

1. GETTING STARTED

The getting started manual explains what TRNSYS is and what programs make the TRNSYS suite. You will learn how to install TRNSYS, how to open and run the examples, how to create a project in the Simulation Studio and how to use the Multizone Building interface (TRNBuild).

1.1. What is TRNSYS?

TRNSYS is a complete and extensible simulation environment for the transient simulation of systems, including multi-zone buildings. It is used by engineers and researchers around the world to validate new energy concepts, from simple domestic hot water systems to the design and simulation of buildings and their equipment, including control strategies, occupant behavior, alternative energy systems (wind, solar, photovoltaic, hydrogen systems), etc.

One of the key factors in TRNSYS' success over the last 25 years is its open, modular structure. The source code of the kernel as well as the component models is delivered to the end users. This simplifies extending existing models to make them fit the user's specific needs.

The DLL-based architecture allows users and third-party developers to easily add custom component models, using all common programming languages (C, C++, PASCAL, FORTRAN, etc.). In addition, TRNSYS can be easily connected to many other applications, for pre- or post-processing or through interactive calls during the simulation (e.g. Microsoft Excel, Matlab, COMIS, etc.). TRNSYS applications include:

- Solar systems (solar thermal and PV)
- Low energy buildings and HVAC systems with advanced design features (natural ventilation, slab heating/cooling, double façade, etc.)
- Renewable energy systems
- Cogeneration, fuel cells
- Anything that requires dynamic simulation!

Some TRNSYS jargon:

A TRNSYS project is typically setup by connecting components graphically in the **Simulation Studio**. Each **Type** of component is described by a mathematical model in the TRNSYS simulation engine and has a set of matching **Proforma's** in the Simulation Studio. The **proforma has a black-box description of a component: inputs, outputs, parameters**, etc.

TRNSYS components are often referred to as **Types** (e.g. Type 1 is the solar collector). The **Multizone building model** is known as **Type 56**.

The Simulation Studio generates a text input file for the TRNSYS simulation engine. That **input file** is referred to as the **deck file**.

In this manual, %TRNSYS16% refers to the TRNSYS 16 installation directory, i.e. C:\Program Files\Trnsys16 if you chose to keep the default location

1.2. The TRNSYS Suite

TRNSYS consists of a suite of programs: The TRNSYS simulation Studio, the simulation engine (TRNDII.dll) and its executable (TRNExe.exe), the Building input data visual interface (TRNBuild.exe), and the Editor used to create stand-alone redistributable programs known as TRNSED applications (TRNEdit.exe).

1.2.1. The TRNSYS Simulation Studio

The main visual interface is the TRNSYS Simulation Studio (formerly known as IISiBat). From there, you can create projects by drag-and-dropping components to the workspace, connecting them together and setting the global simulation parameters.

The Simulation Studio saves the project information in a Trnsys Project File (*.tpf). When you run a simulation, the Studio also creates a TRNSYS input file (text file that contains all the information on the simulation but no graphical information).

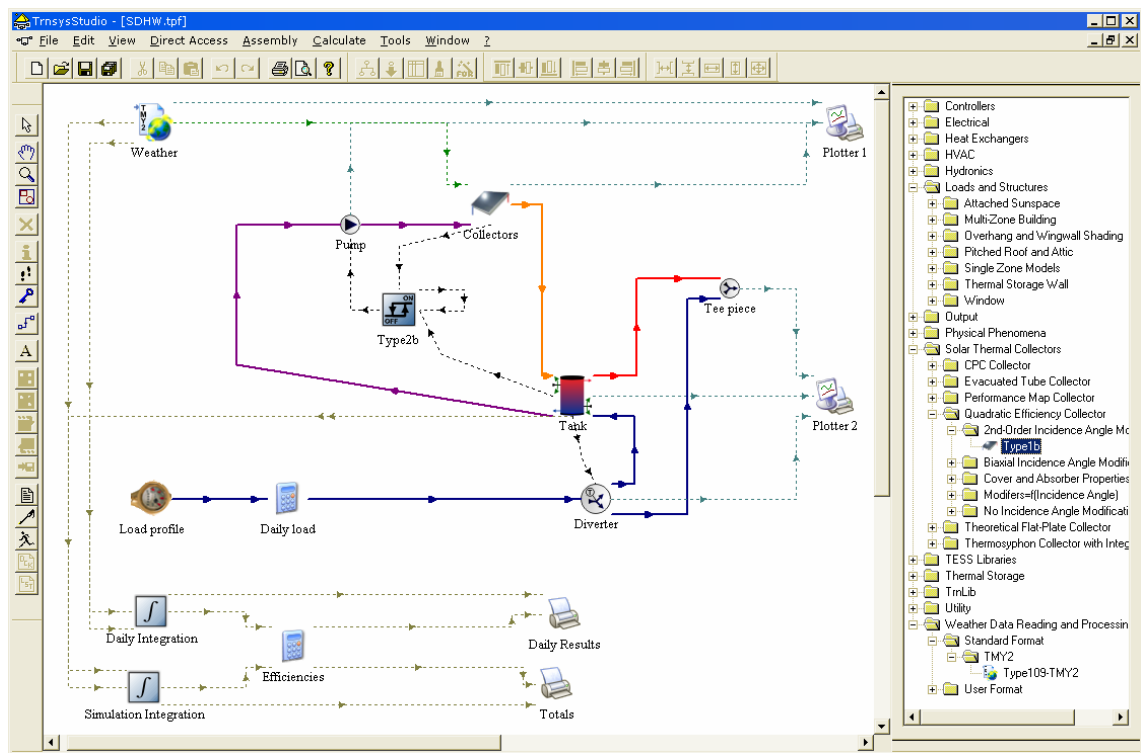


Figure 1–1: The TRNSYS Simulation Studio

The simulation Studio also includes an output manager from where you control which variables are integrated, printed and/or plotted, and a log/error manager that allows you to study in detail what happened during a simulation.

You can also perform many additional tasks from the Simulation Studio: Generate projects using the "New Project Wizard", generate a skeleton for new components using the Fortran Wizard, view and edit the components proformas (a proforma is the input/output/parameters description of a component), view output files, etc.

1.2.2. The TRNSYS Simulation engine

The simulation engine is programmed in Fortran and the source is distributed (see the \SourceCode directory). The engine is compiled into a Windows Dynamic Link Library (DLL), TRNDII. The TRNSYS kernel reads all the information on the simulation (which components are used and how they are connected) in the TRNSYS input file, known as the deck file (*.dck). It also opens additional input files (e.g. weather data) and creates output files.

The simulation engine is called by an executable program, TRNExe, which also implements the online plotter, a very useful tool that allows you to view dozens of output variables during a simulation.

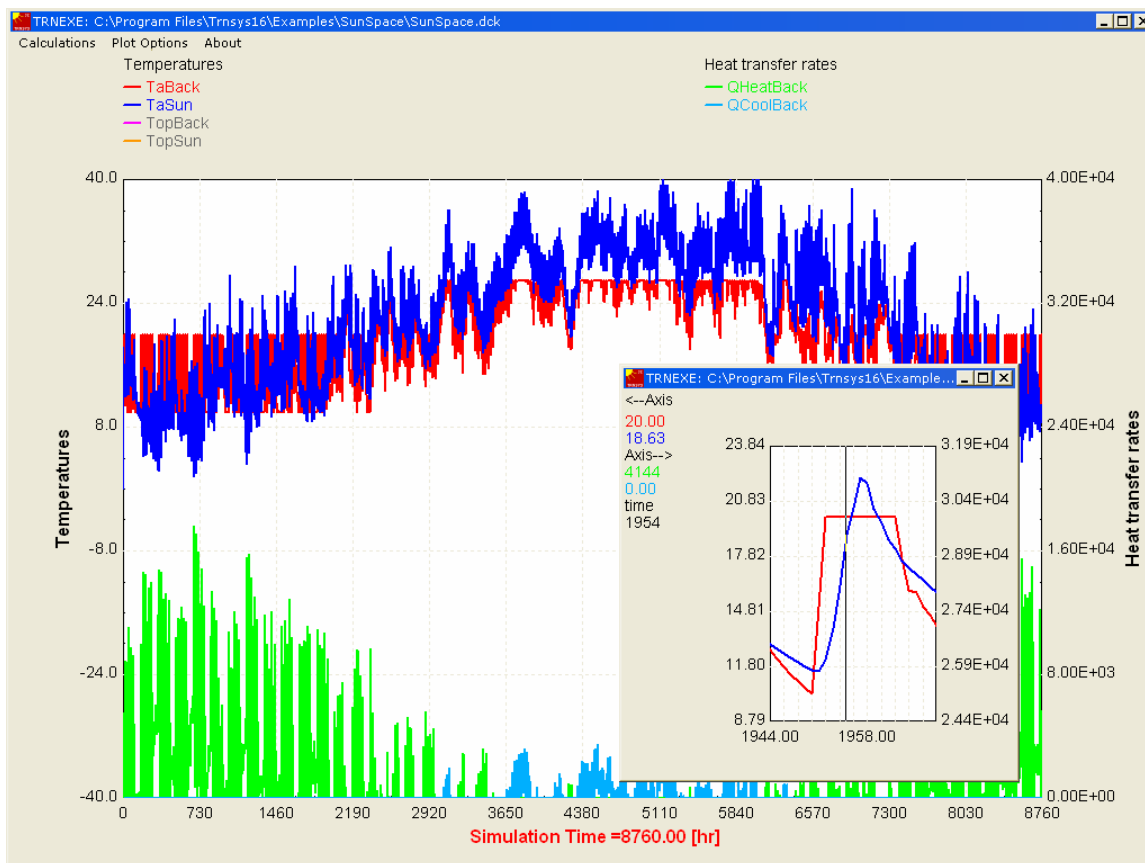


Figure 1–2: The online plotter in TRNExe

The online plotter provides some advanced features such as zooming and display of numerical values of the variables at any time step, as shown in the zoom part of Figure 1–2.

1.2.3. The Building visual interface

TRNBuild (formerly known as Prebid) is the tool used to enter input data for multizone buildings. It allows you to specify all the building structure details, as well as everything that is needed to simulate the thermal behavior of the building, such as windows optical properties, heating and cooling schedules, etc.

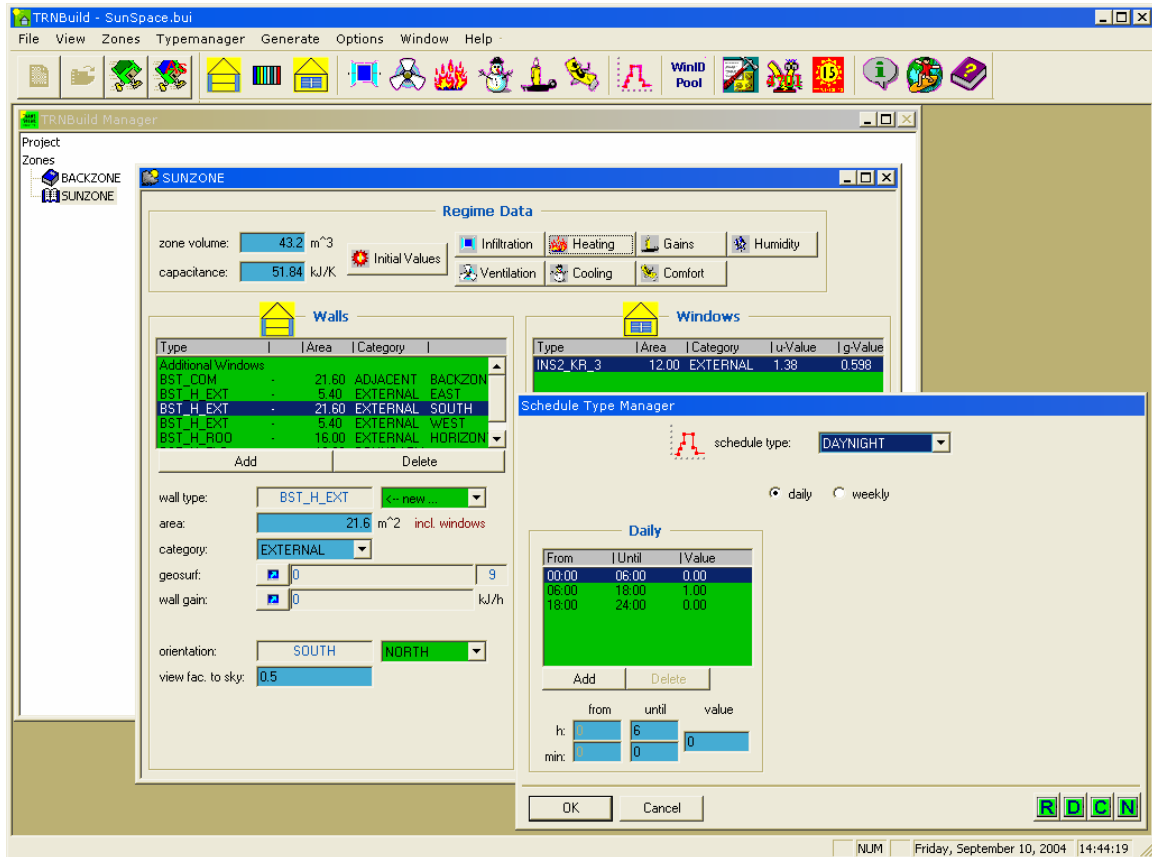


Figure 1–3: TRNBuild

TRNBuild creates a building description file (*.bui) that includes all the information required to simulate the building. Note that this was not the case in TRNSYS 15 where the window library (w4-lib.dat) was always required to run a simulation.

1.2.4. TRNEdit and TRNSED applications

TRNEdit is a specialized editor that can be used to create or modify TRNSYS input files (decks). This is not recommended in general and only advanced users should attempt to modify deck files by hand. Most users should rely on the Simulation Studio to generate and modify deck files.

On the other hand, TRNEdit can be used to create redistributable applications (known as TRNSED applications). Those executables can be freely distributed to end-users who do not have a TRNSYS license in order to offer them a simplified simulation tool. The distributable includes a dedicated visual interface designed by adding special commands to the TRNSYS input

file. Advanced features, such as multiple windows (tabs) and clickable pictures, have been added in TRNSYS 16.

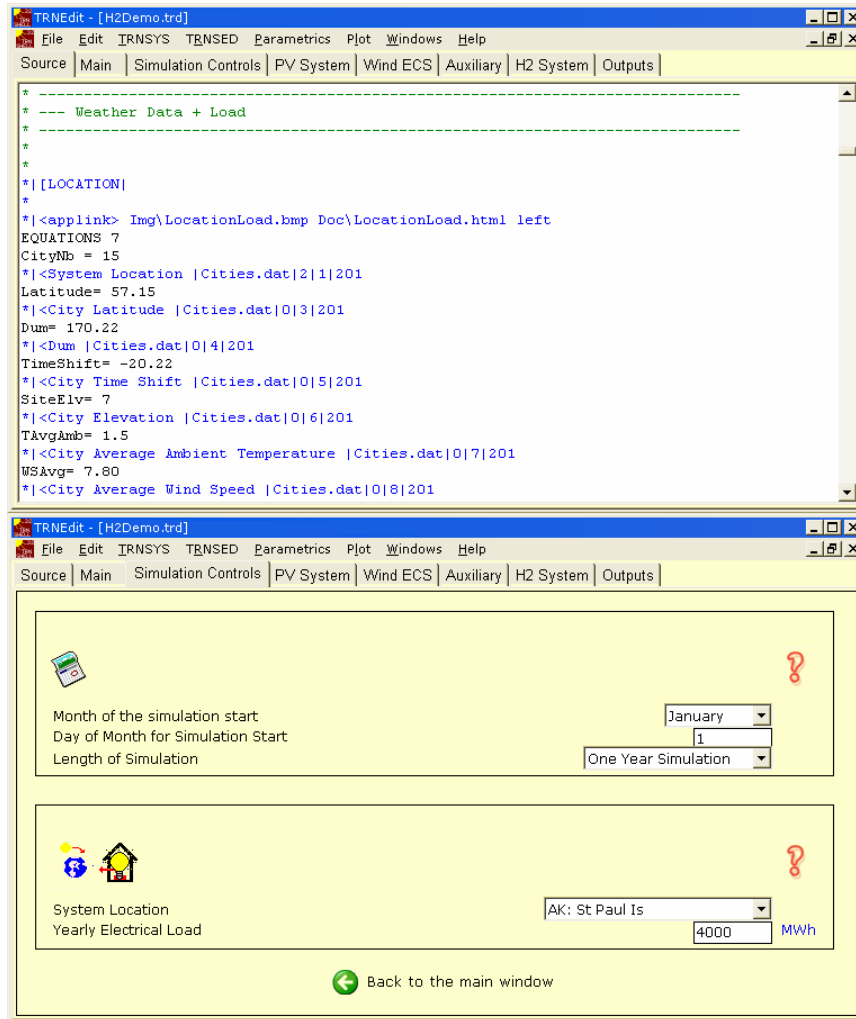


Figure 1-4: TRNEdit – Tabbed view to design TRNSEd applications

For more information on TRNSEd applications licensing, please check the license.txt file in your installation.

