Doc. CEN/TC 228 N568 **CEN/TC 228**

Date: 2006-08

CEN/TC 228 WI 018

CEN/TC 228

Secretariat: DS

Heating systems in buildings - Design of embedded water based surface heating and cooling systems - Part 3: Optimizing for use of renewable energy sources

Einführendes Element — Haupt-Element — Ergänzendes Element

Élément introductif — Élément central — Élément complémentaire

ICS:

Descriptors:

Document type: European Standard
Document subtype:
Document stage: Formal Vote
Document language: E

C:\Mine dokumenter\TC 228\WG 5\WI 00228018\TC version\N568_prEN 15377-3_WI00228018_2006-08.doc STD Version 2.1c

Contents

Page

Foreword

This document (prEN 15377-3:2006) has been prepared by Technical Committee CEN/TC 228 “Heating systems in buildings”, the secretariat of which is held by DS.

This document is part of a series of standards developed for implementation of the European Energy Performance of Buildings Directive (EPBD)

The subjects covered by CEN/TC 228 are the following:

- design of heating systems (water based, electrical etc.);
- installation of heating systems;
- commissioning of heating systems;
- instructions for operation, maintenance and use of heating systems;
- methods for calculation of the design heat loss and heat loads;
- methods for calculation of the energy performance of heating systems.
- methods for design and dimensioning embedded, radiant surface heating and cooling systems

Heating systems also include the effect of attached systems such as hot water production systems.

All these standards are systems standards, i.e. they are based on requirements addressed to the system as a whole and not dealing with requirements to the products within the system.

Where possible, reference is made to other European or International Standards, a.o. product standards. However, use of products complying with relevant product standards is no guarantee of compliance with the system requirements.

The requirements are mainly expressed as functional requirements, i.e. requirements dealing with the function of the system and not specifying shape, material, dimensions or the like.

The guidelines describe ways to meet the requirements, but other ways to fulfil the functional requirements might be used if fulfilment can be proved.

Heating systems differ among the member countries due to climate, traditions and national regulations. In some cases requirements are given as classes so national or individual needs may be accommodated.

In cases where the standards contradict with national regulations, the latter should be followed.

Introduction

The aim of the present document is to give a guide for the design of water based embedded heating and cooling systems to promote the use of renewable energy sources and to provide a method for actively integrating the building mass to reduce peak loads, transfer heating/cooling loads to off-peak times and to decrease systems size.

Part 1 of this series of standards provides methods for calculation of design heating and cooling capacity for this type of systems under steady state conditions. Part 2 provides methods for dimensioning and installation of this type of systems. A section in the present standard describes how the design and dimensioning can be improved to facilitate renewable energy sources.

Peak loads can be reduced by activating the building mass using pipes embedded in the main concrete structure of the building (**T**hermo-**A**ctive-**B**uilding-**S**ystems, TABS). For this type of systems, the steady state calculation of heating and cooling capacity (part 1 of this standard) is not sufficient. Thus, several sections of this standard describe methods for taken into account the dynamic behavior.

The proposed methods are used to calculate and verify that the cooling capacity of the system is sufficient and to calculate the cooling requirements on the water side for sizing the cooling system, including chiller.

The energetic assessment of surface heating and cooling systems may also be carried out according to national guidelines accomplishing the goal of this standard.

1 Scope

This document is applicable to water based surface heating and cooling systems in residential, commercial and industrial buildings.

The methods apply to systems integrated into the wall, floor or ceiling construction without any open air gaps.

The methods do not apply to heated or chilled ceiling panels or beams.

This standard is part 3 of a series of standards:

- Part 1: Steady-state calculation methods for determination of the heating and cooling capacity;
- Part 2: Method for design, dimensioning and installation;
- Part 3:Optimizing for use of renewable energy sources.

The aim of the present Standard is to give a guide for the design for use of renewable energy sources and for the use of Thermo-Active-Building-Systems (TABS).

The tool allows calculation of peak cooling capacity of a thermo-active system, based on heat gains (solar, internal loads, ventilation).

This method also allows calculation of the energy demand on the water side (system) to be used for sizing of the cooling system, like chiller, fluid flow rate, etc.

Steady state heating capacity is calculated according to method B or E, ref. Part 1 of this series of standards.

2 Normative references

The following referenced documents are indispensable for the application of this document. For dated references, only the edition cited applies. For undated references, the latest edition of the referenced document (including any amendments) applies.

prEN 15377-1, *Heating systems in buildings – Design of embedded water based surface heating and cooling systems — Part 1: Determination of the design heating and cooling capacity*

prEN 15377-2, *Heating systems in buildings — Design of embedded water based surface heating and cooling systems — Part 2: Design, dimensioning and installation*

prEN wi 16, *Thermal performance of buildings – Sensible room cooling load calculation – General criteria and validation procedures*

prEN wi 17, *Energy performance of buildings – calculation of energy use for space heating and cooling – General criteria and validation procedures*

prEN wi 31, *Specification of criteria for the internal environment (thermal, lighting, indoor air quality)*

3 Terms, definitions, symbols and abbreviations

For the purposes of this standard, the following apply:

3.1 Data referred to the circuit:

$\dot{m}_{H,sp}$ specific water flow, calculated on the area covered by the circuit [kg/ (m^2 s)];

c_w specific heat of the water [J/ (kg K)];

S_p pipes spacing [m];

d_a external diameter of the pipe [m];

s_p thickness of the pipe wall [m];

L_p length of the circuit [m];

λ_R thermal conductivity of the material of the pipe [W/ (m K)];

P_W^{Max} maximum cooling (<0) or heating (>0) power for a conditioning plant [W];

A_{Floor} area cooled/heated by the circuit [m^2];

θ_w^0 supply water temperature at the beginning of the simulation [°C];

$\theta_w^{\lim}$ minimum (in the cooling case) or maximum (in the heating case) supply water temperature obtainable by the machine [°C].

3.2 Data referred to the room geometry and the boundary conditions:

A_{Walls} overall area of vertical walls, external facade excluded [m^2];

$F_{v\ Floor\text{-}Ext\ Wall}$ view factor floor-external wall;

$F_{v\ Floor\text{-}Ceiling}$ view factor floor-ceiling;

$F_{v\ Floor\text{-}Walls}$ view factor floor-walls;

$R_{add\ Floor}$ additional resistance covering the upper side of the slab [(m^2 K)/W];

$R_{add\ Ceiling}$ additional resistance covering the lower side of the slab [(m^2 K)/W];

R_{Walls} resistance of the surface layer of internal walls [(m^2 K)/W];

$h_{Air\text{-}Floor}$ convective heat transfer coefficient between the air and the floor [W/(m^2 K)];

$h_{Air\text{-}Ceiling}$ convective heat transfer coefficient between the air and the ceiling [W/(m^2 K)];

$h_{Air\text{-}Walls}$ convective heat transfer coefficient between the air and the internal walls [W/(m^2 K)];

$h_{Floor\text{-}Walls}$ radiant heat transfer coefficient between the floor and the internal walls [W/(m^2 K)];

$h_{Floor\text{-}Ceiing}$ radiant heat transfer coefficient between the floor and the ceiling [W/(m^2 K)];

C_{Walls} average specific thermal inertia of the internal walls [J/(m^2 K)]

Δt calculation time step [s].

The following magnitudes shall be known for all the day, so their values during the n-th time step from the beginning of the simulation have to be defined:

$T_{comfort}$ maximum operative temperature allowed for comfort conditions [°C];

$\dot{Q}_{Sun}^n$ solar gain in the room in the present calculation time step [W];

$\dot{Q}^{n}_{Transm}$ incoming heat flux to the room from the external wall in the present calculation time step [W];

$\dot{Q}^{n}_{Air}$ convective heat flux extracted by the air circuit [W];

$\dot{Q}^{n}_{IntRad}$ internal radiant heat gain due to people or electrical equipment in the present calculation time step [W];

$\dot{Q}^{n}_{IntConv}$ internal convective heat gain due to people or electrical equipment in the present calculation time step [W];

f^{n}_{rm} running mode (the value is 1 when the system is running and 0 when the system is switched off) [dimensionless];

3.3 Data referred to the slab and its partitions:

s_1 thickness of the upper part of the slab [m];

s_2 thickness of the lower part of the slab [m];

J_1 number of layers constituting the upper part of the slab [dimensionless];

J_2 number of layers constituting the lower part of the slab [dimensionless];

As a consequence, $J=J_1+J_2$ and J sets of physical properties (ρ_j, c_j, λ_j, δ_j, m_j, R_j) shall be known or chosen, where:

ρ_j density of the material constituting the j-th layer [kg/m^3];

c_j specific heat of the material constituting the j-th layer [J/ (kg K)];

λ_j thermal conductivity of the j-th layer [W/ (m K)];

δ_j thickness of the j-th layer [m], $\delta_j = 0$ if the layer is a mere thermal resistance;

m_j number of partitions of the j-th layer [dimensionless];

R_j thermal resistance summarizing the j-th layer [m^{2}K/W], $R_j > 0$ if the layer is a mere thermal resistance.

Obviously, $\sum_{\mathrm{j=1}}^{\mathrm{J_1}} \delta_{\mathrm{j}} = \mathrm{s}_1$ and $\sum_{j=J_1+1}^{J_1+J_2} \delta_j = s_2$.

3.4 Data referred to the initial temperature profile

The initial value of the supply water temperature (θ_w^0) and the interface temperatures ($\theta_{\mathrm{I,i}}^0$ with $0 \le i \le i_L$) shall be decided. As for the slab, a possible choice could be assigning the same value to all the interfaces, equal to the mean temperature at the start of the simulation.

However, if the simulation covers more than one running cycle, the choice of the initial values is not decisive. In fact, it will influence only the very first time steps of the simulation.

