements attached to 0-junctions can be easily connected to a node.

3. However, since in bond graph models the junction-types toggle between 0- and 1-junctions, such a 1-junction must be designed as well. In order to express a 1-junction, flow and effort variables must be interchanged. This can be obtained by introducing a model type bond.

```
{ bond graph bond }

model type bond

  cut A (x / y), B (y / -x)
  main cut C [A B]
  main path P <A - B>

end
```

That means, all elements ported at a 1-junction must be connected with a model type bond in between.

The DYMOLA code of some basic elements is described below.

```
model type SE

 main cut A(e/.)
 terminal E0

E0 = e

end
```

```
model type SF

 main cut A(./-f)
 terminal F0

F0 = f

end
```

```
model type R

 main cut A(e/f)
 parameter R = 1.0

R * f = e

end
```

```
model type G

 main cut A(e/f)
 parameter G = 1.0

G * e = f

end
```

```
model type C

 main cut A(e/f)
 parameter C = 1.0

C * der(e) = f

end
```

3.2.3 Generating the Target Code

As mentioned previously, DYMOLA works as a front-end for several simulation languages. The DYMOLA compiler produces a program in the desired target language. For this approach ACSL has been chosen as the adequate simulation engine.

The procedure of obtaining the ACSL program will be described step-by-step. The following command sequence will invoke the DYMOLA preprocessor and read the model definition:

```
$ dymola2
> enter model
– @model_name.dym
```

In order to handle a higher number of variables, we had to use the extended version of DYMOLA by invoking the preprocessor after the system prompt ”$” with:

```
$ largedymola2
```

After the DYMOLA prompt ”>” which indicates the interactive mode, it will read the model description by calling for all model types included using the @ operator. Note that submodels need to be included in the order of increasing hierarchy.

Having read the entire model, DYMOLA starts the first step of compilation by referencing the submodel and providing the coupling equations corresponding to the connect statements. The resulting equations can be directed to a file by using the command sequence ”> `outfile` *file_name*” and ”> `output equations`” or viewed on-screen if the first part is omitted.

The next step towards the ACSL program is

```
> partition eliminate
```

This command manipulates all equations by doing the following:

- The `partition` command assigns causalities to each equations, i.e., it determines which equation needs to be solved for what variable. Further, it sorts the equations into an executable order.

- The `eliminate` command gets rid of redundant equations. Expressions of the type $a = b$ will be eliminated and occurrences of a replaced by b. Parameter assigned with the value 0.0 or 1.0 will be reduced to the numerical value in the equation.

Multiplications by 1.0 are reduced, while expressions multiplied by 0.0 are replaced as a whole by 0.0. Further, all equations are solved for causal variables in order to eliminate algebraic loops.

The result of this manipulation can again be viewed or outfiled using the command "`> output solved equations`".

Before DYMOLA produces the program in the target simulation language, we need to add the experiment description. This *control file* basically contains information about the experiment environment.

A basic control file for ACSL will typically be filed under the name of the DYMOLA source code with the extension "`.act`" and consists of following syntax.

```
cmodel
maxtime   variable_name = value
cinterval variable_name = value
TERMT   logical expression

input number of inputs, variable_name (status, assigned variable)

end
```

The term `maxtime` determines the period of simulation time and `cinterval` the *communication interval*, i.e, the interval where output variables have their values reported [12]. These values can be changed during consecutive simulation runs by assigning new values to the adequate variables. `TERMT` specifies the terminate condition which typically is a expression like $T.GE.TMAX$ if T happens to be the simulation time and $TMAX$ the variable name of `maxtime`. The assigned input variables can either have the status `depend` or `independ` depending on their time dependence.

In addition to this, different optional statements can be placed in the control file concerning integration algorithm, step size, schedules, tables, etc. The experiment file for the house description is given in Appendix A. Further, the Thesis of Sunil Charan Idnani about the ACSL interface for DYMOLA [15] and the ACSL user guide and reference manual [12] might be consulted.

3.2.4 Some Improvements for the Future

Although DYMOLA is supposed to be a very handsome tool and powerful for bond graph applications, some improvement are needed to make DYMOLA and the ACSL interface more user-friendly and capable for broader use. Some of the enhancements have already been derived but not yet implemented. Others are suggestions extracted from Dr. Cellier's book and Qingsu Wang's Thesis.

1. A powerful type of global variable should be implemented like the derived approach of *external* and *internal* variables. Externals are similar to parameters, but they provide for an implicit rather than explicit data exchange mechanism. Externals are used to simplify the utilization of global constants or global parameters. For security reasons, the calling model must acknowledge its awareness of the existence of these globals, by specifying them as internals. [1]

2. A *default* statement should exist which would moderate the constraint of assigning terminal variables.

3. An improvement of the algorithm used in the ACSL interface to determine ACSL variables would ease the debugging of the produced ACSL program. Due to the mutilation of DYMOLA variables during the creation of the corresponding ACSL variables, the same assignments can no longer be recognized.

4. DYMOLA is currently able to eliminate variables in equations of the type $a = b$, but should also be able to handle equation of type $a + b = 0.0$.

5. DYMOLA should be able to recognize equations that have been specified twice, and eliminate the duplication automatically to avoid redundant equations. [1]

6. DYMOLA should provide a powerful user interface like a graphic preprocessor, that could support DYMOLA's modelling methodology.

3.3 The DYMOLA Approach

The main *model HOUSE* consists of a variety of smaller models which are related to the constructive elements of a residential building. Thus, the next smaller segments are obviously the rooms which attached to each other will determine the envelope of the structure, called *model type SPACE*. Those compartments, however, contain model types representing construction elements like walls, roof, slab, windows, doors, etc, which can be connected to the space model according to their use and position. Walls with direct solar irradiation are called *model type EXWALL*, which are external walls as opposed to internal walls, with *model type INTWALL*. Once again, we need to decompose these elements following the basic physical laws to describe the phenomena occurring inside. That yields in a couple of submodels which are capable of expressing the mechanisms of heat transfer considering *conduction*, *convection* and *radiation*. The implementation of those physical effects has been derived as follows.

3.3.1 Conduction

The thermal process of conduction is also called heat transfer by diffusion. It refers to the transport of energy in a medium due to a temperature gradient, and the physical mechanism of random atomic and molecular activity. This phenomenon can be expressed in the so called *heat diffusion equation*.

$$\frac{\partial T}{\partial t} = \sigma \Delta^2 T + \dot{Q} \tag{2}$$

For our purpose, i.e. looking at one-dimensional heat transfer through a medium without energy generation $\dot{Q}$, we come up with a partial differential equation (PDE) dependent on the time t and one dimension x.

$$\frac{\partial T}{\partial t} = \sigma \frac{\partial^2 T}{\partial x^2} \tag{3}$$

This enables us to determine the temperature distribution as a function of time and the corresponding heat flux as well. The solution of a PDE like this results from discretizing one axis. In this way we will map the PDE into a set of ordinary differential equations (ODE) as described in detail in Dr. Cellier's book [1]. In this case we used a centered

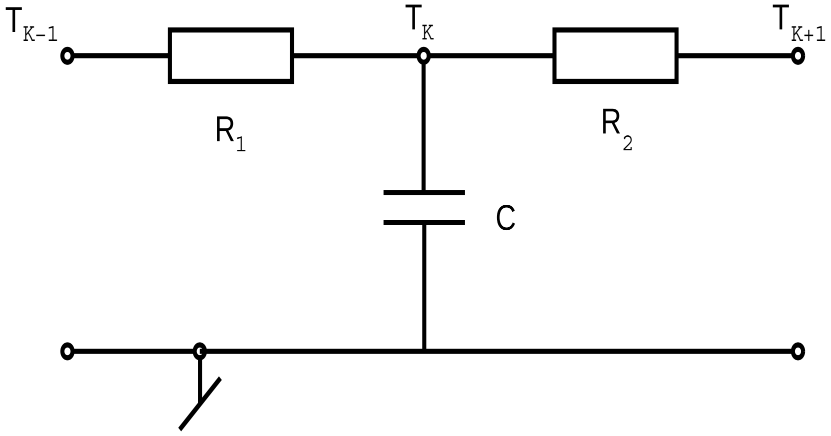


Figure 10: Electrical analogy

difference formula to discretize along the x-axis up to n-times.

$$\frac{dT_k(t)}{dt} = \frac{\sigma}{\Delta x^2}[T_{k+1}(t) - 2T_k(t) + T_{k-1}(t)],\ k \in (1...n) \tag{4}$$

Now, we can use the electrical analogy, which has been proposed 1944 by Gabriel Kron as a convenient way to provide a numerical solution of partial differential equations (PDE's) [16]. In fact, we will express our conductive increment as a chain of resistive and capacitive elements.

How do we derive the adequate Bond Graph Model of this ?

In order to transfer the electrical analogy of a thermal process into a physical model we have to solve some problems first. One of them deals with the fact that a Power Bond is the embodiment of a power flow. We on the other hand would consider the heat flux as the adequate flow variable for our bond graph, but by multiplying with the designated effort variable T we find $Q * T$ does not result in power. That means that our flow variable has to be the *entropy flow* $\dot{S}$ which makes perfect sense because it allows us to follow the energy conservation law as well as the entropy balance.

$$P = \dot{S}T \ with \ \dot{S} = \frac{\dot{Q}}{T} \tag{5}$$

Considering those two basic laws we detect a second flaw. Namely by using the well known

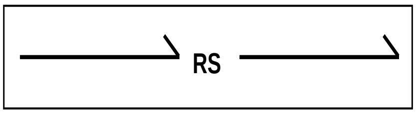


Figure 11: Bond graph for a resistive source

resistor model of the electric circuit for our purpose we violate the energy conservation law.

What happens to the power we apply to a resistor ?

We basically dissipate it and heat up the resistor, but in our thermal resistive element we prefer to describe the heat transfer. That means we have to return this power into our system. In order to accomplish this aim, we introduce a so called *resistive source* [1], as shown in Fig.(11).

Before we start to build a one-dimensional conduction cell we should look at the parameter we will use. For heat conduction the rate equation for a one dimensional plane wall is known as *Fourier's law* [17].

$$\dot{Q}_x = -kA\frac{dT}{dx} = \frac{kA}{l}\Delta T \tag{6}$$

with temperature gradient $\frac{dT}{dx}$, thermal conductivity k $[\frac{W}{m\,K}]$, cross sectional area A $[m^2]$ and segment length l $[m]$. Or written for our purpose:

$$\dot{S}_x = \frac{1}{T}\left(\frac{kA}{l}\right)\Delta T \tag{7}$$

Therefore, we can express our thermal resistor as a modulated resistor (mR) or better a *modulated resistive source* (mRS) with $R = \frac{lT}{kA}$ or what can be more convenient to use a *modulated conductive source* (mGS) with $G = \frac{1}{R} = \frac{kA}{lT}$. The DYMOLA code of the adequate mGS-model is given in Fig.(12).

The heat storage happens to be

$$\Delta\dot{Q} = \rho V\gamma\frac{dT}{dt} \tag{8}$$

with the density ρ $[\frac{Kg}{m^3}]$, the Volume V $[m^3]$ and the specific heat γ $[\frac{J}{Kg\,K}]$. Or again written as:

$$\Delta\dot{S} = \frac{\rho V\gamma}{T}\frac{dT}{dt} \tag{9}$$

```
{ bond graph modulated conductive source }

model type mGS

  cut A (e1 / f1), B (e2 / -f2)
  main cut C[A B]
  main path P<A - B>
  parameter k=1.0, l=1.0
  local G, G1
  external area

  G1 = (k/l)*area
  G = G1/e2
  G*e1 = f1
  f1*e1 = f2*e2

end
```

Figure 12: DYMOLA code for mGS-model

```
{ bond graph modulated capacity }

model type mC

  main cut A (e / f)
  parameter gamma=1.0, m=1.0
  local C

  C = (gamma*m)/(3600.*e)
  C*der(e) = f

end
```

Figure 13: DYMOLA code for the mC-model

By using the DYMOLA model for capacitive elements C again needs to be modulated, so we come up with a *modulated capacity* (mC), comprising $C = \frac{\rho A l \gamma}{T}$ expressed for a segment with cross area A and length l. The DYMOLA code for the mC-model is shown in Fig.(13). It should be mentioned that the time unit for the simulation was selected as $1hour$, $[h]$. Therefore, all following formulas were derived on an hour basis.

That leads us to the bond graph of a one-dimensional cell, Fig.(14). By attaching those conduction cells to each other we can represent the entire diffusion chain.

Here it might be interesting to look closer at the equations. At the 1-junction we will determine the temperature difference $\Delta T_{01} = T_0 - T_1$ between entering and exiting

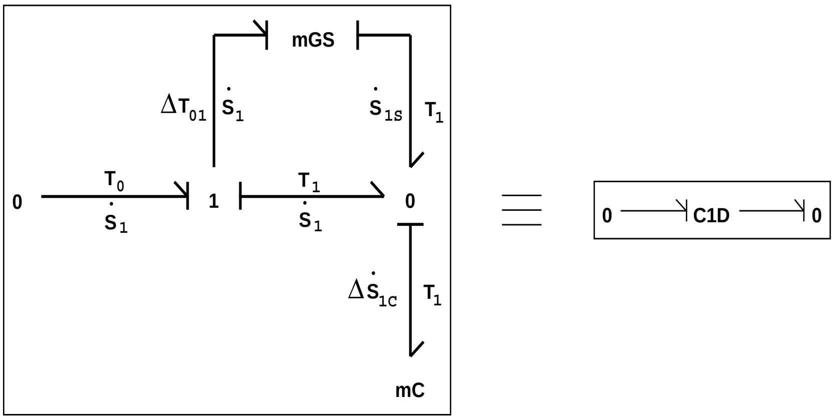


Figure 14: Bond graph for a one-dimensional conductive cell

temperature. This is the driving force of our process; with a $\Delta T_{01} = 0$ nothing would happen. In the mGS model we obtain $\dot{S}_1 = G\Delta T_{01}$ with the modulated conductance G. Then, we return the dissipated power back into the system by applying the first law of thermodynamics at the resistor $\Delta T_{01}\dot{S}_1 = T_1\dot{S}_{1s}$. DYMOLA will solve this equation for $\dot{S}_{1s}$ to determine the additional entropy flow. And now we also realize that we fulfill the second law of thermodynamics, i.e. the entropy balance of a system. At the 0-junction we find not only the entropy flow due to temperature difference, $\dot{S}_1$, but also the entropy flow due to the temperature change from T_0 to T_1, namely $\dot{S}_{1s}$. Last but not least, at the 0-junction we determine the entropy flow of the energy that gets stored in the thermal capacity and calculate the temperature T_1 at this node $\Delta\dot{S}_{1c} = C\frac{dT}{dt}$ according to the given initial condition T_{ic}.

The code of the DYMOLA model type *C1D* can be found in Fig.(15). This model will later be used whenever we have to set up a conduction chain.

3.3.2 Convection

Convection heat transfer is determined by two mechanisms: the energy transfer due to random molecular motion (diffusion) and bulk motion of fluid. The result is the development of a region in the fluid through which the velocity varies from zero at the surface to v_∞ associated with the flow, the velocity boundary layer. This yields in a thermal boundary layer where the temperature varies from surface temperature T_s to T_∞ of the

```
{ bond graph for one dimensional conduction cell }

model type C1D

  submodel (mGS) Gcell (k=k, l=l)
  submodel (mC) Ccell (gamma=gamma, m=m) (ic e=288.0)
  submodel (bond) B1, B2, B3
  node n1, n2

  cut Cx(ex/fx), Ci(ei/ -fi)
  main path P<Cx - Ci>
  parameter k=1.0, l=1.0, gamma=1.0, m=1.0
  internal area
  external area

  connect B1 from Cx to n1
  connect B2 from n1 to n2
  connect Gcell from n2 to Ci
  connect B3 from n1 to Ci
  connect Ccell at Ci

end
```

Figure 15: DYMOLA code for the C1D-model

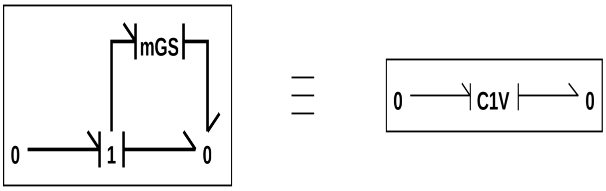


Figure 16: Bond graph for a one-dimensional convective cell

outer flow. In order to model the thermal process of convection accurately we need to set up a model of our fluid or gas to derive the velocity profile through our boundary layer. Considering that due to friction in the fluid/gas at the vicinity of our boundary a heat transforming process takes place, a power exchange between thermal and hydraulic or pneumatic systems follows. However, the hydraulic or pneumatic model provides us with the fluid/gas velocity which can be used to modulate a thermal resistive element in order to combine the diffusion part and the fluid motion part of convection. Then again, we are able to express convection as a modulated resistive source model with $R = av_{fluid} + b$. Thus, we get following bond graph model for a one-dimensional convective cell, Fig.(16).

By looking at *Newton's law of cooling* proposed in most of the heat transfer literature

```
{ bond graph for one dimensional convection cell }

model type C1V

  submodel (mGS) Gcell (k=h, l=1.0)
  submodel (bond) B1, B2, B3
  node n1, n2

  cut Cx(ex/fx), Ci(ei/ -fi)
  main path P<Cx - Ci>
  parameter h=1.0
  internal area
  external area

  connect B1 from Cx to n1
  connect B2 from n1 to n2
  connect Gcell from n2 to Ci
  connect B3 from n1 to Ci

end
```

Figure 17: DYMOLA code for the C1V-model

[17] we also will find the analogy to conduction.

$$\dot{Q} = Ah\Delta T \tag{10}$$

where A is the cross area of our segment and h is the local convection heat transfer coefficient. And we have to consider that h depends on a plurality of parameters $h = f\left(k, c, \mu, \rho, v_{infty}, l, surface\,geometry\right)$ like the fluid properties thermal conductivity k, specific heat c, dynamic viscosity ν, mass density ρ as well as fluid velocity v_∞ and length scale in the fluid direction l. Therefore, we again can write Equ.(10) for our purpose

$$\dot{S} = \frac{Ah}{T}\Delta T \tag{11}$$

and end up with a solution for $R = \frac{T}{Ah}$ or $G = \frac{1}{R} = \frac{Ah}{T}$.

The DYMOLA code for such a convective element is given in Fig.(17). In order to use the same mGS-model as in the conduction cell the conductance G was changed. Rather than having $G = \frac{kA}{lT}$ from Equ.(7) we need to set $G = \frac{Ah}{T}$ following Equ.(11). This was accomplished by setting $k = h$ and $l = 1.0$.

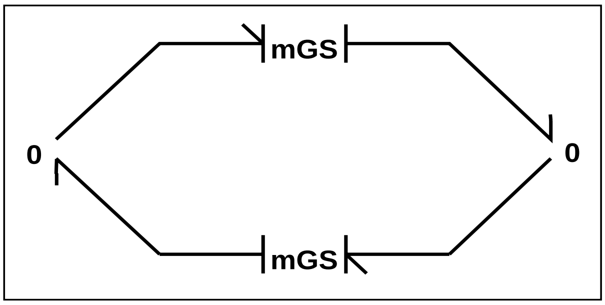


Figure 18: Bond graph for a radiation exchange between two surfaces

3.3.3 Radiation

Thermal radiation is energy emitted by matter that is at a finite temperature. While the transfer of energy by conduction and convection requires the presence of a material medium, radiation does not. In fact, radiation occurs most efficiently in a vacuum. The maximum heat flow at which radiation might be emitted from a surface is given by the Stefan-Boltzmann law

$$\dot{Q} = \sigma A T_s^4 \tag{12}$$

where the Stefan-Boltzmann constant is $\sigma = 5.6710^{-8} \frac{W}{m^2 K^4}$, T_s is the absolute surface temperature and A the surface area. By writing it in terms of the emitted entropy $\dot{S} = \sigma A T^3$ we can use again an mRS or mGS model, respectively, with $G = \sigma A T^2$ to describe the radiation exchange.

At the simulation of our test structure the radiation between surfaces has been omitted because of the considerable low temperatures in a living space. Nevertheless, the radiation exchange might be reconsidered concerning nocturnal radiation of the flat roof surface in order to cool the inside air during the night. Especially high mass buildings tend to store heat during the day and release it at night which is beneficial in winter but not in summer if occupants seek relief from the heat of the day. The approach of radiating to the night sky might support cooling down the building due to very low sky temperatures during the night. The adequate bond graph model will be shown in Fig.(18).

The solar power applied by irradiation was implemented as a flow source in a passive solar system. Inspecting the envelope of the building we will find two basic phenomena

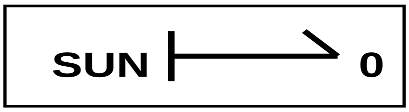


Figure 19: Bond graph for the SUN-model

regarding solar irradiation. One is the absorbed energy of all exposed surfaces and the other is the transmitted energy through a glazing.

3.3.4 Flow Source due to solar exposure – model type SUN

The irradiation on an exposed surface is determined overall by the extraterrestrial radiation, the earth atmosphere, and the effects of orienting a receiving surface.

Extraterrestrial Radiation The sun as a sphere of intense hot gaseous matter has an effective blackbody temperature of about 5762° K and supplies the energy radiated into space by several fusion reactions. The energy emitted by the sun and the spatial relationship to the earth results in a nearly fixed intensity of solar radiation outside of the earth's atmosphere. The solar constant, $I_{SC} = 1353\frac{W}{m^2}$, is the energy from the sun, per unit time, received on an unit area of surface *perpendicular* to the direction of propagation of the radiation, at the earth's mean distance from the sun, outside of the atmosphere.

Because earth's orbit is slightly elliptical, the extraterrestrial radiation intensity ranges from a maximum of $1398\frac{W}{m^2}$ on Jan. 3, when the earth is closest to the sun, to a minimum of $1311\frac{W}{m^2}$ on July 6, when the earth-sun distance reaches its maximum. Duffie & Beckman [18] suggest the formula $I_{0N} = I_{SC}\left(1 + 0.033\cos\frac{360n}{365}\right)$ to determine the extraterrestrial radiation on a plane normal surface on the n^{th} day of the year. ASHRAE [5] prefers tabulated values like the values for A of Table 4. It also should be mentioned that the energy emitted by the sun varies in magnitude ($\approx \pm 1.5\%$) and spectral distribution.

In order to determine the available solar radiation on the earth's surface, we need to consider the effects of the atmosphere in attenuating the sun's radiation. In passing through the earth's surface, the sun beam is reflected, scattered, and absorbed by dust, gas molecules, ozone, and water vapor. The extent of this depletion depends on the atmospheric composition and the length of the traverse of the sun rays. At this point two different approaches have been suggested. Duffie & Beckman predict the solar performance by using past measurements of solar radiation to calculate a clearness index $K_T = \frac{measured\,total\,radiation}{extraterrestrial\,radiation} = \frac{I_N}{I_{0N}}$, where N stands for normal to the beam. ASHRAE provides us with a formula to determine the direct normal solar intensity at the earth's surface on a clear day as

$$I_{DN} = \frac{A}{exp\left(\frac{B}{\sin\beta}\right)} \tag{13}$$

where $A :=$ apparent solar irradiation at sea and $B :=$ atmospheric extinction coefficient are tabulated monthly mean values shown in Table 4. β is the solar altitude.