1.2.5. TRNSYS add-ons

TRNSYS offers a broad variety of standard components, and many additional libraries are available to expand its capabilities:

- TRNLIB: sel.me.wisc.edu/trnsys/trnlib (free component library)
- TRANSSOLAR libraries: www.transsolar.com
- TESS libraries: www.tess-inc.com

6. MULTIZONE BUILDING MODELING WITH TYPE56 AND TRNBUILD

6.1. Introduction

There are 2 main sections in this guide:

- Section 6.2 explains how to use the TRNBuild program to enter multizone building data into TRNSYS.
- Section 6.3 describes the mathematical models and assumptions behind the Type56 multizone building model.

6.2. TRNBuild

6.2.1. Introduction

Due to the complexity of a multizone building the parameters of TYPE 56 are not defined directly in the TRNSYS input file. Instead, two files are assigned containing the required information, the building description (*.BLD) and the ASHRAE transfer function for walls (*.TRN). More information about these files can be found in Section 6.3.

TRNBUILD (formerly known as PREBID) has been developed to provide an easy-to-use tool for creating the *.BLD and *.TRN files. Starting with some basic project data, the user describes each thermal zone in turn. Finally, the desired outputs are selected. All data entered are saved in a so-called building file (*.BUI), a readable ASCII text file. The BUI file is very handy for checking data entered in TRNBuild.

Note: The BUI-file has a rigorous syntax. Editing this file may cause a lot of trouble!

Compared to the previous version PREBID, several improvements have been made to TRNBuild and to the multizone building model itself (TYPE 56):

User interface:

- Automatic segmentation of active layers
- Automatic creation of T56 input files (*.bld,, *.trn, *.inf)
- Chilled ceilings
- Improved library management (walls, layers, gains, schedules and windows),
- Long variable names (all characters allowed except “blank” “:” “;”).
- Rename/copy/paste/delete/new option in all TYPE-Managers (e.g. wall, window, gains)
- Transparent insulation is treated like a normal window with a special data curve

Physical and mathematical modeling:

- Automatic calculation of convective heat transfer coefficients depending on surface temperatures
- New 2-band radiation window model
- Solar and thermal energy as well as moisture is automatically balanced

Note: Despite these improvements, the BUI-file created by PREBID 3 or 4 can be imported into TRNBuild 1.0. However, files can only be saved into a TRNBuild version 1.0 format. Errors and unexpected behavior may occur by loading files which have been created or changed outside of PREBID or TRNBuild!

6.2.2. Getting Started

Assuming that you have installed the TRNSYS package correctly, TRNBuild can be started from TRNEDIT under the UTILITIES menu or directly by double-clicking on the TRNBuild icon in the TRNSYS group. TRNBuild is also housed within the TrnsysStudio environment program. (Note, that TRNBuild 1.0 requires a screen resolution of at least 800x600 and WINDOW NT or 95,98,2000 or newer).

The initial TRNBuild window is shown in Figure 6.2.2-1. The information window will close automatically. The main menu of the initial TRNBuild window houses the following items:

- FILE (open, new, close, or save a *.BUI file)
- VIEW (toolbar, status bar)
- OPTIONS (settings such as library versions and paths for the BIDWIN program)
- WINDOW (cascade, tile, arrange icons, etc.)
- HELP

After you opened a new or existing project three additional items will be available in the main menu:

- ZONES (add and delete zones)
- GENERATE (create BUI file for DIN 4701, run TRNSYS input file)
- TYPEMANAGER (edit previously defined TYPES of walls, windows, gains, ventilation, infiltration, cooling, heating, layers, and schedules)

Many of the features of the main menu are also present in the tool bar. For users with small screens, it might be more convenient to hide the toolbar and status bar.

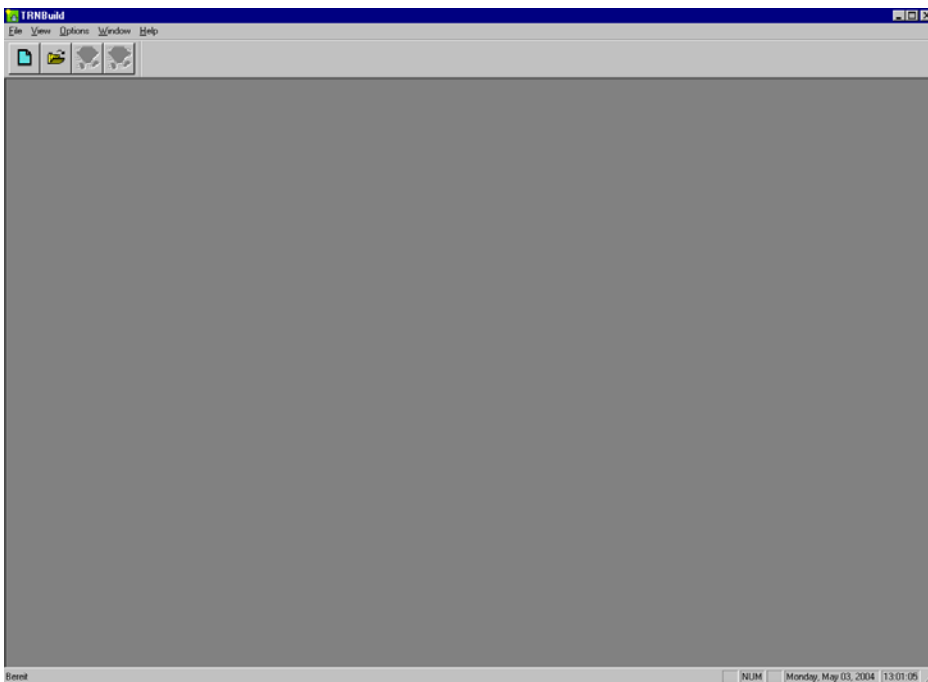


Figure 6.2.2-1: Initial TRNBuild Window

6.2.2.1. Settings

Under the OPTIONS menu, some settings used in TRNBuild can be specified as shown in Figure 6.2.2-2. The Path for standard Libraries accessible in the settings window is only used for:

- TRNSYS: spacer.lib
- TRNFLOW: headers.txt, pollutant.lib

The path and name for all other libraries, for instance windows, walls, layers or schedules, will be preset by the path given here, but can be changed interactively whenever needed, for example when you are describing a new wall or window.

Libraries for common windows, walls, and layers are provided in different languages:

- The United States version contains Standard ASHRAE walls and materials.
- The German version contains materials and walls according to DIN 4108 and VDI 2078, respectively. Glazing materials used on the German market have been added to the German window library.

The location of the TRNSYS application is needed for running DIN 4701 (see Section 6.2.7).

The TRNSYS application name changed from formerly TRNSYS.exe to TRNExe.exe in version 16.

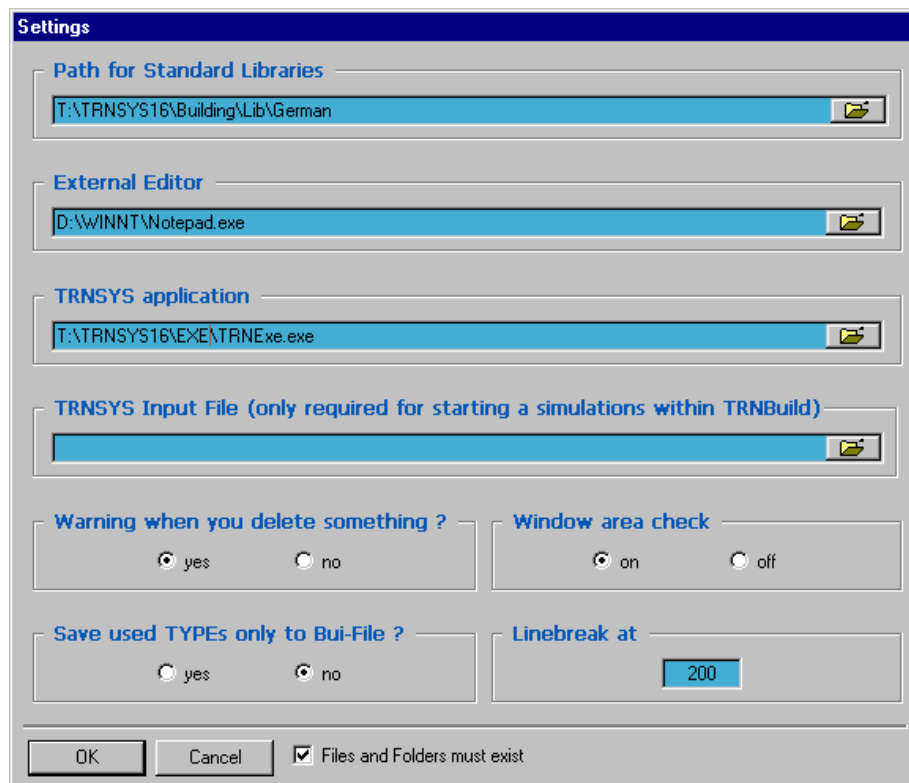


Figure 6.2.2-2: The SETTINGS Window

The new CHECK BOX "Files and Folders must exist" toggles on or off a test for the existence of file paths given in the input windows.

6.2.2.2. Opening and Creating a New File

To open an existing *.BUI file, click on FILE and then click on OPEN. To create a new file, click on FILE and then click on NEW. Once you have opened an existing file or started a new file, the PROJECT INITIALIZATION window opens, as shown in Figure 6.2.2-3. If ZONES have already been defined in this file, the ZONE windows are opened as well.

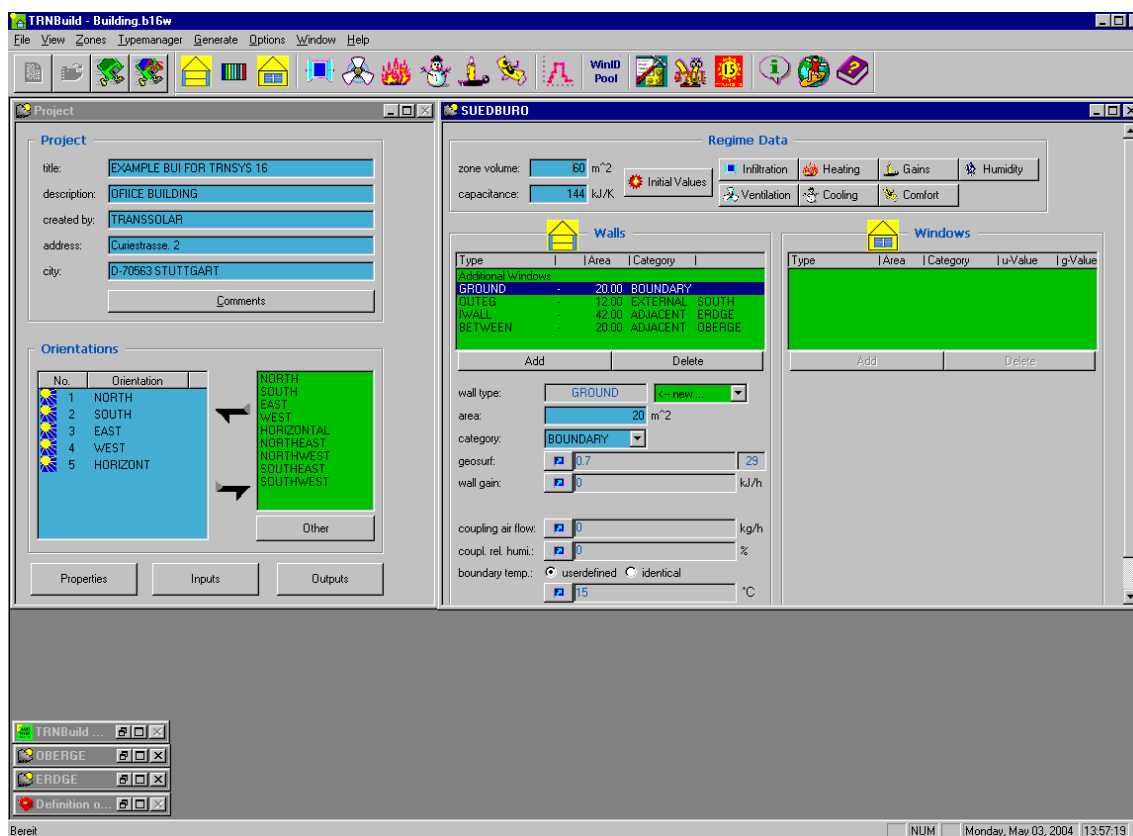


Figure 6.2.2-3: The PROJECT INITIALIZATION and ZONE Windows

6.2.2.3. Specifying the Required Input

In TRNBuild windows, required information is entered in one of several formats: input box, radio button, list box, pull-down menu, or DEF button. The use of each of these formats is explained below:

Input boxes

Input boxes are designed for the user to enter values or text, such as the name of a zone. They are easy to recognize by their light blue background color. (Note: gray colored boxes are used for display only). Some input boxes require a number and will not accept other characters. To enter information in an input box, double-click anywhere in the input box and enter the required information using the keyboard.

Radio buttons

Radio buttons group a set of mutually exclusive options. The selected option is shown with a black dot enclosed within a circle. To select a different radio button option with the mouse, just click on the circle in front of the option name.

List boxes

A list box provides a list of “source” items from which one or more can be selected as shown in Figure 6.2.2-4. Select an item with the mouse, scroll it into view if necessary using the scroll bar. Click on the upper arrow (pointing left) to add an item. The selected item appears in another box on the left (see Figure 6.2.2-4). To delete an item, select the item in the left list and click on the lower arrow (pointing right). Alternatively, the insert and delete key can be used.

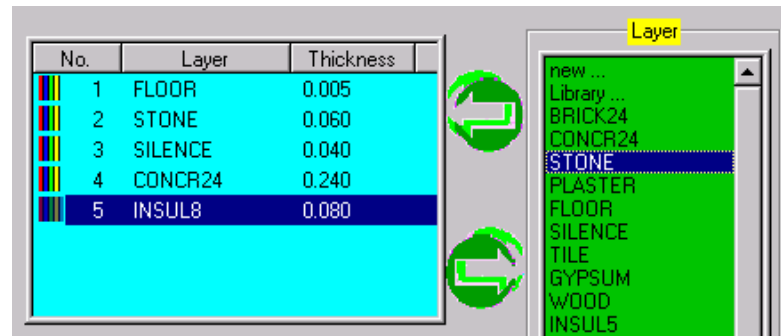


Figure 6.2.2-4: List Box Input

Pull-down menu

A pull-down menu provides a list of items from which only one can be selected. To select an item, click with the mouse on the arrow on the right side and keep the mouse button pressed while looking for the desired item. Release the mouse button when the desired item is highlighted. The pull-down menu reduces again to a single bar and the selected item appears in a display box.

DEF button

A DEF button is used to define items which can be a constant, an input or a schedule (i.e. the shading factor of an internal shading device as shown in Figure 6.2.2-5). In the display box, the defined constant, input or schedule is displayed, respectively. ‘I’ is the abbreviation for input and ‘S’ the abbreviation for schedule.



Figure 6.2.2-5: The DEF Button

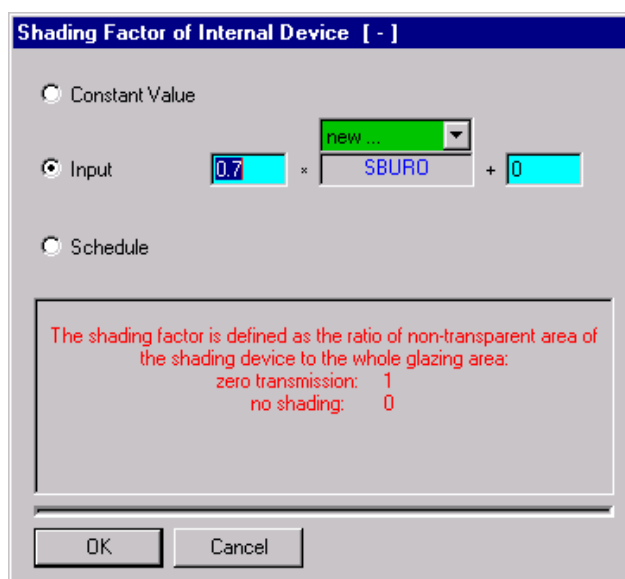


Figure 6.2.2-6: The DEFINITION Window for a Constant, an Input, or a Schedule

After clicking on the DEF button, a definition window opens as shown in Figure 6.2.2-6. A brief online help is provided in an information box. Using a radio button, the user selects whether a constant, an input or a schedule is to be defined:

- For a constant, the user enters a single value.
- For an input, the user selects an input from the pull-down menu as well as a multiplication and addition factor. If the option NEW from the pull-down menu is selected, the user is asked to enter a new unique input name.
- For defining a schedule, the user selects a schedule from the pull-down menu as well as multiplication and addition factors. If the option NEW from the pull-down menu is selected, a window for specifying the schedule appears as shown in Figure 6.2.2-7. This window offers the option to define a daily or a weekly schedule. For a daily schedule, the user specifies the type name first. Then the daily schedule is defined by entering values for the desired time intervals. For a weekly schedule, the user selects a daily schedule for every day of the week by a pull-down menu as shown in Figure 6.2.2-8. If the desired daily schedule has not yet been defined, the option NEW allows the user to define a new daily schedule.