3.5 Calculation of the temperature profile and the heat fluxes in the generic time-step n

The temperature reached at a certain interface at the end of the previous time step is used for calculation of the heat fluxes acting on the building structures and for calculation of the consequent temperatures at the end of the time step in progress. These magnitudes are:

$\dot{q}^{n}_{Conv}$ global specific convective heat gains [W/m^2];

$\dot{q}^{n}_{Rad}$ global specific radiant heat gains [W/m^2];

θ^{n}_{Air} air temperature in the room in the present calculation time step [°C];

θ^{n}_{Walls} mean temperature of the walls in the present calculation time step [°C];

θ^{n}_{Op} operative temperature in the room in the present calculation time step [°C];

θ^{n-1}_{w} supply water temperature at the end of the previous time step [°C];

$\theta^{n-1}_{w\,exit}$ outlet water temperature at the end of the previous time step [°C];

$\theta^{n-1}_{I,i}$ temperature of the i-th interface, with $0 \le i \le i_L$, at the end of the previous time step, [°C];

The results obtained at every time step are:

θ^{n}_{w} supply water temperature at the end of the time step in progress [°C];

$\theta_F{}^n$, $\theta_s{}^n$ temperature of the upper and lower sides of the slab at the end of the time step in progress [°C];

$\theta^{n}_{I,i}$ temperature of the i-th interface, with $0 \le i \le i_L$, at the end of the time step in progress, [°C].

4 Relation to other EPBD standards

The present standard requires input from the following standards: EN 15377-1, EN 15255, EN15265, EN15251.

The present standard provides input data to the following standards: EN15243 (Calculation of room temperatures and of load and energy for buildings with room conditioning systems) and EN ISO13792 (Thermal performance of buildings – Calculation of internal temperatures of a room in summer without mechanical cooling – simplified methods).

5 Optimisation of systems for facilitating the use of renewable energy sources

Transporting energy by water will use less auxiliary energy for pumps and less installation space than carrying the same amount of energy by air. A further optimizing is to use water at temperatures close to room temperature for heating and cooling: Low temperature heating- high temperature cooling.

For normal embedded radiant floor-, wall-, and ceiling heating/cooling systems increasing the pipe spacing density and decreasing temperature difference between supply and return water (delta T) will result in water temperatures closer to room temperatures but this can increase flow rates and pipe lengths leading to higher pressure losses. This forces designers to increase auxiliary energy use for pumps or use larger diameter pipes both negative options. This can partly be compensated by using more circuits of shorter pipe lengths. These factors shall be optimized by using Part 2.: method for design, dimensioning and installation" of this series of standards..

For Thermo Active Building Systems a further optimization regarding use of renewable energy sources is made by reducing the peak load, transferring the load to off peak hours, downsizing of energy generation systems, and increased efficiency of energy generation due to water temperature level. This will increase the possible use of energy sources such as solar collectors, ground source heat pumps, free cooling, ground sources heat exchangers, aquifers etc.

6 The concept of Thermo-Active-Building-Systems (TABS)

A Thermo-Active-Building-System (TABS) is a water based heating and cooling system, where the pipes are embedded in the central concrete core of a building construction (see Figure 1). The heat transfer takes place between the water (pipes) and the concrete, between the concrete core and the surfaces to the room (ceiling, floor) and between the surfaces and the room.

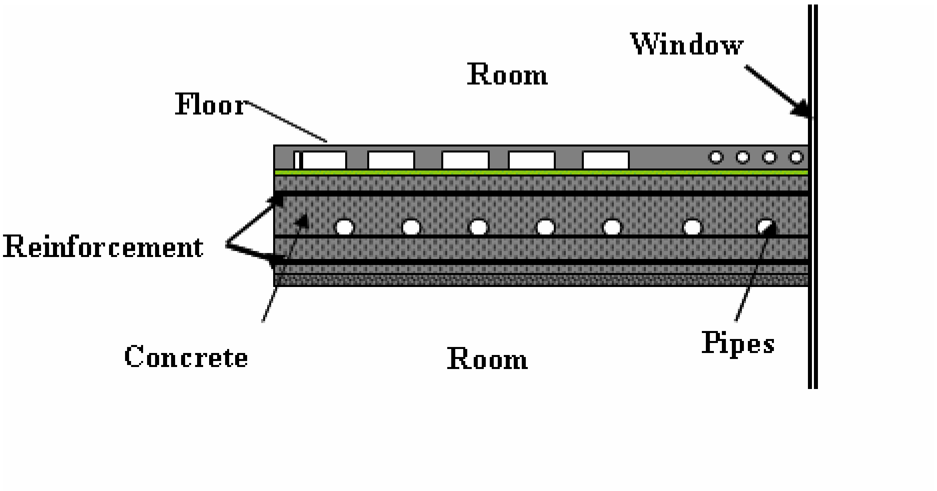
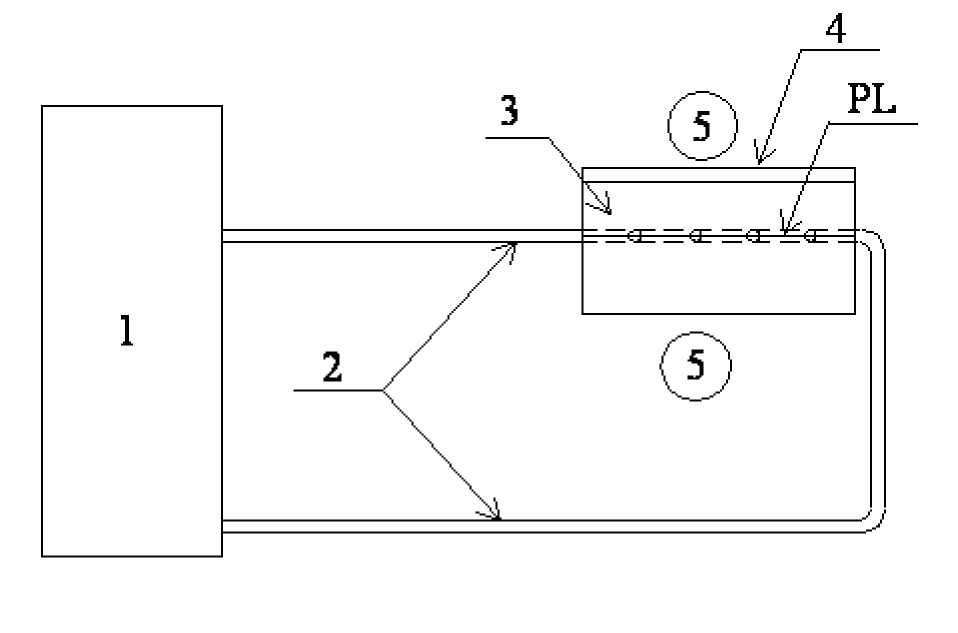


Figure 1 – Thermo-active radiant system

Looking at a typical structure of a thermo-active system, heat is removed by a cooling system (chiller, heat pump, etc.), connected to pipes embedded in the slab. The system can be divided into the following elements (see Figure 2):



where

PL = Pipes level

1 = Cooling system (machine)

2 = Hydraulic circuit

3 = Slab including core level with pipes

4 = Possible additional resistances (floor covering or suspended ceiling)

5 = Room below and room above

Figure 2 – Simple scheme of a thermo-active system

The peak-shaving is the possibility to heat and cool the structures of the building during a period in which the occupants may be absent (during night time), reducing also the peak in the required power (Figure 3). In this way energy consumption may be reduced and lower night time electricity rate can be used. At the same time a reduction of the size of cooling system including chillier is possible

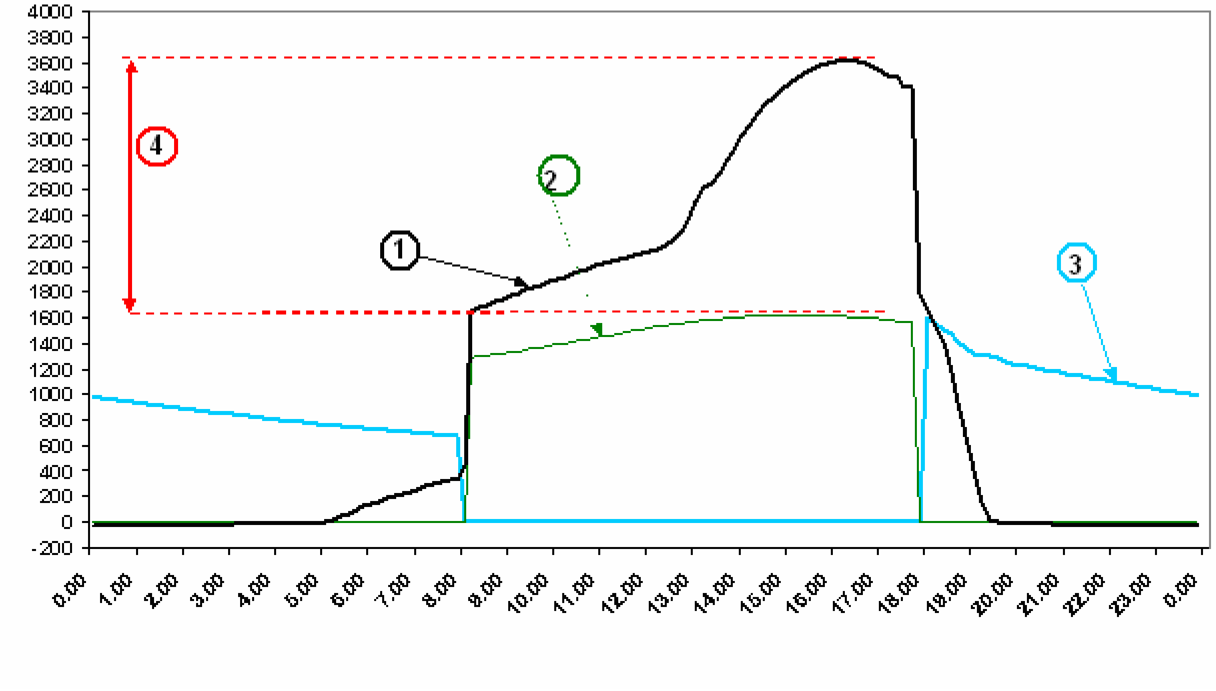


Figure 3 – Example of peak-shaving effect (time vs. cooling power [W])
where: 1) heat gain, 2) power needed for conditioning the ventilation air,
3) power needed on the water side, 4) peak) peak of the required power reduction

TABS may be used both with natural and mechanical ventilation (depending on weather conditions). Mechanical ventilation with dehumidifying may be required depending on external climate and indoor humidity production. In the example in Figure 3, the required cooling power needed for dehumidifying the air during day time is sufficient for cooling the slab during night time.