	A	B	C	Equation of Time, ET
	$[\frac{W}{m^2}]$	(Dimensionless Ratios)		min.
Jan	1230	0.142	0.058	-11.2
Feb	1214	0.144	0.060	-13.9
Mar	1185	0.156	0.071	- 7.5
Apr	1135	0.180	0.097	1.1
May	1103	0.196	0.121	3.3
June	1088	0.205	0.134	- 1.4
July	1085	0.207	0.136	- 6.2
Aug	1107	0.201	0.122	- 2.4
Sep	1151	0.177	0.092	7.5
Oct	1192	0.160	0.073	15.4
Nov	1220	0.149	0.063	13.8
Dec	1233	0.142	0.057	1.6

Table 4: Values for A, B, C and Equation of Time

We will make a simplification by assuming always fine weather conditions, and this will avoid the otherwise necessary crunching a dire amount of weather data, so that the ASHRAE-Method can be chosen. In addition to that, we will find that values for A

and B are lower than the Tucson average because those numbers are a result of research at the University of Minnesota and reflect a mean value for the whole U.S. We also have to consider that by traversing through the atmosphere the spectral distribution changes. Thus, most of the ultraviolet solar radiation gets absorbed by the ozone in upper atmosphere, while parts of the radiation in the shortwave range are scattered by molecules and dust and reach the earth in the form of diffuse radiation, I_d. Obviously the diffuse radiation varies widely with the moisture and dust content of the atmosphere. Some of the energy absorbed by carbondioxide and water vapor in the sky reaches the earth in the form of long-wave atmospheric radiation. Apparently, there can be three contributors influencing the available solar radiation on a terrestrial surface.

$$I = I_D + I_d + I_r \tag{14}$$

where $I_D :=$ direct solar radiation, $I_d :=$ diffuse sky radiation, and $I_r :=$ radiation reflected from surrounding surfaces. The direct solar radiation I_D can be determined as

$$I_D = I_{DN} \cos\theta \tag{15}$$

where θ is the angle of incidence between the incoming solar rays and a line normal to the surface.

The orientation of the surface The geometric relationship between a plane of any arbitrary orientation relative to the earth at any time and the incoming solar beam radiation can be described in terms of several angles:

The declination δ can be found for the n^{th} day of the year as

$$\delta = 23.45 \sin\left(360 \frac{284 + n}{365}\right) \tag{16}$$

The Apparent Solar Time (AST) to Local Standard Time (LST) can be determined as

$$AST = LST + ET + 4(LSM - LON) \tag{17}$$

where $ET :=$ Equation of Time, in min., $LSM :=$ Local Standard time Meridian, $LON :=$ Local Longitude, with 4 minutes of time required for 1° rotation of earth.

Angle	Description	DYMOLA-variable
L	Local latitude, i.e., angular location north or south of the equator, north positive $-90° \leq L \leq 90°$	lat
δ	Declination, i.e., angular position of the sun at solar noon north or south of the equator, north positive $-23.45° \leq \delta \leq +23.45°$	dec
Σ	Tilt angle for the surface related to the horizontal with $\Sigma = 90°$ for a vertical surface and $\Sigma = 0°$ for a horizontal surface, $0° \leq \Sigma \leq 180°$	tilt
γ	Surface azimuth angle, i.e., the deviation of the projection on a horizontal plane of the normal to the surface from the local meridian, with zero due south, east negative, west positive, $-180° \leq \gamma \leq +180°$	sfa
H	Hour angle, i.e., the angular displacement of the sun east or west of the local meridian due to rotation of the earth on its axis at 15° per hour, morning negative, afternoon positive	h
θ	Angle of incidence, i.e., the angle between the beam radiation on a surface and the normal to the surface	inc

Table 5: Solar angle determination

Therefore, on an hourly basis we get the hour angle

$$H = 15^\circ \left(\frac{AST}{60} - 12 \right) \tag{18}$$

and the angle of incidence θ

$$\begin{aligned} \cos\theta &= +\sin\delta \sin L \cos\Sigma \\ &\quad - \sin\delta \cos L \sin\Sigma \cos\gamma \\ &\quad + \cos\delta \cos L \cos\Sigma \cos H \\ &\quad + \cos\delta \sin L \sin\Sigma \cos\gamma \cos H \\ &\quad + \cos\delta \sin\Sigma \sin\gamma \sin H \end{aligned} \tag{19}$$

In order to determine sunrise and sunset we also need to know the solar altitude β, shown as

$$\sin\beta = \cos L \cos\delta \cos H + \sin L \sin\delta \tag{20}$$

Now, after we have determined the direct gain the next step is to calculate the diffuse portion of the available solar radiation.

$$I_d = I_{ds} + I_{dg} \tag{21}$$

where I_{ds}, the diffuse solar radiation from a clear sky, is given approximately by

$$I_{ds} = C I_{DN} F_{SS} \tag{22}$$

C, the diffuse radiation factor, is a monthly tabulated value (Table 4) and $F_{SS} = \frac{(1+\cos\Sigma)}{2}$ is a dimensionless angle factor. I_{dg}, the diffuse ground reflected radiation, can be estimated as

$$I_{dg} = \rho_g I_{DN} (C + \sin\beta) F_{SG} \tag{23}$$

where ρ_g is the reflectance of the foreground, $I_{DN}C$ is the portion of diffuse sky radiation and $I_{DN}\sin\beta$ is the direct gain falling on the ground. $F_{SG} = \frac{(1-\cos\Sigma)}{2}$ determines the angle factor for surface to ground.

All angle factors add up to 1.0. Hence, for a horizontal surface we find $F_{SS} = 1.0$ and $F_{SG} = 0.0$ whereas for a vertical surface we will determine $F_{SS} = 0.5$ and $F_{SG} = 0.5$. That

```
{ bond graph model of the flow source of a sun-exposed surface }

model type SUN

  main cut C (. / -f)
  terminal twall
  parameter tilt=90., abs=1.0, sfa=0., OH=0., W=0., ->
            ro=0.33, cr=1.0, cf=1.0
  constant sc=1353., lsm=105., lon=111., lat=32., ->
           pi=3.14156, one=1.
  local int, h, dec, inc, alt, q, ->
        sfar, latr, tiltr, inc1, inc2, inc3, ->
        sou, idn, dif, dir
  external area, time


 sfar=2.*pi*sfa/360.
 latr=2.*pi*lat/360.
 tiltr=2.*pi*tilt/360.
 h=(2.*pi/24.)*(time-12.+ET(time)/60.+(4./60.)*(lsm-lon))
 dec=(2.*pi/360.)*(23.45*sin(2.*pi*(284.+time/24.)/365.))

 inc      = inc1   +   inc2  +  inc3
 inc1     = sin(dec)*sin(latr)*cos(tiltr) -  ->
            sin(dec)*cos(latr)*sin(tiltr)*cos(sfar)
 inc2     = cos(dec)*cos(latr)*cos(tiltr)*cos(h) +  ->
            cos(dec)*sin(latr)*sin(tiltr)*cos(sfar)*cos(h)
 inc3     = cos(dec)*sin(tiltr)*sin(sfar)*sin(h)
 alt = cos(latr)*cos(dec)*cos(h) + sin(latr)*sin(dec)

 sou = sc*(1.+0.033*cos(2.*pi*(time/24.)/365.))
 idn = sou/EXP(B(time)/BOUND(0.05,10.0,alt))

 dir = idn*inc*(area-OH*W*(alt/sqrt(1.-alt**2.)))
 dif = idn*area*(C(time)*(1.+cos(tiltr))/2.+ ->
       (C(time)+alt)*ro*(1.-cos(tiltr))/2.)

 int = abs*(cr*BOUND(0.,1.E10,dir)+ ->
            cf*BOUND(0.,1.E10,dif))/twall
 q = int*SIGN(one,alt)
 BOUND(0.,1.E10,q) = f

end
```

Figure 20: DYMOLA code for the SUN-model

makes sense because in this case we don't expect a horizontal surface to receive ground reflected radiation.
The DYMOLA realization of the model type SUN is shown in Fig.(20).

Let us now look closer at the code of the flow source model for an exposed surface. The chosen parameters are the ground reflectivity of desert soil, $ro = 0.33$, the solar constant sc, the longitude of the reference meridian for Mountain Standard Time (MST) $lsm = 105°$, and the location of our test-structure in Tucson, AZ at approximately 32° latitude and 111° longitude.

Several *BOUNDs* were used in order to determine values at the right range. The bound of $alt = \sin\beta$ prevents the division by zero when the sun hits the horizon and the bounds for dir and dif allow only positive values of direct and diffuse radiation. The last bound of q was needed to ensure that radiation only occurs between sunrise and sunset. This was accomplished by multiplying the positive value of int with the *sign* of $\sin\beta$ before bounding it.

An *overhang* OH was also provided which approximately results in a reduction of the exposed area when multiplying OH by $\tan\beta = \frac{\sin\beta}{\sqrt{1-\sin^2\beta}}$ and the width W of the shaded wall. Furthermore, the parameters cf and cr have been introduced which serve as 'a switch'. So, one can decide whether to look at the direct gain or diffuse radiation separately or together, respectively. In order to derive the right flow source, i.e., entropy flow, the radiation has to be multiplied by the *absorbtance abs* of the material and must be divided by the adequate reference temperature. It also should be mentioned that different from the source of extraterrestrial radiation, which the ASHRAE method suggests, the formula of Duffie & Beckman has been chosen to determine *sou* because this value tends to be closer to reality at a location like Tucson.

3.3.5 Flow source of a transmitting surface – model type SHGF

The ability of glazing materials to transmit, reflect, or absorb solar radiation depends on the wavelength of the radiation, chemical composition, thickness of the material, and the incidence angle. The heat balance in particular for a sunlit single glazing material, as shown in Fig.(21), is:

$$I_t = q_R + q_S + q_T + q_{RC_o} + q_{RC_i} \tag{24}$$

where q_{RC_o} and q_{RC_i} are rates of heat loss by radiation and convection inward and outward, respectively.

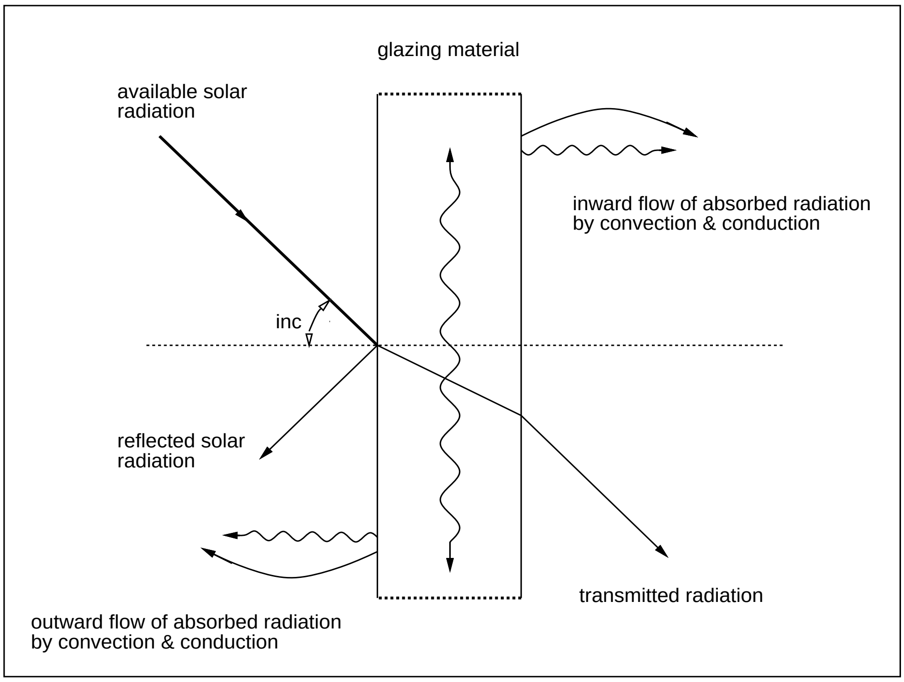


Figure 21: Heat balance for sunlit glazing material

The total rate of heat gain due to radiation is determined by [5]

$$Q_{rad} = \tau I + N_i(\alpha I) \tag{25}$$

where τ := transmissivity , I := available solar radiation, and $N_i(\alpha I)$:= inward flow of absorbed solar radiation. For unshaded single glazing the factor for the inward flow becomes $N_i = \frac{U}{h_o}$ with U := the overall heat transfer coefficient and h_o := the outside film coefficient. Then, Equ.(25) can be written as $Q_{rad} = FI$ by introducing the Solar Heat Gain Coefficient $F = \tau + \frac{U\alpha}{h_o}$ for single glazing material. As we see, F varies with the incident angle since τ and α depend on θ and must be calculated for different fenestration types. In order to avoid calculations for all variations of fenestration, ASHRAE [5] suggests a procedure using a reference glazing material, DSA, double-strength sheet glass, to determine the reference heat gain, designated as the *Solar Heat Gain Factors*

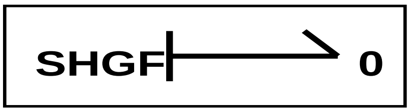


Figure 22: Bond graph of the SHGF-model

(SHGF).The total heat gain is then

$$Q_{rad} = F \cdot I = SC \cdot SHGF \tag{26}$$

where SC, the shading coefficient, is defined as

$$SC = \frac{Solar\,Heat\,Gain\,of\,Fenestration}{Solar\,Heat\,Gain\,of\,Reference\,Glass} = \frac{F\,of\,Fenestration}{F\,of\,DSA} \tag{27}$$

Values of those coefficients have been determined for a wide range of fenestrations. Therefore, and because this method is commonly used for simulation software, the flow source due to irradiation of transmitting material will be provided the same way in this program.

The bond graph of the SHGF-model is given in Fig.(22) and the DYMOLA code for the model type SHGF can be found in Fig.(23).

The solar gain values dir and dif which have been derived as in the model type SUN procedure are used to determine the transmitted radiation rtr and the absorbed radiation rab. The functions $SIGT1$, $SIGA1$ and $SIGT2$, $SIGA2$ calculate transmission and absorption coefficients for direct solar gain and diffuse solar radiation, respectively. Those functions consider variation with the incident angle of the solar-optical properties (transmittance and absorbtance) of the DSA-glass. The functions can be found in the Fortran Library of the program in Appendix A.

Neglecting the heat resistance of the glass, the overall heat transfer coefficient becomes $U = \left(\frac{1}{h_o} + \frac{1}{h_i}\right)^{-1}$ and the inward flow fraction $N_i = \frac{h_i}{h_i + h_o}$ with reference film coefficients $h_i = 8.3\frac{W}{m^2}$ for still air (natural convection) and $h_o = 22.7\frac{W}{m^2}$ for $12\frac{km}{h}$ $(7.5mph)$ average wind speed, a sufficient value for Tucson average. Multiplied by the shading coefficient SC and divided by the reference temperature $tref$ of the node where we want to attach the flow source, we end up with the entropy flow due to irradiation on a transmitting surface.

```
{ bond graph model of the flow source of a transmitting surface }

model type SHGF

  main cut C (. / -f)
  terminal tref
  parameter OH=0., W=0., sfa=0., sc=1., tilt=90., ->
            ro=0.33, cr=1.0, cf=1.0, hi=8.3, ho=22.7
  constant pi=3.14156, lsm=105., lon=111., lat=32., ->
           one=1.0
  local int, h, dec, inc, alt, p, q, rtr, rab, ni, shgf, ->
        sfar, latr, tiltr, inc1, inc2, inc3, idn, dif, dir
  external area, time


 sfar=2.*pi*sfa/360.
 latr=2.*pi*lat/360.
 tiltr=2.*pi*tilt/360.
 h=(2.*pi/24.)*(time-12.+ET(time)/60.+(4./60.)*(lsm-lon))
 dec=(2.*pi/360.)*(23.45*sin(2.*pi*(284.+time/24.)/365.))

 inc      = inc1 + inc2 + inc3
 inc1     = sin(dec)*sin(latr)*cos(tiltr) -  ->
            sin(dec)*cos(latr)*sin(tiltr)*cos(sfar)
 inc2     = cos(dec)*cos(latr)*cos(tiltr)*cos(h) +  ->
            cos(dec)*sin(latr)*sin(tiltr)*cos(sfar)*cos(h)
 inc3     = cos(dec)*sin(tiltr)*sin(sfar)*sin(h)
 alt = cos(latr)*cos(dec)*cos(h) + sin(latr)*sin(dec)

 idn = A(time)/EXP(B(time)/BOUND(0.05,10.0,alt))

 dir = idn*inc*(area-OH*W*(alt/sqrt(1.-alt**2.)))
 dif = idn*area*(C(time)*(1.+cos(tiltr))/2. + ->
                 (C(time)+alt)*ro*(1.-cos(tiltr))/2.)

 rtr = cr*BOUND(0.,1.E10,dir)*SIGT1(inc) + cf*dif*2.*SIGT2(inc)
 rab = cr*BOUND(0.,1.E10,dir)*SIGA1(inc) + cf*dif*2.*SIGA2(inc)
 ni = hi/(hi + ho)
 shgf = rtr + ni*rab

 int = sc*shgf/tref
 BOUND(0.,1.E10,int) = p
 q = p*SIGN(one,alt)
 BOUND(0.,1.E10,q) = f

end
```

Figure 23: DYMOLA code for the SHGF-model

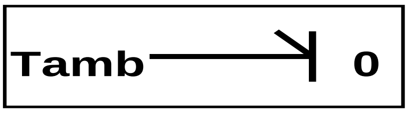


Figure 24: Bond graph of the SE-model

```
{ bond graph effort source, dry-bulb temperature }

model type Tamb

  main cut A (e / .)
  local mt
  external time

  mt = TMEAN(time)
  TOUT(time,mt) = e

end
```

Figure 25: DYMOLA code for the SE-model

3.3.6 Effort Source – The Ambient Temperature

Effort sources according to the bond graph terminology for thermal systems are temperature sources or sinks. Concerning the envelope of a structure it can be found that the living conditions inside of a building are mainly affected by the outside dry-bulb temperature, or ambient temperature. The bond graph of the outside temperature is given in Fig.(24).

In most of the building analysis programs a weather file is provided containing information about radiation, wind speed, and temperatures. Because such a weather file was not available for our purpose, weather conditions were provided using deterministic methods. In case of the outside dry-bulb temperature, a table with monthly mean temperatures $TMEAN$ has been obtained and integrated into the experiment file containing monthly average temperatures extracted from the CALPAS3 weather file. In order to determine hourly temperatures for any day of the year, daily mean temperatures have been derived by interpolation between tabulated values and later multiplied with a factor to obtain an hourly value. Those factors are derived from hourly temperatures of an average day per month. By dividing the hourly data by the mean value for the same month and by taking an average of all values over the year, the right factors have been found. The calculation of hourly temperatures is done by a Fortran routine providing time and mean temperature. Factors and monthly mean temperatures are presented in Fig.(26).

The DYMOLA code is shown in Fig.(25). Table $TMEAN$ can be found in the experiment file and function $TOUT$ in the Fortran Library in Appendix A.

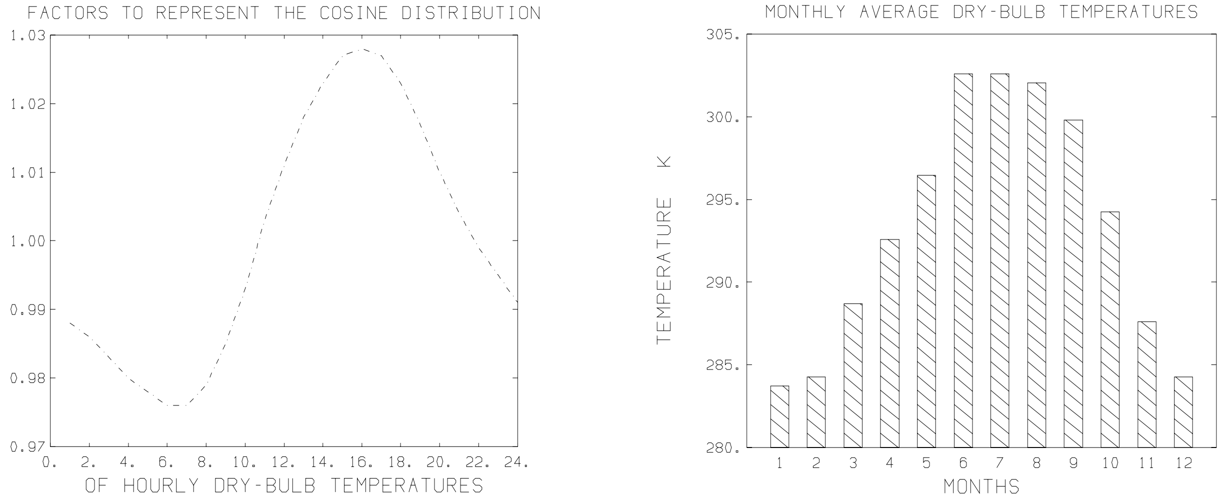


Figure 26: Hourly conversion factors and monthly mean temperatures

3.3.7 The External Walls – model type EXWALL

After having discussed the basic submodels reflecting heat transfer phenomena, we start building construction elements. The bond graph of a wall exposed to solar radiation and outside temperature is shown in Fig.(27).

The *model type EXWALL* consists of the submodels SUN, SE, $C1D$, $C1V$, mC providing the solar radiation, outside temperature, conduction, and inside – outside convective heat transfer, respectively. In order to avoid an algebraic loop, another capacity was added to the conduction chain.

The DYMOLA code of the exwall bond graph is shown in Fig.(28).

The parameters used in that model determine a $40.6cm(16'')$ thick, mud adobe wall with a plaster that provides a absorbtance *abs* of 50%. The thermal properties of adobe

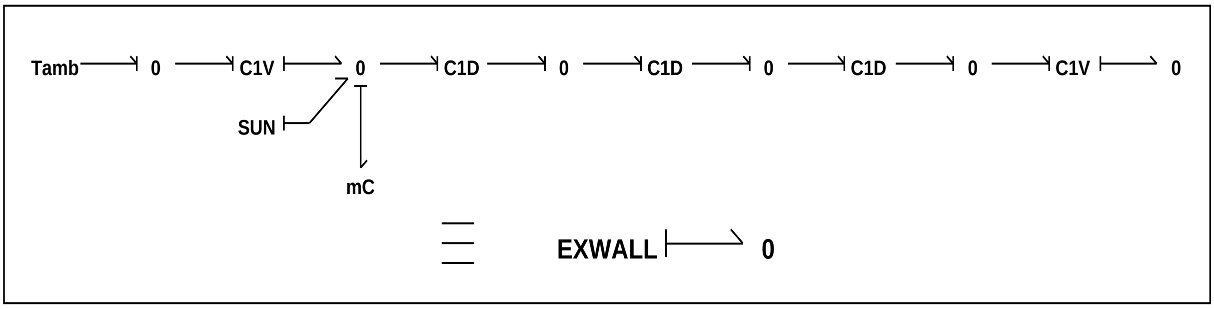


Figure 27: Bond graph for a external wall

```
{ bond graph model of an external wall }

model type EXWALL

  submodel (SUN) rad (tilt=tilt, abs=abs, sfa=sfa, OH=OH, W=W)
  submodel (Tamb) tamb
  submodel (C1D) d1 (k=k1, l=l1, gamma=gamma2, m=m), ->
                 d2 (k=k2, l=l2, gamma=gamma2, m=m), ->
                 d3 (k=k3, l=l3, gamma=gamma3, m=m)
  submodel (C1V) v1 (h=hout), v2 (h=hinv)
  submodel (mC)  c1 (gamma=gamma1, m=m) (ic e=288.0)

 main cut A (e / -f)

 parameter hout=22.7, hinv=8.3, k1=1.7, k2=1.7, k3=1.7, ->
           l1=0.14, l2=0.14, l3=0.14, ->
           gamma1=840., gamma2=840., gamma3=840., ->
           dens=1762., thk=0.41, area=1.0, sfa=0.0, tilt=90. ->
           OH=0.0, W=0.0, abs=0.5
 local m
 internal  area, time
 external  time

 connect (P) d1-d2-d3
 connect v1 from tamb to d1:Cx
 connect d1:Cx at c1
 connect rad at d1:Cx
 connect v2 from d3:Ci to A

 m = area*thk/4.*dens
 rad.twall = v1.ei

end
```

Figure 28: DYMOLA code for the EXWALL-model

	Thermal Conductivity k $[\frac{W}{m^2 K}]$	Specific Heat gamma $[\frac{J}{kgK}]$	Density dens $[\frac{kg}{m^3}]$
Adobe [19, 20]	1.7	840	1762
Steel [5]	45.3	500	7850
Fiberglass [5]	0.0385	657	19.2
Plywood [5]	0.1143	1214	544
Brick [5]		835	2083
Concrete [5]		880	2300

Table 6: Properties of building materials

	Average Film Coefficient [5] $[\frac{W}{m^2 K}]$
Free Convection, Vertical Surface	$h_{in,vertical} = 8.3$
Free Convection, Horizontal Surface	$h_{in,horizontal} = 7.4$
Forced Convection with $12\frac{km}{h}$ $(7mph)$ wind speed	$h_{out} = 22.7$

Table 7: Convection film coefficients

are given in Table 6. The film coefficients for the wall inside and outside convection can be found in Table 7.