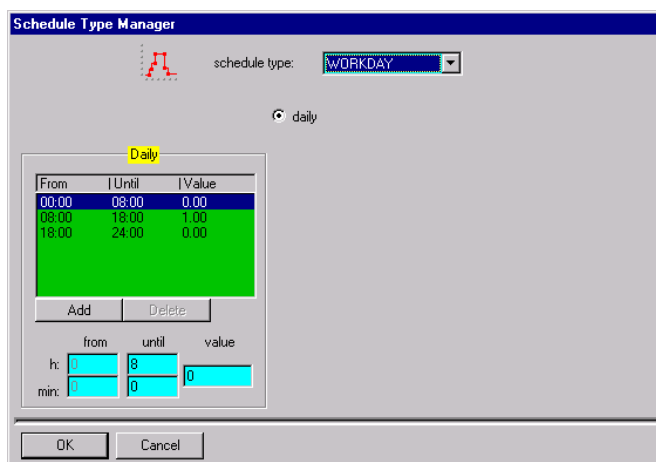


Figure 6.2.2-7: The SCHEDULE Window for a Daily Schedule

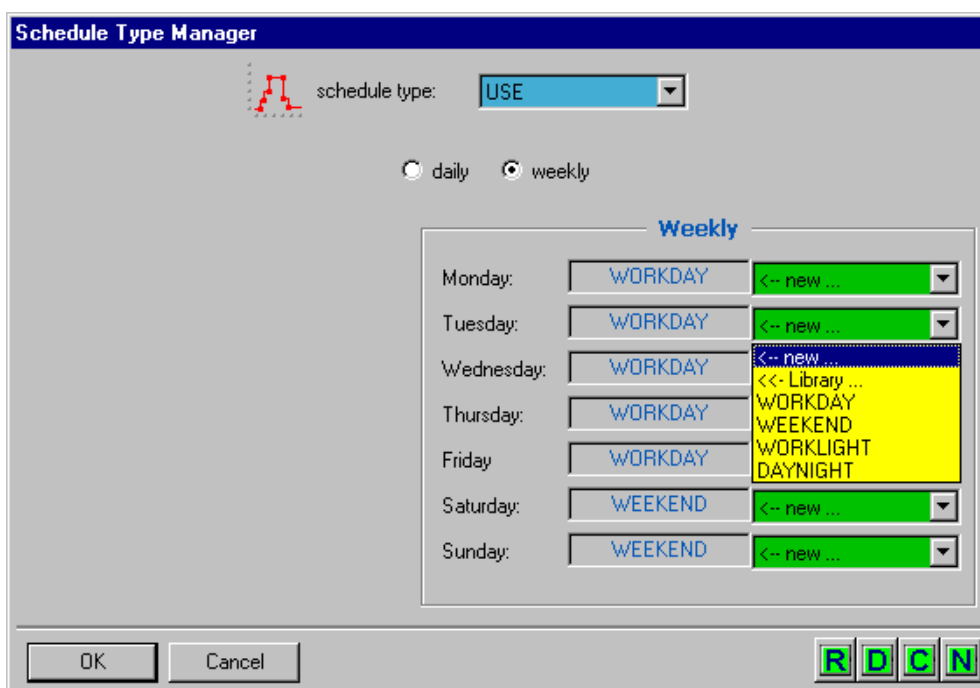


Figure 6.2.2-8: The SCHEDULE Window for a Weekly Schedule

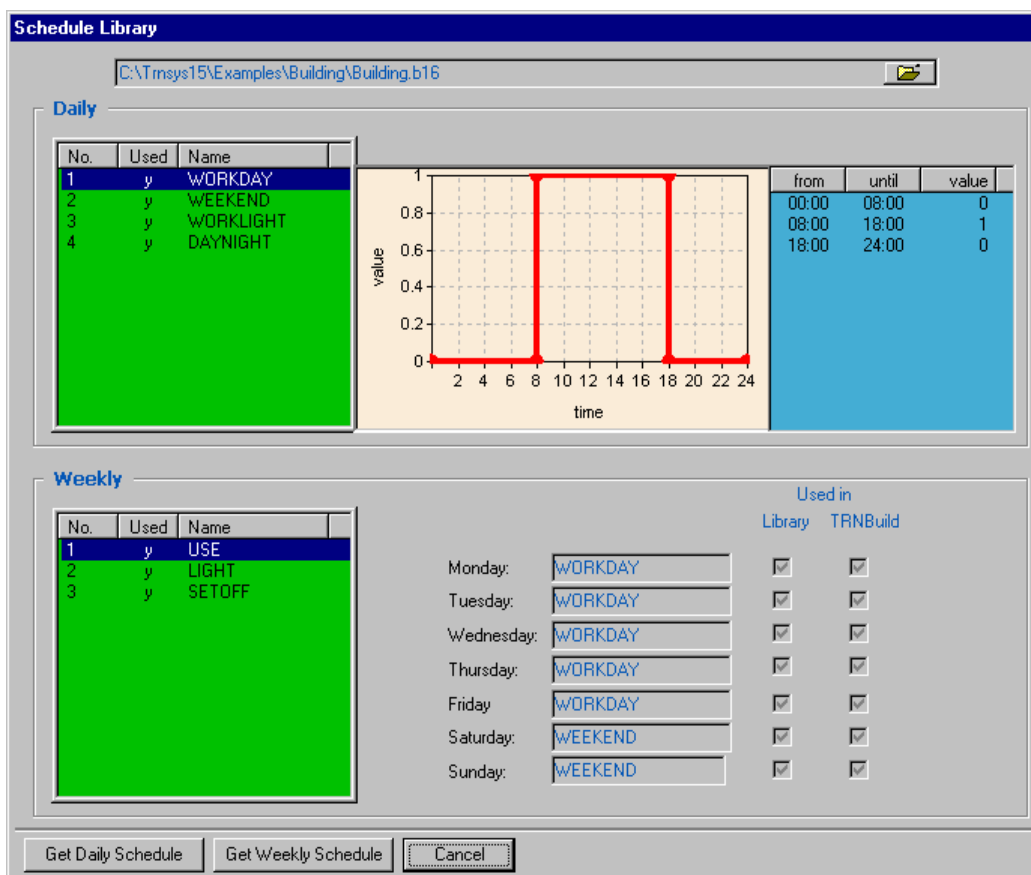


Figure 6.2.2-9: Getting a SCHEDULE for a Library

To take advantage of earlier defined schedules in libraries or any existing bui-file, schedule information can be accessed through the path and file name dialog box at the window top.

6.2.3. The Project Initialization Window

In the project initialization window, the user enters some general information about the project, defines the ORIENTATIONS of walls required by the described building, defines some basic material properties, views the list of required INPUTS to TYPE 56, and selects the desired OUTPUTS of TYPE 56.

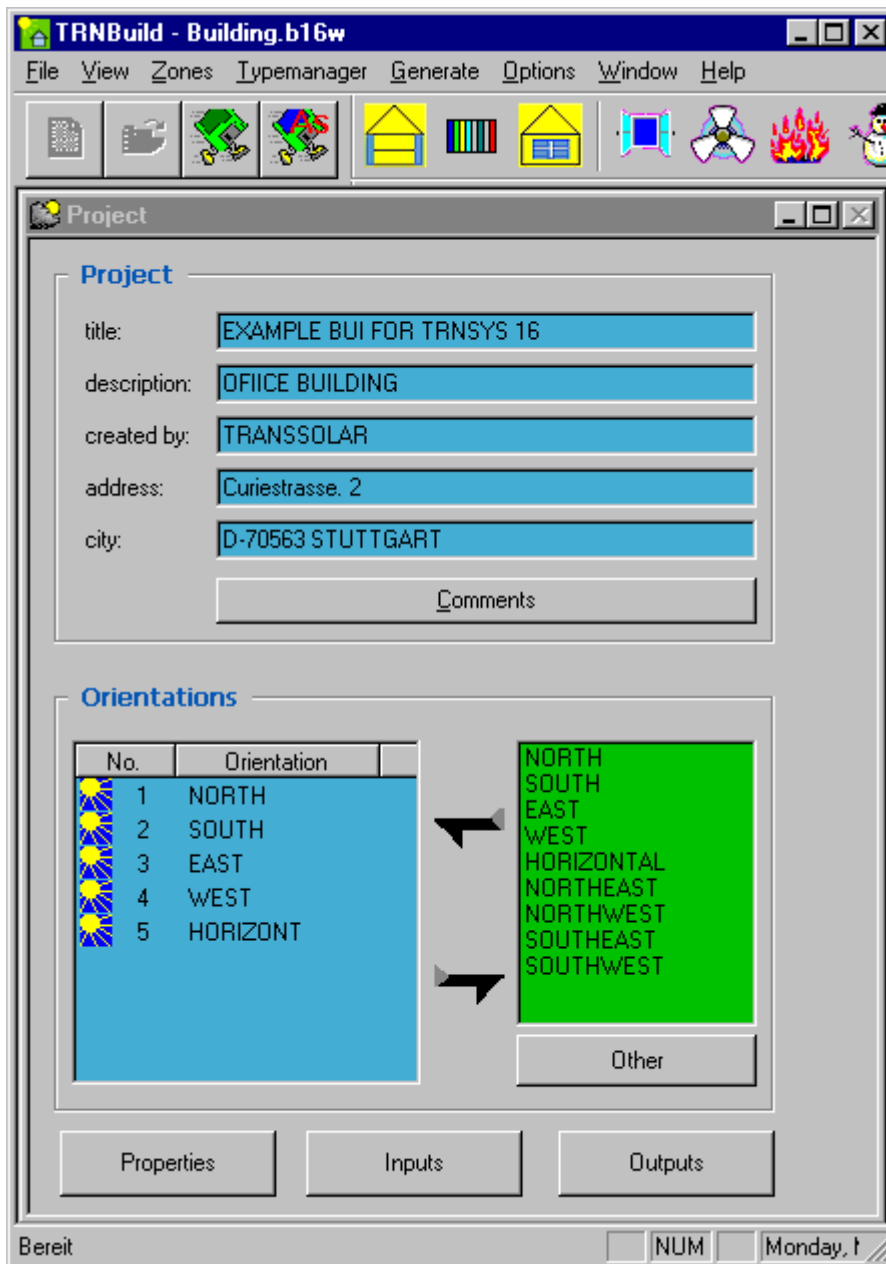


Figure 6.2.3-1: The PROJECT INITIALIZATION window

6.2.3.1. Orientation

All possible orientations of external building walls must be defined here by unique names. The table on the left side contains all orientations defined for this project. The table on the right side provides a list of standard orientations. For adding select the desired standard orientation and click on the upper arrow. The selected orientation will then appear in the left window. If other orientations are needed, click on the button OTHER and a new orientation name can be entered. A defined orientation can be deleted after selection by clicking on the lower arrow.

For each orientation name specified, an input of incident radiation to the Type 56 TRNSYS component will be required. This is generally provided by the Type 16 Radiation Processor.

Note: The incident angle for an orientation must range from 0 to 180 degrees. Due to the fact that the zenith angle of Type 16 (Radiation processor) ranges only between 0 to 90 degrees, the orientation "horizontal" must be defined as a separate surface in Type 16 (Radiation processor.)

6.2.3.2. Properties

Figure 6.2.3-2 shows certain material properties. If the user does not define them, the following default values are used.

Properties	
density of air :	1.204 kg / m ³
specific heat of air:	1.012 kJ / kg K
heat of vaporization of water:	2454 kJ / kg
Stefan Boltzmann Constant:	2.041e-007 kJ / h m ² K ⁴
approx. average surface temp.:	293.15 K
Parameters for internal calculation of heat transfer coefficients	
constant heated floor, if (T _{surf} floor-T _{air} floor) > 0	7.2 kJ / m ² K
exponent heated floor, if (T _{surf} floor-T _{air} floor) > 0	0.31 -
constant cooled floor, if (T _{surf} floor-T _{air} floor) < 0	3.888 kJ / m ² K
exponent cooled floor, if (T _{surf} floor-T _{air} floor) < 0	0.31 -
constant cooled ceiling, if (T _{surf} ceiling-T _{air} ceiling) < 0	3.888 kJ / m ² K
exponent cooled ceiling, if (T _{surf} ceiling-T _{air} ceiling) < 0	0.31 -
constant heated ceiling, if (T _{surf} ceiling-T _{air} ceiling) > 0	7.2 kJ / m ² K
exponent heated ceiling, if (T _{surf} ceiling-T _{air} ceiling) > 0	0.31 -
constant vertical surface:	5.76 kJ / m ² K
exponent vertical surface	0.3 -

OK Cancel

Figure 6.2.3-2: The PROPERTIES window

Heat transfer coefficients depend heavily on the temperature difference between surface and fluid and the direction of heat flow. For example, the flow pattern evolved by a chilled ceiling is similar to that of a heated floor, but completely different to that of a vertical surface. The mathematical formula used for appropriate heat transfer coefficients is of the form

$\alpha_{conv} = \text{const} (T_{surf} - T_{air})^{exp}$. The coefficients const and exp can be changed here to fit different approaches from heat transfer research. Standard values are taken from literature; see TRNSYS Manual, description of TYPE 80 for further details.

Important note: the automatic calculation of heat transfer coefficients has to be activated during the description of a new wall (see Section 6.2.4.2) or in the WALL TYPE MANAGER and is applied only to these explicitly defined walls. In case of heated or chilled building surface, the heat transfer will depend on surface temperature, so automatic heat transfer coefficient calculation is strongly recommended.

For all other walls the standard approach will still be a constant heat transfer coefficient resulting in reduced calculation time.

6.2.3.3. Inputs

By clicking on the INPUTS button, an overview of INPUTs defined within the project is shown (see Figure 6.2.3-3).

New INPUTs can be added here, thus creating a list you want to use in the definition of gains, controller strategies etc. Unused INPUTs can be deleted, but it is not possible to delete INPUTs used in definitions somewhere in the building description.

In certain cases it might be comfortable to change the sequence of INPUTs, which is done by drag and drop (left mouse button).

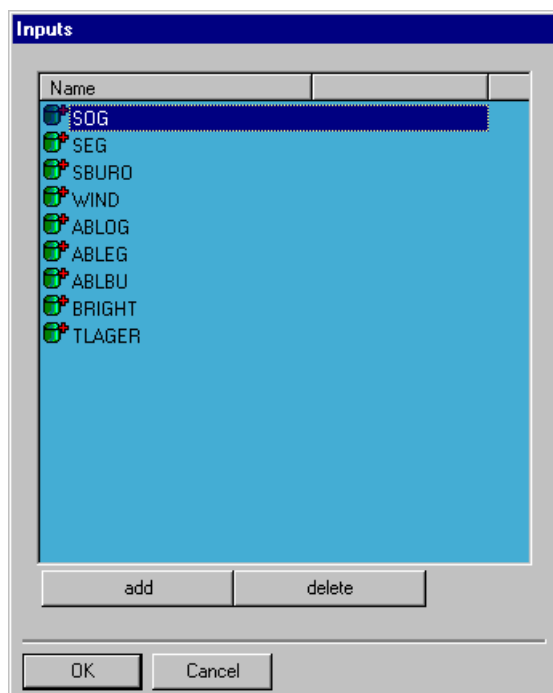


Figure 6.2.3-3: The INPUTS window

6.2.3.4. Outputs

By clicking on the OUTPUTS button, the OUTPUTS window opens as shown in Figure 6.2.3-4. In General, the definition of OUTPUTS is last step of the building description. The user may adjust the time base of the transfer function if necessary. The default value of 1 is adequate for most cases. For heavy constructions 2 up to 4 can be used, 0.5 for light walls. Caution: the start time in the TRNSYS input file (*.DCK) must match to the time base.