The designer needs to know if the capacity at a given water temperature is sufficient to keep the room temperature in a given range. The designer needs also to know the heat flow on the water side to be able to dimension the heat distribution system and the chiller/boiler. The present document provides methods for this.

Some detailed building-systems calculation models have been developed, as for determination of the heat exchanges under non-steady state conditions in a single room, determination of thermal and hygrometric balance of the room air, prediction of comfort conditions, check of condensation on surfaces, availability of control strategies and calculation of the incoming solar radiation. The use of such detailed calculation models is, however, limited due to the high amount of time needed for the simulations. Development of a more user friendly tool is required. Such a tool is provided in the present document, which allows simulation of thermo-active systems in an easy way.

Internal temperature changes only moderately during the day and the aim of a good design of TABS is to maintain comfort within the range of comfort, i.e. –0.5 < PMV < 0.5, during the day, according to prEN wi 31 (see Figure 4).

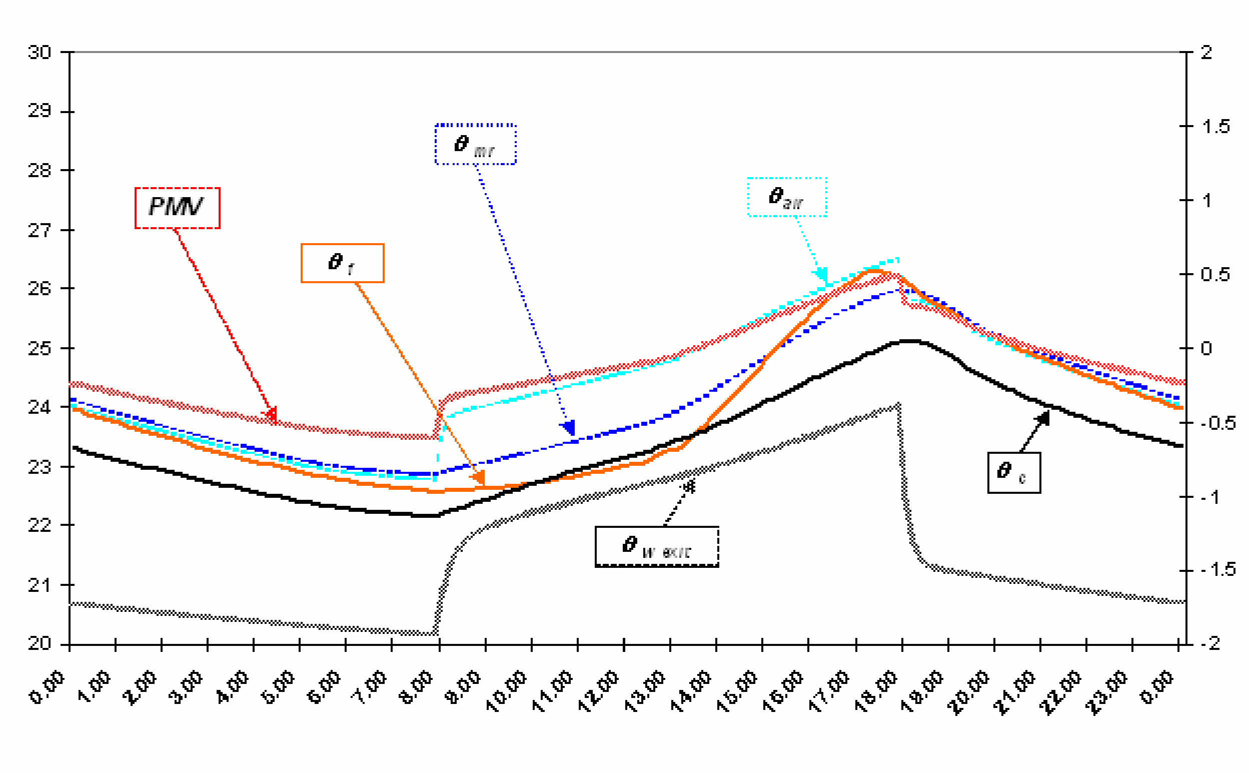


Figure 4 – Example of temperature profiles (left axis) and PMV values (right axis) vs. time where: θ_{mr} mean radiant temperature, θ_{air} air temperature, θ_f floor temperature, θ_c ceiling temperature, $\theta_{w\ exit}$ water return temperature, PMV (Predicted Mean Vote)

The diagrams in Figure 5 show an example of the relation between internal heat gains, water supply temperature, heat transfer on the room side, hours of operation and heat transfer on the water side. The diagrams correspond to a concrete slab with raised floor (R=0.45 [m^{2}K/W]) and a permissible room temperature range of 21 °C to 26 °C.

The upper diagram shows on the y-axis the maximum permissible total heat gain in space (internal gains plus solar gains) [W/m^2], and on the x-axis the required water supply temperature. The lines in the diagram correspond to different hours of operation (8h, 12h, 16h, 24h) and different maximum amount of energy supplied per day [Wh/m^2 d].

The lower diagram shows the cooling power [W/m^2] required on the water side (for dimensioning of chiller) for thermally activated slabs as a function of supply water temperature and operation time. Further, the amount of energy rejected per day is indicated Wh/(m^2 d).

The example shows, that by a maximum internal heat gain of 38 W/m^2 and 8 hour operation, a supply water temperature of 18,2 °C is required. If, instead, the system is in operation for 12 hours, a supply water temperature of 19,3 °C is required. In total, the amount of energy rejected from the room is appr. 335 Wh/m^2 per day. The required cooling power on the water side is by 8 hours operation 37 W/m^2 and by 12 hours operation only 25 W/m^2. Thus, by 12 hours operation, the chiller can be much smaller. The total heat rejection on the water side is appr. 300 Wh/m^2 per day.

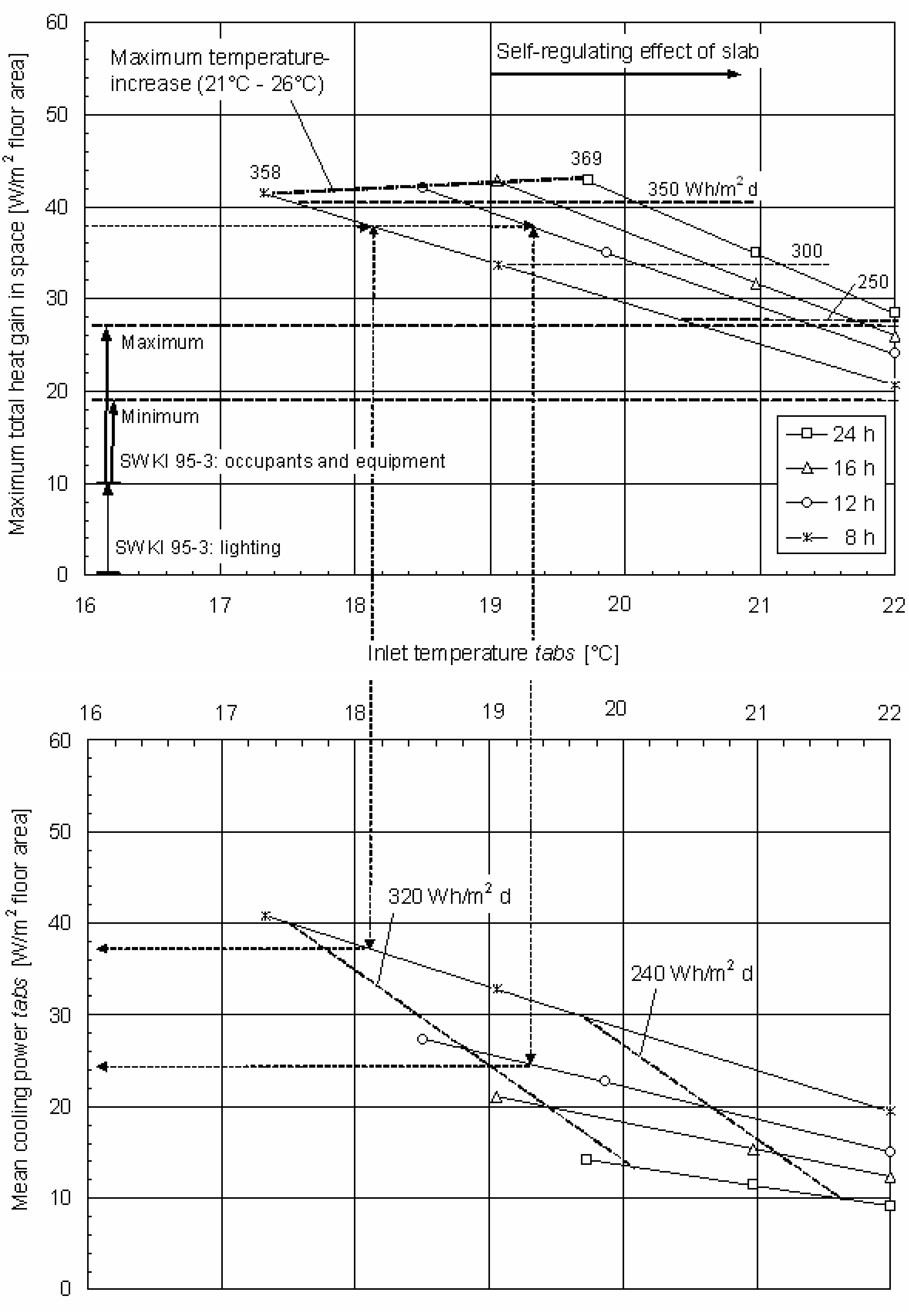


Figure 5 – Working principle of TABS

7 Calculation methods

The following calculation methods can be applied:

— Rough sizing method based on a standard calculation of the cooling load (accuracy 20-30%). To be used based on the knowledge of the peak value for heat gains see 7.1

— Simplified method using diagrams for sizing based on 24 hours values of heat gains, see 7.2 (accuracy of this method 15-20%);

— Simplified model based on finite difference method (accuracy 10-15%). Detailed dynamic simulation for the thermal conduction in the slab via FDM. Based on the knowledge of the 24 values of the variable cooling loads of the room and the temperatures of the air see 7.3

— Detailed simulation models (accuracy 6-10%). Overall dynamic simulation model for the radiant system and the room, see 7.4

7.1 Rough sizing method

The cooling system shall be sized for 70 % of the peak cooling load (ref. prEN 15377-1 and prEN 15377-2). In this case, calculation of the cooling load has to be carried out using an operative temperature of 24°C.