The diffusion chain through the wall contains three conduction cells C1D. Therefore, the 0.41m thick wall was divided by three to obtain $l1 - l3$. However, in order to incorporate the additional mC-model $c1$ the mass of the wall was sliced into four equal parts. It can be noted that *time* and *area* are by design global variables. However, while time reaches through the entire model of the house, the area is the same throughout the model type exwall. So, the area of the external wall must only be assigned once when calling for the submodel (EXWALL). Because we assume that the exwall is always exposed to outside temperature and irradiation, only one cut is provided which connects the model to the inside of the building.

3.3.8 The Roof – model type ROOF

The bond graph for the roof is very similar to the model of the external wall except for the different constructive structure. The roof construction is a rather low mass component comprising a $3.8cm\,(1\frac{1}{2}'')$ deep corrugated steel roof and a $20cm\,(8'')$ fiberglass batt insulation. The fiberglass batt is covered by $1.6cm\,(\frac{5}{8}'')$ plywood. The properties of all building materials can be found in Table 6 as well as the values for the average film coefficients in Table 7. The roof is exposed to the north $sfa = 180°$, $7°$ tilted and has an absorbtance $abs = 0.2$ due to the metallic surface. The DYMOLA code is given in Fig.(29).

We have decided to use one C1D-model for each construction layer and express the fiberglass batt with two mC-elements. Besides that the code is the same as for the external wall.

3.3.9 The Internal Wall – model type INTWALL

The model type INTWALL is basically the model for the external wall which is not exposed to solar radiation and outside temperature. This model describes the heat transfer and heat storage for a wall connecting two spaces. The bond graph can be found in Fig.(30) and Fig.(31) shows the DYMOLA code.

The parameters are verifying a $30cm(12'')$ thick adobe wall with average film coefficient for free convection on a vertical surface. Because the internal wall is connected to two adjacent spaces two cuts are provided with a path in between.

3.3.10 The Frame Wall – model type FRAME

In order to integrate elements with neglectable heat storage like window frames, doors, etc, a model type FRAME has been designed, Fig.(32).

This model only contains a mGS type model which is used to determine the entropy flow due to the temperature difference between the inside and outside, air providing an overall heat transfer coefficient U. The frame wall model can be used for sun exposed elements as well as for inside structures by manipulating the absorbtance abs. The adequate

```
{ bond graph model of the roof }

model type ROOF

  submodel (SUN) rad (tilt=tilt, abs=abs, sfa=sfa, OH=OH, W=W)
  submodel (Tamb)  tamb
  submodel (C1D) d1 (k=k1, l=l1, gamma=gamma2, m=m2), ->
                 d2 (k=k2, l=l2, gamma=gamma2, m=m2), ->
                 d3 (k=k3, l=l3, gamma=gamma3, m=m3)
  submodel (C1V) v1 (h=hout), v2 (h=hinh)
  submodel (mC)  c1 (gamma=gamma1, m=m1) (ic e=288.0)

 main cut A (e / -f)

 parameter hout=22.7, hinh=7.4, k1=45.3, k2=0.0385, k3=0.1143, ->
           l1=0.038, l2=0.2, l3=0.016, ->
           gamma1=500., dens1=7850., gamma2=657., dens2=19.2, ->
           gamma3=1214., dens3=544., area=1.0, sfa=180., tilt=7., ->
           OH=0.0, W=0.0, abs=0.2
 local     m1, m2, m3
 internal area, time
 external time


 connect (P) d1-d2-d3
 connect v1 from tamb to d1:Cx
 connect d1:Cx at c1
 connect rad at d1:Cx
 connect v2 from d3:Ci to A

 m1 = area*l1*dens1
 m2 = 0.5*area*l2*dens2
 m3 = area*l3*dens3
 rad.twall = v1.ei

end
```

Figure 29: DYMOLA code for the ROOF-model

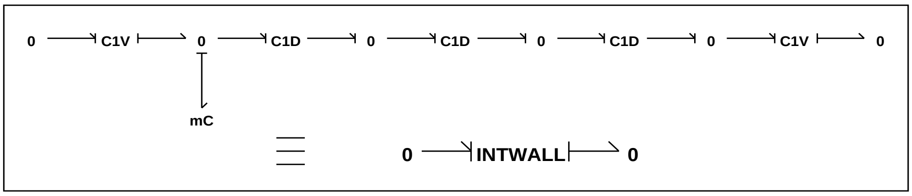


Figure 30: Bond graph of the INTWALL-model

```
{ bond graph model of an internal wall }

model type INTWALL

  submodel (C1D) d1 (k=k1, l=l1, gamma=gamma2, m=m), ->
                 d2 (k=k2, l=l2, gamma=gamma2, m=m), ->
                 d3 (k=k3, l=l3, gamma=gamma3, m=m)
  submodel (C1V) v1 (h=hinv), v2 (h=hinv)
  submodel (mC)  c1 (gamma=gamma1, m=m) (ic e=288.0)

  cut A (ea / fa), B (eb / -fb)
  main cut C[A B]
  main path F<A - B>

  parameter hinv=8.3, k1=1.7, k2=1.7, k3=1.7, ->
            l1=0.1, l2=0.1, l3=0.1, ->
            gamma1=840., gamma2=840., gamma3=840., ->
            dens=1762., thk=0.3, area=1.0
  local m
  internal area, time
  external time

  connect (P) d1-d2-d3
  connect A at v1:Cx
  connect v1:Ci at d1:Cx
  connect d1:Cx at c1
  connect d3:Ci at v2:Cx
  connect v2:Ci at B

  m = area*thk/4.*dens

end
```

Figure 31: DYMOLA code for the INTWALL-model

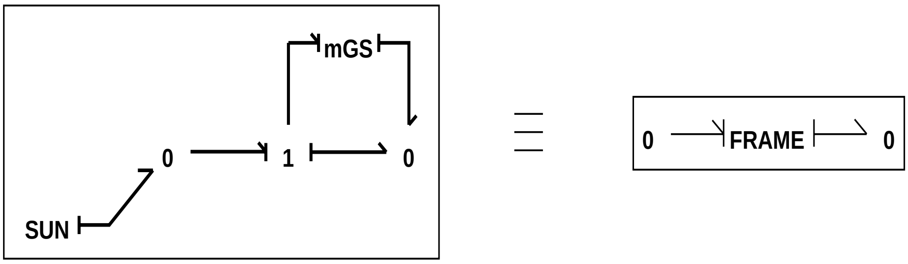


Figure 32: Bond graph of the FRAME-model

```
{ bond graph model of a low mass element }

model type FRAME

  submodel (SUN) rad (tilt=tilt, abs=abs, sfa=sfa, OH=OH, W=W)
  submodel (mGS) Gcell (k=u, l=1.0)
  submodel (bond) B1, B2, B3
  node n1, n2, n3

  cut A (ea / fa), B (eb / -fb)
  main cut C[A B]
  main path F<A - B>

  terminal tamb
  parameter area=1.0, tilt=90., sfa=180., abs=0.2, OH=0.0, W=0.0, ->
            u=0.189
  internal area, time
  external time

  connect A at n1
  connect B1 from n1 to n2
  connect B2 from n2 to B
  connect B3 from n2 to n3
  connect Gcell from n3 to B
  connect rad at n1

  rad.twall = tamb

end
```

Figure 33: DYMOLA code for the FRAME-model

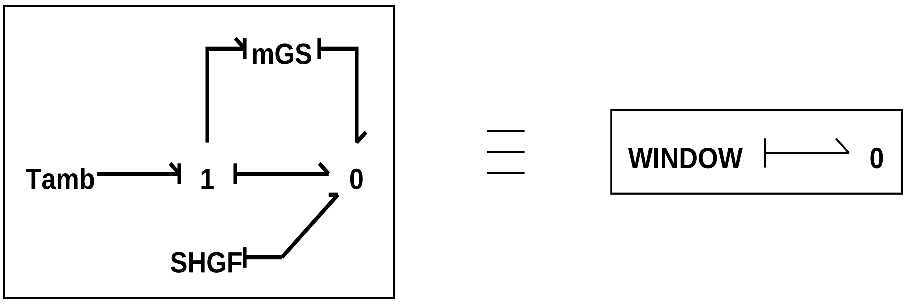


Figure 34: Bond graph for a sunlit glazing surface

```
{ bond graph model of the heat transfer through a glazing }
{ material due to radiation and temperature difference    }

model type WINDOW

  submodel (SHGF) beam (OH=OH, W=W, sfa=sfa, sc=sc)
  submodel (Tamb) tamb
  submodel (mGS) Gcell (k=u, l=1.0)
  submodel (bond) B1, B2, B3
  node n1, n2, n3

 main cut A (e / -f)

 parameter area=1.0, sfa=0.0, sc=0.85, OH=0.0, W=0.0, u=3.4
 internal area, time
 external time

 connect tamb at n1
 connect B1 from n1 to n2
 connect B2 from n2 to A
 connect B3 from n2 to n3
 connect Gcell from n3 to A
 connect beam at A

 beam.tref = Gcell.e2

end
```

Figure 35: DYMOLA code for the WIN-model

DYMOLA code appears in Fig.(33).

3.3.11 The Window – model type WIN

Heat flow through fenestration areas are affected by solar irradiation and transmittance as well as by outdoor – indoor temperature difference. The bond graph to this is shown in Fig.(34).

The parameter have been found for a $\frac{1}{8}''$ double glazed clear glass without shades with average outside wind speed at $12\frac{km}{h}(7mph)$ as $SC = 0.85$ $U = 3.4\frac{W}{m^2K}$ [5]. The corresponding DYMOLA program can be found in Fig.(35).

3.3.12 The Slab – model type SLAB

The slab under a building is on one hand important because of its great mass and heat capacity. Heat storage occurs and is dependent on inside room temperatures and the thermal capacity of the slab material. On the other hand, a heat loss through the slab is inevitable due to its earth contact. Whereas it is obviously not easy to determine the temperature underneath the house, many simulation programs consider only the loss through the slab edge due to the outside temperature T_{amb}. The bond graph for the slab is shown in Fig.(36).

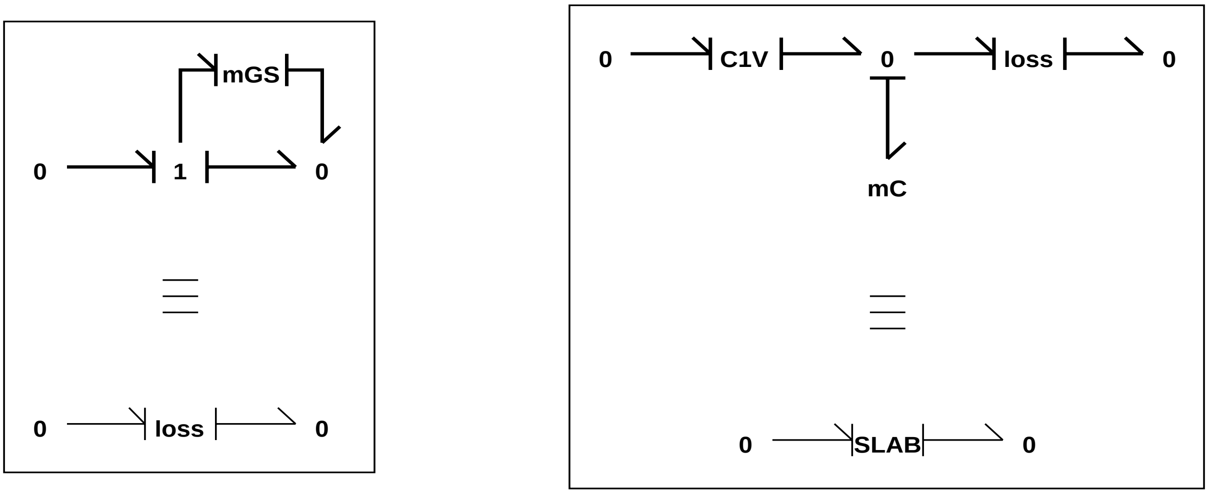


Figure 36: Bond graph for a slab

```
{ bond graph the loss through the slabedge }

model type loss

  submodel (mGS) Gcell (k=u, l=1.0)
  submodel (bond) B1, B2, B3
  node n1, n2

  cut Cx(ex/fx), Ci(ei/ -fi)
  main path P<Cx - Ci>
  parameter u=1.0, perim=1.0, thk=1.0
  internal area
  local area

  connect B1 from Cx to n1
  connect B2 from n1 to n2
  connect Gcell from n2 to Ci
  connect B3 from n1 to Ci

  area = perim*thk

end
```

```
{ bond graph model for the loss through slab }

model type SLAB

  submodel (C1V) v1 (h=hinh, l=1.0)
  submodel (loss) slabloss (u=u, perim=perim, thk=thk)
  submodel (mC) c (gamma=gamma, m=m) (ic e=288.0)
  node n1

  cut A (ea / fa), B (eb / -fb)
  main cut C[A B]
  main path F<A - B>

  parameter hinh=7.4, u=5.11, area=1.0, gamma=835., ->
            dens=2083., thk=0.2, perim=1.0
  local  m
  internal area

  connect v1 from A to n1
  connect c at n
  connect slabloss from n to B

  m = area*thk*dens

end
```

Figure 37: DYMOLA code for the SLAB-model

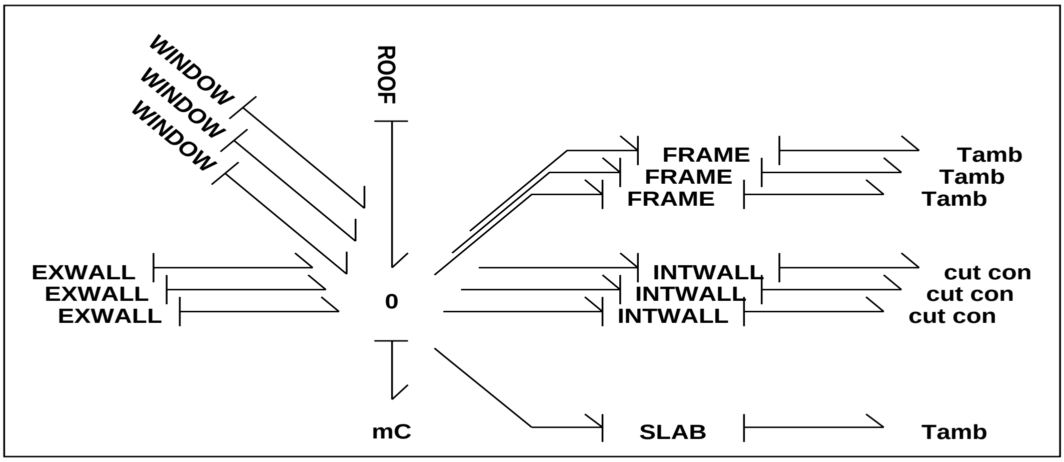


Figure 38: Bond graph for a space

The DYMOLA code for the model type loss and the SLAB-model are given in Fig.(37).

The model type loss was necessary because the perimeter rather than the area of the slab is decisive. However, the area is defined as a global variable throughout the slab. For the bond graph model, an overall heat transfer coefficient has been used for the edge loss, $U = 5.11\frac{W}{m^2 K}$. The properties of the brick slab inside the building and the concrete slab which is used for the sunspace can be found in Table 6. The model type slab is later connected between the inside space conditions and the ambient temperature.

3.3.13 The entire House

The full description of the house according to the floor plan contains three interior rooms and an attached sunspace. The previously derived models for the construction elements were used to express the model types SPACE.

Each space is modelled with a capacity *croom* which represents the air mass inside the space. All elements at the boundaries of each space are connected to that capacity. In this way all needed submodels are referenced and assigned with the adequate properties. Cuts are provided which correspond to the adjacent spaces, namely $con12$, $con13$ and $con1ss$. The bond graph of any given space can be seen in Fig.(38).

The DYMOLA code for all four spaces is given in Fig.(39), Fig.(40), Fig.(41) and Fig.(42).

```
{ bond graph model of room 1 covering the south side of the adobe building }

model type SPACE1

 submodel (ROOF) r1 (area=72.5)
 submodel (EXWALL) southwall (area=15.7, sfa=0.0, OH=0.3, W=12.), ->
                   northwall (area=6.60, sfa=180., OH=0.3, W=12.), ->
                   eastwall  (area=11.3, sfa=270., OH=0.3, W=8.), ->
                   westwall  (area=20.7, sfa=90., OH=0.3, W=8.)
 submodel (WINDOW) southwin1 (area=2.23, sfa=0.0), ->
                   southwin2 (area=2.23, sfa=0.0), ->
                    northwin1 (area=1.5, sfa=180.)
 submodel (INTWALL) intw2 (area=6.6) ->
                    intw3 (area=15.2) ->
                    sswall (area=12.9, thk=0.41, l1=0.14, l2=0.14, l3=0.14)
 submodel (FRAME) outdoor (area=2.17, abs=0.6, u=3.57), ->
                  indoor1 (area=2.17, abs=0.0, u=3.57), ->
                  indoor2 (area=2.17, abs=0.0, u=3.57), ->
                  ssdoor (area=2.17, abs=0.0, u=3.57), ->
                  sswin (area=1.1, abs=0.0, OH=3.1, W=5.4, u=3.4)
 submodel (SLAB) sl1 (area=72.5, perim=23.7)
 submodel (Tamb) tamb
 submodel (mC)  croom (gamma=gammaroom, m=mroom) (ic e=288.0)

 cut con12 (e2 / -f2), con13 (e3 / -f3), con1ss (es / -fs)

 parameter gammaroom=1000., mroom=265.
 node n1

 internal time
 external time

 connect southwall at n1
 connect northwall at n1
 connect eastwall at n1
 connect westwall at n1
 connect r1 at n1
 connect southwin1 at n1
 connect southwin2 at n1
 connect northwin1 at n1
 connect n1 to intw2 to con12
 connect n1 to intw3 to con13
 connect n1 to sswall to con1ss
 connect n1 to ssdoor to con1ss
 connect n1 to sswin to con1ss
 connect tamb to outdoor to n1
 connect n1 to indoor1 to con12
 connect n1 to indoor2 to con13
 connect n1 to sl1 to tamb
 connect n1 at croom

 sswin.tamb = tamb.e
 outdoor.tamb = tamb.e
 indoor1.tamb = tamb.e
 indoor2.tamb = tamb.e
 ssdoor.tamb = tamb.e

end
```

Figure 39: DYMOLA code for the SPACE1-model

```
{ bond graph model of room 2 at the north-west corner of the adobe house }

model type SPACE2

 submodel (ROOF) r2 (area=15.0)
 submodel (EXWALL) northwall (area=9.0 , sfa=180., OH=0.3, W=12.), ->
                   eastwall  (area= 9.4, sfa=270., OH=0.3, W=8.)
 submodel (WINDOW) northwin (area=1.5, sfa=180.)
 submodel (INTWALL) int (area=6.6)
 submodel (SLAB) sl2 (area=15.0, perim=7.9)
 submodel (Tamb) tamb
 submodel (mC)  croom (gamma=gammaroom, m=mroom) (ic e=288.0)

 cut con12 (e1 / f1), con23 (e3 / -f3)

 parameter gammaroom=1000., mroom=54.
 node n1

 internal time
 external time

 connect northwall at n1
 connect eastwall at n1
 connect r2 at n1
 connect northwin at n1
 connect n1 to int to con23
 connect n1 to sl2 to tamb
 connect con12 at n1
 connect n1 at croom

end
```

Figure 40: DYMOLA code for the SPACE2-model

```
{ bond graph model of room 3 at the north side between room 2 and }
{ room 1 of the adobe house }

model type SPACE3

 submodel (ROOF) r3 (area=15.0)
 submodel (EXWALL) northwall (area=9.0 , sfa=180., OH=0.3, W=12.)
 submodel (WINDOW) northwin (area=1.5, sfa=180.)
 submodel (SLAB) sl3 (area=15.0, perim=4.2)
 submodel (Tamb) tamb
 submodel (mC) croom (gamma=gammaroom, m=mroom) (ic e=288.0)

 cut con13 (e1 / f1), con23 (e2 / f2)

 parameter gammaroom=1000., mroom=54.
 node n1

 internal time
 external time

 connect northwall at n1
 connect r3 at n1
 connect northwin at n1
 connect n1 to sl3 to tamb
 connect con13 at n1
 connect con23 at n1
 connect n1 at croom

end
```

Figure 41: DYMOLA code for the SPACE3-model

```
{ bond graph model of the solar/screen porch attached on the south side }
{ of the adobe house }

model type SUNSPACE

 submodel (WINDOW) southwin (area=10.30, sfa=0., u=4.9), ->
                   eastwin (area=6.20, sfa=270., u=4.9), ->
                   westwin (area=6.20, sfa=90., u=4.9)
 submodel (FRAME) southwall (area=1.0, sfa=0., abs=0.2, u=5.11), ->
                  eastwall (area=3.0, sfa=270., abs=0.2, u=5.11), ->
                  westwall (area=3.0, sfa=90., abs=0.2, u=5.11), ->
                  ssroof (area=17.2, sfa=0., tilt=15., abs=0.3, u=0.23)
 submodel (SLAB) sl (area=17.2,perim=11.7,gamma=880.,dens=2300.,thk=0.1)
 submodel (Tamb) tamb
 submodel (mC)  croom (gamma=gammaroom, m=mroom) (ic e=288.0)

 cut con1ss (e1 / f1)

 parameter gammaroom=1000., mroom=48.6
 node n1

 internal time
 external time

 connect tamb to southwall to n1
 connect tamb to eastwall to n1
 connect tamb to westwall to n1
 connect tamb to ssroof to n1
 connect southwin at n1
 connect eastwin at n1
 connect westwin at n1
 connect n1 at con1ss
 connect n1 to sl to tamb
 connect n1 at croom

 southwall.tamb = tamb.e
 eastwall.tamb = tamb.e
 westwall.tamb = tamb.e
 ssroof.tamb = tamb.e

end
```

Figure 42: DYMOLA code for the SUNSPACE-model

```
{ bond graph model of the adobe house }

model HOUSE

  submodel (SPACE1) room1
  submodel (SPACE2) room2
  submodel (SPACE3) room3
  submodel (SUNSPACE) room4

  input Ti, tbase
  output t1, t2, t3, t4, to, tw1, tw2, tw3, tw4, ->
         h1, h2, h3, h4

  local time
  internal time

  connect room1:con12 to room2:con12
  connect room1:con13 to room3:con13
  connect room1:con1ss to room4:con1ss
  connect room2:con23 to room3:con23

  time = Ti + tbase
  room1::croom.e = t1
  room2::croom.e = t2
  room3::croom.e = t3
  room4::croom.e = t4
  room1::tamb.e = to
  room1::eastwall::c1.e = tw1
  room1::eastwall::d1::Ccell.e = tw2
  room1::eastwall::d2::Ccell.e = tw3
  room1::eastwall::d3::Ccell.e = tw4
  room1::croom.f * room1::croom.e = h1
  room2::croom.f * room2::croom.e = h2
  room3::croom.f * room3::croom.e = h3
  room4::croom.f * room4::croom.e = h4

end
```

Figure 43: DYMOLA code for the entire HOUSE

The last step of assembling the test building means attaching the spaces together and resulting in the model HOUSE, Fig.(43). We use the connection cuts of the spaces to put the whole structure together and provide the needed input and output variables. For the time dependent calculations in our submodels we need to get the simulation time Ti. In order to start calculations at any time of the year, we assign a starting point $tbase$, which can be changed between simulation runs. The values of the output variables can be collected with each simulation run. It has been found meaningful to obtain room temperature $t1$ –$t4$, outside temperature to, temperatures following the distribution over a external wall $tw1$ – $tw4$, and heat flow rates into and out of each space.