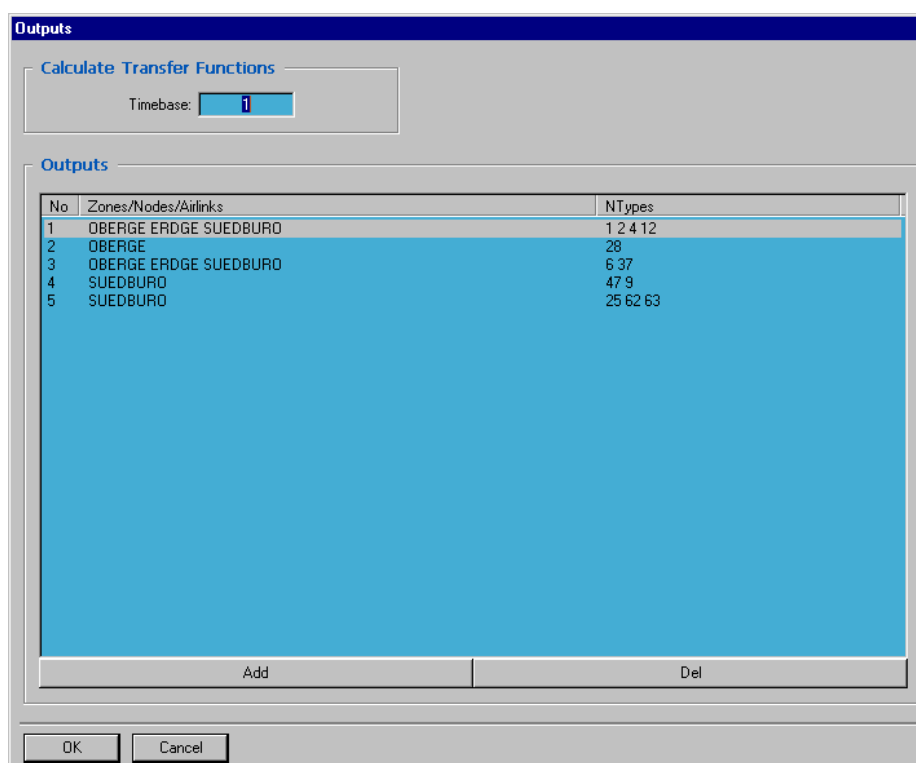


Figure 6.2.3-4: The OUTPUT window

In addition, the user can edit, add or delete outputs of TYPE 56, so-called NTYPES. Default outputs are provided (zone air temperatures (NTYPE 1) and sensible energy demands (NTYPE2) for all specified zones). To add a new output, the user clicks on the ADD button and the NEW OUTPUT window opens (see Figure 6.2.3-5). The user now has the following options:

- to use a DEFAULT setting (zone air temperatures (NTYPE 1) and sensible energy demands (NTYPE 2) for all specified zones)
- to select a single or a group of zones use the arrows between the upper boxes or the insert and delete key, respectively. Selected items appear in the left boxes. For specifying the desired outputs use the lower list boxes. Selected outputs appear in the lower left box, simultaneously selected items appear in a different color in the lower right box to prevent from multiple selection.
- in addition to outputs for single or multi-zones, balances are available for thermal energy, solar radiation and humidity, extremely helpful for consistency check of thermal modeling.
- If a so-called surface output is selected, the desired surfaces must be defined by a double click on the NTYPE of the left box. Afterwards, the selected surface numbers are displayed too. Also, for NTYPE 28 (values of schedules) the schedules must specified by a double click on the NTYPE of the left box.

For editing a previously defined OUTPUT, double-click on the output in the lower left Ntypes overview window.

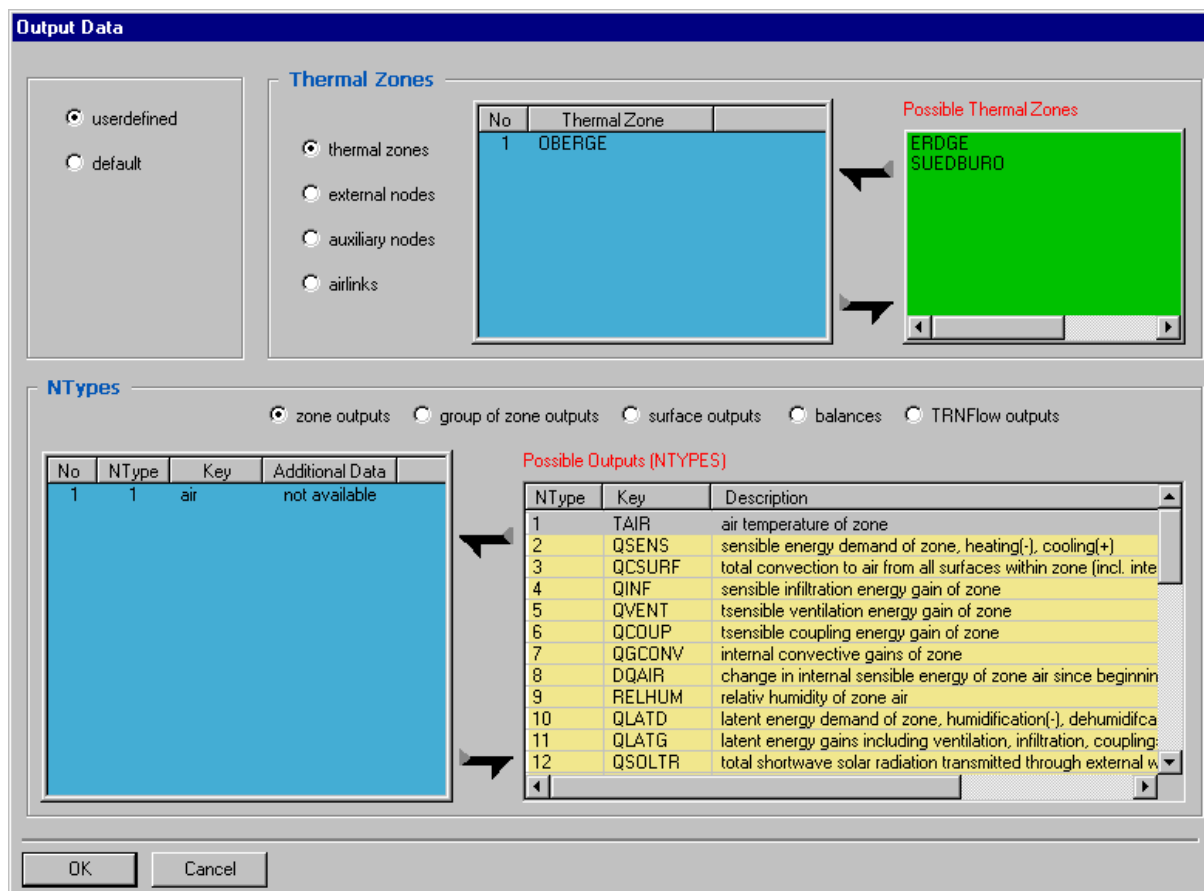


Figure 6.2.3-5: Adding a new user-defined OUTPUT

6.2.4. The Zone Window

The ZONE window contains all information describing a thermal zone of the building as shown in Figure 6.2.4-1. To add a new zone, click on ADD ZONE under the ZONES menu. An active zone can be deleted by clicking on DELETE ACTIVE ZONE under the ZONES menu.

In addition, you may use the zone manager by a right mouse click. For opening an existing zone click on the zone name within the zone manager. Since Windows 95 and 98 have a bug concerning the resource handling you may have up to 6 zone windows open at once only.

6.3. *Type56 User's Manual*

6.3.1. *TYPE 56: MULTI-ZONE BUILDING - General Description*

This component models the thermal behavior of a building divided into different thermal zones. In order to use this component, a separate pre-processing program must first be executed. The TRNBUILD program reads in and processes a file containing the building description and generates two files that will be used by the TYPE 56 component during a TRNSYS simulation. The file containing the building description processed by TRNBUILD can be generated by the user with any text editor or with the interactive program TRNBUILD. The required notation is described fully in the TRNBUILD documentation in the following section. [3 parameters are required by the TYPE 56 component.](#) The first parameter is the FORTRAN logical unit for the data file with the building data (*.BUI). Since Version 16 this file contains also the thermal and optical data for all windows used in a project. The Data is taken from the Program Window library This ASCII file is distributed with TYPE 56 and may be extended using the DOE2 Output of WINDOW 4.1 or 5 program developed by Lawrence Berkeley Laboratory, USA. The second parameter is set to 1 if time-dependent convective heat transfer coefficients are used (for example in combination with floor panel heating systems). The third parameter gives the weighting factor between air and mean surface temperature for the calculation of an operative room temperature. The optional parameters 4 – 6 give FORTRAN logical unit numbers for the standard output files generated by TYPE 56. The inputs and outputs of TYPE 56 depend upon the building description and options within the TRNBUILD program. TRNBUILD generates an information file describing the outputs and required inputs of TYPE 56.

There are two ways to model the equipment for heating, cooling, humidification, and dehumidification. The two methods are similar to the "energy rate" and "temperature level" control modes available in the TYPE 12 and 19 load models. With the "energy rate" method, a simplified model of the air conditioning equipment is implemented within the TYPE 56 component. The user specifies the set temperatures for heating and cooling, set points for humidity control, and maximum cooling and heating rates. These specifications can be different for each zone of the building. If the user desires a more detailed model of the heating and cooling equipment, a "temperature level" approach is required. In this case, separate components are required to model the heating and/or cooling equipment. The outputs from the TYPE 56 zones can be used as inputs to the equipment models, which in turn produce heating and cooling inputs to the TYPE 56 zones.

Note: Only one unit of TYPE 56 is allowed per simulation.
--

6.3.2. *General Mathematical Description of the Thermal Model*

The general case, which does not include the simplified model of the heating and cooling equipment, is presented first. If separate equipment components are used, they can be coupled to the zones as either internal convective gains or ventilation gains. Following this, the simplified method of providing heating and cooling equipment within the TYPE 56 component is described.

Another section will cover the use of a simulation timestep that is not equal to the timebase on which the wall transfer function relationships are based. Finally, descriptions of the optical and thermal window model, the way in which solar and internal radiation are distributed within each zone, the moisture balance calculations and the integrated model for thermo-active walls are given.

6.3.2.1. Thermal Zone

The building model in TYPE 56 is a non-geometrical balance model with one air node per zone, representing the thermal capacity of the zone air volume and capacities which are closely connected with the air node (furniture, for example). Thus the node capacity is a separate input in addition to the zone volume.

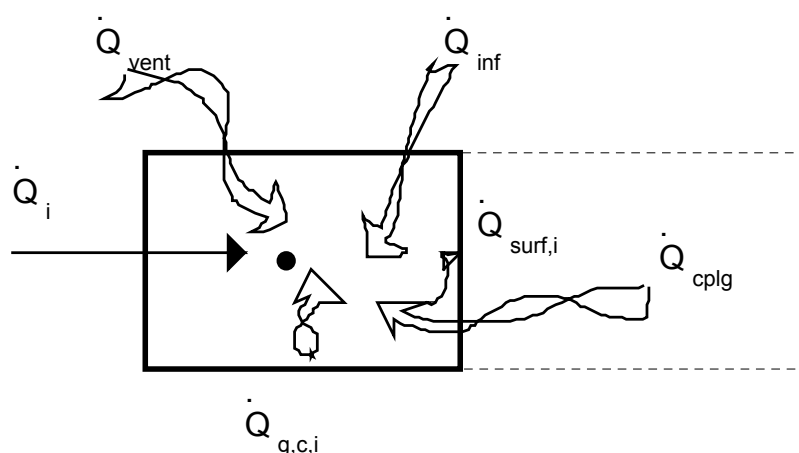


Figure 6.3.2-1: Heat balance on the zone air node

6.3.2.1.1. CONVECTIVE HEAT FLUX TO THE AIR NODE

$$\dot{Q}_i = \dot{Q}_{surf,i} + \dot{Q}_{inf,i} + \dot{Q}_{vent} + \dot{Q}_{g,c,i} + \dot{Q}_{cplg,i} \quad \text{Eq. 6.3.2-1}$$

where $\dot{Q}_{surf,i}$ is the infiltration gains (air flow from outside only), given by

$$\dot{Q}_{inf,i} = V \cdot \rho \cdot c_p \cdot (T_{outside} - T_{air}) \quad \text{Eq. 6.3.2-2}$$

$\dot{Q}_{vent,i}$ is the ventilation gains (air flow from a user-defined source, like an HVAC system, given by

$$\dot{Q}_{vent,i} = \dot{V} \cdot \rho \cdot c_p \cdot (T_{ventilation,i} - T_{air}) \quad \text{Eq. 6.3.2-3}$$

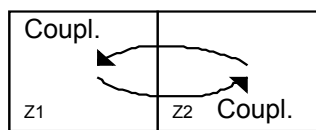
$\dot{Q}_{g,c,i}$ is the internal convective gains (by people, equipment, illumination, radiators, etc.), and

$\dot{Q}_{cplg,i}$ is the gains due to (connective) air flow from zone I or boundary condition, given by

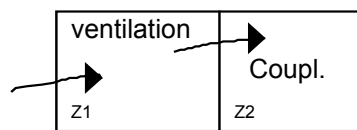
$$\dot{Q}_{cplg,i} = \dot{V} \cdot \rho \cdot c_p \cdot (T_{zone,i} - T_{air}) \quad \text{Eq. 6.3.2-4}$$

6.3.2.1.2. COUPLING

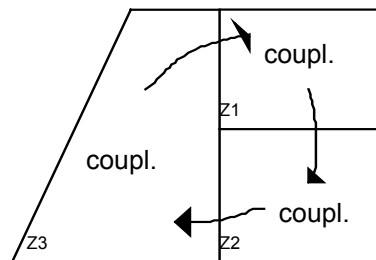
The coupling statement allows the definition an air mass flow a zone receives from another zone, considered as a heat flow from or to the air node. The statement does not automatically define the air flow back to the adjacent zone as would occur in an interzonal air exchange. To consider this return flow, the corresponding coupling must be defined in the adjacent zone to receive the same air flow in return. The reason for this convention is to allow the user to describe cross ventilation or a ventilation circle within 3 or more zones (e.g., thermosyphon through a 2 story winter-garden,).



interzonal airchange



cross ventilation



Ventilation circle

Note: There is no air balance check in TYPE 56. The user can empty or overload a zone by couplings. Be sure that the specified air flows into a zone by coupling, ventilation, and infiltration are physically meaningful.

6.3.2.1.3. RADIATIVE HEAT FLOWS (ONLY) TO THE WALLS AND WINDOWS

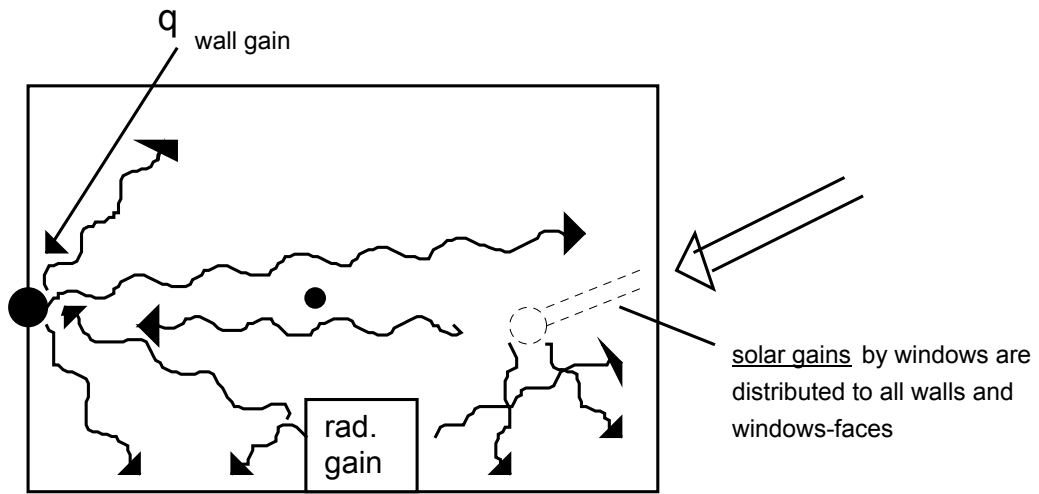


Figure 6.3.2-2: Radiative energy flows considering one wall with its surface temperature node.

$$\dot{Q}_{r,w_i} = \dot{Q}_{g,r,i,w_i} + \dot{Q}_{sol,w_i} + \dot{Q}_{long,w_i} + \dot{Q}_{wall-gain} \quad \text{Eq. 6.3.2-5}$$

where \dot{Q}_{r,w_i} is the radiative gains for the wall surface temperature node, \dot{Q}_{g,r,i,w_i} is the radiative zone internal gains received by wall, \dot{Q}_{sol,w_i} is the solar gains through zone windows received by walls, \dot{Q}_{long,w_i} is the longwave radiation exchange between this wall and all other walls and windows ($\epsilon_i = 1$), and $\dot{Q}_{wall-gain}$ is the user-specified heat flow to the wall or window surface. All of these quantities are given in kJ/h.