7.2 Simplified sizing by diagrams

In this case, calculation of the heat gains has to be carried out by means of 24 hourly calculations with an operative temperature of 24°C. If heat gains are approximated, 10% of the solar gain has to be added each hour in order to take into account the gains due to external window. This method is based on the assumption that the entire conductive slab is at a constant temperature during the whole day. This average temperature of the slab is calculated by the method itself and is connected with the supply water temperature of the running time of the circuit.

The following magnitudes are involved by the present method:

- Q, which is the specific daily heat load on the room during the design day: it is the sum of the above mentioned 24 hourly values of heat gains, divided by the floor area [kWh/m^2]. The pattern of the load profile must be known.

- $_{comfort}$: maximum operative temperature allowed for comfort conditions [°C].

- Direction of the room to determine when the peak load from heat gains will happened [East (morning), South (noon) or West (afternoon)].

- Number of active surfaces. It distinguishes whether the slab works by heat transfer both on the floor side and on the ceiling side or only the ceiling side is active [Figure 6].

- h: number of hours of fluid flow through the circuits [h];

- R_{int}, thermal resistance of the slab: it is the thermal resistance that connects the conductive parts (figure 7) near the pipes level to the pipes level [m^{2}K/W]. In other words, it is assumed that the conductive part of the slab is maintained at a constant temperature during the occupied period (Figure 8)

- $_{s}$ average slab temperature [°C] dependant on activated ceiling only , or ceiling and floor), the running mode (24h or 8h) and the shape of the internal load profile(lunch break or not). The average surface temperature of the slab is achieved through coefficients included in the method by the equation:

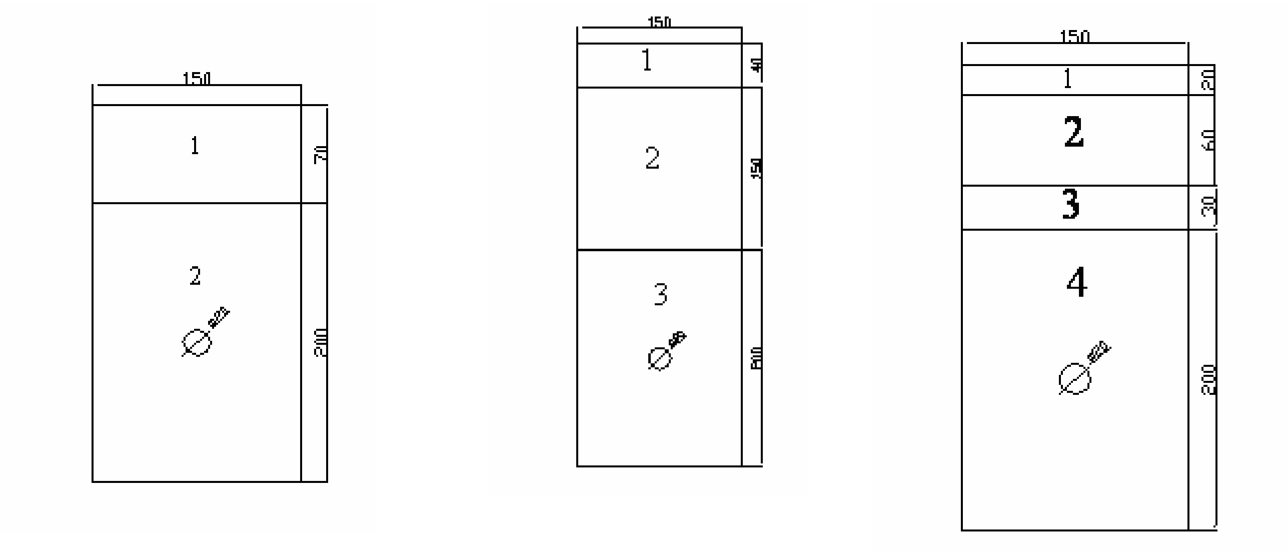
$$\theta_s = \theta_{comfort} + coeff \cdot Q \qquad °C \qquad (1)$$

where values of the coefficient are given in Table 1 and Table 2;

- R_t thermal resistance of the circuit, obtained by the Resistance Method [m^2K/W]. This thermal resistance depends on the characteristics of pipe wall resistance, pipe diameter and pipe spacing (Figure 10).

- $_w$, which is the required temperature of the supply water [°C]. It is obtained through the equation

$$\theta_w = \theta_s - \frac{Q \cdot (R_{\text{int}} + R_t) \cdot 1000}{h} \qquad °C \qquad (2)$$



where

1 = Concrete

2 = Reinforced concrete

Example of slab acting through 2 surfaces

where

1 = Wood

2 = Air

3 = Reinforced concrete

Example of slab acting through 1 surface

where

1 = Wood

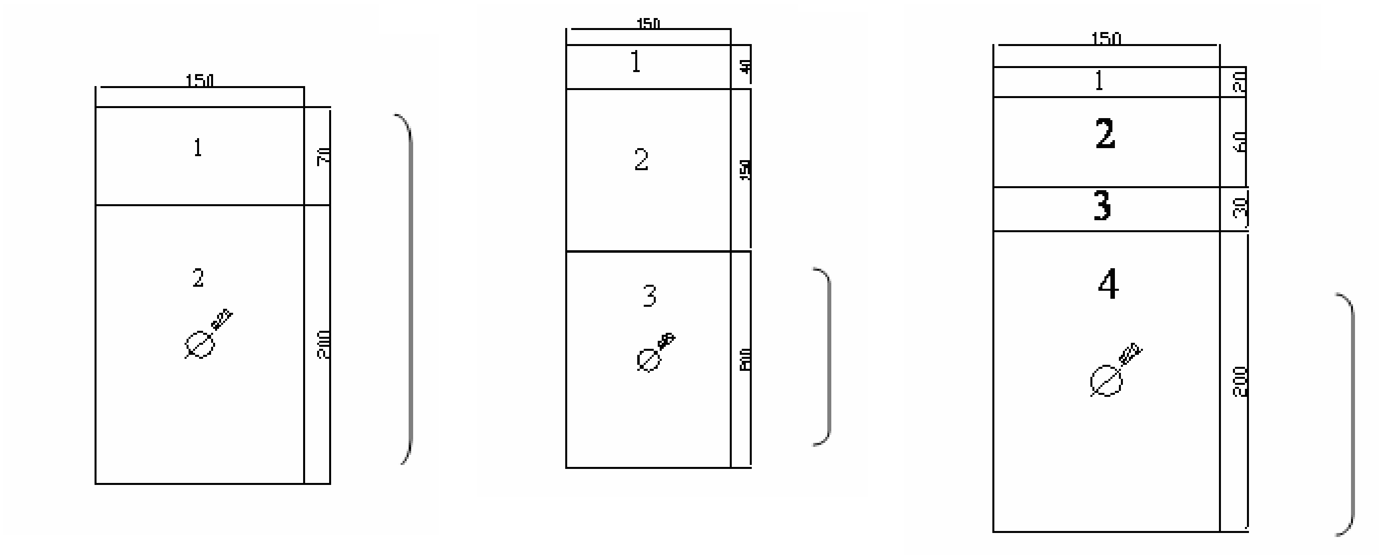
2 = Concrete

3 = Fibreglass

4 = Reinforced concrete

Example of slab acting through 1 surface

Figure 6 – Number of active surfaces



where

1 = Concrete

2 = Reinforced concrete

Conductive region:

Materials 1 and 2

where

1 = Wood

2 = Air

3 = Reinforced concrete

Conductive region:

Material 3

where

1 = Wood

2 = Concrete

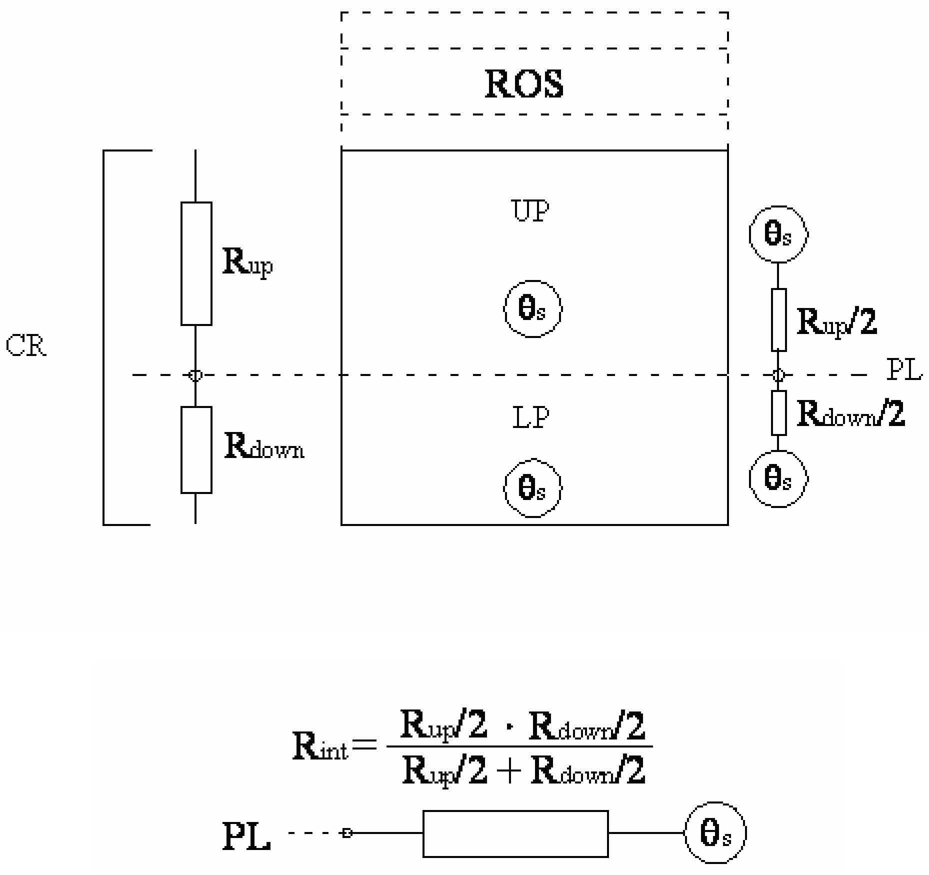
3 = Fibreglass

4 = Reinforced concrete

Conductive region:

Material 4

Figure 7 – Examples of conductive regions


ROS
UP
LP
CR
PL
R_{up}
R_{down}
θ_s
$R_{up}/2$
$R_{down}/2$
$R_{int} = \frac{R_{up}/2 \cdot R_{down}/2}{R_{up}/2 + R_{down}/2}$
PL

where

CR = Conductive region

UP = Upper part of the conductive region

LP = Lower part of the conductive region

ROS = Rest of the slab

PL = Pipes level

Figure 8 – Resistance diagram

The coefficients for calculation of the average temperature of the slab are given in two tables, Table 1 and Table 2, depending on the shape of the internal heat gains profile.