4 Results and Analysis

Comparing the predictions of different approaches in energy building analysis programs is rather tricky. This is basically due to the great number of parameters that influence the thermal system and play an important role in the simulation environment. By trying to group those interfering mechanisms we find three of major influences: the *boundary conditions* of our simulation, the *structure interpretation* of the building, and the *methodology* of the approach.

4.1 Boundary Conditions

The way our simulation interacts with its environment is in general expressed in the *weather conditions*. Concerning the performance of the structural envelope the outside conditions are utilized to determine the boundary conditions of the heat equation. In detail our building boundaries are mainly influenced by outside dry-bulb temperature and radiation. But also wind and wet-bulb temperature affect the living conditions in the interior of our space. The wet-bulb temperature has an impact on the application of an evaporative cooler and determines the humidity and, therefore, the subjective perception of temperature, while the wind increases the heat transfer coefficients and natural ventilation.

The choice of boundary conditions is therefore very important and the difference of weather data used will result in varying building performance.

CALPAS3 as well as DOE-2 fall back on collected weather data retrieved from a separate data file. Those data files were derived from observations over several years and represent average hourly temperatures and solar radiation values.

The DYMOLA approach, however, uses a totally deterministic method by providing monthly average dry-bulb temperatures and obtaining the values in between by linear interpolation. The hourly data are derived by superimposing a cosine-distribution.

4.2 Structure Interpretation

The way the building is modelled and the range of available models for construction elements are important in that they affect the accuracy of the results as well as flexibility in describing the structure. The structure interpretation comprises not only the input mechanisms but also the possibilities of obtaining output reports and having access to state and flow variables at any desired time.

4.2.1 Input Procedure

CALPAS3 as well as DOE-2 provide a so-called *input-file* that can be prepared with any text editor using the required keywords and syntax. In this way the user describes his/her building in a very convenient code without having ever seen the models. Depending on the sophistication of the provided options, it can be done in a decent time.

The disadvantage of those input-files is certainly that they require a detailed description about the keywords and their assigned values. Because the user does not know anything about the models hidden behind these keywords the building description relies on the explanations according to the manual. Considering that the manual for DOE-2 comprises three folders with several hundred pages, it requires quite some experience to find the right answers. It has taken approximately 10 days to describe the test structure in DOE-2 and half a week to do the same in CALPAS3, without having any experience with either of the two programs.

However, it should be mentioned that both input-files are very simple and suppress many options in order to keep the discrepancies small and focus on options which are comparable in all approaches. Those options are the general thermal performance regarding conduction/convection heat transfer through the envelope and solar irradiation on exposed and transmitting surfaces.

In the DYMOLA interpretation of the test structure, we had to deal with a different modelling technique. Models which are described through physical equations were assembled into hierarchical representations of construction elements. This way of object-oriented modelling allows the use of those models as often as needed. The data describing the properties and parameters must be provided every time the model is invoked. Therefore, the data is assigned to the models during the process of assembling the building model.

So can we already find differences in the geometrical description of our building. CALPAS3 only allows a modelling of the building envelope and treats the internal wall exclusively from the heat storage perspective, while DOE-2 and the DYMOLA approach put emphasis on the space description with the internal wall as separations between the spaces. In our test structure we therefore distinguish between three spaces and internal heat transfer between them. For the temperature analysis the CALPAS3 building inside temperature has been compared with the temperature of the biggest space *room 1*.

Note that DOE-2 and CALPAS3 use conventional units to express material properties and heat and energy transfer, while the DYMOLA approach stays with SI-Units in order to be consistent with common opinion and because the author has never gotten used to conventional units. However, that meant to convert some properties, where no information in SI-Units could be obtained, and all results in order to compare them.

4.2.2 Output Procedure

In order to meet the demands of a broad clientèle, output-files need to be provided containing every possible combination of output data. The choice of outputs should be easy to derive and be very flexible in the way the data is organized.

CALPAS3 and DOE-2 provide options in their input-files to specify a user composed report or to go with a default report. The reports are divided into monthly, daily or hourly outputs. The report tells about temperatures, hourly heat flows, loads or peak loads and the day of their occurrence. It also summarizes loads and shows highest and lowest temperatures. Yearly or run time summaries together with information about cooling and heating loads over the run time period can be obtained.

CALPAS3 puts more emphasis on balancing the gains and losses and transfers from or into sunspace or storage elements. Furthermore, it only provides a certain number of reports which can be defined in the input-file and are well explained in the manual. For most of the applications those reports are sufficient, but information can only be retrieved for whole construction groups. The heat transfer through one particular wall can not be found, only the heat gain or loss through the whole envelope.

DOE-2 provides reports for all its sub-procedures where basically load analysis data can be obtained. Summaries of peak loads and monthly loads for spaces and building as well as load analysis by components and design parameters can be determined. All other informations can be retrieved as hourly data in an hourly report which has to be created by the user providing a schedule telling which days and hours need to be reported and a variable-list classifying the desired data. The result is a long output-file containing every variable that has ever been calculated. However, it was a little bit clumsy to find the right number standing for the desired variable without having an accurate listing of variables.

The extraction of output data in the DYMOLA description is very convenient and flexible. Rather than defining an output report, every variable can be collected during the simulation process if classified as an output variable in the building model.

4.2.3 Methodologies

The methodologies of all three programs differ in terms of modelling technique, physical expression, and simulation tool.

CALPAS3 as well as DOE-2 follow the approach of a *procedural* language and therefore behave very statically during the modelling process. They also need to have a input file in order to make the building description fit the predesigned models.

DYMOLA, however, is a object-oriented modelling language, i.e., the building will be expressed by using a pool of basic models and assemble them in a hierarchical way.

The three programs are using basically all different approaches to express the physics of heat transfer. We find three proposals to implement the heat diffusion equation Equ.(2) into a simulation language.

CALPAS3 distinguishes between non-massive and storage elements. The non-massive elements are modelled by providing an overall heat transfer coefficient and applying Fourier's law, Equ.(6). The storage, high-mass elements are simulated using a backward difference equation. The transient heat transfer problem is solved by calculating the air temperature with an estimate for the mass temperature and then updating the mass temperature.

DOE-2 uses thermal response factors to evaluate the transient heat transfer. This approach expresses the heat equation by applying the Laplace transform and assumes the boundary conditions represented by a series of pulse functions.

The DYMOLA approach implements the heat equation by discretizing the partial differential equation in space and solving for a set of ordinary differential equations. Hereby, the electrical analogy resembles the transport and storage phenomena occurring in each finite space step. Bond graphs were used to represent the heat transport. The accuracy can be influenced by choosing bigger or smaller space increments.

It is obvious that different algorithms lead to different results in performance as well as in computation times. On the other hand, DYMOLA uses continuous-time simulation while CALPAS3 and DOE-2 follow the discrete-time simulation. A continuous system calculates every computation step, while a discrete system calculates on a hourly basis. However, the DYMOLA approach with bond graphs describing the heat transfer equations will achieve more accuracy than a discrete simulation, but for this we require more computation time.

4.3 Radiation Heat Transfer Analysis

As we have seen in previous chapters, a very important perspective for the thermal performance of our building is the influence of solar radiation. It happens to be the main energy source concerning passive solar heating mechanisms.

Solar radiation has been modelled for irradiation on surfaces and transmitting glazing materials in a similar way for the DYMOLA model. In order to evaluate radiation heat flow, the heat gain through transmitting materials has been chosen because all three programs provide data about transmission of radiation through transparent elements. It will be interesting to see how the three programs perform because of the different approaches they use.

CALPAS3 calculates a *Glass Solar Factor* which expresses the glass type, orientation, ground-reflectance, shading, tilt angle, etc., of the specified window and multiplies it by the solar beam radiation given in the weather data. The diffuse radiation portion will be the result of the multiplication of a diffuse radiation factor (DFSE) and the diffuse radiation on a horizontal surface extracted from the weather data. Added together we derive the solar radiation through the specified glass.

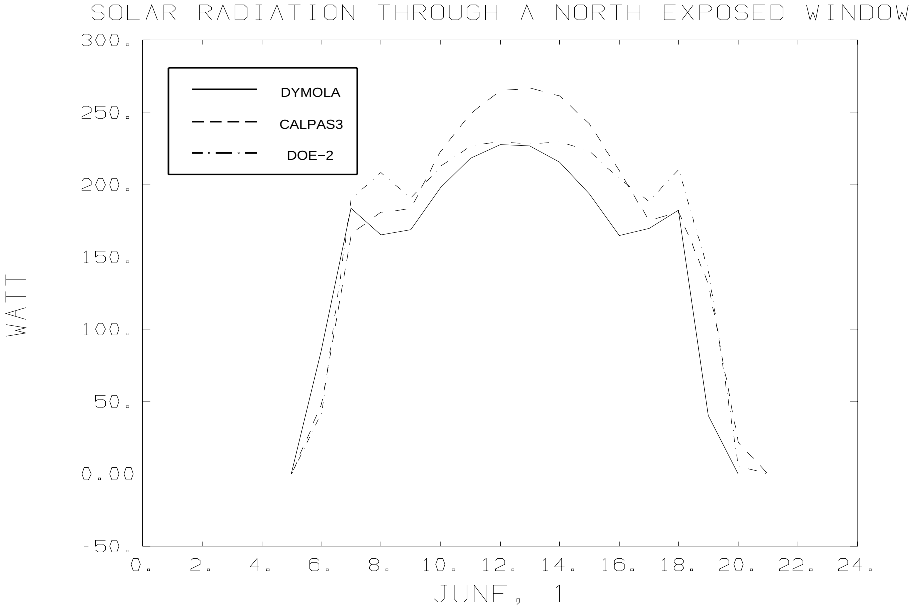


Figure 44: Solar radiation through a north exposed window

The procedure of DOE-2 that extracts hourly data of the solar gain determines the available solar radiation in a similar way as the ASHRAE method but uses measured data for direct and diffuse radiation, extracted from the weather data. After absorbtion and transmission coefficients have been calculated for the specified class type, the total heat gain will be calculated and shading effects will be added.

The DYMOLA model, however, determines the transmitted radiation of a window by calculating the *Solar Heat Gain Factor* according to the method given in the ASHRAE Fundamentals [5] and multiplies by a glass-specific shading coefficient. In those calculations solar radiation on a clear day is assumed and tabulated values are used resembling average U.S. data.

Three windows were selected in order to investigate the results determined by the different approaches. Results obtained for a north and a south window of the building and a east exposed window of the sunspace will be shown in Fig.(44), Fig.(45) and Fig.(46).

All windows have the same properties of double-glazed, 3/16” clear glass with the ASHRAE shading coefficient of 0.85 and perform at a randomly picked day, namely June 1.

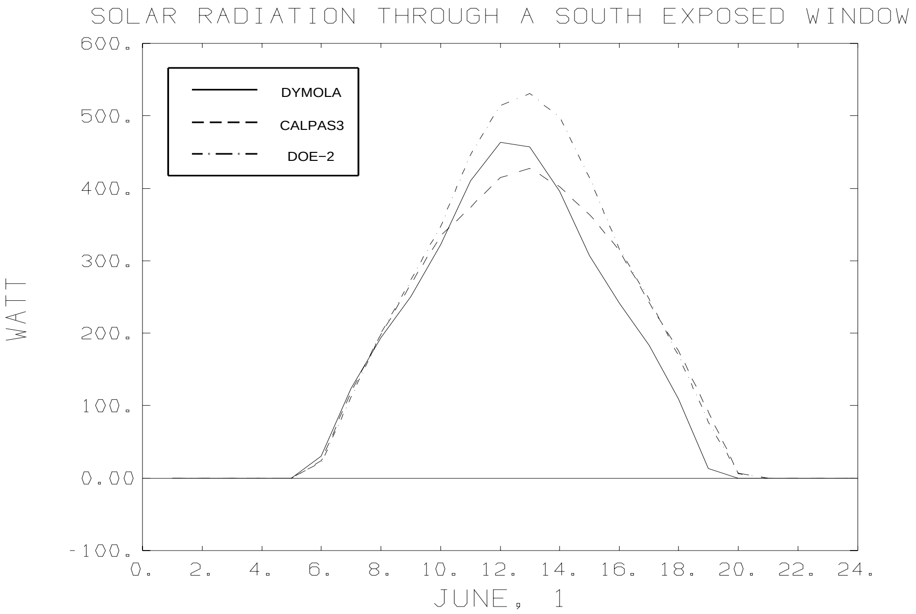


Figure 45: Solar radiation through a south exposed window

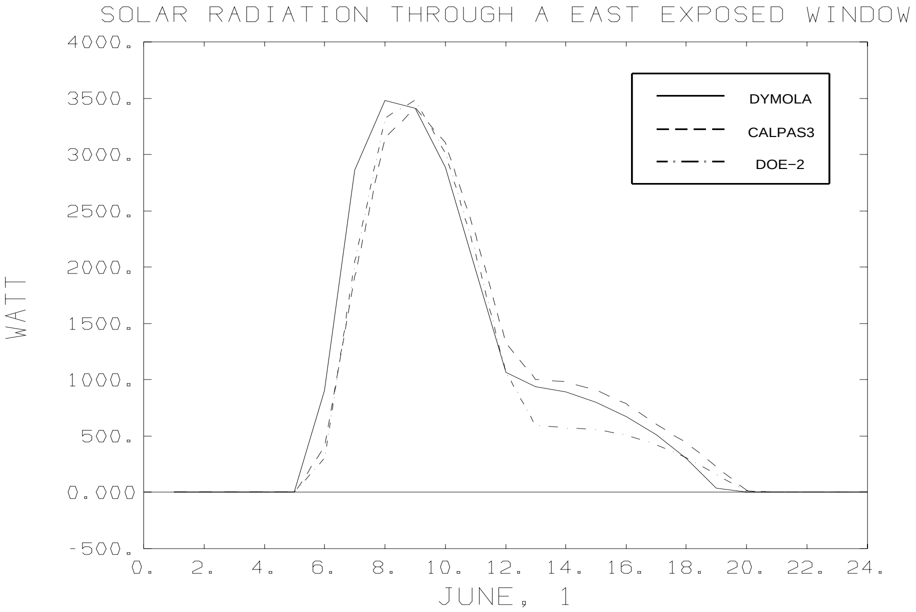


Figure 46: Solar radiation through a east exposed window

We realize that the shape of all three curves is sufficiently similar, even though they do not fit exactly. However, considering the different approaches we could not expect exact agreement. The graphs do not look smooth and some of the points seem to be awry. This is due to the way those data were collected. CALPAS3 and DOE-2 only calculate heat flows on a hourly basis, therefore, the communication interval for the DYMOLA/ACSL simulation has been set to one hour and hourly values have been obtained.

Looking at the north exposed window we find the two peaks in the morning and evening which are due to the influence of direct gain during these hours. The dome in between represents exclusively the diffuse radiation occurring during this time.

At the east exposed window we find the same, high direct gain in the morning hours and only diffuse radiation in the afternoon. The direct solar beam drops down at solar noon.

The south exposed window has diffuse and direct radiation superimposed with a peak at solar noon. The reason why these curves are not symmetrical has to do with the fact that the *Equation of Time* shifts solar noon to 12.24 pm and data are collected on hourly basis.

4.4 Temperature Analysis

The temperatures determined by all three programs have been a main part of the validation for the new DYMOLA approach. Temperatures have been derived for all spaces of our test structure. However, it has been found that in order to determine the space temperatures in all programs, some manipulation had to be applied, which will be explained next.

CALPAS3 provides by default a backup heating and cooling system which is used to supply the living space with desired conditions. The purpose of investigating the thermal behavior of the building envelope, however, requires the suppression of any auxiliary heating or cooling system. This has been achieved by setting the thermostat to its maximal and minimal border. The heating-setpoint has therefore been fixed to 0° Fahrenheit (F) while the cooling-setpoint were frozen at $150°F$. The house, sunspace and outside dry-bulb temperature has then been obtained by the hourly condition report for the months of June and December.

The determination of space temperatures in DOE-2 skipping the heating and cooling system is also fairly tricky. The *loads analysis* program determines the space load according to a design space temperature. Therefore, a *system* has to be designed and the zone conditions in this part must be manipulated. This has been done with *design-heat-temperature* = $10°F$ and *design-cool-temperature* = $180°F$. The second problem was to provide the hourly space temperatures in one of the output reports. DOE-2 is capable of hourly reports of any variable appearing on the report *variable-list* but finding the right number for the desired variable without having an updated manual is fairly demanding.

In DYMOLA the output of any variable is rather simple because every value that has ever been used can be addressed from the main model by calling for it providing the path through all submodels. This might result in a longish expression sometimes but one will always succeed having done it properly, e.g.

$$room1 :: eastwall :: d1 :: Ccell.e = tw2$$

call for the effort variable e of the capacity submodel $Ccell$ in a one-dimensional cell $d1$ of the *eastwall* in $room1$ and assigns it to the output variable $tw2$.

In this way we found hourly temperatures of all three programs for the provided spaces for the months June and December. Because CALPAS3 calculates only one temperature for the inside of the house, while the other programs provide temperatures for all three spaces, the room temperature of the biggest room $room1$ has been taken to compare the house inside conditions.

After collecting the data the temperatures have been extracted from the output files and written in matrix form in a CTRL-C [13] file. This was necessary in order to convert the CALPAS3 and DOE-2 temperatures into degrees Kelvin (K) and compare the results using the drawing procedures of CTRL-C.

In Fig.(47) the outside dry-bulb temperatures are shown for the first three days of June. One may realize that the DYMOLA approximation of a cosine-distribution to determine the hourly temperatures reflects the measured data of CALPAS3 and DOE-2 fairly well. Furthermore, we realize that CALPAS3 and DOE-2 obviously use the same weather data.

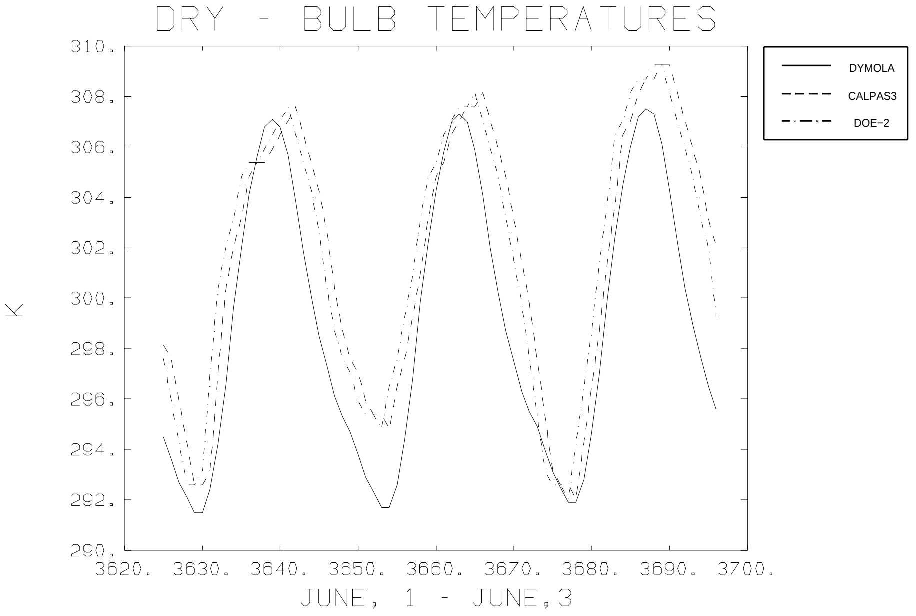


Figure 47: Dry-bulb temperatures for June 1 – June 3

The y-axis represents the temperatures in *absolute* temperature (K), while the x-axis indicates the time in *julian* hours.

Fig.(48) shows the inside conditions of our building while Fig.(49) reflects the sunspace temperatures. The shapes of those graphs look again very similar, even though the values differ sometimes from each other. The graphs for the conditions in December are given in Appendix C.

The house conditions are determined by high mass effects, that means we find moderate temperatures and only a small temperature range between day and night. The thermal behavior is influenced by a time lag due to the high heat capacity of the thick mass walls, slab and other mass elements. By looking at the chart of the room temperatures for June, and even better for the December conditions, we will realize that the temperature ascent in the morning after passing the lowest point is almost identical. The descent from the highest point, however, looks different. The temperature course, best seen in the graph for December, changes after sunset and the DYMOLA model shows a different time constant in the temperature drop especially compared with DOE-2's. The reason for this might be the heat stored in the mass walls and than released after a time

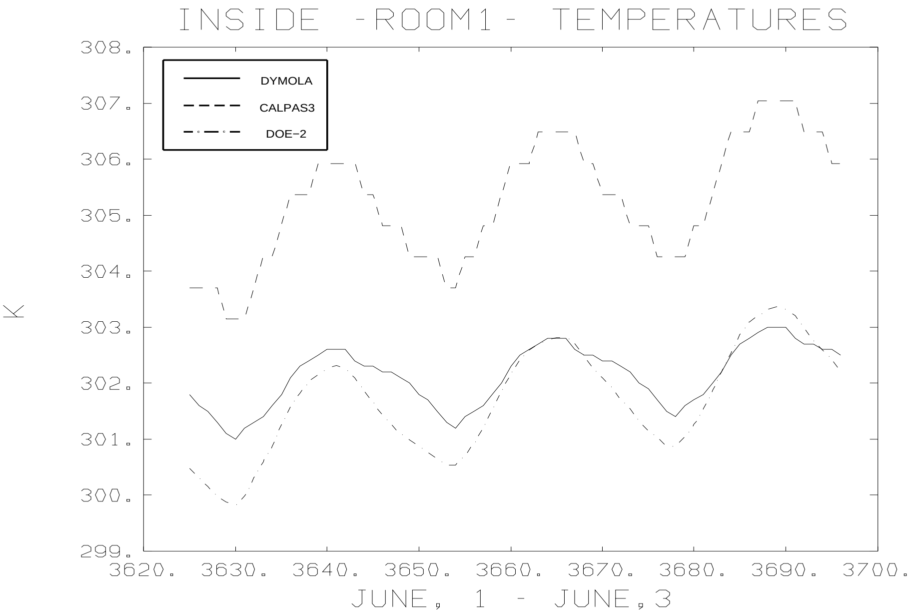


Figure 48: Inside temperature for June 1 – June 3

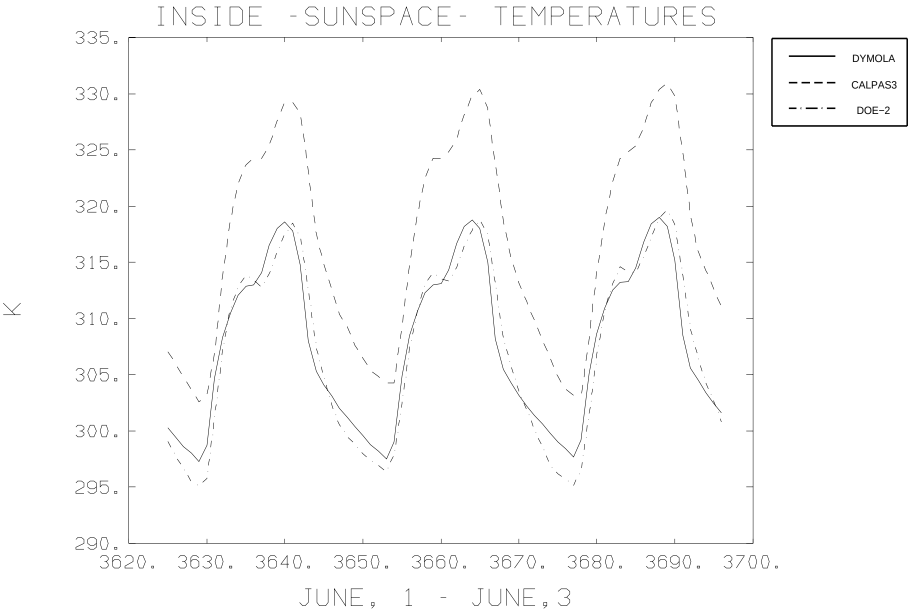


Figure 49: Sunspace temperatures for June 1 – June 3

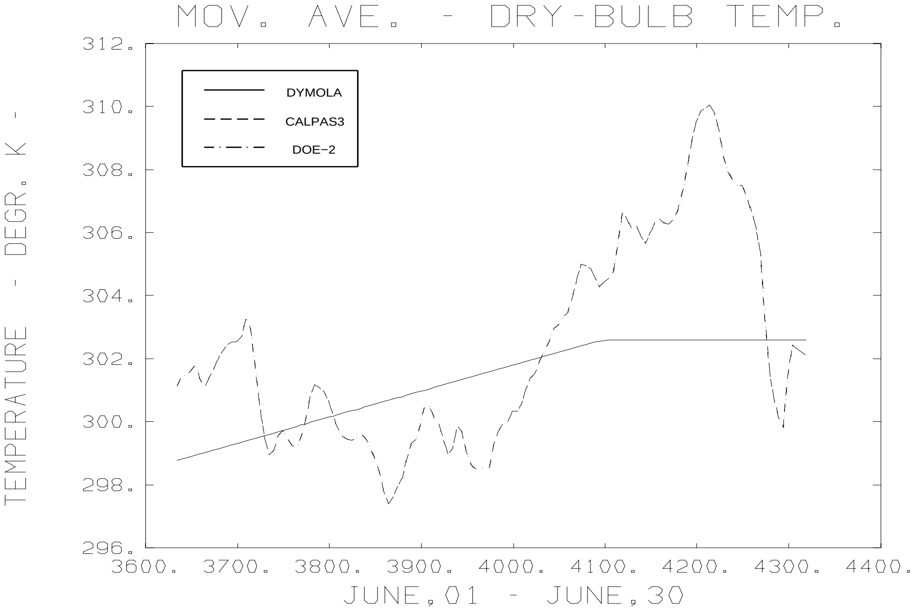


Figure 50: Moving average of the dry-bulb temperatures over every day of June

lag, and it has been supposed that CALPAS3 and especially the new DYMOLA model would handle this behavior much better than DOE-2 does.