6.3.2.1.4. INTEGRATION OF WALLS AND WINDOWS

Figure 6.3.2-3 shows the heat fluxes and temperatures that characterize the thermal behavior of any wall or window. The nomenclature used in this figure is defined as follows:

- $S_{s,i}$ Radiation heat flux absorbed at the inside surface (solar and radiative gains)
- $S_{s,o}$ Radiation heat flux absorbed at the outside surface (solar gains)

$\dot{q}_{r,s,i}$	Net radiative heat transfer with all other surfaces within the zone
$\dot{q}_{r,s,o}$	Net radiative heat transfer with all surfaces in view of the outside surface
$\dot{q}_{w,g,i}$	User defined heat flux to the wall or window surface
$\dot{q}_{s,i}$	Conduction heat flux from the wall at the inside surface
$\dot{q}_{s,o}$	Into the wall at the outside surface
$\dot{q}_{c,s,i}$	Convection heat flux from the inside surface to the zone air
$\dot{q}_{c,s,o}$	Convection heat flux to the outside surface from the boundary/ambient
$T_{s,i}$	Inside surface temperature
$T_{s,o}$	Outside surface temperature

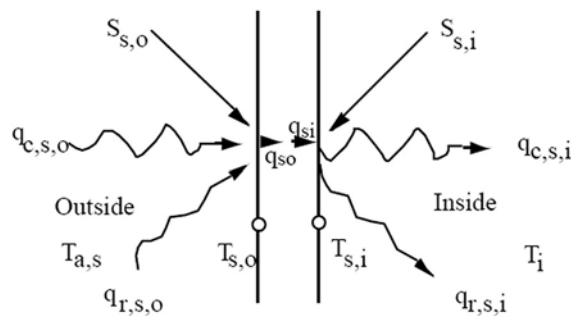


Figure 6.3.2-3: Surface Heat Fluxes and Temperatures

The walls are modeled according to the transfer function relationships of Mitalas and Arseneault [1,2,6] defined from surface to surface. For any wall, the heat conduction at the surfaces are:

$$\dot{q}_{s,i} = \sum_{k=0}^{n_{b_s}} b_s^k T_{s,o}^k - \sum_{k=0}^{n_{c_s}} c_s^k T_{s,i}^k - \sum_{k=1}^{n_{d_s}} d_s^k \dot{q}_{s,i}^k \quad \text{Eq. 6.3.2-6}$$

$$\dot{q}_{s,o} = \sum_{k=0}^{n_{a_s}} a_s^k T_{s,o}^k - \sum_{k=0}^{n_{b_s}} b_s^k T_{s,i}^k - \sum_{k=1}^{n_{d_s}} d_s^k \dot{q}_{s,o}^k \quad \text{Eq. 6.3.2-7}$$

These time series equations in terms of surface temperatures and heat fluxes are evaluated at equal time intervals. The superscript k refers to the term in the time series. The current time is

$k=0$, the previous time is for $k=1$, etc. The time-base on which these calculations are based is specified by the user within the TRNBUILD description. The coefficients of the time series (a's, b's, c's, and d's) are determined within the TRNBUILD program using the z-transfer function routines of reference [2].

A window is thermally considered as an external wall with no thermal mass, partially transparent to solar, but opaque to long-wave internal gains. Long-wave absorption is considered to occur only at the surfaces. In the energy balance calculation of the TYPE 56, the window is described as a 2-node model shown in Figure 6.3.2-4. The detailed optical and thermal window model is described in Section 6.3.2.2. Eq. 6.3.2-6 to Eq. 6.3.2-17 are valid for a window with:

$$a_s^o = b_s^o = c_s^o = d_s^o = U_{g,s}$$

$$a_s^k = b_s^k = c_s^k = d_s^k = 0 \text{ for } k > 0.$$

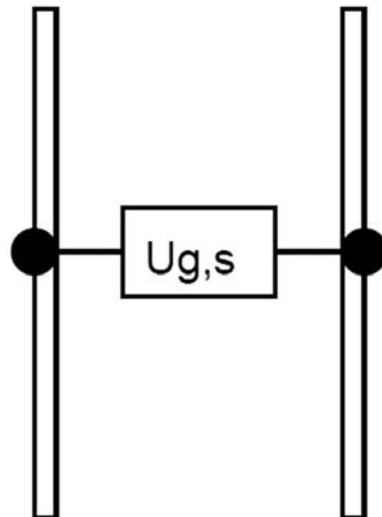


Figure 6.3.2-4: Two-node window model used in th TYPE56 energy balance equation.

6.3.2.1.5. *TRANSFER FUNCTION METHOD BY MITALAS*

The method of the transfer function or response factors can be described as the method to tell the "thermal history" of the wall. The wall is considered as a black box. The number of time-steps (k) related to the time-base (defined by the user) shows whether the wall is a heavy wall with a high thermal mass ($k \leq 20$) or if only a few time-steps have to be considered to describe the thermal behavior of this wall. If the time-base of the considered wall is higher than the time-constant, the calculation of the Transfer-function matrix coefficients is stopped. Therefore such a "thin" wall can be replaced by a resistance definition neglecting the thermal mass. As an example, the Figure 6.3.2-5 shows the different material layers of a wall. The wall example consists of three layers with concrete, mineral wool and gypsum from outside to inside.

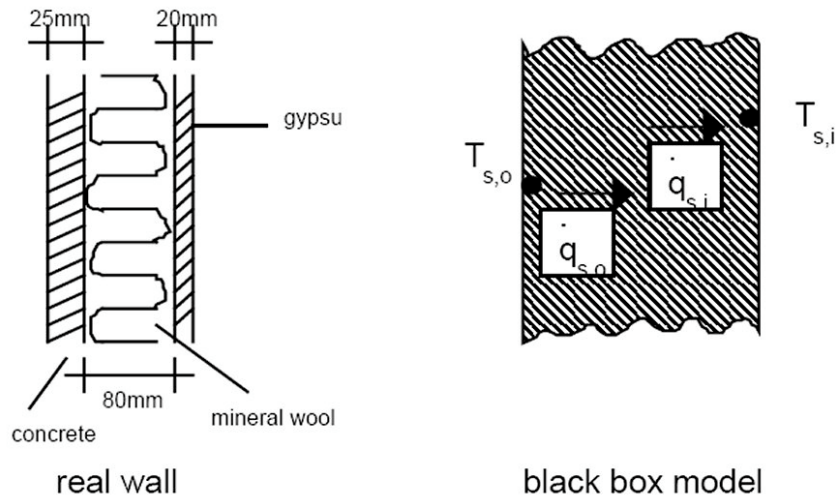


Figure 6.3.2-5: Real wall and black box model of the wall

Material data	Thickness [m]	Conductivity [kJ/h·m·K]	Capacity [kJ/kg·K]	Density [kg/m ³]
Concrete	0.025	7.56	1.0	2400
Mineral Wool	0.08	0.144	1.0	40
Gypsum	0.02	2.52	1.0	1400

Using the transfer function method, the TRNBUILD-program calculates the transfer function coefficients, listed below for the example wall.

----- WALL TYPE EXAMPLE -----

THERMAL CONDUCTANCE, U= 1.76429 kJ/h m²K; U-value= 0.45239 W/m²K

TRANSFERFUNCTION COEFFICIENTS

K	A	B	C	D
0	3.0402072E+01	8.6597596E-01	6.2473097E+01	1.0000000E+00
1	-2.8791436E+01	8.7958309E-01	-6.1044043E+01	-5.5725114E-03
2	1.4382785E-01	8.9032318E-03	3.2541274E-01	1.0083948E-07
3	-1.0589132E-06	4.0042651E-07	-4.7183532E-06	
SUM	1.7544627E+00	1.7544627E+00	1.7544627E+00	9.9442759E-01

Figure 6.3.2-6: Transfer function equation-system (Mitalas)

For the test wall, the coefficient table looks like that above. In addition to the transfer function coefficients, the listing contains a calculation of the heat conduction value U of the wall construction and the total heat transfer coefficient k considering a constant combined (convective+radiative) heat transfer (α_i, α_o) for the inside and outside surface.

Note: these combined heat transfer coefficients are not used during the simulation. For the calculation of the U-values stated in the information file (*.INF), the following combined heat transfer coefficients are used:

$$\frac{1}{\alpha_i} = 0.13 \left[W/m^2 K \right]^{-1}$$

$$\frac{1}{\alpha_o} = 0.04 \left[W/m^2 K \right]^{-1}$$

A dynamic simulation considering the thermal mass of a wall element does not use these values explicitly, but is describing the thermal resistance implicit in transient heat flows into and out of the wall surfaces. The latest validation report for the transfer function method and the whole TYPE 56 building model was prepared by Peter Voit [7] using detailed measurements from the CEC research program PASSYS.

6.3.2.1.6. THE LONG-WAVE RADIATION

The long-wave radiation exchange between the surfaces within the zone and the convective heat flux from the inside surfaces to the zone air are approximated using the star network given by Seem [3] and represented in Figure 6.3.2-7. This method uses an artificial temperature node (T_{star}) to consider the parallel energy flow from a wall surface by convection to the air node and by radiation to other wall and window elements. Comparisons to the detailed building model JOULOTTA from the University of Lund, Sweden done by S. Holst, ZAE Munich [4], show a good agreement for the surface temperatures. A single node model using a combined convective and radiative heat transfer coefficient shows much higher differences (IEA Task 13 report,[7]).

$$R_{STAR,i} = f(\alpha_i, A_{Surf,i}) = \frac{1}{Q_{surf,i}} (T_{Star} - T_i) \quad \text{Eq. 6.3.2-8}$$

Methods to calculate the resistances $R_{equiv,i}$ and $R_{star,i}$ can be found in reference [3]. Area ratios are used in these calculations to find the absorption factors between all surfaces. The star temperature can be used to calculate a net radiative and convective heat flux from the inside wall surface:

$$\dot{q}_{comb,s,i} = \dot{q}_{c,s,i} + \dot{q}_{r,s,i} \quad \text{Eq. 6.3.2-9}$$

then,

$$\dot{q}_{comb,s,i} = \frac{1}{R_{equiv,i} A_{s,i}} (T_{s,i} - T_{star}) \quad \text{Eq. 6.3.2-10}$$

where $\dot{q}_{comb,s,i}$ is the combined convective and radiative heat flux, and $A_{s,i}$ is the inside surface area.

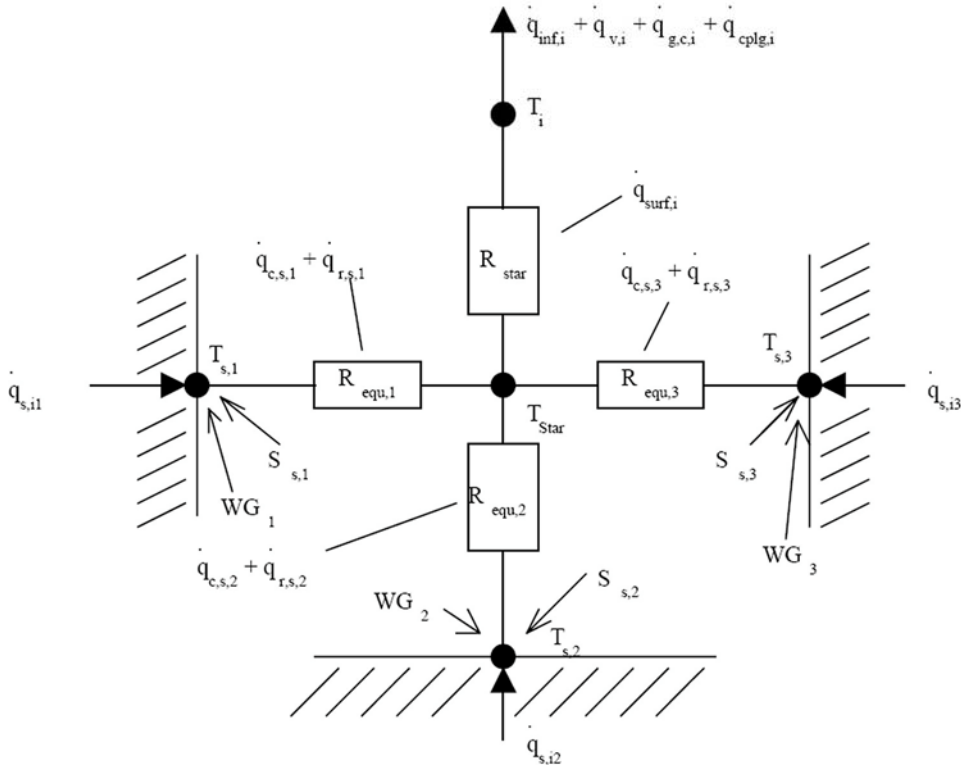


Figure 6.3.2-7: Star network for a zone with three surfaces.

For external surfaces the long-wave radiation exchange at the outside surface is considered explicitly using a fictive sky temperature, T_{sky} , which is an input to the TYPE 56 model and a view factor to the sky, f_{sky} , for each external surface. The total heat transfer is given as the sum of convective and radiative heat transfer:

$$\dot{q}_{comb,s,o} = \dot{q}_{c,s,o} + \dot{q}_{r,s,o} \quad \text{Eq. 6.3.2-11}$$

with

$$\dot{q}_{c,s,o} = h_{conv,s,o} (T_{a,s} - T_{s,o}) \quad \text{Eq. 6.3.2-12}$$

$$\dot{q}_{r,s,o} = \sigma \varepsilon_{s,o} (T_{s,o}^4 - T_{fsky}^4) \quad \text{Eq. 6.3.2-13}$$

$$T_{fsky} = (1 - f_{sky}) T_{a,s} + f_{sky} T_{sky} \quad \text{Eq. 6.3.2-14}$$

where

$\dot{q}_{comb,s,o}$	Combined convective and radiative heat flux to the surface
$\dot{q}_{c,s,o}$	Convective heat flux to the surface
$\dot{q}_{r,s,o}$	Radiative heat flux to the surface
$h_{conv,s,o}$	Convective heat transfer coefficient at the outside surface
f_{sky}	Fraction of the sky seen by the outside surface ¹
T_{sky}	Fictive sky temperature used for long-wave radiation exchange
$\varepsilon_{s,o}$	Long-wave emissivity of outside surface ($\varepsilon = 0.9$ for walls; value read from window library for windows)
σ	Stephan-Boltzmann constant

Energy balances at the surfaces give:

$$\dot{q}_{s,i} = \dot{q}_{comb,s,i} + S_{s,i} + Wallgain \quad \text{Eq. 6.3.2-15}$$

$$\dot{q}_{s,o} = \dot{q}_{comb,s,o} + S_{s,o} \quad \text{Eq. 6.3.2-16}$$

For internal surfaces $S_{s,i}$ can include both solar radiative and long-wave radiation generated from internal objects such as people or furniture.

Wallgain is a user-defined energy flow to the inside wall or window surfaces. It can describe solar gains changing during the day due to different sun positions or might be used as a simple way to model a floor heating or a chilled ceiling system. For external surfaces, $S_{s,o}$ consists of solar radiation only.

6.3.2.1.7. EXTERNAL WALLS

It is possible to combine Eq. 6.3.2-7 to Eq. 6.3.2-16 to express the inside surface heat flux for an external wall as a function of the boundary air temperatures:

¹ For a vertical wall with no buildings nearby, a reasonable value for f_{sky} is 0.5. If there are buildings in front of the wall obstructing the view of the sky, the value for f_{sky} would be lower than 0.5. For a horizontal roof with only the sky in view, f_{sky} would be 1.0.

$$\dot{q}_{s,i} = B_s T_{a,s} - C_s T_{a,s} + D_s \quad \text{Eq. 6.3.2-17}$$

where

$$B_s = \frac{e_s h_{s,o}}{(1 - f_s)} \quad \text{Eq. 6.3.2-18}$$

$$C_s = \frac{f_s}{(f_s - 1)} \left(\frac{1}{R_{equiv,i} A_{s,i}} \right) \quad \text{Eq. 6.3.2-19}$$

$$D_s = \frac{f_s S_{s,i} + e_s (S_{s,o} - k_{s,o}) + K_{s,i}}{(1 - f_s)} \quad \text{Eq. 6.3.2-20}$$

$$e_s = \frac{b_s^o}{a_s^o + h_{s,o}} \quad \text{Eq. 6.3.2-21}$$

$$f_s = (b_s^o e_s - c_s^o) R_{equiv,i} A_{s,i} \quad \text{Eq. 6.3.2-22}$$

The values for $K_{s,i}$ and $K_{s,o}$ are defined by the transfer function equations:

$$K_{s,i} = \sum_{k=0}^{n_{b_s}} b_s^k T_{s,o}^k - \sum_{k=0}^{n_{c_s}} c_s^k T_{s,i}^k - \sum_{k=1}^{n_{d_s}} d_s^k \dot{q}_{s,i}^k \quad \text{Eq. 6.3.2-23}$$

$$K_{s,o} = \sum_{k=0}^{n_{a_s}} a_s^k T_{s,o}^k - \sum_{k=0}^{n_{b_s}} b_s^k T_{s,i}^k - \sum_{k=1}^{n_{d_s}} d_s^k \dot{q}_{s,o}^k \quad \text{Eq. 6.3.2-24}$$

6.3.2.1.8. WALLS WITH BOUNDARY CONDITIONS

Eq. 6.3.2-17 also applies for a wall with a known boundary temperature, $T_{b,s}$, with $T_{b,s}$ substituted for $T_{a,s}$.