Table 1: Constant internal heat gains from 8:00 to 18:00

		Exposure of the room		
	Kind of floor	**EAST**	**SOUTH**	**WEST**
		Coefficient for calculation of average slab temperature		
Continuous running mode (24 h)	Floor and ceiling C2	-4.6816	-5.3696	-5.935
	Only ceiling C1	-6.3022	-7.2237	-7.7982
Intermittent running mode (8 h)	Floor and ceiling I2	-5.5273	-6.1701	-6.7323
	Only ceiling I1	-7.2853	-7.8562	-8.5791

Table 2: Constant internal heat gains from 8:00 to 12:00 and from 14:00 to 18:00

		Exposure of the room		
	Kind of floor	**EAST**	**SOUTH**	**WEST**
		Coefficient for calculation of average slab temperature		
Continuous running mode (24 h)	Floor and ceiling	-6.279	-7.1094	-7.3681
	Only ceiling	-7.9663	-8.7989	-8.7455
Intermittent running mode (8 h)	Floor and ceiling	-8.1474	-8.758	-9.3264
	Only ceiling	-10.029	-10.685	-10.967

Once $\theta_{\mathrm{comfort}}$ is defined, the tables can be summarized by diagrams. For instance, if $\theta_{comfort}$ = 26°C, the diagram for constant internal heat gains from 8:00 to 18:00 is given in Figure 9.

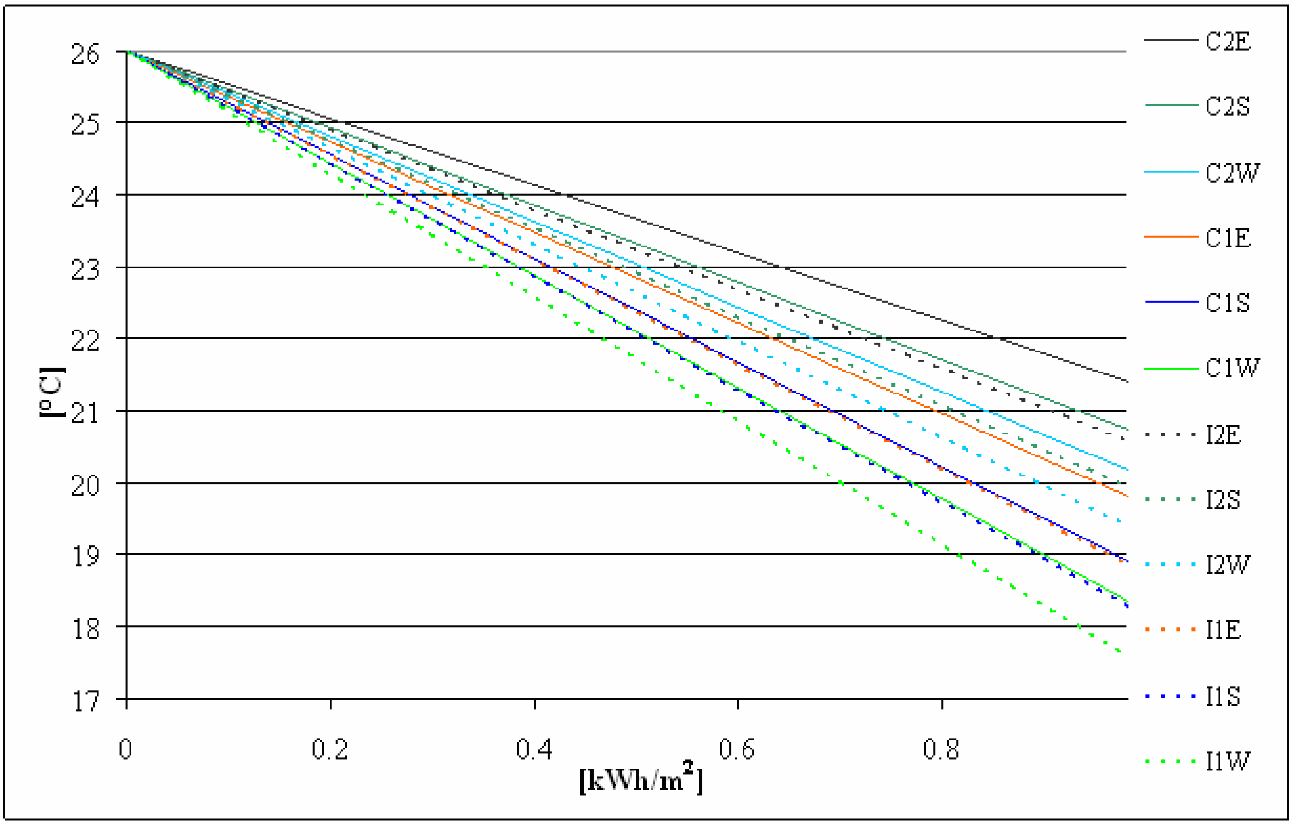
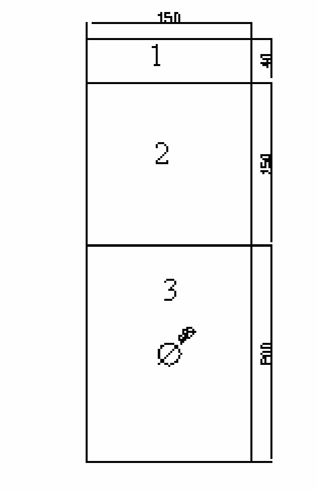


Figure 9 – Diagram for determining θ_s (y-axis) as a function of the specific daily energy (x-axis), exposure of the room (E = East, S = South, W = West), operation condition of the circuit (C = continuous, I = intermittent, 8 hours), and number of active surfaces (1 or 2) in the case of constant internal heat gains during the day (line type will be changed to be readable in B&W)

Example

- Q: 0.6 kWh/m^2; shape of thermal loads: 2 peaks
- $\theta_{comfort}$: 26°C
- Exposure of the room: SOUTH
- Kind of floor:



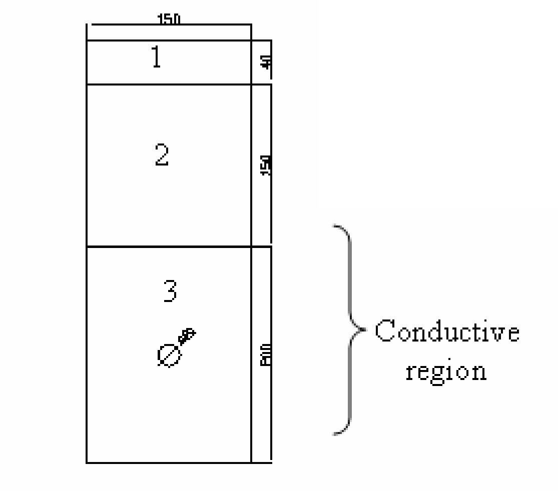
where

1 = Wood

2 = Air

3 = Reinforced concrete

- h: 8h
- R$_{int}$:



If λ of the conductive region = 1.9 W/(m K), then

R$_{up}$ = R$_{down}$ = 0.1/1.9 = 0.053 m^{2}K/W and R$_{int}$ = 0.013 m^{2}K/W

- $\theta_s = 26 - 10.685 \cdot 0.6 = 19.6°C$
- R$_t$: 0.07 m^{2}K/W
- $\theta_w = 19.6 - \frac{0.6 \cdot (0.013 + 0.07) \cdot 1000}{8} = 13.38°C$

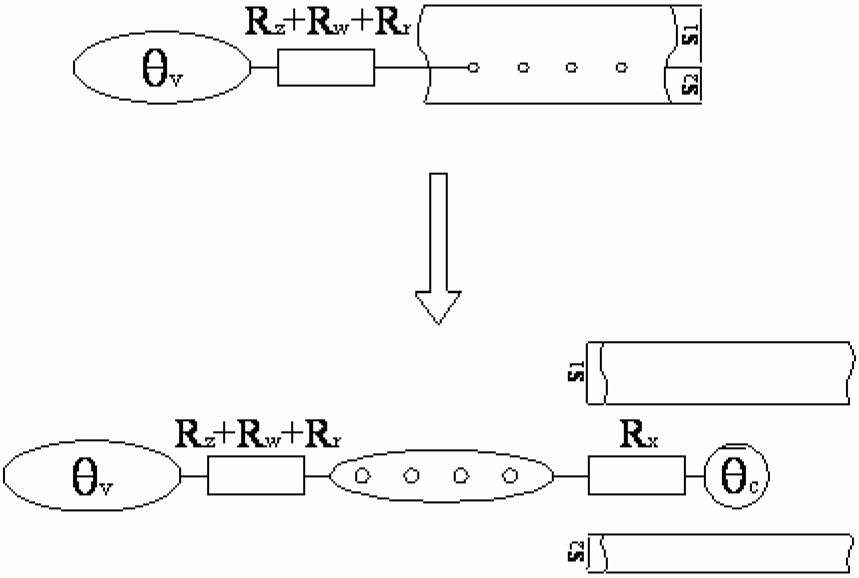
7.3 Simplified model based on finite difference method (FDM)

7.3.1 Cooling system

The limited power of the cooling system shall be taken into account. In fact, it will not be able to keep a constant supply water temperature, since it depends on the amount of heat flux previously exchanged with the slab and on the maximum power of the chiller. A new inlet water temperature after each time step is calculated by taking into account the heat fluxes at the end of the previous time step.