Contrary to that is the behavior of the low mass sunspace. It follows the outside temperature without delay and solar irradiation on the glass areas instantly increases the sunspace temperature. It can be seen that the slopes of these graphs are very similar. This basically means that all programs use the same time constant for the heat storage mainly by the inside air volume of the sunspace.

In order to investigate the achieved results in a wider range the hourly data have been averaged using the moving average function of SAPS-II which is available as a CTRL-C library. The result of a moving average of the dry-bulb temperature for June is given in Fig.(50).

Now we see the dramatic difference between the chaotic performance of the measured data of CALPAS3 and DOE-2 and the linear interpolated mean temperatures used in the DYMOLA model.

By looking at the moving average over house and sunspace temperatures we find again the thermal behavior of high and low mass structures. The average house temperatures

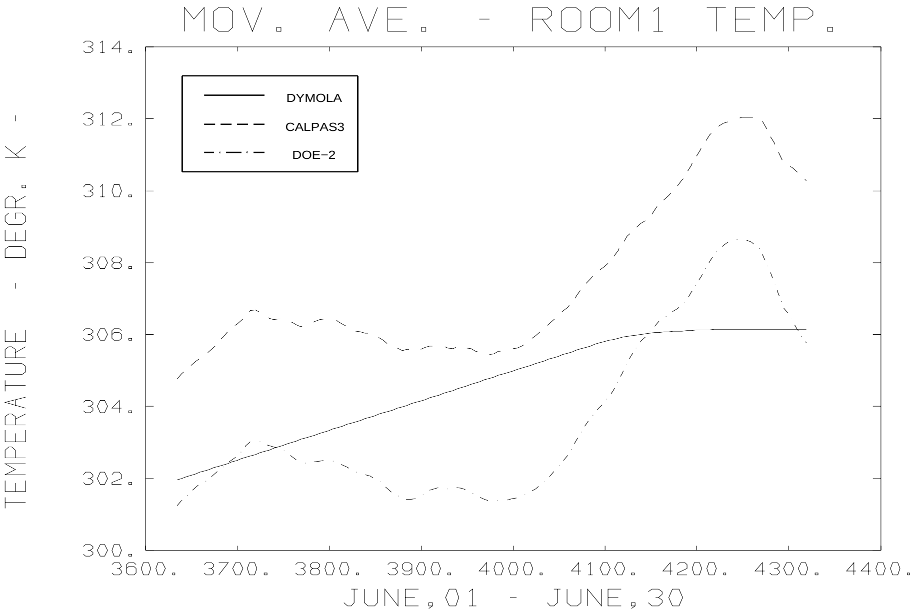


Figure 51: Moving average of the inside conditions in June

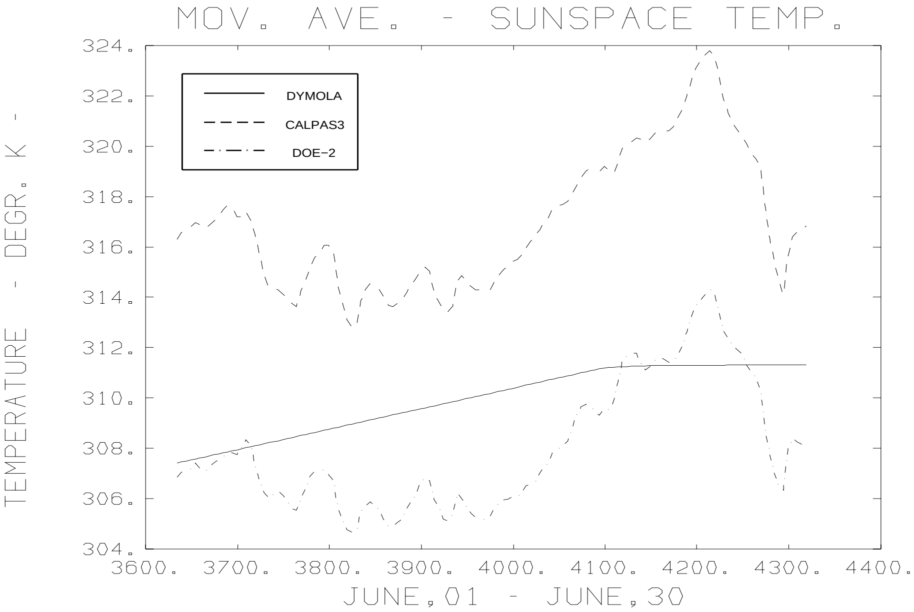


Figure 52: Moving average of the sunspace temperatures in June

in general follow the outside temperature but the mass effect lets the curve look really smooth, and sudden outside temperature drops or increases affect the house inside conditions with delay. We also see that the DYMOLA model does not show us this behavior due to the fact that the mean temperature changes its slope only once.

For this reason another model for the outside dry-bulb temperature has been developed. The purpose is to achieve more changes in the behavior of the mean temperature and more randomness in its values. The use of random numbers, however, is limited by the chosen integration algorithm. We have selected a *Gear* algorithm with variable step size. Unfortunately, the step size control of an algorithm with variable step size will goof up if we apply random numbers.

Therefore, we again come up with a deterministic method.

```
{ bond graph for the modified effort source, dry-bulb temperature }

model type mSE

  main cut A (e / .)
  parameter pi=3.1416
  local mt
  external time

  mt = TMEAN(time) * ( 1. + 0.004 * sin(2.*pi*time/120.) + ->
       0.0025 * cos(2.*pi*(time - 60.)/240.) + ->
       0.0002 * sin(2.*pi*(time - 30.)/360.))
  TOUT(time,mt) = e

end
```

In this model the mean temperature is superimposed by different sine and cosine waves varying in amplitude and frequency.

The results using this approach are shown in Fig.(53), Fig.(54) and Fig.(55). Now, we see that the DYMOLA model shows the same expected thermal behavior as we have already seen for the other two programs. The amplitude decreases and it gets damped by the mass elements. In addition to this the temperature changes inside the house appear with a time lag, while the sunspace conditions do not show a delay. This can be seen in the comparison of DYMOLA moving average dry-bulb, sunspace, and room temperatures in Fig.(56).

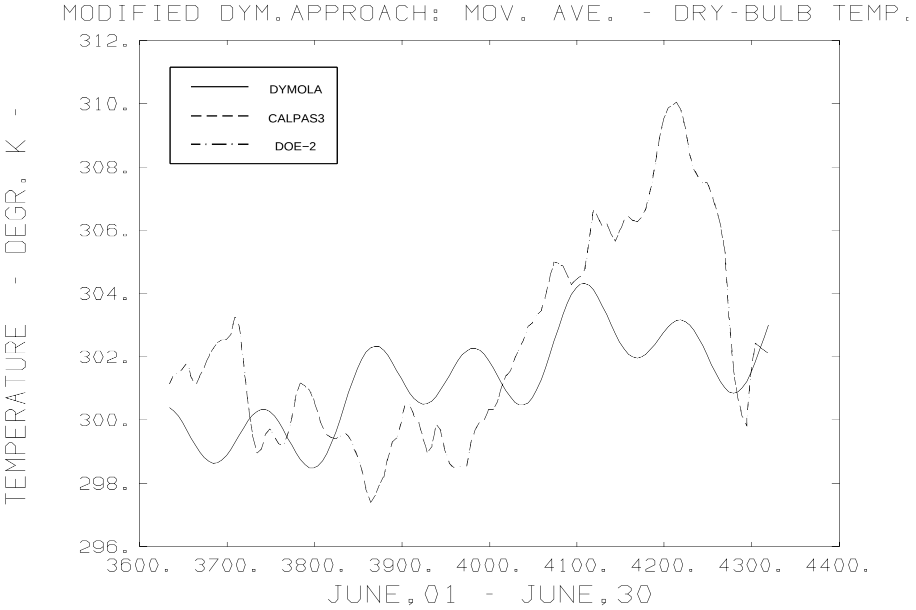


Figure 53: Modified DYMOLA approach: moving average of the dry-bulb temperatures in June

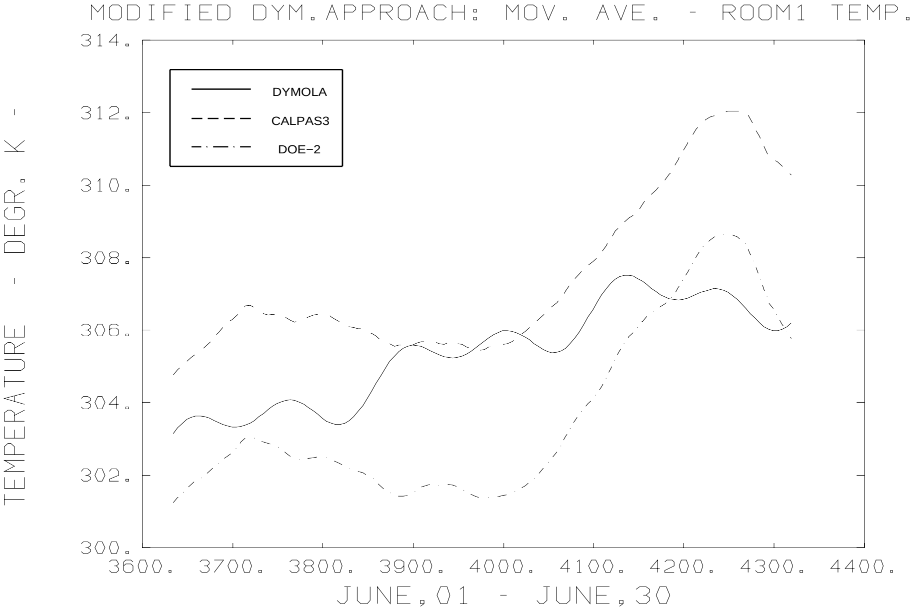


Figure 54: Modified DYMOLA approach: moving average of the inside conditions in June

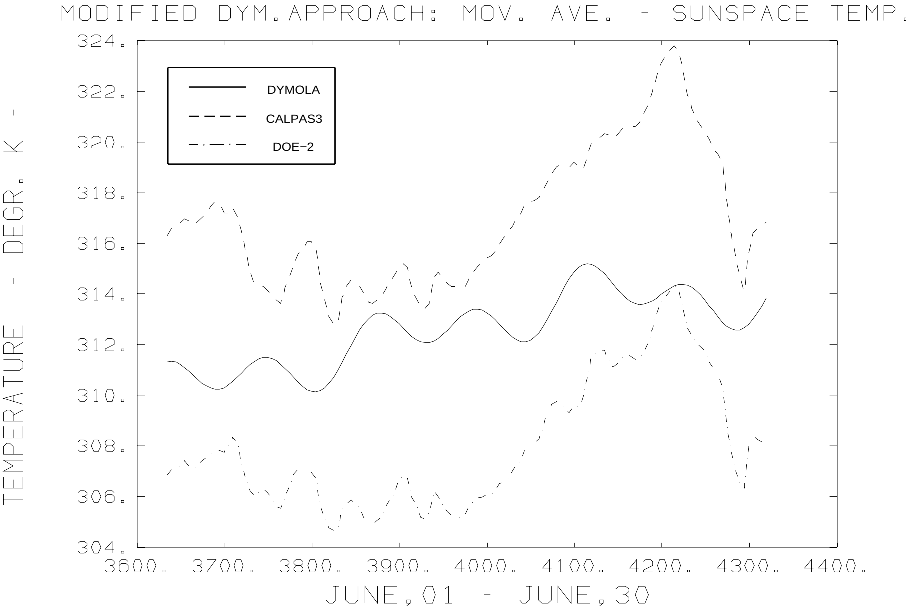


Figure 55: Modified DYMOLA approach: moving average of the sunspace temperatures in June

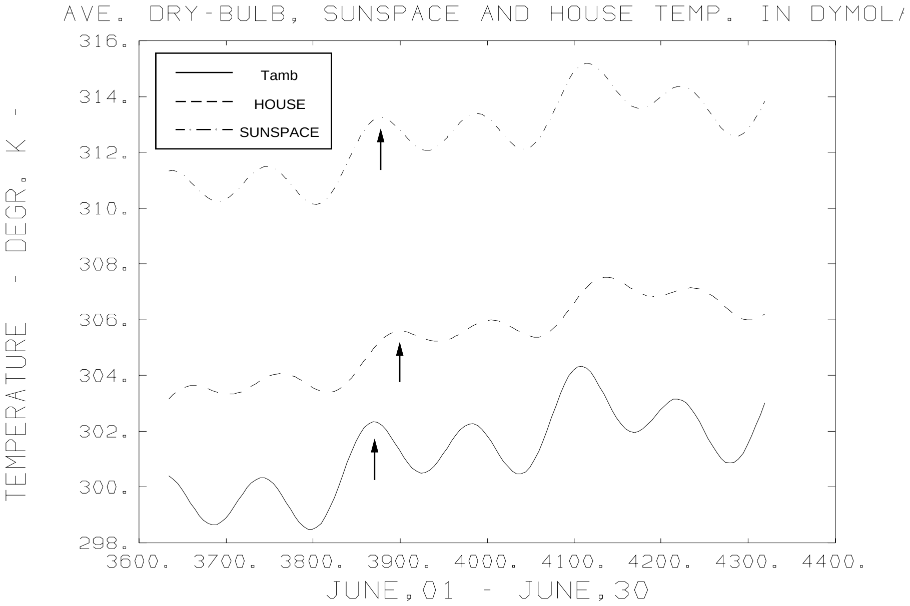


Figure 56: Modified DYMOLA approach: comparison of moving average for dry-bulb, sunspace and house temperatures in DYMOLA

4.5 Load Analysis

The determination of the loads profile of a structure is essential for the devise of distribution systems and plants which have to accomplish the heating and cooling demand of the building. The loads represent the effort required to achieve desired conditions inside the living space. Peak loads need to be determined in order to design the margin of safety of the supplying device.

Therefore, loads have been investigated by all programs in order to establish space conditions between $22°C$ and $27°C$, or $71°F$, $295°K$ and $80°F$, $300°K$, respectively. The data again have been extracted from the output-files and written in matrix form for the CTRL-C input. After the conversion into the same units (*watt* has been selected as a convenient unit), the values have been plotted using the CTRL-C graphic routines.

Before we start comparing data, however, we should look at the way loads analysis can be done in the three approaches.

In CALPAS3 the temperature set-points for heating and cooling have been kept at $71°$ and $80°F$ and the cooling and heating load is given in the report about *hourly house energy balance*. The values for mechanical system energy transfer are printed for the current hour and also indicate whether heating or cooling operation occurs.

DOE-2 has been designed for the accurate analysis of the supplying system and device. Therefore, it calculates loads for the distributing system and the plant according to the building loads for a design temperature. The design heating temperature and design cooling temperature set on $71°$ and $80°F$ obviously leads to system heating and cooling loads. For this reason, loads obtained by DOE-2 were not expected to fit the optimal.

In the DYMOLA model, an entropy source or sink has been modelled attached to the capacity of $room1$. The DYMOLA code for this *flow source* is given below.

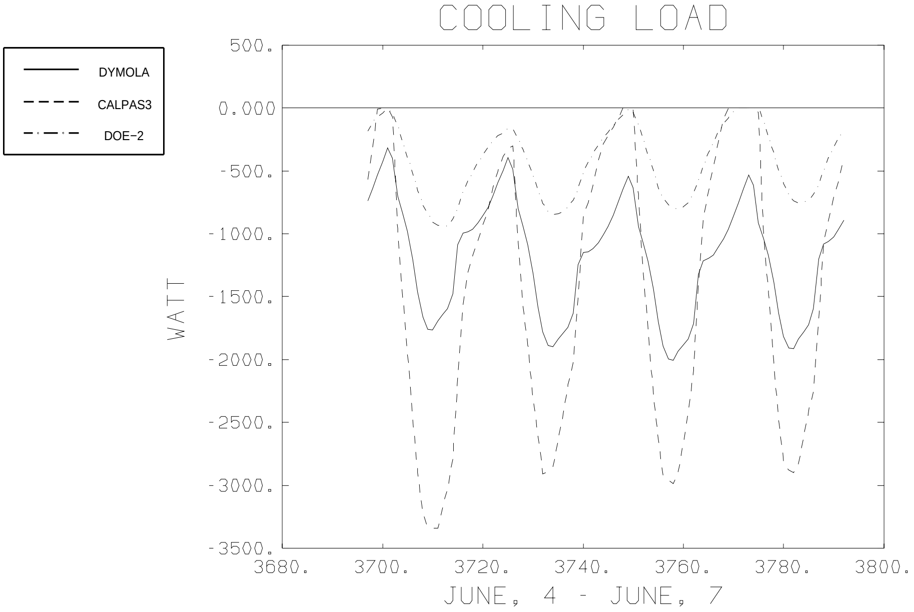


Figure 57: Cooling load in June

```
{ bond graph model of a flow source }

model type SF

  main cut A (. / -f)
  terminal troom
  local theat, tcool, yh, yc

  tcool = troom - 300.
  theat = troom - 295.
  yc = 1.0 - 2.0 * 1.0/(1.0+EXP(-tcool))
  yh = 1.0 - 2.0 * 1.0/(1.0+EXP(-theat))
  (BOUND(0.,10.0,yh) + BOUND(-10.,0.,yc)) * 100000./troom = f

end
```

What we used is a sigmoid function between 1. and −1., which has been bound between 0. and 1. for heating and between −1. and 0. for cooling. The device acting as heater or cooler, respectively, has a power of $100kW$.

The load profiles of the three programs are given in Fig.(57) for the cooling load in June and in Fig.(58) for the heating load in December.

The graph for the June cooling loads has similar shape for all three results. We can find the same behavior of the DYMOLA approach as in the temperature analysis. The slope changes during the late afternoon hours due to the effect of the high mass structure. While

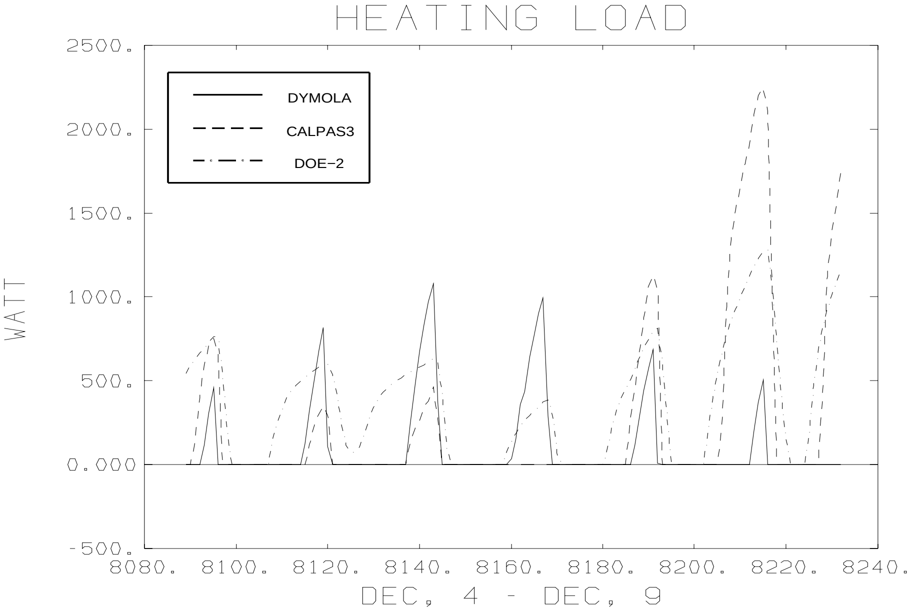


Figure 58: Heating load in December

the temperatures increase as the mass elements release heat, the cooling load increases, too. With the higher temperatures the cooling load of CALPAS3 is also the highest.

As we assumed the DOE-2 values differ especially if we look at the heating load in December. The reason for this is the different control mechanism DOE-2 provides in the system part of the program. We found it therefore interesting to look at the loads profile obtained for a design-temperature of $T_{design} = 73°F$ or $23°C, 296°K$, respectively. The December heating and cooling loads for these design conditions is given in Fig.(59).

4.6 Temperature Profile of Transient Conduction

The way how the DYMOLA model represents the heat diffusion equation enables us to focus better on the phenomena of heat conduction. We remember that the heat equation was expressed by discretizing the space axis and simulating over time using a set of ordinary differential equations.

Therefore, we can determine the temperature of every node in our discretized space axis as a function of time. The result is a time dependent temperature profile of the element where conduction occurs. The high mass, adobe exterior walls are modelled with

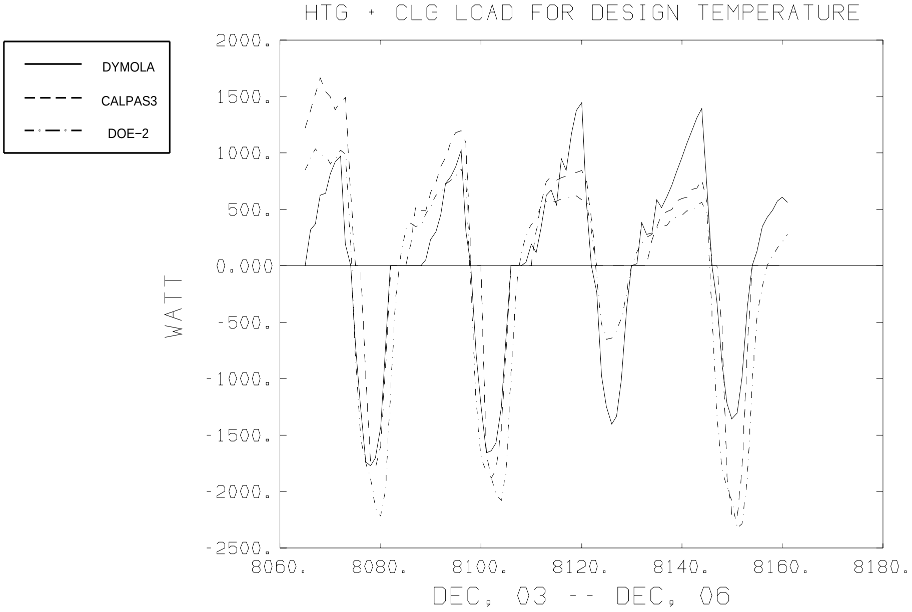


Figure 59: Heating and cooling loads in December for design conditions

three nodes inside the wall, two nodes representing the inside and outside surfaces, and nodes for the dry-bulb and room temperature. Fig.(60) shows the 3-dimensional plot of this temperature profile.