6.3.2.1.9. ADJACENT, INTERNAL WALLS AND WALLS WITH IDENTICAL BOUNDARY CONDITIONS

For walls adjacent to another zone, internal walls, or walls adjacent zones with identical conditions, Eq. 6.3.2-17 applies, but with:

$$\text{Adjacent zone:} \quad T_{a,s} = T_{star,j}$$

$$\text{Internal wall:} \quad T_{a,s} = T_{star,i}$$

$$\text{Adjacent identical} \quad T_{a,s} = T_{star,i}$$

and

$$B_s = \frac{e_s}{(1-f_s)} \left(\frac{1}{R_{equiv,j} A_{s,j}} \right) \quad \text{Eq. 6.3.2-25}$$

$$e_s = \frac{b_s^o}{a_s^o + \frac{1}{R_{equiv,j} A_{s,j}}} \quad \text{Eq. 6.3.2-26}$$

Note: For an internal wall, both sides must be considered for the area A_s .

It is also possible to specify a boundary condition for the outside surface temperature rather than an air temperature by setting $HBACK \leq 0.001$. In this case, $T_{a,s} = T_{s,o} = T_{b,s}$. Eq. 6.3.2-17 applies, but with:

$$B_s = \frac{b_s^o}{1 + c_s^o R_{equiv,i} A_{s,i}} \quad \text{Eq. 6.3.2-27}$$

$$C_s = \frac{c_s^o}{1 + c_s^o R_{equiv,i} A_{s,i}} \quad \text{Eq. 6.3.2-28}$$

$$D_s = \frac{K_{s,i} + c_s^o R_{equiv,i} A_{s,i} S_{s,i}}{1 + c_s^o R_{equiv,i} A_{s,i}} \quad \text{Eq. 6.3.2-29}$$

6.3.2.1.10. TOTAL GAINS FROM SURFACES IN A ZONE

The total gain to zone i from all surfaces is the sum of the combined heat transfers, or from Eq. 6.3.2-16 and Eq. 6.3.2-17, is:

$$\begin{aligned} \dot{Q}_{surf,i} = \sum A_s \dot{q}_{comb,i} = & \sum_{j=1}^{Adj.Zones} \sum_{i=1}^{surface\ to\ j} A_s B_s T_{star,j} + \sum_{ext.surfaces} A_s B_s T_a + \sum_{int.walls} A_s B_s T_{star} \\ & + \sum_{known\ bound} A_s B_s T_{b,s} - \sum_{surface\ in\ zone\ i} A_s (C_s T_{star,i} - D_s - S_{s,i}) \end{aligned} \quad \text{Eq. 6.3.2-30}$$

where A_s is the inside area of surface s.

Both sides of an internal wall are considered as inside surfaces and must be included twice in Eq. 6.3.2-30.

An energy balance on the star node in Figure 6.3.2-7 also shows that:

$$\dot{Q}_{surf,i} = \frac{1}{R_{star,i}} (T_{star,i} - T_i) \quad \text{Eq. 6.3.2-31}$$

6.3.2.1.11. INFILTRATION, VENTILATION, AND CONVECTIVE COUPLING

Infiltration and ventilation rates are given in terms of air changes per hour for each zone. The mass flow rate is the product of the zone air volume, air density, and air change rate. Infiltration occurs always from outdoor conditions, while ventilation occurs from a specified (possibly variable) temperature. Equal amounts of air are assumed to leave the zone at the zone temperature. The energy gains to any zone i due to infiltration and ventilation are:

$$\dot{Q}_{inf,i} = \dot{m}_{inf,i} C_p (T_a - T_i) \quad \text{Eq. 6.3.2-32}$$

$$\dot{Q}_{v,i} = \sum_k^{nvent} \dot{m}_{v,k,i} C_p (T_{v,k} - T_i) \quad \text{Eq. 6.3.2-33}$$

where

- $\dot{m}_{inf,i}$ mass flow rate of infiltration air
- $\dot{m}_{v,k,i}$ mass flow rate of ventilation air of ventilation type k
- C_p specific heat of the air
- $T_{v,k}$ temperature of ventilation air of ventilation type k
- T_a ambient air temperature

For each wall or window separating zones of floating temperature or each wall having a known boundary condition, it is possible to specify a convective coupling. This coupling is the mass flow rate that enters the zone across the surface. An equal quantity of air is assumed to leave the zone at the zone temperature. The energy gain due to the convective coupling is the sum of all such gains for all walls or windows in the zone.

$$\dot{Q}_{cplg,i} = \sum_{adj.zones} \sum_{surfaces\ s\ to\ j} \dot{m}_{cplg,s} C_p (T_j - T_i) + \dots + \sum_{knownbound} \dot{m}_{cplg,s} C_p (T_{b,s} - T_i) \quad \text{Eq. 6.3.2-34}$$

where $\dot{m}_{cplg,s}$ is the mass flow rate of air entering zone i across walls or windows.

6.3.2.1.12. FLOATING ZONE TEMPERATURE (NO HEATING OR COOLING)

The rate of change of internal energy for any free floating zone is equal to the net heat gain or

$$C_i \frac{d}{dt} T_i = \dot{Q}_i \quad \text{Eq. 6.3.2-35}$$

where C_i is the thermal capacitance of zone i (minimal = $V_i \rho C_p$ with V_i = zone volume).

The net heat gain, \dot{Q}_i , is a function of T_i and the temperatures of all other zones adjacent to zone i .

Note: To simplify the solution of the set of equations, \dot{Q}_i is considered constant during any timestep, evaluated at average values of the zone temperatures. In this case, the solution to the differential equation for final temperature for a given time interval is

$$T_{i,\tau} = T_{i,\tau-\Delta t} + \frac{\overline{\dot{Q}_{i\Delta t}}}{C_i} \quad \text{Eq. 6.3.2-36}$$

where

Δt = The simulation time-step

$T_{i,\tau-\Delta t}$ = The zone temperature at the beginning of the time-step.

The temperature variation is linear, such that the average is:

$$T_i = \frac{T_{i,\tau} + T_{i,\tau-\Delta t}}{2} \quad \text{Eq. 6.3.2-37}$$

If Eq. 6.3.2-37 is solved for $T_{i,\tau}$ and the result substituted into Eq. 6.3.2-36, along with the individual expressions representing the net heat gain, the following is obtained:

$$\begin{aligned} \frac{2 \cdot C_i \cdot (\bar{T}_i - T_{i,\tau-\Delta t})}{\Delta t} &= \sum_{j=1}^{adjac. \ surfaces} \sum_{zones \ i \ to \ j} \dot{m}_{cplg,s} \cdot C_p \cdot \bar{T}_j + \dot{m}_{inf,i} \cdot C_p \cdot T_a + \sum_{known \ boundaries} \dot{m}_{cplg,i} \cdot C_p \cdot T_{b,s} \\ &- \left(\frac{1}{R_{star,i}} + \left(\sum_{known \ boundaries} \dot{m}_{cplg,i} + \sum_{j=1}^{adjac. \ surfaces} \sum_{zones \ i \ to \ j} \dot{m}_{cplg,s} + \dot{m}_{inf,i} + \sum_k^{nvent} \dot{m}_{v,k,i} \right) C_p \right) \bar{T}_i \\ &+ \left(\frac{1}{R_{star,i}} \cdot \bar{T}_{star,i} + \sum_k^{nvent} \dot{m}_{v,k,i} \cdot C_p \cdot T_{v,k} + Q_{g,c,i} \right) \end{aligned}$$

$$\text{Eq. 6.3.2-38}$$

Eq. 6.3.2-30 and Eq. 6.3.2-31 can be equated and regrouped to find:

$$\begin{aligned} & \left(\frac{1}{R_{star,i}} - \sum_{int.walls} A_s B_s + \sum_{surf.ini} A_s C_c \right) \bar{T}_{star,i} - \left(\sum_{adj.zone\ walls\ to\ j} \sum A_s B_s \right) \bar{T}_{star,j} - \frac{1}{R_{star,i}} \bar{T}_i \\ & = \left(\sum_{exterior\ surfaces} A_s B_s \right) T_a + \sum_{known\ boundaries} A_s B_s T_{b,s} + \sum_{surface\ in\ zone\ i} A_s (D_s + S_{s,i}) \end{aligned} \quad \text{Eq. 6.3.2-39}$$

The set of energy balances given by Eq. 6.3.2-38 and Eq. 6.3.2-39, written for all zones, results in a linear set of equations in average zone temperatures and average star temperatures. In matrix form,

$$[X][\bar{T}] = [Z] \quad \text{Eq. 6.3.2-40}$$

This matrix can be partitioned such that

$$[X] = \begin{bmatrix} X_{11} & X_{12} \\ X_{21} & X_{22} \end{bmatrix} \quad \text{Eq. 6.3.2-41}$$

$$[T] = \begin{bmatrix} \bar{T}_1 \\ \bar{T}_2 \end{bmatrix} = \begin{bmatrix} \bar{T} \\ \bar{T}_{star} \end{bmatrix} \quad \text{Eq. 6.3.2-42}$$

$$[Z] = \begin{bmatrix} Z_1 \\ Z_2 \end{bmatrix} \quad \text{Eq. 6.3.2-43}$$

where

$$X_{11,ii} = \left(\sum_{i\ to\ j}^{surfaces} \dot{m}_{cplg,s} + \dot{m}_{inf,i} + \sum_k^{nvent} \dot{m}_{v,k,i} \right) \cdot C_p + \frac{2C_i}{\Delta t} + \frac{1}{R_{star,i}} + \sum_{known\ boundaries} \dot{m}_{cplg,i} \cdot C_p \quad \text{Eq. 6.3.2-44}$$

$$X_{11,ij} = - \sum_{j=1}^{adjac.\ surfaces\ zones\ i\ to\ j} \sum \dot{m}_{cplg,s} \cdot C_p \quad \text{for } i \neq j \quad \text{Eq. 6.3.2-45}$$

$$X_{12,ii} = \frac{-1}{R_{star,i}} \quad \text{Eq. 6.3.2-46}$$

$$X_{12,ij} = 0 \quad \text{for } i \neq j \quad \text{Eq. 6.3.2-47}$$

$$X_{21,ii} = - \frac{1}{R_{star,i}} \quad \text{Eq. 6.3.2-48}$$

$$X_{21,ij} = 0 \quad \text{Eq. 6.3.2-49}$$

$$X_{22,ii} = - \sum_{\text{int.walls}} A_S \cdot B_S + \sum_{\text{surf in zone } i} A_S \cdot C_S + \frac{1}{R_{star,i}} \quad \text{Eq. 6.3.2-50}$$

$$X_{22,ij} = - \sum_{\text{adj. walls zones } i \text{ to } j} A_S B_S \quad \text{Eq. 6.3.2-51}$$

$$Z_{1,i} = \dot{m}_{inf,i} \cdot C_p \cdot T_a + \sum_{\text{known boundaries}} \dot{m}_{cp1g,s} \cdot C_p \cdot T_{b,s} + \sum_k^{nvent} \dot{m}_{v,k,i} \cdot C_p \cdot T_{v,k} + \frac{2C_i \cdot T_{i,\tau-\Delta t}}{\Delta t} + Q_{g,c,i} \quad \text{Eq. 6.3.2-52}$$

$$Z_{2,i} = \left(\sum_{\text{ext.surf.}} A_S B_S \right) T_a + \sum_{\text{known boundaries}} A_S B_S T_{b,s} + \sum_{\text{surf. in zone } i} A_S (D_s + S_{s,i}) \quad \text{Eq. 6.3.2-53}$$

For the case of all zones in floating temperature,

$$[\bar{T}] = [X]^{-1} [Z] \quad \text{Eq. 6.3.2-54}$$

The final temperature for each zone i is

$$T_{i,\tau} = 2\bar{T}_i - T_{i,\tau-\Delta t} \quad \text{Eq. 6.3.2-55}$$

Note: The mean temperatures during the time-step for each zone are output by TYPE 56. This follows the TRNSYS standard for providing results as mean values during the last time-step.

6.3.2.1.13. SIMPLIFIED HEATING AND COOLING

It is possible to determine the energy requirement for zones controlled in an idealized way. Therefore the heating and cooling energy flow is directly connected to the zone air temperature node. The output of the heating and/or cooling equipment is a function of the zone temperature as shown in Figure 6.3.2-8, where

P_i Power output for zone i (negative for heating, positive for cooling)

$P_{max,i}$ Absolute value of the maximum power for zone i

$T_{set,i}$ Set temperatures for heating or cooling in zone i

For the simulation of heating equipment that produces a partially radiative gain to the zone, the radiative fraction of the supplied heating power may be defined. This fraction of the power is supplied as internal radiative gains and distributed to the walls of the zone. As the set temperature for the heating equipment is related to the air temperature of the zone, *the radiative fraction of the heating power cannot be higher than 0.99* in order to have a convective part left to ensure stable control of the heating equipment.

The zone temperature is free floating in the comfort region where the power is zero. If the temperature of a free floating zone is within the heating or cooling regions at the end of a timestep, power is applied throughout the timestep so that the final zone temperature just reaches T_{set} . If the power required is greater than the maximum specified, then the maximum power is applied throughout the timestep and the zone temperature is again free floating.

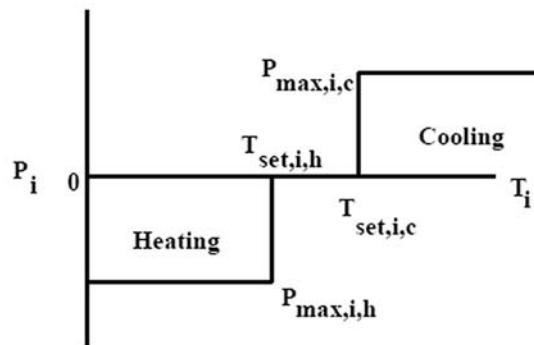


Figure 6.3.2-8: Power output versus temperature

The temperature change of the zone air, when power is supplied, is assumed to be linear. If power is required and enough is available to maintain the final zone temperature at $T_{set,i}$, then the final and average zone temperatures are known.

$$T_t = T_{set,i} \tag{Eq. 6.3.2-56}$$

$$T_{req,i} = \frac{T_{\tau-\Delta t} + T_{set,i}}{2} \tag{Eq. 6.3.2-57}$$

where, $T_{req,i}$ is the average zone temperature over the time-step if less than maximum power is required.

It is necessary to consider the general case of zones that are in different control regions. With the inclusion of the control laws, the equations remain linear. For the zones that have floating temperatures, the solution for average zone temperatures and star temperatures is again of the form

$$[\bar{T}] = [X']^{-1} [Z'] \tag{Eq. 6.3.2-58}$$

The coefficients of the X' matrix and Z' vector depend upon the control region. In the comfort zone, with no energy requirement:

$$X'_{ij} = X_{ij} \text{ for all } i \text{ and } j \quad \text{Eq. 6.3.2-59}$$

$$Z'_i = Z_i \quad \text{Eq. 6.3.2-60}$$

For zones whose temperature falls below the point for maximum heating or above that for maximum cooling

$$X'_{ij} = X_{ij} \text{ for all } i \text{ and } j \quad \text{Eq. 6.3.2-61}$$

$$Z'_i = Z_i + P_{max,i,h} \text{ or } Z'_i = Z_i - P_{max,i,c} \quad \text{Eq. 6.3.2-62}$$

For zones that fall within the heating or cooling regions and require less than maximum power, the final temperature is assumed to be equal to the heating or cooling set temperature and the average room temperature is then $T_{req,i}$. Eq. 6.3.2-35 can be rewritten to include the power requirements.

$$C_i \frac{d}{dt} T_i = \dot{Q}_i - P_i \quad \text{Eq. 6.3.2-63}$$

P_i and \dot{Q}_i are considered constant over the time-step and \dot{Q}_i is evaluated at the average zone temperature. Substituting into Eq. 6.3.2-63 yields:

$$\begin{aligned} \bar{P}_i - \frac{1}{R_{star,i}} \cdot T_{star,i} - \sum_{j=1}^{adjac. \ surfaces} \sum_{zones \ i \ to \ j} m_{cplg} \cdot C_p \cdot \bar{T}_j = \\ - \left[\frac{1}{R_{star,i}} + \left(\dot{m}_{inf,i} + \sum_k^{nvent} \dot{m}_{v,k,i} + \sum_{j=1}^{adjac. \ surfaces} \sum_{zones \ i \ to \ j} m_{cplg} + \sum_{known \ boundaries} \dot{m}_{cplg} \right) C_p \right] \bar{T}_{req,i} \quad \text{Eq. 6.3.2-64} \\ - \frac{C_i}{\Delta t} (T_{set,i} - T_{\tau-\Delta t}) + \dot{m}_{inf,i} \cdot C_p \cdot T_a + \sum_k^{nvent} \dot{m}_{v,k,i} \cdot C_p \cdot T_{v,k} + Q_{g,c,i} + \sum_{known \ boundaries} \dot{m}_{cplg,s} \cdot C_p \cdot T_{b,s} \end{aligned}$$

Eq. 6.3.2-64 is substituted into the set of energy balances on all zones for any zone that is in the less than maximum heating or cooling region. The solution given by Eq. 6.3.2-58 is valid with the following substitutions for zones evaluated with Eq. 6.3.2-64.

$$X'_{11,ij} = X_{11,ij} \text{ for } i \neq j \quad \text{Eq. 6.3.2-65}$$

$$X'_{11,ii} = 1.0 \quad \text{Eq. 6.3.2-66}$$

$$X'_{12,ij} = X_{12,ij} \quad \text{Eq. 6.3.2-67}$$

$$X'_{22,ii} = X_{22,ii} \quad \text{Eq. 6.3.2-68}$$

Eq. 6.3.2-39 is corrected by adding $\left(\frac{1}{R_{star,i}} T_{req,i} \right)$ to both sides of the equation, then

$$X'_{21,ii} = 0 \quad \text{Eq. 6.3.2-69}$$

$$Z'_{2,i} = Z_{2,i} - X_{11,ii} T_{req,i} \quad \text{Eq. 6.3.2-70}$$

For any zone i connected to any zone m at a fixed set-point, the X' matrix and the Z' vector are modified as

$$X'_{11,im} = 0 \quad \text{Eq. 6.3.2-71}$$

$$Z'_i = Z'_i - X'_{im} T_{req,m} \quad \text{Eq. 6.3.2-72}$$

The solution given by Eq. 6.3.2-58 using the adjusted matrix entries is valid with one further note. The temperature vector actually contains the required power instead of the average zone temperature, for those zones in the less than heating or cooling regions.