7.3.2 Hydraulic circuit

The Resistance Method is applied. It sets up a straightforward relation, expressed in terms of resistances, between the inlet water temperature and the average temperature at the pipes plane, $\bar{\theta}_c$. So the slab may be split into two smaller slabs. In this way, the upper slab (which is above the pipes plane) and the lower slab (which is below the pipes plane) are considered separately (see Figure 10).



Include legend

Figure 10 – Background concept of the Resistance Method

Slab

The Resistance Method allows splitting of the slab into two parts, which are analyzed through an explicit finite difference method.

Room

An air node is taken into account coupled with the upward and downward surface of the slab and with a fictitious wall-node, via three resistances. Besides, the two surfaces of the slab are coupled together via a resistance taking into account the radiation exchange between them, and each slab surface is connected through a resistance to the wall-node (see Figure 11, 12 and 13).

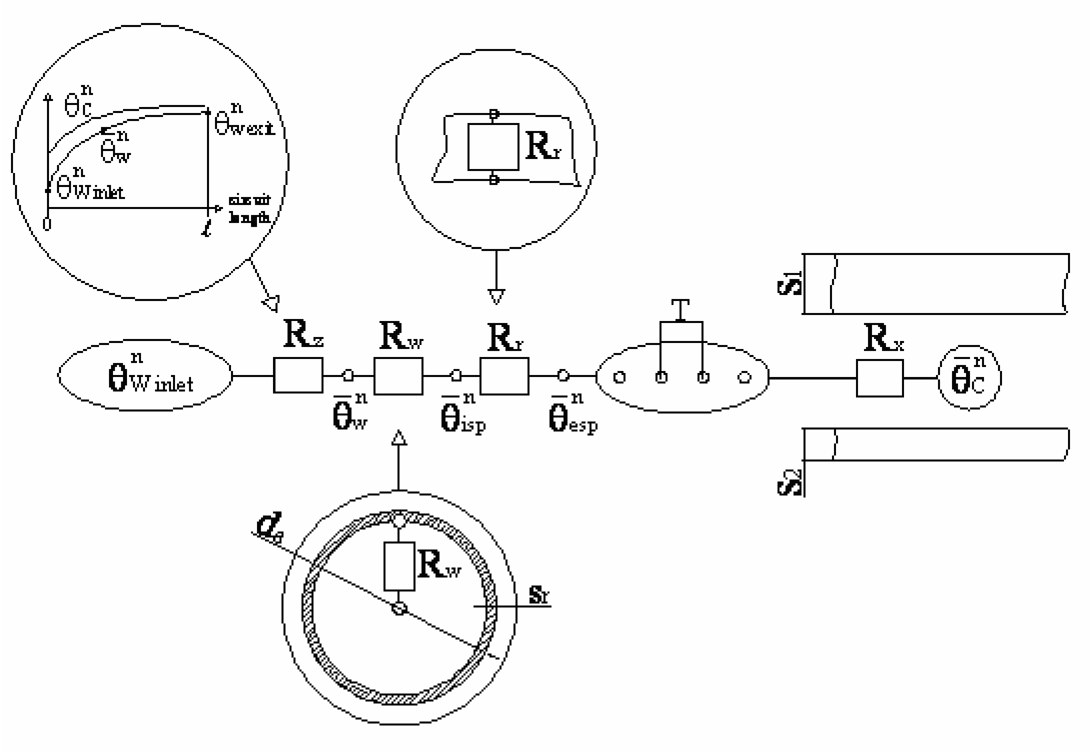
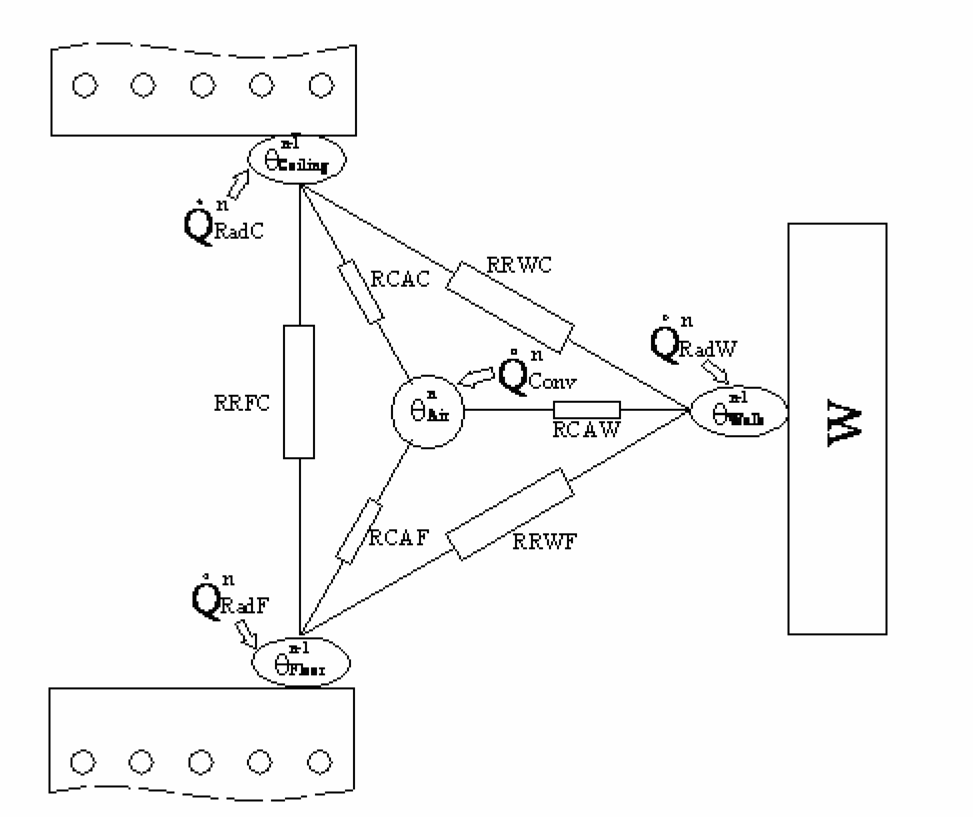
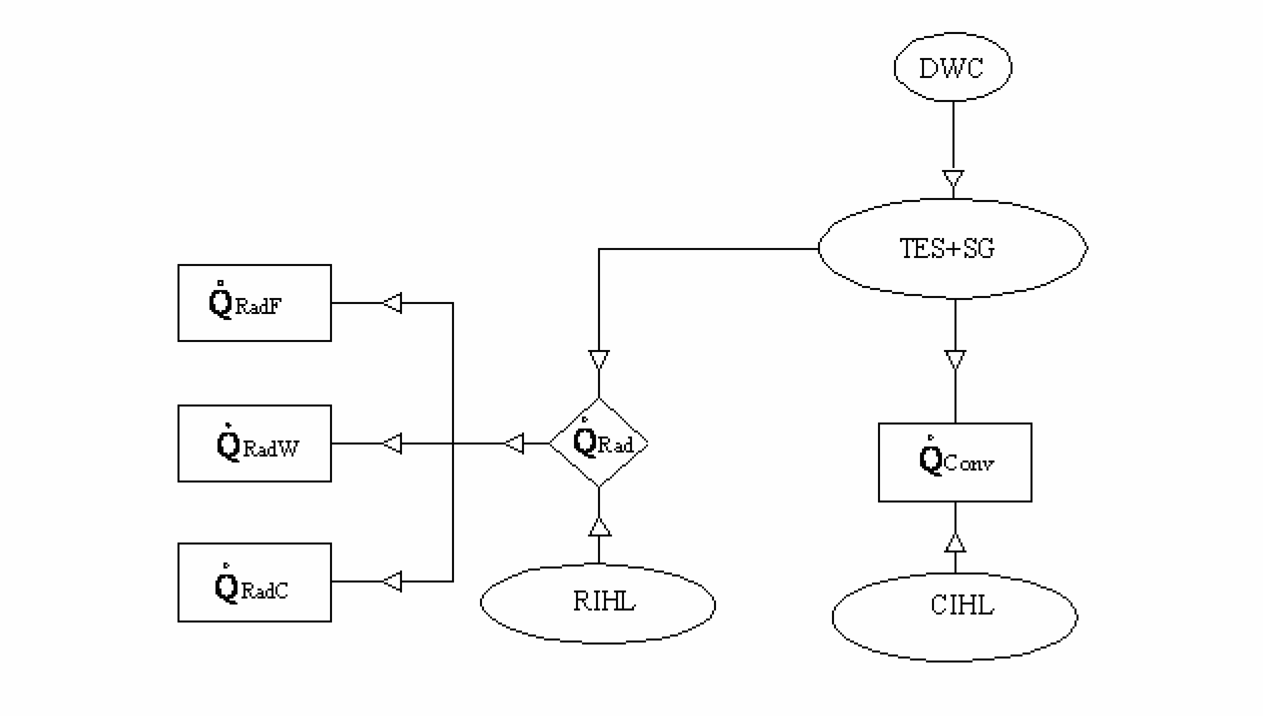


Figure 11 – General scheme of the Resistance Method



where W is the node describing the internal walls

Figure 12 – Scheme of the heat loads network



where

DWG = Design weather conditions

TES = Transmission through the external surface

SG = Solar gain

CIHL = Convective internal heat loads

RIHL = Radiant internal heat loads

Figure 13 – Heat loads involved acting on the room and how they take part in the calculations

Limits of the method

The following limitations shall be met:

- pipes distance: from 0.15 to 0.3 m
- usual concrete slab structures have to be considered, λ = 1.15-2.0 W/ (m K), with upward additional materials, which might be acoustic insulation or raised floor. No discontinuous light fillings can be considered in the structures of the lower and upper slabs.

If these conditions are not fulfilled, a detailed simulation program has to be applied for dimensioning the thermo-active system (see 6.4).