The peaks of the outside surface temperature are determined by the increase of radiation on a east exposed wall during the afternoon. Furthermore, we find the effect of mass resulting in a moderate temperature course of the inside contrary to the outside temperature.

4.7 Computation Time

The computation time plays an important role in the evaluation of the different approaches. The performance of the methodologies behind each program can not only be judged by accuracy in the results and sufficient capabilities but also by the expected user acceptance. For a user working with this program, however, the computation time is a high ranking criterion. The time a program requires a computer's CPU for the simulation costs money and it can not be used for other purposes. Considering that optimization of a building requires many simulation runs in order to determine the right parameter, it is

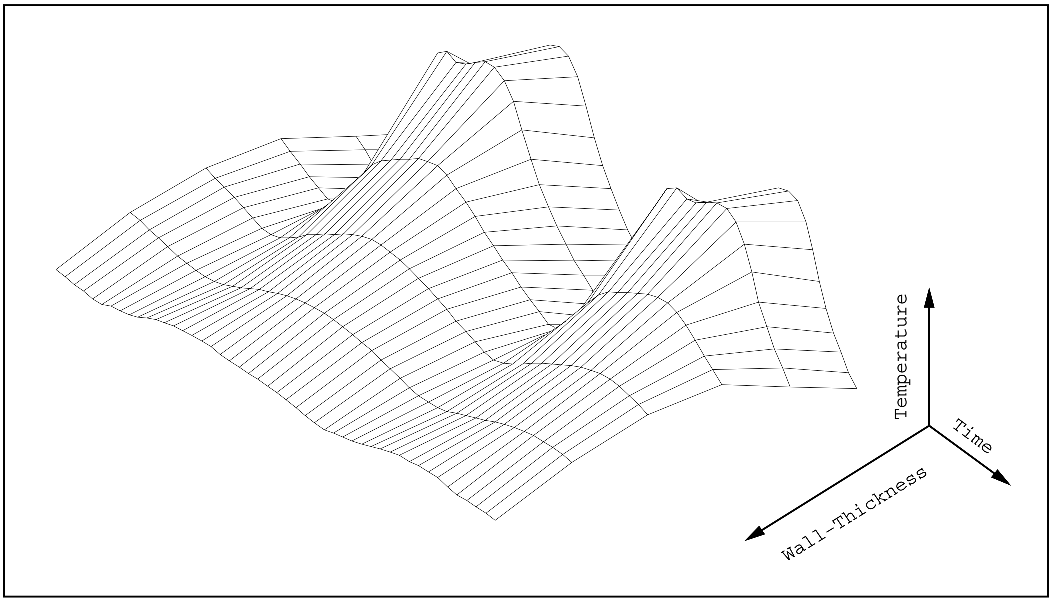


Figure 60: Temperature profile of a east exposed external wall

not affordable to compute excessively long.

The comparison of the three programs is difficult because they ran on three different computers and sometimes have options to speed the calculations up while accepting less accuracy in the results.

CALPAS3 is certainly the fastest of the three programs, and provides the choice to calculate solar geometry and transmission factors either monthly or hourly. We had CALPAS3 running on an IBM-compatible Personal Computer with a Intel-80386 CPU and a Intel-80287 mathematical coprocessor. The time it needed to check the input-file for errors and initialize the values was approximately 15 – 20 seconds. The simulation over a month and the computation of an average output-file took another 3 – 4 minutes. With the reduction to monthly calculated solar data it took 30 seconds less.

DOE-2 was running on an IBM-compatible Personal Computer with a Intel-80386 CPU and it took about 3 minutes to translate the input-file and calculate the weighting factors. The building simulation needed $\approx$ 2 minutes for a whole month and the system simulation another 1 minute. To write the results in hourly or monthly reports DOE-2 worked $\approx$ 1 minute for a default monthly report and two hourly reports.

The derived DYMOLA source code had to be compiled using the DYMOLA version on the VAX 4000. DYMOLA produces a ACSL program which then will be compiled into FORTRAN code and a executable file, called ACSLCC.EXE. The results were then obtained in CTRL-C by using the ACSL – CTRL-C interface which runs the ACSLCC.EXE and writes values in matrix form.

The time it takes DYMOLA to compile and produce the ACSL program was 6 minutes and ≈ 20 seconds CPU-time. For the preparation of the ACSLCC.EXE by compiling the ACSL program and attaching functions of the fortran library *func.lib* approximately 20 seconds CPU-time was needed. The simulation and obtainment of the results in CTRL-C finally could be done in ≈ 5 minutes CPU-time.

5 Conclusion

A Building Analysis Program was developed using bond graphs to describe the thermal behavior of the building. The bond graphs were directly coded into DYMOLA and the model was assembled in a hierarchical structure using modular components.

The performance of this program was studied on a test building and compared to the results of two compatible building analysis programs. Points of consideration were the capabilities of all programs to model the test building, the accuracy of the results, and computation times.

Bond graphs were found to be very capable in modelling heat transfer phenomena, especially heat conduction. They are well suited to express the heat diffusion equation and combine the topological representation as well as the computational structure. In this way, bond graphs can be easily used to model thermal systems very accurately.

DYMOLA is a very powerful tool for the modelling of modular and hierarchical systems. The cut concept and other features support the coding of bond graphs, while the code preserves readability. However, some of the options DYMOLA provides are not yet implemented. In order to improve comprehension during the model development in Chapter 3, some of these features were incorporated. The program which suffices the current DYMOLA version is given in Appendix A.

The obtained results show that the new approach is capable for building analysis. The difference in performance results depend on the different approaches the three programs use. However, note that despite CALPAS3 and DOE-2 use a similar simulation approach and the same weather file, the results of the DYMOLA program is always found in between them.

Computation times are quite various. Firstly, all programs run on different computers and secondly, computational effort varies from one program to another. CALPAS3 and DOE-2 combine the procedural programming structure of their program with discrete-time simulation and end up with moderate computation times within several minutes. DYMOLA is an object-oriented modelling language and at the model expansion the length of the program increases with the number of submodels. The new approach uses a integration algorithm to solve the differential equations and the complexity of the system determines the computation time.

The advantages of the new approach are the high accuracy due to the bond graph technique and the powerful modelling capabilities. This allows the construction of flexible models which can describe a variety of thermal systems. However, in building analysis, the accuracy requirement is not as high as we can accomplish with the new approach. This is basically due to the fact that there are too many uncertainties in the description of a structure and the accuracy relies on the faithful choice of parameters. Some of those parameters, e.g., the infiltration rate through gaps and cracks, influence the thermal performance strongly but can not be derived properly. Those effects therefore undermine the accuracy of the calculations and were neglected in this case study. For applications where high accuracy is required and longer computation times are accepted, the new approach will certainly be an improvement.

References

[1] François E. Cellier (1991), *Continuous System Modeling*, Springer-Verlag, N.Y.

[2] BSG – Berkeley Solar Group (1982), *CALPAS3 – User's Manual*, BSG, Berkeley, CA.

[3] U.S. Department of Energy, Lawrence Berkeley Laboratory, Los Alamos National Laboratory (1979), *DOE-2 – User Guide and Reference Manual*

[4] Acrosoft International, Inc. (1987), *MICRO–DOE-2 Version 2.1C – User's Manual*, Denver, CO.

[5] ASHRAE Handbook (1981), *1981 Fundamentals*, American Society of Heating, Refrigerating and Air-Conditioning Engineers, Atlanta, GA.

[6] T. Kusuda (1969), *Thermal Response Factors for Multi-Layer Structures of Various Heat Conduction Systems*, ASHRAE Transactions 1969, **75**, Part 1, 246.

[7] Henry M. Paynter (1961), *Analysis and Design of Engineering Systems*, M.I.T. Press, Cambridge, MA.

[8] Alan Blundell (1982), *Bond Graphs for Modelling Engineering Systems*, John Wiley & Sons, N.Y.

[9] Dean Karnopp and Ronald C. Rosenberg (1968), *Analysis and Simulation of Multiport Systems – The Bond Graph Approach to Physical System Dynamics*, M.I.T. Press, Cambridge, MA.

[10] Hilding Elmqvist (1978), *A Structured Model Language for Large Continuous Systems*, Ph.D. Dissertation, Report CODEN:LUTFD2/(TRFT-1015), Dept. of Automatic Control, Lund Institute of Technology, Lund, Sweden.

[11] Granino A. Korn (1989), *Interactive Dynamic-System Simulation*, McGraw-Hill, N.Y.

[12] Edward L. Mitchell and Joseph S. Gauthier (1986), *ACSL: Advanced Continuous Simulation Language – User Guide and Reference Manual*, Mitchell & Gauthier Assoc., Concord, MA.

[13] Systems Control Technology, Inc. (1985), *CTRL-C, A Language for the Computer-Aided Design of Multivariable Control Systems, User's Guide*, Palo Alto, CA.

[14] Qingsu Wang (1989), *Management of Continuous System Models in DEVS-SCHEME: Time Windows for Event Based Control*, MS Thesis, Dept. of Electrical and Computer Engineering, University of Arizona, Tucson, AZ.

[15] Sunil C. Idnani (1991), *An ACSL Interface for DYMOLA*, MS Thesis, Dept. of Electrical and Computer Engineering, University of Arizona, Tucson, AZ.

[16] Gabriel Kron (1944), *Numerical Solution of Ordinary and Partial Differential Equations by Means of Equivalent Circuits*, J. App. Phys. **16**, 172.

[17] Frank P. Incropera and David P. De Witt (1985), *Introduction to Heat Transfer*, John Wiley & Sons, N.Y.

[18] John A. Duffie and William A. Beckman (1980), *Solar Engineering of Thermal Processes*, John Wiley & Sons, N.Y.

[19] H.Alan Fine et al. (1977), *The Thermal Properties of Adobe*, ASHRAE Transactions 1977, **83**, Part 2, 80.

[20] Benjamin T. Rogers (1978), *Effects of Moisture Content on the Thermal Properties of Sun Dried Adobe*, Proc. ISES 1978, **2.2**, 41.

A The DYMOLA Program

A.1 The house.dym file

```
{ bond graph bond }

model type bond

  cut A (x / y), B (y / -x)
  main cut C [A B]
  main path P <A - B>

end

{ bond graph effort source, dry-bulb temperature }

model type SE

  main cut A (e / .)
  terminal time
  local mt

  mt = TMEAN(time)
  TOUT(time,mt) = e

end

{ bond graph for the modified effort source, dry-bulb temperature }

model type mSE

  main cut A (e / .)
  terminal time
  parameter pi=3.1416
  local mt

  mt = TMEAN(time) * ( 1. + 0.004 * sin(2.*pi*time/120.) + ->
       0.0025 * cos(2.*pi*(time - 60.)/240.) + ->
       0.0002 * sin(2.*pi*(time - 30.)/360.))
  TOUT(time,mt) = e

end

{ bond graph model of a flow source }

model type SF

  main cut A (. / -f)
  terminal troom
  local theat, tcool, yh, yc

  tcool = troom - 300.
  theat = troom - 295.
  yc = 1.0 - 2.0 * 1.0/(1.0+EXP(-tcool))
  yh = 1.0 - 2.0 * 1.0/(1.0+EXP(-theat))
  (BOUND(0.,10.0,yh) + BOUND(-10.,0.,yc)) * 100000./troom = f

end

{ bond graph modulated conductive source }
```

```
model type mGS

  cut A (e1 / f1), B (e2 / -f2)
  main cut C[A B]
  main path P<A - B>
  terminal k, area, l
  local G, G1

  G1 = (k/l)*area
  G = G1/e2
  G*e1 = f1
  f1*e1 = f2*e2

end

{ bond graph modulated capacity }

model type mC

  main cut A (e / f)
  terminal gamma, m
  local C

  C = (gamma*m)/(3600.*e)
  C*der(e) = f

end

{ bond graph for one dimensional conduction cell }

model type C1D

  submodel (mGS) Gcell
  submodel (mC) Ccell (ic e=288.0)
  submodel (bond) B1, B2, B3
  node n1, n2

  cut Cx(ex/fx), Ci(ei/ -fi)
  main path P<Cx - Ci>

  connect B1 from Cx to n1
  connect B2 from n1 to n2
  connect Gcell from n2 to Ci
  connect B3 from n1 to Ci
  connect Ccell at Ci

end

{ bond graph for one dimensional convection cell }

model type C1V

  submodel(mGS) Gcell
  submodel(bond) B1, B2, B3
  node n1, n2

  cut Cx(ex/fx), Ci(ei/ -fi)
  main path P<Cx - Ci>

  connect B1 from Cx to n1
  connect B2 from n1 to n2
  connect Gcell from n2 to Ci
  connect B3 from n1 to Ci
```

```
end

{ bond graph model of the flow source of a sun-exposed surface }

model type SUN

   main cut C (. / -f)
   terminal twall, area, tilt, abs, sfa, OH, W, time
   parameter ro=0.33, cr=1.0, cf=1.0
   constant sc=1353., lsm=105., lon=111., lat=32., ->
            pi=3.14156, one=1.
   local int, h, dec, inc, alt, q, ->
         sfar, latr, tiltr, inc1, inc2, inc3, ->
         sou, idn, dif, dir


 sfar=2.*pi*sfa/360.
 latr=2.*pi*lat/360.
 tiltr=2.*pi*tilt/360.
 h=(2.*pi/24.)*(time-12.+ET(time)/60.+(4./60.)*(lsm-lon))
 dec=(2.*pi/360.)*(23.45*sin(2.*pi*(284.+time/24.)/365.))

 inc      = inc1   +   inc2  +  inc3
 inc1     = sin(dec)*sin(latr)*cos(tiltr) -  ->
            sin(dec)*cos(latr)*sin(tiltr)*cos(sfar)
 inc2     = cos(dec)*cos(latr)*cos(tiltr)*cos(h) +  ->
            cos(dec)*sin(latr)*sin(tiltr)*cos(sfar)*cos(h)
 inc3     = cos(dec)*sin(tiltr)*sin(sfar)*sin(h)
 alt = cos(latr)*cos(dec)*cos(h) + sin(latr)*sin(dec)

 sou = sc*(1.+0.033*cos(2.*pi*(time/24.)/365.))
 idn = sou/EXP(B(time)/BOUND(0.05,10.0,alt))

 dir = idn*inc*(area-OH*W*(alt/sqrt(1.-alt**2.)))
 dif = idn*area*(C(time)*(1.+cos(tiltr))/2.+ ->
       (C(time)+alt)*ro*(1.-cos(tiltr))/2.)

 int = abs*(cr*BOUND(0.,1.E10,dir)+ ->
            cf*BOUND(0.,1.E10,dif))/twall
 q = int*SIGN(one,alt)
 BOUND(0.,1.E10,q) = f

end

{ bond graph model of the flow source of a transmitting surface }

model type SHGF

   main cut C (. / -f)
   terminal tref, area, OH, W, sfa, sc, time
   parameter tilt=90., ro=0.33, cr=1.0, cf=1.0, ->
             hi=8.3, ho=22.7
   constant pi=3.14156, lsm=105., lon=111., lat=32., ->
            one=1.0
   local int, h, dec, inc, alt, p, q, rtr, rab, ni, shgf, ->
         sfar, latr, tiltr, inc1, inc2, inc3, idn, dif, dir


 sfar=2.*pi*sfa/360.
 latr=2.*pi*lat/360.
 tiltr=2.*pi*tilt/360.
 h=(2.*pi/24.)*(time-12.+ET(time)/60.+(4./60.)*(lsm-lon))
 dec=(2.*pi/360.)*(23.45*sin(2.*pi*(284.+time/24.)/365.))
```

```
 inc      = inc1 + inc2 + inc3
 inc1     = sin(dec)*sin(latr)*cos(tiltr) -  ->
            sin(dec)*cos(latr)*sin(tiltr)*cos(sfar)
 inc2     = cos(dec)*cos(latr)*cos(tiltr)*cos(h) +  ->
            cos(dec)*sin(latr)*sin(tiltr)*cos(sfar)*cos(h)
 inc3     = cos(dec)*sin(tiltr)*sin(sfar)*sin(h)
 alt = cos(latr)*cos(dec)*cos(h) + sin(latr)*sin(dec)

 idn = A(time)/EXP(B(time)/BOUND(0.05,10.0,alt))

 dir = idn*inc*(area-OH*W*(alt/sqrt(1.-alt**2.)))
 dif = idn*area*(C(time)*(1.+cos(tiltr))/2. + ->
                 (C(time)+alt)*ro*(1.-cos(tiltr))/2.)

 rtr = cr*BOUND(0.,1.E10,dir)*SIGT1(inc) + cf*dif*2.*SIGT2(inc)
 rab = cr*BOUND(0.,1.E10,dir)*SIGA1(inc) + cf*dif*2.*SIGA2(inc)
 ni = hi/(hi + ho)
 shgf = rtr + ni*rab

 int = sc*shgf/tref
 BOUND(0.,1.E10,int) = p
 q = p*SIGN(one,alt)
 BOUND(0.,1.E10,q) = f

end

{ bond graph model of an external wall }

model type EXWALL

  submodel (SUN) rad
  submodel (SE) tamb
  submodel (C1D) d1,d2,d3
  submodel (C1V) v1,v2
  submodel (mC)  c1 (ic e=288.0)

  main cut A (e / -f)

  parameter hout=22.7, hinv=8.3, k1=1.7, k2=1.7, k3=1.7, ->
            l=1.0, l1=0.14, l2=0.14, l3=0.14, ->
            gamma1=840., gamma2=840., gamma3=840., ->
            dens=1762., thk=0.41, area=1.0, sfa=0.0, tilt=90. ->
            OH=0.0, W=0.0, abs=0.5
  local m

  connect (P) d1-d2-d3
  connect v1 from tamb to d1:Cx
  connect d1:Cx at c1
  connect rad at d1:Cx
  connect v2 from d3:Ci to A

  m = area*thk/4.*dens
  tamb.time = rad.time
  rad.area = area
  rad.tilt = tilt
  rad.sfa = sfa
  rad.OH = OH
  rad.W = W
  rad.abs = abs
  rad.twall = v1.ei
  v1::Gcell.k = hout
  v2::Gcell.k = hinv
  d1::Gcell.k = k1
  d2::Gcell.k = k2
```

```
  d3::Gcell.k = k3
  v1::Gcell.area = area
  v2::Gcell.area = area
  d1::Gcell.area = area
  d2::Gcell.area = area
  d3::Gcell.area = area
  v1::Gcell.l = l
  v2::Gcell.l = l
  d1::Gcell.l = l1
  d2::Gcell.l = l2
  d3::Gcell.l = l3
  c1.gamma = gamma1
  c1.m = m
  d1::Ccell.gamma = gamma2
  d1::Ccell.m = m
  d2::Ccell.gamma = gamma2
  d2::Ccell.m = m
  d3::Ccell.gamma = gamma3
  d3::Ccell.m = m

end

{ bond graph model of the roof }

model type ROOF

  submodel (SUN) rad
  submodel (SE) tamb
  submodel (C1D) d1,d2,d3
  submodel (C1V) v1, v2
  submodel (mC)  c1 (ic e=288.0)

  main cut A (e / -f)

  parameter hout=22.7, hinh=7.4, k1=45.3, k2=0.0385, k3=0.1143, ->
            l=1.0, l1=0.038, l2=0.2, l3=0.016, ->
            gamma1=500., dens1=7850., gamma2=657., dens2=19.2, ->
            gamma3=1214., dens3=544., area=100.7, sfa=180., tilt=7., ->
            OH=0.0, W=0.0, abs=0.2
  local     m1, m2, m3


  connect (P) d1-d2-d3
  connect v1 from tamb to d1:Cx
  connect d1:Cx at c1
  connect rad at d1:Cx
  connect v2 from d3:Ci to A

  m1 = area*l1*dens1
  m2 = 0.5*area*l2*dens2
  m3 = area*l3*dens3
  tamb.time = rad.time
  rad.area = area
  rad.tilt = tilt
  rad.sfa = sfa
  rad.OH = OH
  rad.W = W
  rad.abs = abs
  rad.twall = v1.ei
  v1::Gcell.k = hout
  v2::Gcell.k = hinh
  d1::Gcell.k = k1
  d2::Gcell.k = k2
  d3::Gcell.k = k3
```

```
  v1::Gcell.area = area
  v2::Gcell.area = area
  d1::Gcell.area = area
  d2::Gcell.area = area
  d3::Gcell.area = area
  v1::Gcell.l = l
  v2::Gcell.l = l
  d1::Gcell.l = l1
  d2::Gcell.l = l2
  d3::Gcell.l = l3
  c1.gamma = gamma1
  c1.m = m1
  d1::Ccell.gamma = gamma2
  d1::Ccell.m = m2
  d2::Ccell.gamma = gamma2
  d2::Ccell.m = m2
  d3::Ccell.gamma = gamma3
  d3::Ccell.m = m3

end

{ bond graph model of an internal wall }

model type INTWALL

  submodel (C1D) d1,d2,d3
  submodel (C1V) v1,v2
  submodel (mC)  c1 (ic e=288.0)

  cut A (ea / fa), B (eb / -fb)
  main cut C[A B]
  main path F<A - B>

  parameter hinv=8.3, k1=1.7, k2=1.7, k3=1.7, ->
            l=1.0, l1=0.1, l2=0.1, l3=0.1, ->
            gamma1=840., gamma2=840., gamma3=840., ->
            dens=1762., thk=0.3, area=1.0
  local m

  connect (P) d1-d2-d3
  connect A at v1:Cx
  connect v1:Ci at d1:Cx
  connect d1:Cx at c1
  connect d3:Ci at v2:Cx
  connect v2:Ci at B

  m = area*thk/4.*dens
  v1::Gcell.k = hinv
  v2::Gcell.k = hinv
  d1::Gcell.k = k1
  d2::Gcell.k = k2
  d3::Gcell.k = k3
  v1::Gcell.area = area
  v2::Gcell.area = area
  d1::Gcell.area = area
  d2::Gcell.area = area
  d3::Gcell.area = area
  v1::Gcell.l = l
  v2::Gcell.l = l
  d1::Gcell.l = l1
  d2::Gcell.l = l2
  d3::Gcell.l = l3
  c1.gamma = gamma1
  c1.m = m
```

```
  d1::Ccell.gamma = gamma2
  d1::Ccell.m = m
  d2::Ccell.gamma = gamma2
  d2::Ccell.m = m
  d3::Ccell.gamma = gamma3
  d3::Ccell.m = m

end

{ bond graph model of a low mass element }

model type FRAME

  submodel (SUN) rad
  submodel (mGS) Gcell
  submodel (bond) B1, B2, B3
  node n1, n2, n3

  cut A(ea / fa), B (eb / -fb)
  main cut C[A B]
  main path F<A - B>

  terminal tamb
  parameter area=1.0, tilt=90., sfa=180., abs=0.2, OH=0.0, W=0.0, ->
            l=1.0, u=0.189

  connect A at n1
  connect B1 from n1 to n2
  connect B2 from n2 to B
  connect B3 from n2 to n3
  connect Gcell from n3 to B
  connect rad at n1

  rad.area = area
  rad.tilt = tilt
  rad.sfa = sfa
  rad.OH = OH
  rad.W = W
  rad.abs = abs
  rad.twall = tamb
  Gcell.k = u
  Gcell.l = l
  Gcell.area = area

end

{ bond gragh model of the heat transfer through a glazing }
{ material due to radiation and temperature difference    }

model type WINDOW

  submodel (SHGF) beam
  submodel (SE) tamb
  submodel (mGS) Gcell
  submodel (bond) B1, B2, B3
  node n1, n2, n3

  main cut A (e / -f)

  parameter area=1.0, sfa=0.0, sc=0.85, OH=0.0, W=0.0, u=3.4, l=1.0

  connect tamb at n1
  connect B1 from n1 to n2
  connect B2 from n2 to A
```

```
  connect B3 from n2 to n3
  connect Gcell from n3 to A
  connect beam at A

  beam.tref = Gcell.e2
  tamb.time = beam.time
  beam.area = area
  beam.sfa = sfa
  beam.sc = sc
  beam.OH = OH
  beam.W = W
  Gcell.k = u
  Gcell.l = l
  Gcell.area = area

end

{ bond graph model for the loss through slab }

model type SLAB

  submodel (C1D) d1
  submodel (C1V) v1

  cut A (ea / fa), B (eb / -fb)
  main cut C[A B]
  main path F<A - B>

  parameter hin=7.4, u=5.11, l=1.0, area=1.0, gamma=835. ->
            dens=2083., thk=0.2, perim=1.0
  local  m

  connect A to d1 to v1 to B

  m = area*thk*dens
  d1::Gcell.k = hin
  d1::Gcell.area = area
  d1::Gcell.l = l
  d1::Ccell.m = m
  d1::Ccell.gamma = gamma
  v1::Gcell.k = u
  v1::Gcell.area = perim*thk
  v1::Gcell.l = l

end

{ bond graph model of room 1 covering the south side of the adobe building }

model type SPACE1

 submodel (ROOF) r1 (area=72.5)
 submodel (EXWALL) southwall (area=15.7, sfa=0.0, OH=0.3, W=12.), ->
                   northwall (area=6.60, sfa=180., OH=0.3, W=12.), ->
                   eastwall  (area=11.3, sfa=270., OH=0.3, W=8.), ->
                   westwall  (area=20.7, sfa=90., OH=0.3, W=8.)
 submodel (WINDOW) southwin1 (area=2.23, sfa=0.0), ->
                   southwin2 (area=2.23, sfa=0.0), ->
                    northwin1 (area=1.5, sfa=180.)
 submodel (INTWALL) intw2 (area=6.6) ->
                    intw3 (area=15.2) ->
                    sswall (area=12.9, thk=0.41, l1=0.14, l2=0.14, l3=0.14)
 submodel (FRAME) outdoor (area=2.17, abs=0.6, u=3.57), ->
                  indoor1 (area=2.17, abs=0.0, u=3.57), ->
                  indoor2 (area=2.17, abs=0.0, u=3.57), ->
```

```
                 ssdoor (area=2.17, abs=0.0, u=3.57), ->
                 sswin (area=1.1, abs=0.0, OH=3.1, W=5.4, u=3.4)
 submodel (SLAB) sl1 (area=72.5, perim=23.7)
 submodel (mSE) tamb
 submodel (mC)  croom (ic e=288.0)
 submodel (SF) aircond

 cut con12 (e2 / -f2), con13 (e3 / -f3), con1ss (es / -fs)

 parameter gammaroom=1000., mroom=265.
 node n1

 terminal t

 connect southwall at n1
 connect northwall at n1
 connect eastwall at n1
 connect westwall at n1
 connect r1 at n1
 connect southwin1 at n1
 connect southwin2 at n1
 connect northwin1 at n1
 connect n1 to intw2 to con12
 connect n1 to intw3 to con13
 connect n1 to sswall to con1ss
 connect n1 to ssdoor to con1ss
 connect n1 to sswin to con1ss
 connect tamb to outdoor to n1
 connect n1 to indoor1 to con12
 connect n1 to indoor2 to con13
 connect n1 to sl1 to tamb
 connect n1 at croom
 connect n1 at aircond

 croom.gamma = gammaroom
 croom.m = mroom
 aircond.troom = croom.e
 r1::rad.time = t
 southwall::rad.time = t
 northwall::rad.time = t
 eastwall::rad.time = t
 westwall::rad.time = t
 southwin1::beam.time = t
 southwin2::beam.time = t
 northwin1::beam.time = t
 sswin::rad.time = t
 outdoor::rad.time = t
 indoor1::rad.time = t
 indoor2::rad.time = t
 ssdoor::rad.time = t
 sswin.tamb = tamb.e
 outdoor.tamb = tamb.e
 indoor1.tamb = tamb.e
 indoor2.tamb = tamb.e
 ssdoor.tamb = tamb.e
 tamb.time = t

end

{ bond graph model of room 2 at the north-west corner of the adobe house }

model type SPACE2

 submodel (ROOF) r2 (area=15.0)
```

```
 submodel (EXWALL) northwall (area=9.0 , sfa=180., OH=0.3, W=12.), ->
                  eastwall  (area= 9.4, sfa=270., OH=0.3, W=8.)
 submodel (WINDOW) northwin (area=1.5, sfa=180.)
 submodel (INTWALL) int (area=6.6)
 submodel (SLAB) sl2 (area=15.0, perim=7.9)
 submodel (mSE) tamb
 submodel (mC)  croom (ic e=288.0)

 cut con12 (e1 / f1), con23 (e3 / -f3)

 parameter gammaroom=1000., mroom=54.
 node n1

 terminal t

 connect northwall at n1
 connect eastwall at n1
 connect r2 at n1
 connect northwin at n1
 connect n1 to int to con23
 connect n1 to sl2 to tamb
 connect con12 at n1
 connect n1 at croom

 croom.gamma = gammaroom
 croom.m = mroom
 r2::rad.time = t
 northwall::rad.time = t
 eastwall::rad.time = t
 northwin::beam.time = t
 tamb.time = t

end

{ bond graph model of room 3 at the north side between room 2 and }
{ room 1 of the adobe house }

model type SPACE3

 submodel (ROOF) r3 (area=15.0)
 submodel (EXWALL) northwall (area=9.0 , sfa=180., OH=0.3, W=12.)
 submodel (WINDOW) northwin (area=1.5, sfa=180.)
 submodel (SLAB) sl3 (area=15.0, perim=4.2)
 submodel (mSE) tamb
 submodel (mC) croom (ic e=288.0)

 cut con13 (e1 / f1), con23 (e2 / f2)

 parameter gammaroom=1000., mroom=54.
 node n1

 terminal t

 connect northwall at n1
 connect r3 at n1
 connect northwin at n1
 connect n1 to sl3 to tamb
 connect con13 at n1
 connect con23 at n1
 connect n1 at croom

 croom.gamma = gammaroom
 croom.m = mroom
 r3::rad.time = t
```

```
 northwall::rad.time = t
 northwin::beam.time = t
 tamb.time = t

end


{ bond graph model of the solar/screen porch attached on the south side }
{ of the adobe house }

model type SUNSPACE

 submodel (WINDOW) southwin (area=10.30, sfa=0., u=4.9), ->
                   eastwin (area=6.20, sfa=270., u=4.9), ->
                   westwin (area=6.20, sfa=90., u=4.9)
 submodel (FRAME) southwall (area=1.0, sfa=0., abs=0.2, u=5.11), ->
                  eastwall (area=3.0, sfa=270., abs=0.2, u=5.11), ->
                  westwall (area=3.0, sfa=90., abs=0.2, u=5.11), ->
                  ssroof (area=17.2, sfa=0., tilt=15., abs=0.3, u=0.23)
 submodel (SLAB) sl (area=17.2,perim=11.7,gamma=880.,dens=2300.,thk=0.1)
 submodel (mSE) tamb
 submodel (mC)  croom (ic e=288.0)

 cut con1ss (e1 / f1)

 parameter gammaroom=1000., mroom=48.6
 node n1

 terminal t

 connect tamb to southwall to n1
 connect tamb to eastwall to n1
 connect tamb to westwall to n1
 connect tamb to ssroof to n1
 connect southwin at n1
 connect eastwin at n1
 connect westwin at n1
 connect n1 at con1ss
 connect n1 to sl to tamb
 connect n1 at croom

 croom.gamma = gammaroom
 croom.m = mroom
 southwall::rad.time = t
 eastwall::rad.time = t
 westwall::rad.time = t
 southwin::beam.time = t
 eastwin::beam.time = t
 westwin::beam.time = t
 ssroof::rad.time = t
 southwall.tamb = tamb.e
 eastwall.tamb = tamb.e
 westwall.tamb = tamb.e
 ssroof.tamb = tamb.e
 tamb.time = t

end

{ bond graph model of the adobe house }

model HOUSE

  submodel (SPACE1) room1
  submodel (SPACE2) room2
```

```
  submodel (SPACE3) room3
  submodel (SUNSPACE) room4

  input time, tbase
  output t1, t2, t3, t4, to, tw1, tw2, tw3, tw4, ->
         h1, h2, h3, h4, ac

  local t

  connect room1:con12 to room2:con12
  connect room1:con13 to room3:con13
  connect room1:con1ss to room4:con1ss
  connect room2:con23 to room3:con23

  t = time + tbase
  room1.t = t
  room2.t = t
  room3.t = t
  room4.t = t
  room1::croom.e = t1
  room2::croom.e = t2
  room3::croom.e = t3
  room4::croom.e = t4
  room1::tamb.e = to
  room1::eastwall::c1.e = tw1
  room1::eastwall::d1::Ccell.e = tw2
  room1::eastwall::d2::Ccell.e = tw3
  room1::eastwall::d3::Ccell.e = tw4
  room1::croom.f * room1::croom.e = h1
  room2::croom.f * room2::croom.e = h2
  room3::croom.f * room3::croom.e = h3
  room4::croom.f * room4::croom.e = h4
  room1::aircond.f * room1::croom.e = ac

end
```