In order to determine the proper control regions for all zones, temperatures are calculated for the case of no heating or cooling. This allows a first estimate of the control. For zones where heating or cooling is required, the energy required to maintain the final zone temperature at the set temperature is determined. If the required energy is less than the maximum power available, then the zone is considered to be within the less than maximum heating or cooling region. Otherwise, the heating or cooling output is equal to the maximum. Elements of the X' matrix and Z' vector are set according to the control situation. The system of equations represented by the matrix Eq. 6.3.2-58 is solved. This process is repeated until the control is not changing. The energy requirements are then evaluated for the zones maintained at fixed setpoints.

6.3.2.1.14. SIMULATION TIMESTEP VERSUS WALL TIMEBASE

Eq. 6.3.2-17 gives the heat transfer at the inside surface of a wall given the inside and outside air temperatures. Since it is based upon the time series (Eq. 6.3.2-6 and Eq. 6.3.2-7) it is only strictly correct when these temperatures represent averages over the time-base interval of the time series. The differential equation describing the rate of change of internal energy of the zone air, on the other hand, is solved over the simulation time-step. This may be less than or equal to the wall time-base. If the simulation time-step is less than the wall time-base, then the average heat transfer at the inside wall surface over the time-base is

$$\overline{\dot{q}_{s,i,\Delta t_b}} = B_s \overline{T_{i,\Delta t_b}} + C_s \overline{T_{a,s,\Delta t_b}} + D_s \quad \text{Eq. 6.3.2-73}$$

where Δt_b refers to the wall time-base.

The average time-base air temperatures are

$$\bar{T}_{i,\Delta t_b} = \frac{\sum_{\text{over timebase}} T_{i,\Delta t} \Delta t}{\Delta t_b} \quad \text{Eq. 6.3.2-74}$$

$$\bar{T}_{a,s,\Delta t_b} = \frac{\sum_{\text{over timebase}} T_{a,s,\Delta t} \Delta t}{\Delta t_b} \quad \text{Eq. 6.3.2-75}$$

If Eq. 6.3.2-74 and Eq. 6.3.2-75 are substituted into Eq. 6.3.2-73, then

$$\dot{q}_{s,i,\Delta t_b} = \sum_{\text{over timebase}} \left(B_s \bar{T}_{i,\Delta t} + C_s \bar{T}_{a,s,\Delta t} + D_s \right) \frac{\Delta t}{\Delta t_b} = \frac{\sum_{\text{over timebase}} \dot{q}_{s,i,\Delta t} \Delta t}{\Delta t_b} \quad \text{Eq. 6.3.2-76}$$

The average heat flux over the wall time-base is equal to the average of the heat fluxes evaluated at the average temperatures over the simulation time-step. For the solution of the set of equations each time-step, the heat flux for any wall, as given by Eq. 6.3.2-17, is determined using the current conditions. However, the temperatures and heat fluxes characterizing the previous time history of the wall represent averages over the wall time-base. This leads to temperature steps in the period of the time-base value.

For heavy and thick walls the time-base can be set to two or more hours. If there are very thin walls in the same building, the TRNBUILD program stops with an error message. This is caused by the time constant of the light construction. If the time-constant is lower than the transfer function time-base, the response description will produce an error. In this case, thin walls should be replaced by a description using resistance layers. This procedure yields a heat flux over the wall time-base as given by Eq. 6.3.2-76.

7.2.15. *The SOLVER Statement*

TRNSYS is outfitted with two methods for solving the coupled system of algebraic and differential equations that model a given system: the “successive substitution” method and “Powell’s” method.

A SOLVER command has been added to TRNSYS to select the computational scheme. The optional SOLVER card allows the user to select one of two algorithms built into TRNSYS to numerically solve the system of algebraic and differential equations. The format of the SOLVER card is

```
SOLVER k
```

where k is either the integer 0 or 1. If a SOLVER card is not present in the TRNSYS input file, SOLVER 0 is assumed. If k = 0, the SOLVER statement takes two additional parameters, RFmin and RFmax (see section):

```
SOLVER 0 RFmin RFmax
```

The two solution algorithms are:

- 0: Successive Substitution
- 1: Powell’s Method (Powell, 1970a and 1070b)

Descriptions of these algorithms can be found in the references and most other numerical methods books. They are briefly described in the next sections.

7.2.15.1. Solver 0: Successive substitution

With successive substitution, the outputs of a given model are substituted for the inputs of the next model in the system. The performance of that next model is recomputed and its outputs are then substituted for the inputs of the next model. This substitution continues at a given time step until all connected outputs have stopped changing (i.e. their change is smaller than the limits fixed by the TOLERANCES statement). At that point the TRNSYS kernel deems that convergence has been reached and proceeds on to simulate the next time step.

Generally speaking, TRNSYS calls all components at least once per time step. Then, the TRNSYS solver keeps track of which components must be called during the same time step: only components for which at least one input has changed beyond the fixed tolerances are called. When all components have converged, or when the maximum number of iterations has been reached (see the LIMITS statement), TRNSYS continues to the next time step.

Note: Some components have a different behavior, depending on their value of INFO(9). For example, printers are not part of the normal iterative process, they are only called after all other components have converged. Please see Volume 08, Programmer's Guide, for more details.

Although relatively simple, the successive substitution computational scheme used in most TRNSYS simulations has proven to be reliable and efficient for simulating systems with energy storage such as solar domestic hot water systems, buildings, and HVAC systems. These systems typically have less than 50 coupled differential equations and 100 simultaneous nearly-linear algebraic equations with few recyclic loops and controller decisions. The limitations of the computational scheme become apparent when TRNSYS is used to solve sets of non-linear algebraic equations without differential equations. Equations of this type occur in systems for which the energy storage is negligible, such as for a photovoltaic array directly coupled to a load or a refrigeration system operating at steady-state conditions. The successive substitution solution method does not efficiently solve non-linear algebraic equations and may, in fact, not be able to find a solution if the equations are highly non-linear. If the algebraic loops that cause the numerical problems are clearly identified, the ACCELERATE statement can be used to solve the problem. In other cases, adding numerical relaxation to solver 0 can improve its robustness and speed, depending on the type of numerical problems which is involved. Both methods are described here below

THE ACCELERATE STATEMENT

An ACCELERATE command was added to TRNSYS version 13 to improve convergence in problems with recyclic information flow. The ACCELERATE (see Section 7.2.10) command allows the user to break a selected INPUT-OUTPUT connection and replace it with a single-variable Newton's method solution algorithm. Although useful in many circumstances, the ACCELERATE command may be unsatisfactory for two reasons; 1) it requires the user to identify the appropriate INPUT-OUTPUT connection and 2) it implements a single-variable solution method when in many situations, a multiple-variable method is required. Another way in which numerical convergence problems have been handled in past versions is by the user coding convergence-enhancing techniques within the component models. For example, 'combined-component' models have been developed for TRNSYS wherein the non-linear equations describing two or more pieces of equipment are solved internally in a single component model. Although 'combined-component' models may eliminate numerical problems, they reduce component modularity and require the user to implement solution techniques, thereby defeating the original purpose of TRNSYS.

NUMERICAL RELAXATION

Numerical relaxation has been proven to significantly improve the performance of TRNSYS solver 0 for some classes of problems. The coupling of airflow and temperatures in a building are a typical example of problems that cannot be solved easily using successive substitution, and the implementation of numerical relaxation in the TRNSYS kernel was decided after its successful application in TRNFLOW (Weber et al., 2003).

A simple example can be used to illustrate the purpose of numerical relaxation. Consider the following system of equations

$$\begin{cases} y = ax + b \\ y = x \end{cases} \quad \text{Eq. 7.2.15-1}$$

Solver 0 will only solve this system for $-1 < a < 1$. In a TRNSYS simulation, this system of equations will occur if 2 Units are connected in a recyclic information loop:

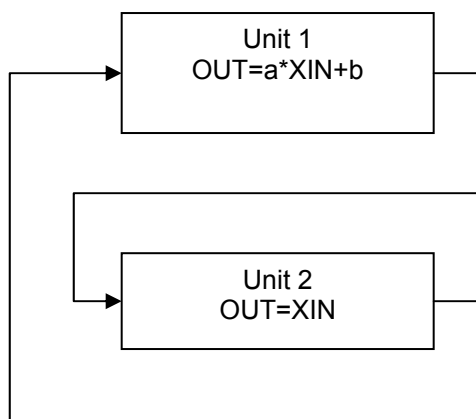


Figure 7–4: Simple linear equations in TRNSYS

- Unit 1: $OUT_{1,1} = a * XIN_{1,1} + b$
- Unit 2: $OUT_{2,1} = XIN_{2,1}$
- Connections: $XIN_{2,1} = OUT_{1,1}$ and $XIN_{1,1} = OUT_{2,1}$

Figure 7–5 shows how Solver 0 would attempt to solve such problems with $a = -1$ and $b = 1$. Starting with a guess value of 0, the successive values of the output of Unit 1 are: 1, -1, 3, -5, 11, -21, etc. The solution ($x=y=1/3$) will never be reached.

The relaxation method consists in adding some damping to the output values to restrict their change. The general formulation is:

$$OUT_{n+1} = OUT_n + RF (OUT_{n+1} - OUT_n) \quad \text{Eq. 7.2.15-2}$$

Where RF is the relaxation factor ($0 \leq RF \leq 1$). If $RF=1$, There is no relaxation (Solver 0). If $RF=0$, there is no solver at all, OUT is kept constant.

If we take the same example ($a=-2$, $b=1$), the successive substitution will converge to the solution if RF is less than $2/3$. For example, if $RF=0.4$ the successive values of the output are: 0.400, 0.320, 0.336, 0.333 (see Figure 7–6).

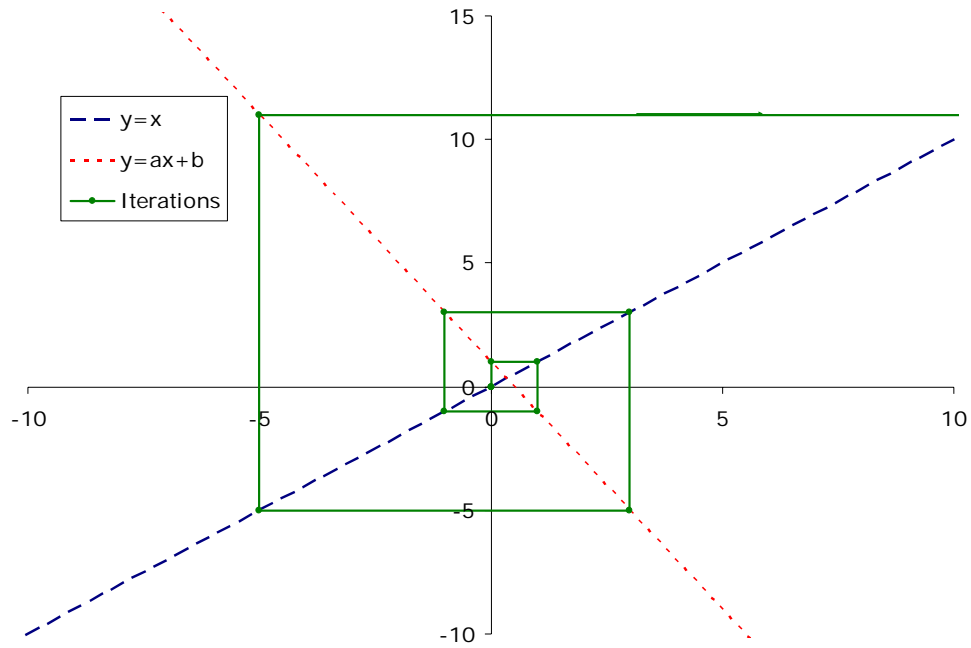


Figure 7–5: Solver 0 without relaxation

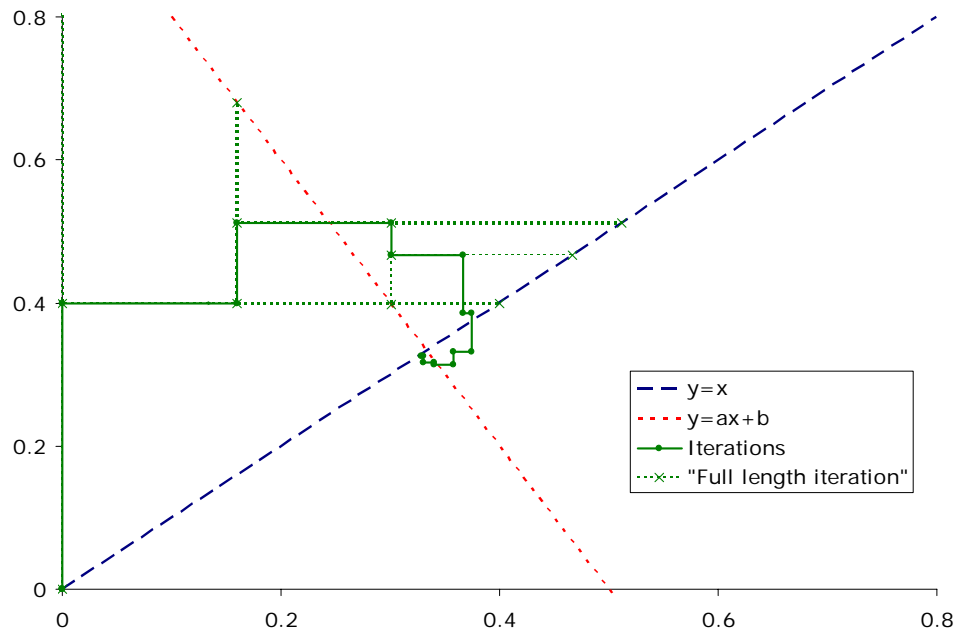


Figure 7–6: Solver 0 with numerical relaxation

Note that if $\alpha > 1$, it is not possible to find a positive value of RF that will lead to convergence. It is necessary to use a negative RF.

IMPLEMENTATION OF THE RELAXATION IN TRNSYS

The relaxation factor is chosen for each output independently and then the output is modified according to.

The implemented rules to modify the relaxation factor were proposed by EMPA:

If $((OUT_{n+1} - OUT_n)(OUT_n - OUT_{n-1}) < 0)$

$$RF_{n+1} = \frac{RF_n}{2}$$

Else

$$RF_{n+1} = 1.5 RF_n$$

In all cases, RF is restricted to $[RF_{min} ; RF_{max}]$

Eq. 7.2.15-3

In other words, the relaxation factor is reduced (*0.5) when the solution oscillates and it is increased (*1.5) when the solution keeps progressing in the same direction.

7.2.15.2. Solver 1: Powell's method

Two reasons led to the development of Solver 1: the need for backsolving and the oscillations resulting from discrete controller outputs.

Notes:

Backsolving requires all TRNSYS components used in a simulation to comply with certain rules, which is rarely the case. For this reason, this feature should be considered for research applications only.

Discrete-state controllers should also be adapted to use solver 1. An example is given by Type 2, the ON-OFF controller with hysteresis. Users are invited to use Type 2 as a template if they want to develop controllers for solver 1. It is important to note that a mismatch between the selected solver and the one that is expected by the component will lead to unpredictable results (hence the error message generated by Type 2). It is generally recommended to use Solver 0 (without and with numerical relaxation) and to consider reduced time steps and time delays before switching to Solver 1.