A.2 The house.act file

```
{ experiment file for model house }

cmodel

maxtime tmax = 24.0
cinterval cint = 1.0
termt (t.ge.tmax)

input 2, time(depend, T), tbase(independ,0.)

initial
ialg = 2
nsteps nstp = 100

table ET, 1, 14  /   0.0, 480. , 1224., 1896., 2640.   ...
                 , 3360., 4104., 4824., 5568., 6312.   ...
                 , 7032., 7776., 8496., 8760.          ...
                 , - 6.4, -11.2, -13.9, - 7.5,   1.1   ...
                 ,   3.3, - 1.4, - 6.2, - 2.4,   7.5   ...
                 ,  15.4,  13.8,   1.6, - 6.4  /

table A, 1, 14  /   0.0, 480. , 1224., 1896., 2640.   ...
                , 3360., 4104., 4824., 5568., 6312.   ...
```

```
                , 7032., 7776., 8496., 8760.          ...
                , 1232., 1230., 1214., 1185., 1135.   ...
                , 1103., 1088., 1085., 1107., 1151.   ...
                , 1192., 1220., 1233., 1232.  /

table B, 1, 14  /   0.0, 480. , 1224., 1896., 2640.   ...
                , 3360., 4104., 4824., 5568., 6312.   ...
                , 7032., 7776., 8496., 8760.          ...
                , 0.140, 0.142, 0.144, 0.156, 0.180   ...
                , 0.196, 0.205, 0.207, 0.201, 0.177   ...
                , 0.160, 0.149, 0.142, 0.140  /

table C, 1, 14  /   0.0, 480. , 1224., 1896., 2640.   ...
                , 3360., 4104., 4824., 5568., 6312.   ...
                , 7032., 7776., 8496., 8760.          ...
                , 0.057, 0.058, 0.060, 0.071, 0.097   ...
                , 0.121, 0.134, 0.136, 0.122, 0.092   ...
                , 0.073, 0.063, 0.057, 0.057  /

table TMEAN, 1, 14 /    0.0, 480., 1224., 1896., 2640.   ...
                    , 3360., 4104., 4824., 5568., 6312.  ...
                    , 7032., 7776., 8496., 8760.         ...
                    , 284.00, 283.71, 284.26, 288.71     ...
                    , 292.59, 296.48, 302.59, 302.59     ...
                    , 302.04, 299.81, 294.26, 287.59     ...
                    , 284.26, 284.0 /
end


end
```

A.3 The fortran library

```
       FUNCTION SIGT1(CINC)
C
C THIS FUNCTION RETURNS THE TRANSMISSION OF DIRECT SOLAR RADIATION
C INCIDENT AT AN ANGLE INC FOR DSA GLASS
C
      INTEGER I, L
      REAL    T(6)
      DATA T/-0.00885, 2.71235, -0.62062, -7.07329, 9.75995,
     #-3.89922/
      SIGT1=0.
C
      DO 10 I=1,6
           L = I-1
          SIGT1 = SIGT1 + T(I)*(CINC**L)
   10 CONTINUE
      RETURN
      END
C
      FUNCTION SIGT2
C
C THIS FUNCTION RETURNS THE TRANSMISSION OF DIFFUSE SOLAR RADIATION
C FOR DSA GLASS
C
      INTEGER I, L
      REAL    T(6)
      DATA T/-0.00885, 2.71235, -0.62062, -7.07329, 9.75995,
     #-3.89922/
      SIGT2=0.
C
```

```
      DO 10 I=1,6
      L = I-1
      SIGT2 = SIGT2 + T(I)/(L + 2)
   10 CONTINUE
      RETURN
      END
C
      FUNCTION SIGA1(CINC)
C
C THIS FUNCTION RETURNS THE ABSORPTION OF DIRECT SOLAR RADIATION
C INCIDENT AT AN ANGLE INC FOR DSA GLASS
C
      INTEGER I, L
      REAL    A(6)
      DATA A/0.01154, 0.77674, -3.94657, 8.57881,-8.38135,
     #3.01188/
      SIGA1=0.
C
      DO 10 I=1,6
      L = I-1
      SIGA1 = SIGA1 + A(I)*(CINC**L)
   10 CONTINUE
      RETURN
      END
C
      FUNCTION SIGA2
C
C THIS FUNCTION RETURNS THE ABSORPTION OF DIFFUSE SOLAR RADIATION
C FOR DSA GLASS
C
      INTEGER I, L
      REAL    A(6)
      DATA A/0.01154, 0.77674, -3.94657, 8.57881,-8.38135,
     #3.01188/
      SIGA2=0.
C
      DO 10 I=1,6
      L = I-1
      SIGA2 = SIGA2 + A(I)/(L + 2)
   10 CONTINUE
      RETURN
      END
C
C
      FUNCTION TOUT(T, AVT)
C
C THIS FUNCTION RETURNS THE HOURLY OUTSIDE TEMPERATURES FOR
C GIVEN TIME (T) AND DAILY MEANTEMPERATURE (AVT)
C
      REAL   H, V, W
      REAL   P(25)
      DATA P/.988, .986, .983, .98, .978,
     #.976, .976, .979, .985, .993, 1.003,
     #1.011, 1.018, 1.023, 1.027, 1.028, 1.027,
     #1.023, 1.017, 1.01, 1.004, .999, .995, .991,
     #.988/
C
      H = (QMOD (T, 24.))*24. + 1.
      IF (H.EQ.INT(H)) GO TO 100
      V = P(INT(H)) + (P(INT(H)+1.) - P(INT(H))) * (H - INT(H))
      TOUT = V*AVT
      RETURN
C
  100 CONTINUE
```

```
      TOUT = P(H)*AVT
      RETURN
      END
C
```

B The CALPAS3 and DOE-2 Input Files

```
;   Model Description in CALPAS3  ;
;   ****************************  ;


 *title         structure 1
 site           location = adobebuilding at erl, tucson
 greflect       jangr=0.33             febgr=0.33           margr=0.33        &
                aprgr=0.33             maygr=0.33           jungr=0.33        &
                julgr=0.33             auggr=0.33           sepgr=0.33        &
                octgr=0.33             novgr=0.33           decgr=0.33
 hout           4
 *house         *flrarea=1084.4        vol=10844            cair=1895
 roof           area=1084.4            azm=180              tilt=7            &
                uval=0.03              absrp=0.2            inside=air
 wall           name=southwall         *area=167.3          *azm=0            &
                tilt=90                uval=1.1             absrp=0.5         &
                inside=exwall
 wall           name=northwall         *area=252.5          *azm=180          &
                tilt=90                uval=1.1             absrp=0.5         &
                inside=exwall
 wall           name=eastwall          *area=222.2          *azm=-90          &
                tilt=90                uval=1.1             absrp=0.5         &
                inside=exwall
 wall           name=westwall          *area=222.2          *azm=90           &
                tilt=90                uval=1.1             absrp=0.5         &
                inside=exwall
 wall           name=slabedge          *area=134.6          *azm=180          &
                tilt=90                uval=0.9             absrp=0           &
                inside=air
 slab           *area=1000             *thkns=7                               &
                *material=brickcmn     htahs=1.3                              &
                rsurf=0.0
 intwall        *area=470              *thkns=6                               &
                *material=adobe        htahs=1.5            rsurf=0.0
 exwall         *thkns=16              *material=adobe      htahs=1.5         &
                rsurf=0.0
 glass          name=north1            *area=16             *azm=180          &
                tilt=90                nglz=2               uval=0.6          &
                glstyp=2
 sgdistwntr     air=1                  slb=0
 sgdistsmr      air=1                  slb=0
 glass          name=north2            *area=16             *azm=180          &
                tilt=90                nglz=2               uval=0.6          &
                glstyp=2
 sgdistwntr     air=1                  slb=0
 sgdistsmr      air=1                  slb=0
 glass          name=north3            *area=16             *azm=180          &
                tilt=90                nglz=2               uval=0.6          &
                glstyp=2
 sgdistwntr     air=1                  slb=0
 sgdistsmr      air=1                  slb=0
 glass          name=south1            *area=24             *azm=0            &
                tilt=90                nglz=2               uval=0.6          &
                glstyp=2
 sgdistwntr     air=1                  slb=0
 sgdistsmr      air=1                  slb=0
 shading        *wheight=4             *wwidth=6            ohdepth=1         &
                ohwd=3                 ohlx=3               ohrx=35           &
                ohflap=0.2
 glass          name=south2            *area=24             *azm=0            &
```

```
               tilt=90                nglz=2                uval=0.6          &
               glstyp=2
sgdistwntr     air=1                  slb=0
sgdistsmr      air=1                  slb=0
shading        *wheight=4             *wwidth=6             ohdepth=1         &
               ohwd=3                 ohlx=13               ohrx=25           &
               ohflap=0.2
infil          acbase=0               actd=0                acwind=0
intgain        intgain=0              sched=res             air=1
vent           *type=none
tstatswntr     theat=72               tdsrd=73              tcool=74          &
               theatnight=72
tstatssmr      theat=72               tdsrd=73              tcool=74          &
               theatnight=72

sunspace       *flrarea=182.5         vol=1430              cair=300
ssroof         area=184               azm=0                 tilt=15           &
               uval=0.04              absrp=0.3
sswall         name=southwall         *area=10              *azm=0            &
               tilt=90                uval=0.9              absrp=0.2
sswall         name=eastwall          *area=30              *azm=-90          &
               tilt=90                uval=0.9              absrp=0.2
sswall         name=westwall          *area=30              *azm=0            &
               tilt=90                uval=0.9              absrp=0.2
sswall         name=slabedge          *area=56              *azm=180          &
               tilt=90                uval=0.9              absrp=0
ssmasswall     *area=110              *thkns=16                               &
               *material=adobe        htass=0               htahs=1.5         &
               rsurf=0                hogls=0               hotass=1.5        &
               hgtass=1.5
ssmwglass      *azm=0                 tilt=90               nglz=2            &
               uglass=0.85            glstyp=2
sgdistwntr     ssair=1                ssslb=0
sgdistsmr      ssair=1                ssslb=0
ssslab         *area=185              *thkns=4                                &
               *material=conc100      htass=1.3                               &
               rsurf=0
ssglass        name=eastglass         *area=60              *azm=-90          &
               tilt=90                nglz=2                uval=0.85         &
               glstyp=2
sgdistwntr     ssair=1                ssslb=0
sgdistsmr      ssair=1                ssslb=0
ssglass        name=westglass         *area=60              *azm=90           &
               tilt=90                nglz=2                uval=0.85         &
               glstyp=2
sgdistwntr     ssair=1                ssslb=0
sgdistsmr      ssair=1                ssslb=0
ssinfil        acbase=0               actd=0                acwind=0
ssvent         *type=none
sststatswntr   theat=0                tvent=150
sststatssmr    theat=0                tvent=150
sscoupling     uatahs=15.2            *vent=none

warmup         *wudays=14
solarcalc      *freq=daily
solarprint     *firstday=jun-01       lastday=jun-02                          &
               report1=wall           report2=glssmr
printhourly    *firstday=jun-01       lastday=jun-30                          &
               report1=hseb           report2=sseb          report3=cond      &
               report4=temp
*end
```

```
             $ ---  Model Description in DOE-2 ---  $
             $      **************************      $


INPUT LOADS ..
TITLE     LINE-1 *ADOBEBUILDING, RESIDENCE  RUN 1 *
          LINE-2 *---MASSWALL ADOBE DESIGN--- *
          LINE-3 *ENVIRONMENTAL RESEARCH LAB *
          LINE-4 *TUCSON, AZ * ..

                          $ -----HEADING----- $

          ABORT                 ERRORS ..
          DIAGNOSTIC            WARNINGS  ..
          RUN-PERIOD            JAN 01 1991 THRU DEC 31 1991 ..
          BUILDING-LOCATION     LAT=32.07  LON=110.56  ALT=2555.1  T-Z=7  D-S=NO
                                HOL=NO  AZ=0.0  ..
          LOADS-REPORT          SUMMARY=(LS-A)
                                VERIFICATION=(LV-A)  ..

                          $ -----MATERIALS----- $

ADOBE-16  =MAT      TH=1.333 COND=.3000 DENS=110 S-H=.2  $16'ADOBEWALL$ ..
ADOBE-12  =MAT      TH=1.000 COND=.3000 DENS=110 S-H=.2  $12'ADOBEWALL$ ..
DRYWALL-1 =MAT      TH=.0417 COND=.0925 DENS=50  S-H=.26 $1/2 IN DRYWALL$ ..
ROOF-INS-1=MAT      RES=29                               $R30 INSULATION$ ..
PLYW-1    =MAT      TH=.0417 COND=.0667 DENS=34  S-H=.29 $1/2-IN PLYWOOD$ ..
STEEL-ROOF=MAT      TH=.125  COND=26.2  DENS=489 S-H=.12 $1 1/2' STEEL$ ..
CONCRETE-1=MAT      TH=.3333 COND=.408  DENS=100 S-H=.22 $4-IN CONCRETE$ ..
BRICKCMN  =MAT      TH=.5833 COND=.420  DENS=120 S-H=.2  $7' BRICKSLAB$ ..

                          $ -----GLAZING----- $

GT-WIN-1  =GLASS-TYPE           P=2   S-C=.85  G-C=.6   ..

                          $ -----CONSTRUCTIONS----- $

LAY-1     =LAYERS               MAT=(ADOBE-16) I-F-R=0.67  ..
MASS-1    =CONS                 LAYERS=LAY-1 ABS=.5  ROUGHNESS=2 ..

LAY-2     =LAYERS               MAT=(STEEL-ROOF,ROOF-INS-1,PLYW-1,DRYWALL-1)
                                I-F-R=.77 ..
ROOF-1    =CONS                 LAYERS=LAY-2  ABS=.2  ROUGHNESS=6 ..

LAY-3     =LAYERS               MAT=(BRICKCMN)  I-F-R=.67 ..
SLAB-1    =CONS                 LAYERS=LAY-3  ..

LAY-4     =LAYERS               MAT=(CONCRETE-1)  I-F-R=0.67 ..
SLAB-2    =CONS                 LAYERS=LAY-4 ..

LAY-5     =LAYERS               MAT=(ADOBE-12)  I-F-R=.77 ..
INSIDE-1  =CONS                 LAYERS=LAY-5  ROUGHNESS=4 ..

DR-1      =CONS                 U=.629  ABS=.78 ROUGHNESS=4  ..
INS-WL-1  =CONS                 U=.9    ABS=.2  ROUGHNESS=4  ..
SSR-1     =CONS                 U=.04   ABS=.3  ROUGHNESS=3 ..

                        $ -----BUILDING SHADE----- $

          BUILDING-SHADE        X=40.66  Y=0  Z=9  H=1.5  W=10.66
                                AZ=180  TILT=7  $FRONT OVERHANG$  ..
```

```
          BUILDING-SHADE      X=40.66  Y=28.16  Z=8.33  H=1.5  W=10.66
                              AZ=180  TILT=7   $REAR OVERHANG$  ..

                       $ -----SPACE DESCRIPTION----- $

COND-1    =SPACE-CONDITIONS   FLOOR-WEIGHT=0
                              ZONE-TYPE=UNCONDITIONED ..

$******* ROOM 1 ********$

ROOM-1    =SPACE              X=0  Y=0  Z=0  AZ=0  A=781.2  V=7812
                              S-C=COND-1   ..

S-WL-1     =E-W               H=9.33  W=23  AZ=180 CONS=MASS-1  G-R=.33 ..
WIN-1       =WI               H=4.0  W=6.0  G-T=GT-WIN-1  X=3  Y=3   ..
WIN-2       =WI               H=4.0  W=6.0  G-T=GT-WIN-1  X=13  Y=3   ..

E-WL-1     =E-W               H=8.33  W=14.8  CONS=MASS-1 AZ=90  G-R=.33
                              X=40.66  ..

NE-WL-1    =I-W               A=130  CONS=INSIDE-1  N-T=ROOM-2 ..
DOOR-I-1    =DOOR             H=6.67 W=3.5 CONS=DR-1 X=2.0 INF-COEF=150  ..

NM-WL-1    =I-W               LIKE NE-WL-1  N-T=ROOM-3  ..
DOOR-I-2    =DOOR             LIKE DOOR-I-1 X=7.8 ..

NW-WL-1    =E-W               H=8.33  W=13.33 CONS=MASS-1 AZ=0 G-R=.33  X=13.33
                              Y=26.66 ..
WIN-3       =WI               H=4.0  W=4.0  X=6.33  Y=3  G-T=GT-WIN-1  ..
DOOR-1      =DOOR             H=6.67  W=3.5  CONS=DR-1  X=1.0 ..

W-WL-1     =E-W               LIKE E-WL-1   X=0   Y=26.66  AZ=270 W=26.66 ..

EM-WL-1    =I-W               A=110 CONS=INSIDE-1 N-T=ROOM-3 ..

FLOOR-1    =U-F               CONS=SLAB-1  A=781.2  $PERIMETER AREA = 95.45$
                              U-EFFECTIVE = .015  ..

RF-1      =ROOF               H=19.2   W=40.66  X=13.33 Y=26.66 Z=9  AZ=0
                              TILT=7 CONS=ROOF-1 ..

$******* ROOM 2 ********$

ROOM-2    =SPACE              X=26.33  Y=12.33 Z=0 AZ=0 A=160.3 V=1603
                              S-C=COND-1  ..

E-WL-2     =E-W               H=8.33 W=12.33 CONS=MASS-1 AZ=90 G-R=.33
                              X=14.33  ..
N-WL-2     =E-W               H=8.33 W=13.8 CONS=MASS-1 AZ=0 G-R=.33  X=14.33
                              Y=12.33 ..

W-WL-2     =I-W               LIKE EM-WL-1  N-T=ROOM-3 ..
FLOOR-2    =U-F               CONS=SLAB-1  A=160.3  $PERIMETER AREA = 26.2$
                              U-EFFECTIVE = .019  ..

RF-2      =ROOF               H=11.6   W=13.88 X=14.33 Y=12.33 Z=8.33  AZ=0
                              TILT=7 CONS=ROOF-1 ..

$******* ROOM 3 ********$

ROOM-3    =SPACE              X=13.33 Y=12.33 Z=0 AZ=0 A=160.3 V=1603
                              S-C=COND-1 ..
```

```
N-WL-3     =E-W               H=8.33 W=13.8 CONS=MASS-1 AZ=0 G-R=.33  X=13.8
                              Y=12.33 ..
WIN-5       =WI               H=4.0  W=4.0  X=4.33 Y=3  G-T=GT-WIN-1  ..

FLOOR-3    =U-F               CONS=SLAB-1  A=160.3  $PERIMETER AREA = 13.8$
                              U-EFFECTIVE = .010  ..

RF-3      =ROOF               H=11.6  W=13.88  X=13.8 Y=12.33 Z=8.33  AZ=0
                              TILT=7 CONS=ROOF-1 ..


                     $ --- SUNSPACE DESCRIPTION --- $

SUNSP-1   =SPACE              SUNSPACE=YES
                              AREA=182.5  VOLUME=1430
                              X=23  Y=-10.33  AZIMUTH=0
                              FLOOR-WEIGHT=0 ..

SSLEFT     =EXTERIOR-WALL     H=9.5  W=10.33  X=0  Y=10.33  AZ=270  Z=0
                              CONS=INS-WL-1  ..
SSLEFTWIN   =WINDOW           H=6.5  W=9      X=.5 Y=.5
                              G-T=GT-WIN-1 ..

SSRIGHT    =EXTERIOR-WALL     H=9.5  W=10.33  X=17.66  Y=0  AZ=90  Z=0
                              CONS=INS-WL-1  ..
SSRIGHTWIN  =WINDOW           LIKE SSLEFTWIN ..

SSFRONT    =EXTERIOR-WALL     H=7.5 W=17.66  X=0  Y=0  AZ=180  Z=0
                              CONS=INS-WL-1  ..
SSFRONTWIN  =WINDOW           H=6.5  W=16  X=.5  Y=.5
                              GLASS-TYPE=GT-WIN-1 ..


SSROOF     =ROOF              H=10.5  W=17.6  X=0  Y=0  Z=7
                              AZ=180  TILT=15
                              CONS=SSR-1  ..


SSINTWAL-1 =INTERIOR-WALL     H=8 W=17.66  Y=10.33  AZ=0
                              CONS=MASS-1  NEXT-TO=ROOM-1  ..
INTWINDOW   =WINDOW           H=4.0 W=3.0 X=12  Y=3
                              GLASS-TYPE=GT-WIN-1  ..
INTDOOR     =DOOR             H=6.67 W=3.5 X=1.33 CONS=DR-1 ..

SSFLOOR    =UNDERGROUND-FLOOR AREA=185  CONS=SLAB-2
                              U-EFFECTIVE=.055  ..

                   $ --- HOURLY REPORTS --- $

HR-SCH-1  =SCHEDULE           THRU MAY 31 (ALL) (1,24) (0)
                              THRU JUN 30 (ALL) (1,24) (1)
                              THRU NOV 30 (ALL) (1,24) (0)
                              THRU DEC 31 (ALL) (1,24) (1) ..

HR-SCH-2  =SCHEDULE           THRU MAY 31 (ALL) (1,24) (0)
                              THRU JUN 01 (ALL) (1,24) (1)
                              THRU NOV 30 (ALL) (1,24) (0)
                              THRU DEC 01 (ALL) (1,24) (1)
                              THRU DEC 31 (ALL) (1,24) (0) ..

OUT-1     =REPORT-BLOCK       VARIABLE-TYPE=GLOBAL
                              VARIABLE-LIST=(4) ..
OUT-2     =REPORT-BLOCK       VARIABLE-TYPE=BUILDING
                              VARIABLE-LIST=(1,2,19,20) ..
```

```
OUT-3     =REPORT-BLOCK       VARIABLE-TYPE=ROOM-1
                              VARIABLE-LIST=(44) ..
OUT-4     =REPORT-BLOCK       VARIABLE-TYPE=SUNSP-1
                              VARIABLE-LIST=(44) ..
OUT-5     =REPORT-BLOCK       VARIABLE-TYPE=WIN-1
                              VARIABLE-LIST=(15) ..
OUT-6     =REPORT-BLOCK       VARIABLE-TYPE=WIN-3
                              VARIABLE-LIST=(15) ..
OUT-7     =REPORT-BLOCK       VARIABLE-TYPE=SSRIGHTWIN
                              VARIABLE-LIST=(15) ..
OUT-8     =REPORT-BLOCK       VARIABLE-TYPE=SSFRONTWIN
                              VARIABLE-LIST=(15,16) ..
OUT-9     =REPORT-BLOCK       VARIABLE-TYPE=W-WL-1
                              VARIABLE-LIST=(1,5) ..

LDS-REP-1 =HOURLY-REPORT      REPORT-SCHEDULE=HR-SCH-1
                              REPORT-BLOCK=(OUT-1,OUT-2) ..
LDS-REP-2 =HOURLY-REPORT      REPORT-SCHEDULE=HR-SCH-1
                              REPORT-BLOCK=(OUT-3,OUT-4,OUT-9) ..
LDS-REP-3 =HOURLY-REPORT      REPORT-SCHEDULE=HR-SCH-2
                              REPORT-BLOCK=(OUT-6,OUT-5,OUT-7,OUT-8) ..


END ..
COMPUTE LOADS ..
INPUT SYSTEMS ..


                     $ ---- SCHEDULES ----$

HEAT-1    =SCHEDULE           THRU DEC 31 (ALL) (1,24) (10) ..

COOL-1    =SCHEDULE           THRU DEC 31 (ALL) (1,24) (180) ..

                    $ ---- ZONE DATA ----$

ROOM-1    =ZONE               DESIGN-HEAT-T=10
                              DESIGN-COOL-T=180
                              ZONE-TYPE=CONDITIONED
                              THERMOSTAT-TYPE=TWO-POSITION
                              HEAT-TEMP-SCH=HEAT-1
                              COOL-TEMP-SCH=COOL-1 ..

ROOM-2    =ZONE               DESIGN-HEAT-T=10
                              DESIGN-COOL-T=180
                              ZONE-TYPE=UNCONDITIONED ..

ROOM-3    =ZONE               DESIGN-HEAT-T=10
                              DESIGN-COOL-T=180
                              ZONE-TYPE=UNCONDITIONED ..

SUNSP-1   =ZONE               DESIGN-HEAT-T=10
                              DESIGN-COOL-T=180
                              ZONE-TYPE=UNCONDITIONED ..

SYS-1     =SYSTEM             SYSTEM-TYPE=SUM
                              ZONE-NAMES=(ROOM-1,ROOM-2,ROOM-3,SUNSP-1)
                              HEAT-SOURCE=GAS-FURNACE ..

                     $ --- HOURLY REPORTS --- $

HR-SCH-1  =SCHEDULE           THRU MAY 31 (ALL) (1,24) (0)
                              THRU JUN 30 (ALL) (1,24) (1)
                              THRU NOV 30 (ALL) (1,24) (0)
```

```
                              THRU DEC 31 (ALL) (1,24) (1) ..

HR-SCH-2  =SCHEDULE           THRU MAY 31 (ALL) (1,24) (0)
                              THRU JUN 15 (ALL) (1,24) (1)
                              THRU NOV 30 (ALL) (1,24) (0)
                              THRU DEC 15 (ALL) (1,24) (1)
                              THRU DEC 31 (ALL) (1,24) (0) ..

OUT-1     =REPORT-BLOCK       VARIABLE-TYPE=ROOM-1
                              VARIABLE-LIST=(6) ..
OUT-2     =REPORT-BLOCK       VARIABLE-TYPE=SUNSP-1
                              VARIABLE-LIST=(6) ..
OUT-3     =REPORT-BLOCK       VARIABLE-TYPE=ROOM-2
                              VARIABLE-LIST=(6) ..
OUT-4     =REPORT-BLOCK       VARIABLE-TYPE=ROOM-3
                              VARIABLE-LIST=(6) ..
OUT-5     =REPORT-BLOCK       VARIABLE-TYPE=ROOM-1
                              VARIABLE-LIST=(1,2,32,33) ..

SYS-REP-1 =HOURLY-REPORT      REPORT-SCHEDULE=HR-SCH-1
                              REPORT-BLOCK=(OUT-1,OUT-2,OUT-3,OUT-4) ..
SYS-REP-2 =HOURLY-REPORT      REPORT-SCHEDULE=HR-SCH-2
                              REPORT-BLOCK=(OUT-5) ..
END ..
COMPUTE SYSTEMS ..
STOP ..
```

C The Floor Plan of the Test Building

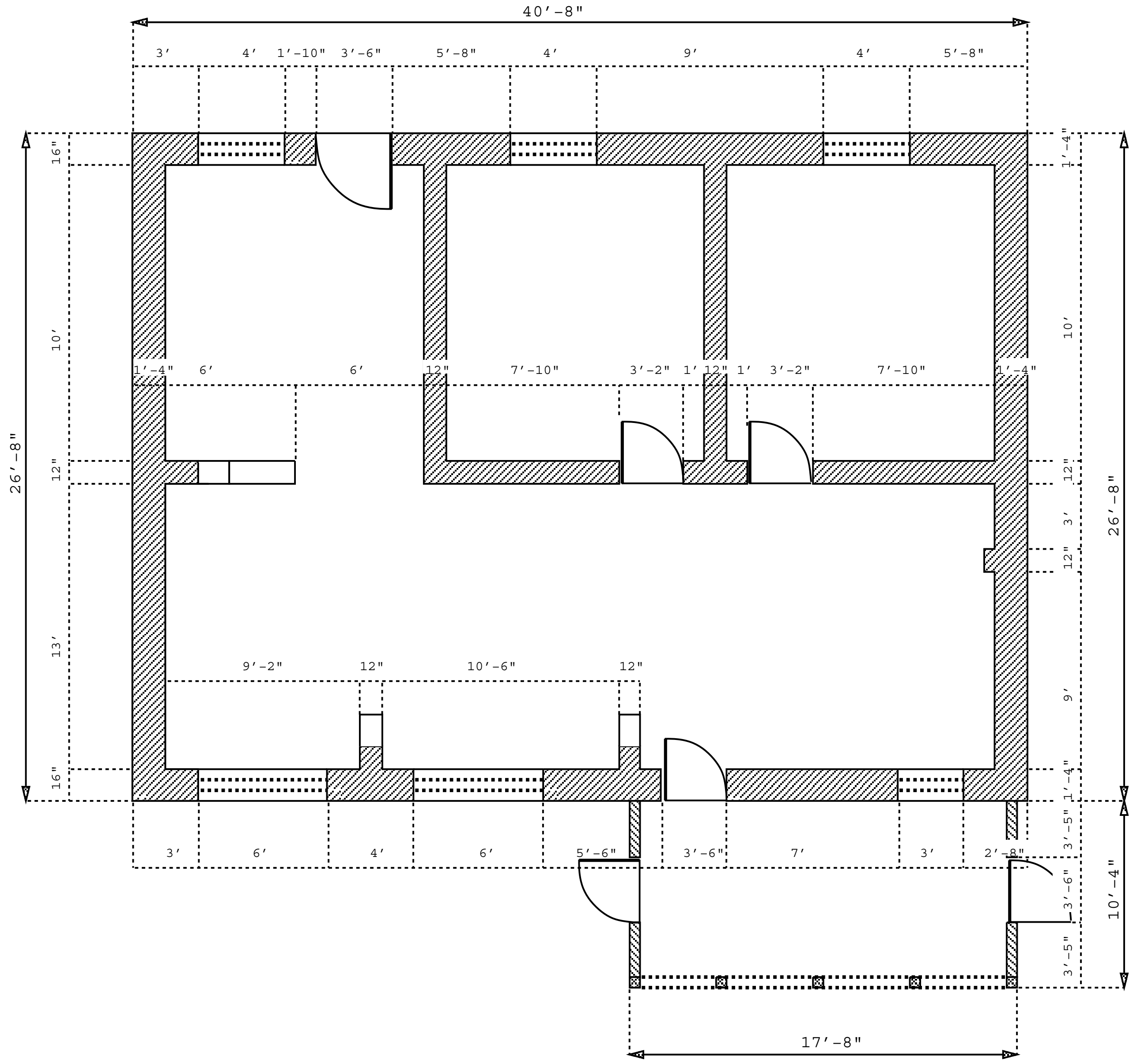
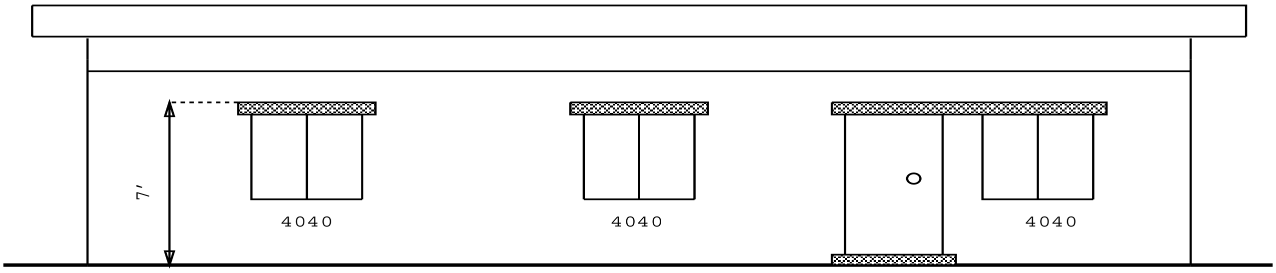
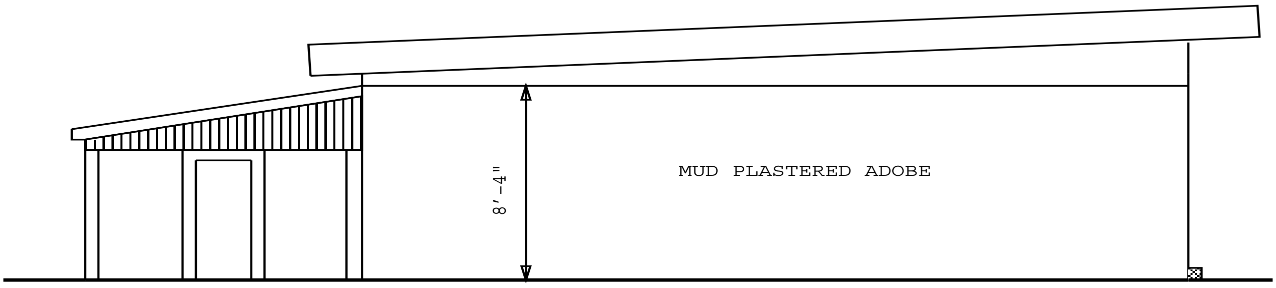


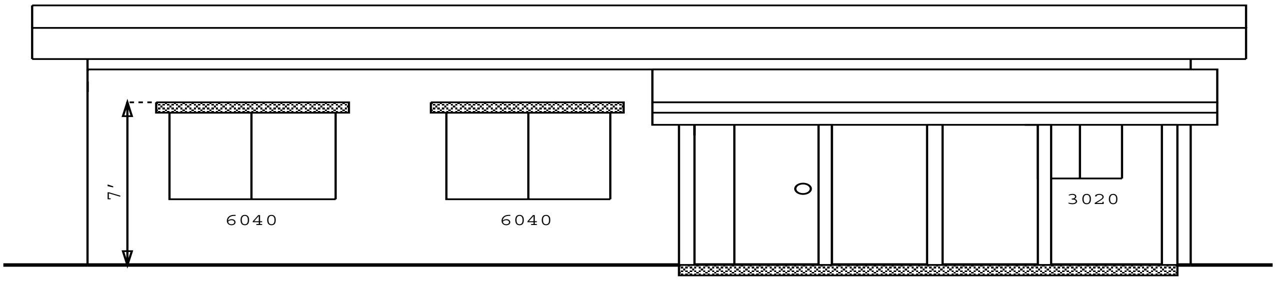
Figure 61: Floor plan of the test structure



NORTH ELEVATION



EAST ELEVATION



SOUTH ELEVATION

Figure 62: Perspectives of the test structure

D Charts of the December Conditions

D.1 Conditions for Dec 15 – Dec 17

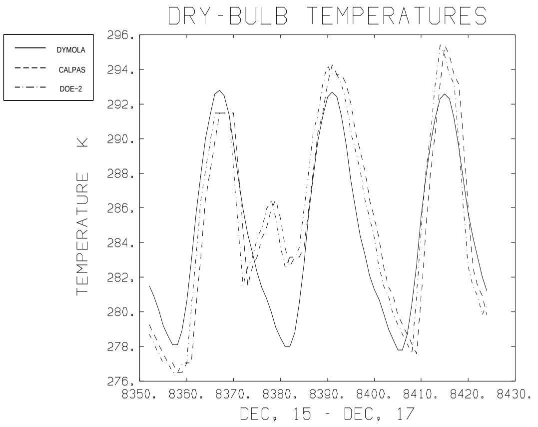


Figure 63: Dry-bulb temperatures

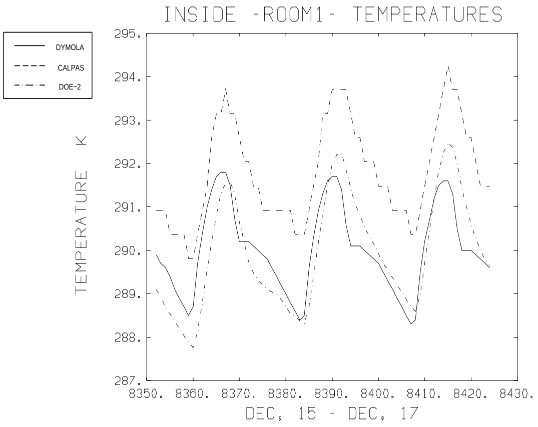


Figure 64: Inside temperatures

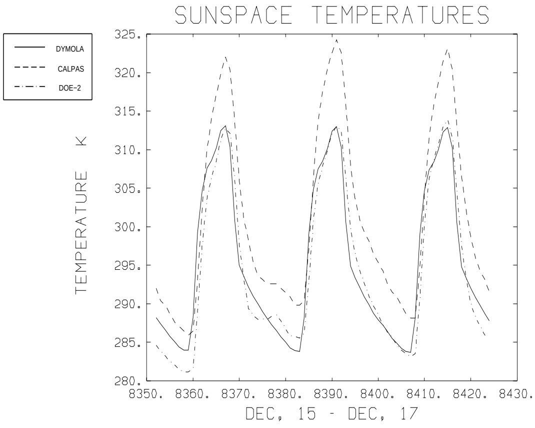


Figure 65: Sunspace temperatures

D.2 Moving average over every day of December

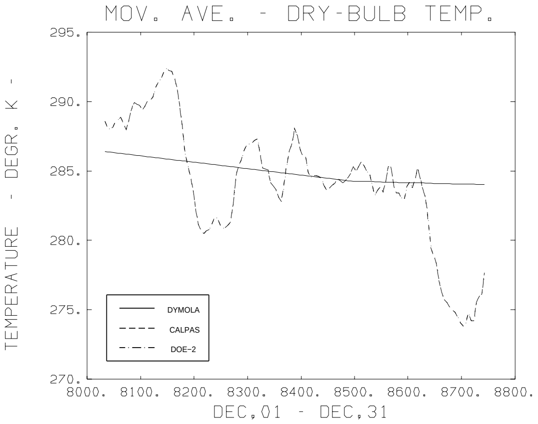


Figure 66: Dry-bulb temperatures

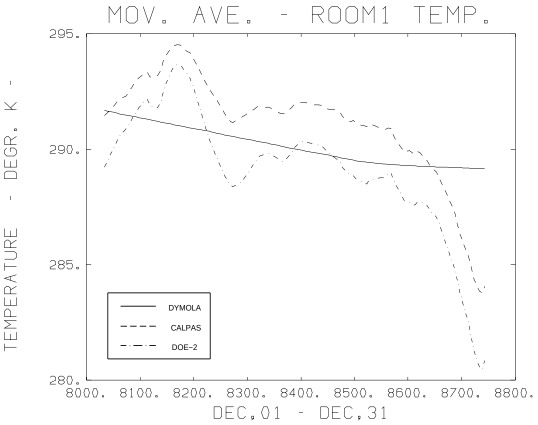


Figure 67: Inside conditions

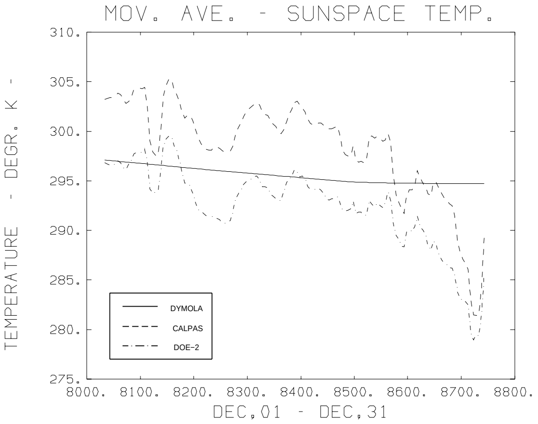


Figure 68: Sunspace conditions