BACKSOLVING AND DISCRETE VARIABLES IN TRNSYS

The distinction between INPUTS and OUTPUTS in TRNSYS places a limitation on the generality of the component models. When a model is formulated, a decision must be made as to what are the INPUTS and what are the OUTPUTS. For example, a model may be developed to calculate an energy flow for a given mass flow rate and temperatures. However, in a different application, the energy flow rate may be known and the model must calculate the mass flow rate. The same physical model is used in either situation, but the INPUTS and OUTPUTS are different. Normally, TRNSYS requires different component models (or modes) to handle the change in information flow. However, the Powell's Method solver, when implemented, is able to backsolve the TRNSYS equations; effectively solving an input for a given output as discussed below.

The equations within any TRNSYS component model can be reduced to the following general form:

$$\dot{X}_i = D_i(X_i, \dot{X}_i, U_i, t) \quad \text{Eq. 7.2.15-4}$$

$$\dot{X}_i = F_i(U_i, t) \quad \text{Eq. 7.2.15-5}$$

where U_i and X_i are, respectively, the vector of INPUTS and OUTPUTS for the i^{th} module, t is time, and D_i and F_i are functions representing the differential and algebraic equations, respectively.

An analogous set of equations can be written for the combined set of equations in all components,

$$\dot{X} = D(X, \dot{X}, U, t) \quad \text{Eq. 7.2.15-6}$$

$$\dot{X} = F(U, t) \quad \text{Eq. 7.2.15-7}$$

where **U** and **X** are vectors containing the INPUTS and OUTPUTS for all components. Each INPUT in a TRNSYS model is an OUTPUT from another component, an equation, or a specified value. This additional information can be written as a matrix equation:

$$\mathbf{AU} + \mathbf{BX} + \mathbf{D} = 0 \qquad \text{Eq. 7.2.15-8}$$

where **A** and **B** are coupling matrixes with element values of 0 and ± 1 , and **D** is a vector of boundary conditions. The system of algebraic-differential equations can now be solved to determine **U(t)** and **X(t)**.

The number of simultaneous non-linear equations resulting from this new computational scheme can become very large. To minimize computational effort, it is necessary to employ methods for decreasing the number of simultaneous equations. Some TRNSYS inputs are always known, either because they are constants or they depend explicitly on time. Moreover, it is possible for all of the inputs in some TRNSYS components to be of this type. In this case, the outputs of these components can be calculated independently of other components. Once the outputs of these components are known, the inputs to which they are connected are now known, possibly allowing additional components to be evaluated independently. This process is repeated until only the components that require simultaneous solution remain. This process is a first step in equation blocking in which a large set of equations is broken into smaller sets that are more easily solved.

The remaining set of equations is solved numerically. There are a number of well-developed numerical methods for these types of systems (Broyden, 1965; Gerald and Wheatley, 1984). In TRNSYS 14 and beyond, differential equations are solved by a backwards differential method (Gear, 1967) so that the combined algebraic and differential systems can be converted to an algebraic system which can be solved at each time step. This method is extremely robust, even for stiff problems. In addition, this method allows use of variable time steps, although TRNSYS does not currently employ variable time steps. Before solving the resulting system of algebraic equations, TRNSYS numerically calculates the Jacobian matrix and permutes it to lower triangular form. This permutation facilitates equation blocking which breaks the original system into smaller sets of equations that can be solved more efficiently (Duff et al., 1986). Each block of equations is solved using Powell's method (Powell, 1970a and 1970b). This method combines Newton's method and steepest descent approach and is one of the more robust and efficient algorithms for solving non-linear equations.

A major advantage of the computational scheme is that the solution method does not care whether INPUTS or OUTPUTS are specified as long as a valid set of simultaneous equations are defined. As a result, TRNSYS has the ability to solve backward problems, i.e problems in which one or more OUTPUTS are specified and the corresponding INPUTS must be determined. An example of a backward solution problem is provided in the Examples section.

Unfortunately, there is also a disadvantage of the Powell's method computational scheme. Because the alternative solver determines the Jacobian matrix numerically, TRNSYS usually requires more component calls each time step than the scheme used in TRNSYS 13 - provided that TRNSYS 13 is able to solve the problem. The computational scheme in TRNSYS 14 and above is far more robust and is consequently able to solve a greater variety of problems without experiencing computational difficulties. However, because some problems can be more efficiently solved with the successive substitution computational scheme, both computational schemes are made available to the user in TRNSYS version 14 and above. In most cases, users should try to solve their TRNSYS input file using the simpler (and most times quicker) successive substitution method. If problems are encountered, the alternative solver should be employed.

A problem that plagued TRNSYS 13 and other general equation solvers in the past is the presence of discontinuities or discrete states, particularly when they demonstrate hysteresis properties. Many existing TRNSYS components have equations of this type. The most common

example is the ON/OFF Differential Controller provided as TYPE 2 in the standard TRNSYS library. This component calculates a control signal, which is either 0 or 1, based on the differences between two input temperatures and the control signal from the previous calculation. The functional relation between the controller OUTPUT and its INPUTS is shown in Figure 7–7. If the temperature difference is less than ΔT_{\min} , the control signal is 0. If the temperature difference is greater than ΔT_{\max} , the control signal is 1. However, if the temperature difference is between ΔT_{\min} and ΔT_{\max} , the output control signal is either 0 or 1, depending on its previous value. This type of mathematical behavior can lead to oscillations in the calculations preventing convergence of the numerical solution to the system of equations. In fact, there may not be a stable numerical solution, depending on the selected values of ΔT_{\min} and ΔT_{\max} .

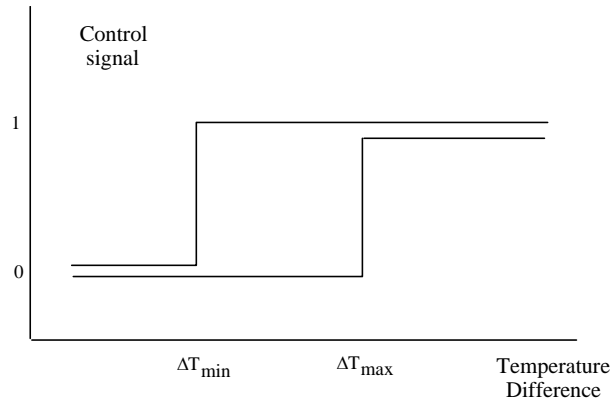


Figure 7–7: TRNSYS Type 2 controller with hysteresis properties

There are many examples of ON/OFF controls in the TRNSYS component library, such as the Flow Diverter (Type 11), the Energy/Degree-Day Space Heating or Cooling Load (Type 12) and the Pressure Relief Valve (Type 13). Other examples of discontinuities which appear in TRNSYS component models are the shift from turbulent to laminar flow regime which causes a discrete change in a heat transfer coefficient or friction factor, and the inter-node flow control logic for a stratified water storage tank. During the iteration process the component OUTPUTS can switch states resulting, in effect, in a different set of equations with a different Jacobian. Previous versions of TRNSYS handled the convergence problems that result from this behavior by ‘sticking’ the state after a specified number of iterations. For example, the TYPE 2 ON/OFF Differential Controller has a parameters called NSTK which is the number of state changes allowed before the controller state is frozen at its current value. Although freezing the controller state eliminates most convergence problems, it may lead to incorrect results in some cases, particularly in short term simulations.

The Powell’s Method solver in TRNSYS 14 provided a very different approach for handling these problematic discontinuities. The TRNSYS executive program, rather than the component models, directly controls discrete variables. At the start of each time step, the values of all discrete variables are known. TRNSYS forces these values to remain at their current state until a converged solution to the equations is obtained or an iteration limit is encountered. During these calculations, the component models calculate the ‘desired’ value of the discrete variables but they do not actually change the settings. After a converged solution is obtained, or the maximum number of iterations is exceeded, TRNSYS compares the current discrete variable values with the ‘desired’ values. If they do not differ and a converged solution has been obtained, the calculations are completed for the time step. Otherwise, TRNSYS changes the values of the discrete variables to the ‘desired’ setting and repeats this process, taking care to not repeat the calculations with the same set of discrete variables used previously. If a solution is not obtained after all combinations of discrete variables have been tried, TRNSYS issues a warning message and continues on to the next time step. This method of dealing with discrete variables eliminates

numerical oscillations between iterations and finds the correct solution to the equations, provided that a solution truly exists. Minor changes in the component models utilizing discrete variables are required to take advantage of this computational scheme. These changes have been made in many of the TRNSYS library components. For these reasons, the Powell's Method mode of the Type 2 controller should be used when the Powell's Method TRNSYS equation solver (SOLVER=1). Refer to the TYPE 2 documentation for more details.

To use the backsolving technique described in this section, follow the guidelines listed below:

1. the Powell's Method TRNSYS equation solver should be used (SOLVER 1)
2. the Powell's Method control mode should be used (TYPE 2 - MODE 0)
3. initial values should be reasonable and non-zero
4. the input to be solved should be specified as -1,0 in the input file with an initial value equal to the value the input will take if no backwards solution is found.
5. the specified OUTPUT should be set in an equation with the form:
EQUATION 1
[5,3] = 65 (set the third output from UNIT 5 to a constant value of 65)
6. the specified OUTPUT can be of any form acceptable to the TRNSYS equation processor ([5,3] = 65 + MOD(TIME) + 0.5 * [1,1] for example)
7. the NOCHECK statement should be used with the following variables:
 - all INPUTS to the TYPE 2 controllers (required to eliminate convergence problems)
 - all INPUTS which do not need to be checked for convergence at each iteration. For example flow rates at all components in a flow loop (recommended to reduce the set of equations to be solved)
8. set the differential equation solver to Modified-Euler method (DFQ 1)

Although the backsolving technique is extremely useful, in many cases it can not be employed due to the formulation of many component routines. Due to internal convergence enhancement algorithms, all users writing component routines should consider the backsolving technique during component formulation.

9. WEATHER DATA

9.1. Introduction

The weather files distributed with TRNSYS 16 are organized in directories according to the data sources. Currently there are two data sources:

- The US-TMY2 data sets from NREL (redistributed with NREL's permission). The dataset includes 237 locations in the US (contiguous states + Alaska and Hawaii), and 2 files for Puerto-Rico and Guam.
- Selected worldwide stations from Meteonorm (distributed under license from Meteotest). More than 1000 locations are included, in more than 150 countries.

More details on the datasets and their owners are given in the next sections.

9.1.1. Important notice

Conditions for using the weather files included with TRNSYS 16 are given in the license agreement. The relevant section is reproduced here:

Weather files distributed with TRNSYS are copyrighted by their respective owners. They may not be redistributed in any form without prior consent from their owner. Neither the weather files owners nor The TRNSYS developers can accept any liability concerning the use of the weather files. The user shall be responsible for checking the correctness and plausibility of the results by applying state-of-the-art control procedures.

The TMY2 data sets for the US and their manual were produced by the National Renewable Energy Laboratory's (NREL's) Analytic Studies Division under the Resource Assessment Program, which is funded and monitored by the U.S. Department of Energy's Office of Solar Energy Conversion.

The weather files in the "Meteonorm" directory (and subdirectories) were generated using Meteonorm Version 5. MeteoNorm is published by METEOTEST (<http://www.meteotest.com>) – authors: METEOTEST, Jan Remund, Stefan Kunz).

The use of weather files distributed with TRNSYS implies the acceptance of the above-mentioned license agreement.

9.1.2. Overview of available data

Figure 9–1 gives an idea of the available locations. Blue triangles represent US-TMY2 stations, black and white dots represent Meteonorm stations. Black dots represent stations for which recorded solar radiation data is available (see section 9.3 for more details).

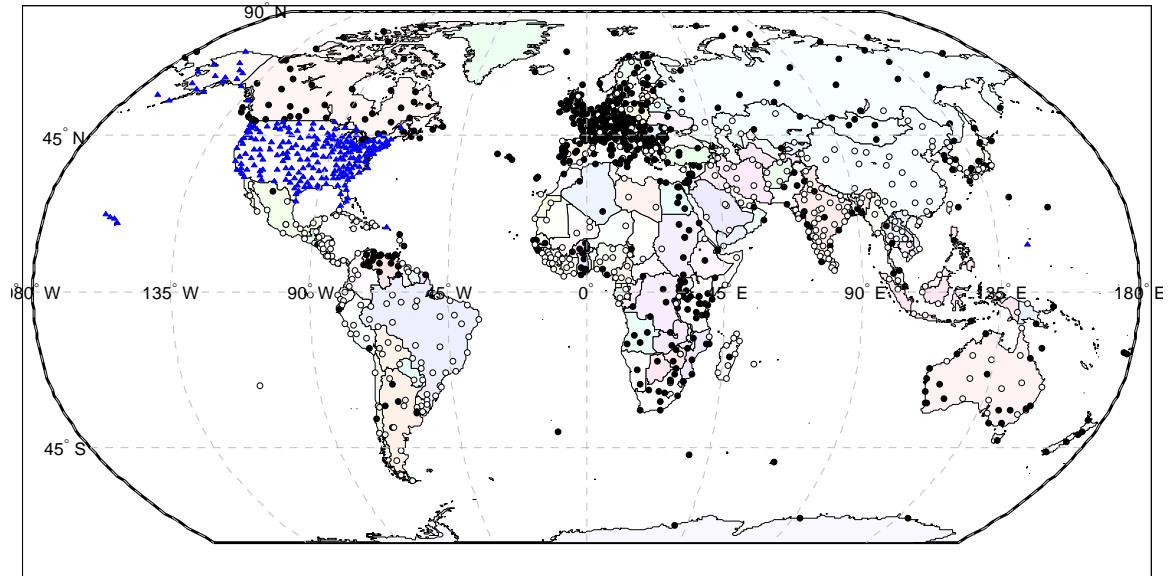


Figure 9–1: Overview of available weather locations

Table 9–1 summarizes the available data according to the data sources and world regions.

Table 9–1: Summary of available weather locations

Source	World Region	Available locations
US-TMY2	United States only	237 locations in the US (contiguous states + Alaska and Hawaii), and 2 files for Puerto-Rico and Guam.
Meteonorm	Africa	174 locations in 46 countries
	Asia	227 locations in 43 countries
	Australia-Oceania	55 locations in 9 countries
	Central and South America	141 locations in 29 countries
	Europe	369 locations in 35 countries
	North America (excluding US)	67 locations in 3 countries

9.3. *Meteonorm data*

9.3.1. *Introduction*

The data file distributed with TRNSYS 16 that were generated using Meteonorm are distributed under license from Meteotest. All files were generated using default options in Meteonorm V 5.0.13. The "TMY2" output format is used because it is easily read by Type 89 and Type109.

9.3.1.1. *Important warning*

The selected data set includes locations for which no local solar radiation data is available. In those cases, Meteonorm interpolates data but the quality of hourly weather files is not as good as for locations where actual solar radiation data was recorded. **Those stations are identified by white dots on the maps and by a "0" in the "SR" column in the file lists. Special care should be exercised when using those files.** For all weather files included with TRNSYS, it is always the responsibility of users to check the correctness and plausibility of the results.

9.3.2. *File names and organization*

All data files are in the subdirectory Weather\Meteonorm. There are 5 subdirectories corresponding to the 5 world regions considered in Meteonorm.

- **Africa**
 - This directory includes all weather data files for Africa.
- **Asia**
 - This directory includes all weather data files for Asia. The separation between Asia and Europe, as well as Asia and "Australia / Oceania", are the ones adopted in Meteonorm so users looking for weather files in Western Asia or South-Eastern Asia may have to look in the above-mentioned regions.
- **Australia-Oceania**
 - This directory includes all weather data files for Australia and Oceania. The separation between this region and Asia is the ones adopted in Meteonorm so users looking for weather files in Oceania may have to look in the "Asia" directory.
- **Central-South-America**
 - This directory includes all weather data files for Central and South America. Note that Mexico is listed in the "North-America" category in Meteonorm. This directory also includes all weather data files for Antarctica, independently of their longitude.
- **Europe**
 - This directory includes all weather data files for Europe. This covers the region separately identified in Meteonorm by "Switzerland". The separation between Europe and Asia is the one adopted in Meteonorm so users looking for weather files in Eastern Europe may have to look in the "Asia" directory.
- **North-America**

- This directory includes all weather data files for Northern America (including Mexico). Note that no weather data files are given for the US. Users should use the US-TMY2 data files for locations in the US (contiguous states + Alaska), Puerto-Rico and Guam.

Available stations are listed in the following sections. The filenames have the following format:

CC-ST-Location-nnnnn.tm2 **(e.g. CA-QC-Montreal-716270.tm2)**

or

CC-Location-nnnnn.tm2 **(e.g. DE-Stuttgart-107370.tm2)**

With:

- CC ISO-3266 2-letter country code
 Note that some islands belonging to other countries have their own code (e.g. RE for La Réunion) while others use the country code of the "mainland" (e.g. ES for the Spanish island of Gran Canaria)
- ST The 2-letter code for the State if relevant (Canada only)
- Location name of the city or weather station (e.g. "Stuttgart", or " Abidjan-Airp")
- nnnnn World Meteorological Organization (WMO) Identification Number of the location (if relevant)
- .tm2 The files follow the TMY2 format which can be read by Type 89 and Type 109