Under the above mentioned conditions, a cooling load calculation or a simulation for a convective system can be carried out for an entire 24 h period and with an internal temperature of 24°C. The results of this calculation, to be taken into account as input for the present simplified model, are the solar gains and the heat fluxes into the room from the external surface.

7.4 Dynamic building simulations program

For all cases, which are not in the range of validation of the simplified methods, TABS calculations have to be carried out by means of a detailed dynamic building-system model.

These TABS calculations have to take into account the water flow into the pipes, the heat conduction between upward and downward surface of the slab and the pipe level, heat conduction of each wall, mutual radiation between internal surfaces, convection with air, and the thermal balance of the air.

Whenever results of TABS calculations are reported, the computer program applied shall be specified.

Note: TC156WG6 may develop a verification method.

Annex A
(normative)

Calculation method

A.1. Pipes level

R_t is the total resistance (m^2 K)/W between the inlet water temperature and the pipes level temperature, determined by the Resistance Method. R_t can be calculated through the equation

$$R_t = R_z + R_w + R_r + R_x \qquad \text{(a1)}$$

where:

$$R_z = \frac{1}{2 \cdot \dot{m}_{H,sp} \cdot c_w} \qquad R_w = \frac{T^{0.13}}{8 \cdot \pi} \left(\frac{d_a - 2 \cdot s_r}{\dot{m}_{H,sp} \cdot L_R} \right)^{0.87}$$

$$R_p = \frac{T \cdot \ln\left(\dfrac{d_a}{d_a - 2 \cdot s_r} \right)}{2 \cdot \pi \cdot \lambda_r} \qquad R_x = \frac{T \cdot \ln\left(\dfrac{T}{\pi \cdot d_a} \right)}{2 \cdot \pi \cdot \lambda_b}$$

Two conditions shall be fulfilled for application of these equations:

— the equation for R_x is valid only if $s_1 / T > 0.3$, $s_2 / T > 0.3$ and $d_a / T < 0.2$

— the equation for R_z is valid only if $\dot{m}_{H,sp} \cdot c_w \cdot \left(R_w + R_p + R_x \right) \geq \frac{1}{2}$

If both conditions are fulfilled, the equation $R_t = R_z + R_w + R_p + R_x$ can be applied.

The machine model is expressed in an explicit way, so the inequality $R_t \cdot \dot{m}_{H,sp} \cdot c_w > 1$ shall be fulfilled in order to avoid calculations instability.

A.2. Subdivision of the slab

The slab is composed by J=J$_1$+J$_2$ material layers. As a consequence, J sets of physical properties (ρ_j, c_j, λ_j) shall be known. Besides, each layer has its own thickness, δ_j, thus, for geometrical consistency:

$$\sum_{j=1}^{J_1} \delta_j = s_1 \text{ and } \sum_{j=J_1+1}^{J_1+J_2} \delta_j = s_2 \, .$$

For the calculations, each material layer is subdivided into a number of smaller layers. For each material layer, the number of layers, m$_j$, into which it is divided for the calculations, shall be decided.

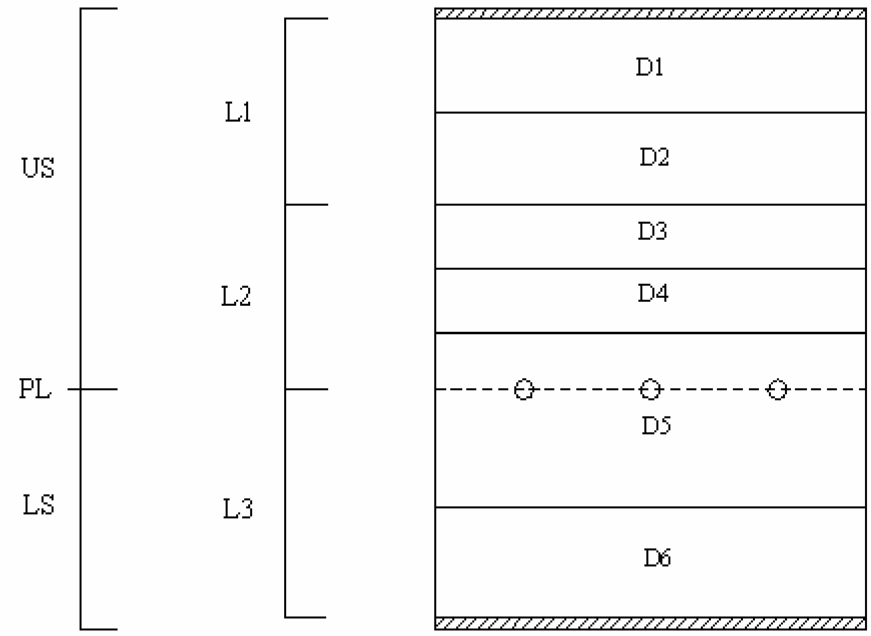


Figure A.1 – Example of subdivision of the slab, where D = Division, L = Layer, US = Upper part of the slab, LS = Lower part of the slab, PL = Pipes level

Each division inherits the physical properties from the material layer to which it belongs. Thus, if the k-th division belongs to the j-th layer, then $\lambda_{D,k} = \lambda_j$, $\rho_{D,k} = \rho_j$ and $c_{D,k} = c_j$.

Divisions are used as thermal nodes in this method. The heat fluxes and temperatures pertaining the divisions are calculated for studying the capability of the system. In order to perform such calculations, each division is characterized by four main physical magnitudes:

— thermal inertia $C_{D,k}$, which is calculated by taking into account the thickness of the division $\tau_{D,k} = \frac{\delta_j}{m_j}$:

$$C_{D,k} = \rho_{D,k} \cdot c_{D,k} \cdot \tau_{D,k} = \rho_j \cdot c_j \cdot \frac{\delta_j}{m_j}$$

— thermal resistance $RU_{D,k}$, which connects the present division with the boundary of the upper division:

$$RU_{D,k} = \frac{\left(\frac{\tau_{D,k}}{2}\right)}{\lambda_{D,k}} = \frac{\delta_j}{2 \cdot m_j \cdot \lambda_j}$$

— thermal resistance $RL_{D,k}$, which connects the present division with the boundary of the lower division;

$$RL_{D,k} = RU_{D,k} = \frac{\left(\frac{\tau_{D,k}}{2}\right)}{\lambda_{D,k}} = \frac{\delta_j}{2 \cdot m_j \cdot \lambda_j}$$

— heat transfer coefficient $HC_{D,k}$ between the present division and the circuit; the value is $HC_{D,k} = \frac{1}{R_t}$, if the element borders on the pipes level, otherwise the value is 0.

As is seen, two divisions border on the pipes level and, thus, they share the thermal resistance R_t, and it is not possible to determine the heat flux passing through R_t by means of a single division. In order to avoid this difficulty, the two divisions bordering on the pipes level are joined together, and constitute one single division in the calculations.

As a consequence, the divisions involved in the calculations are $\sum_{i=1}^{J} m_j - 1$. For clarity, this sum of divisions is named i_L, thus $i_L = \sum_{i=1}^{J} m_j - 1$.

Where a mere resistance layer is present, only R_j shall be declared.

The division crossing the pipes level is the i_P-th division, where $i_P = \sum_{i=1}^{J_1} m_j$. The temperature of this division is important for the connection between slab and circuit, as seen in A.1, where the pipes level temperature is named $\overline{\theta}_C^n$. As a definition, $\overline{\theta}_C^n = \theta_{I,i_P}^n$, so that only θ_{I,i_P}^n is used.

As a consequence, the slab of Figure A.1 can be converted into the following RC network:

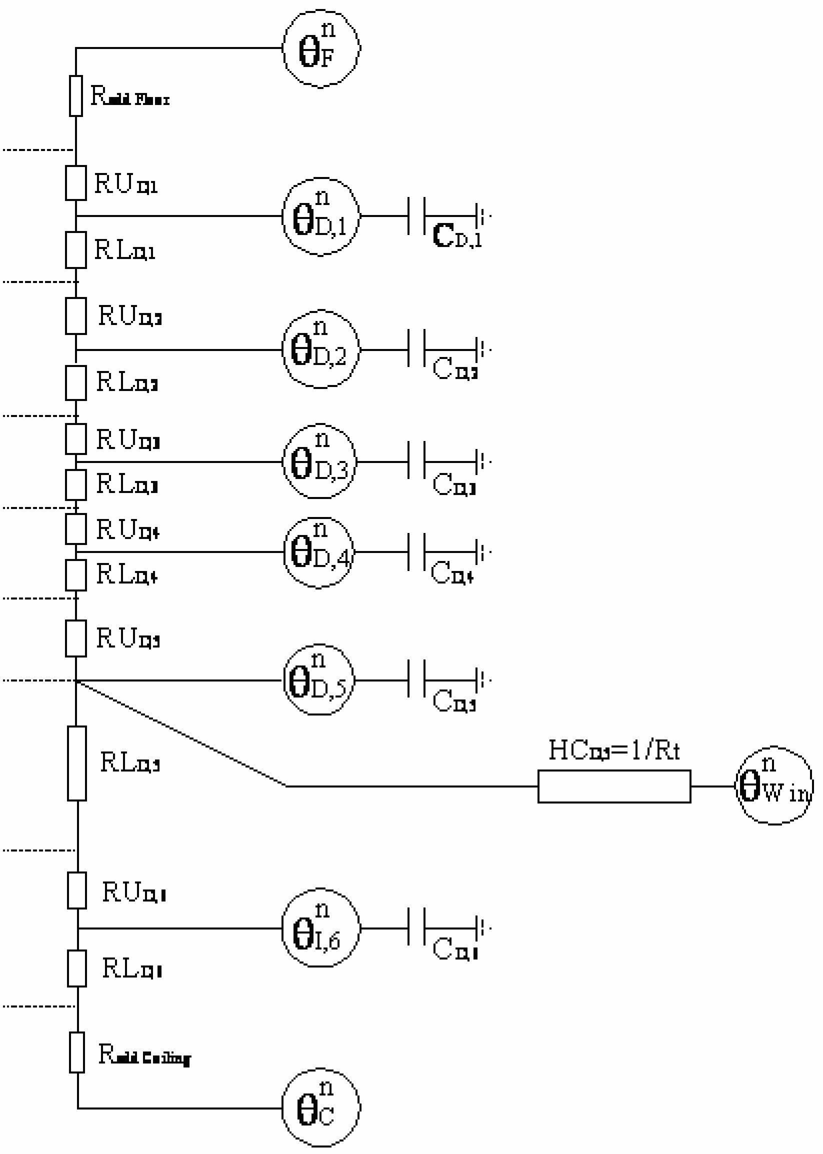


Figure A.2 – Equivalent RC network

Consequently, the following values characterizing the divisions are defined:

Division "*i*", with $1 < i < i_L$ and $i \neq i_P$:

$$C_{D,i} = \rho_j \cdot c_j \cdot \frac{\delta_j}{m_j}$$

$$RU_{D,i} = \frac{\delta_j}{2 \cdot m_j \cdot \lambda_j} + \frac{R_{j-1}}{2}$$ if D,i is the upper element of the j-th layer and the (j-1)-th layer is a mere resistance

$$RU_{D,i} = \frac{\delta_j}{2 \cdot m_j \cdot \lambda_j}$$

if D,i is neither the upper element of the j-th layer nor the lower one, or if the (j-1)-th layer is not a mere resistance

$$RL_{D,i} = \frac{\delta_j}{2 \cdot m_j \cdot \lambda_j} + \frac{R_{j+1}}{2}$$

if D,i is the lower element of the j-th layer and the (j+1)-th layer is a mere resistance

$$RL_{D,i} = \frac{\delta_j}{2 \cdot m_j \cdot \lambda_j}$$

if D,i is neither the upper element of the j-th layer nor the lower one, or if the (j+1)-th layer is not a mere resistance

$$HC_{D,i} = 0$$

Interface “i_P”:

$$C_{D,i_P} = \rho_{J_1} \cdot c_{J_1} \cdot \frac{\delta_{J_1}}{m_{J_1}} + \rho_{J_1+1} \cdot c_{J_1+1} \cdot \frac{\delta_{J_1+1}}{m_{J_1+1}}$$

$$RU_{D,i_P} = \frac{\delta_{J_1}}{2 \cdot m_{J_1} \cdot \lambda_{J_1}} + \frac{R_{J_1-1}}{2}$$

if D,i is the upper element of the J1-th layer and the (J1-1)-th layer is a mere resistance

$$RU_{D,i_P} = \frac{\delta_{J_1}}{m_{J_1} \cdot \lambda_{J_1}}$$

if D,i is neither the upper element of the J1-th layer nor the lower one, or if the (J1-1)-th layer is not a mere resistance

$$RL_{D,i_P} = \frac{\delta_{J_1+1}}{2 \cdot m_{J_1+1} \cdot \lambda_{J_1+1}} + \frac{R_{J_1+2}}{2}$$

if D,i is the lower element of the (J1+1)-th layer and the (J1+2)-th layer is a mere resistance

$$RL_{D,i_P} = \frac{\delta_{J_1+1}}{m_{J_1+1} \cdot \lambda_{J_1+1}}$$

if D,i is neither the upper element of the (J1+1)-th layer nor the lower one, or if the (J1+2)-th layer is not a mere resistance

$$HC_{D,i_P} = \frac{1}{R_t}$$

A.3. Choice of the calculation time step:

The calculation time step shall be chosen in order to avoid calculations instability. A safe value of the calculation time step is evaluated around 40 s.

A.4. Calculations for the generic n-th time step

The values of $\dot{Q}^n_{Sun}$, $\dot{Q}^n_{Transm}$, $\dot{Q}^n_{Air}$, $\dot{Q}^n_{IntRad}$ and $\dot{Q}^n_{IntConv}$ shall be known for the whole day. $\dot{Q}^n_{Sun}$ and $\dot{Q}^n_{Transm}$ can be calculated by other software (through commercial software enabling calculation of the cooling loads of a room with a constant room temperature equal to 24°C). $\dot{Q}^n_{IntRad}$, $\dot{Q}^n_{IntConv}$ and $\dot{Q}^n_{Air}$ depend on the people and the equipment in the room and on the possible air circuit, and are thus known.

For every time step, the running strategy of the circuit f_{rm}^{n} shall be decided before the simulation is started, and the supply water temperature θ_W^n is an input as well. These parameters are chosen by the designer and by performing the simulation with different sets of parameters, it is possible to approach the best combination of running strategy of the circuit and supply water temperature.

For beginning of the simulation, initial values of temperatures of the slab, $\theta_{D,i}^0$ (with $1 \le i \le i_L$), temperature of the air, θ_{Air}^0 , temperature of the walls, θ_{Walls}^0 , supply water temperature, θ_W^0 , and outlet water temperature, $\theta_{W\,exit}^0$, shall be defined. These are only initial values and do not influence the subsequent results, as long as the simulation time is sufficiently long.

The following shortcuts are useful in the subsequent calculations:

$$RCAC = \frac{1}{h_{Air-Ceiling}} + R_{add\ Ceiling} + RL_{D,i_L}$$

$$RRWC = \frac{1}{h_{Ceiling-Walls}} + R_{add\ Ceiling} + RL_{D,i_L} + R_{Walls} \cdot \frac{A_{Floor}}{A_{Walls}}$$

$$RRFC = \frac{1}{h_{Floor-Ceiling}} + R_{add\ Floor} + R_{add\ Ceiling} + RL_{D,i_L} + RU_{D,1}$$

$$RCAW = \frac{1}{h_{Air-Walls}} + R_{Walls}$$

$$RRWF = \frac{1}{h_{Floor-Walls}} + R_{add\ Floor} + RU_{D,1} + R_{Walls} \cdot \frac{A_{Floor}}{A_{Walls}}$$

$$RCAF = \frac{1}{h_{Air-Floor}} + R_{add\ Floor} + RU_{D,1}$$

$$with$$

$$h_{Ceiling-Walls} = 4 \cdot \sigma \cdot 300^3 \cdot F_{v\ Floor-Walls}$$

$$h_{Floor-Ceiling} = 4 \cdot \sigma \cdot 300^3 \cdot F_{v\ Floor-Ceiling}$$

For the n-th time step, the following calculations shall be executed:

Determination of supply water temperature:

$$\theta_W^n = \theta_{W\,exit}^{n-1} + \frac{P_W^{Max}}{\dot{m}_{H,sp} \cdot c_w \cdot A_{Floor}} \ ,\ if\ \ \theta_{W\,exit}^{n-1} + \frac{P_W^{Max}}{\dot{m}_{H,sp} \cdot c_w \cdot A_{Floor}} > \theta_W^{\lim}$$

$$\theta_W^n = \theta_W^{\lim} \ ,\ if\ \theta_{W\,exit}^{n-1} + \frac{P_W^{Max}}{\dot{m}_{H,sp} \cdot c_w \cdot A_{Floor}} < \theta_W^{\lim}$$

Calculation of the heat loads acting towards the room:

$$\dot{Q}^{n}_{Conv} = 0.15 \cdot \dot{Q}^{n}_{Transm} + \dot{Q}^{n}_{IntConv}$$

$$\dot{Q}^{n}_{Rad} = 0.85 \cdot \dot{Q}^{n}_{Transm} + \dot{Q}^{n}_{Sun} + \dot{Q}^{n}_{IntRad}$$

Calculation of the air temperature necessary in order to transfer all convective gains to the surfaces surrounding the room:

$$\theta^{n}_{Air} = \frac{\dot{Q}^{n}_{Conv} - \dot{Q}^{n}_{Air} + \dfrac{A_{Walls}}{RCAW} \cdot \theta^{n-1}_{Walls} + \dfrac{A_{Floor}}{RCAF} \cdot \theta^{n-1}_{D,1} + \dfrac{A_{Floor}}{RCAC} \cdot \theta^{n-1}_{D,i_L}}{\dfrac{A_{Walls}}{RCAW} + \dfrac{A_{Floor}}{RCAF} + \dfrac{A_{Floor}}{RCAC}}$$

Calculation of the heat loads acting on the surfaces:

$$\dot{Q}^{n}_{RadW} = \dot{Q}^{n}_{Rad} \cdot \frac{A_{Walls}}{2 \cdot A_{Floor} + A_{Walls}}$$

$$\dot{Q}^{n}_{RadF} = \dot{Q}^{n}_{Rad} \cdot \frac{A_{Floor}}{2 \cdot A_{Floor} + A_{Walls}}$$

$$\dot{Q}^{n}_{RadC} = \dot{Q}^{n}_{Rad} \cdot \frac{A_{Floor}}{2 \cdot A_{Floor} + A_{Walls}}$$

$$\dot{Q}^{n}_{RadWF} = \frac{\left(\theta^{n-1}_{Walls} - \theta^{n-1}_{D,1}\right)}{RRWF} \cdot A_{Floor}$$

$$\dot{Q}^{n}_{RadWC} = \frac{\left(\theta^{n-1}_{Walls} - \theta^{n-1}_{D,i_L}\right)}{RRWC} \cdot A_{Floor}$$

$$\dot{Q}^{n}_{RadFC} = \frac{\left(\theta^{n-1}_{D,1} - \theta^{n-1}_{D,i_L}\right)}{RRFC} \cdot A_{Floor}$$

$$\dot{Q}^{n}_{ConvW} = \frac{\left(\theta^{n}_{Air} - \theta^{n-1}_{Walls}\right)}{RCAW} \cdot A_{Walls}$$

$$\dot{Q}^{n}_{ConvF} = \frac{\left(\theta^{n}_{Air} - \theta^{n-1}_{D,1}\right)}{RCAF} \cdot A_{Floor}$$

$$\dot{Q}^{n}_{ConvC} = \frac{\left(\theta^{n}_{Air} - \theta^{n-1}_{D,i_L}\right)}{RCAC} \cdot A_{Floor}$$

$$\dot{q}^{n}_{OnFloor} = \frac{\dot{Q}^{n}_{RadF} + \dot{Q}^{n}_{RadWF} - \dot{Q}^{n}_{RadFC} + \dot{Q}^{n}_{ConvF}}{A_{Floor}}$$

$$\dot{q}^{n}_{OnCeiling} = \frac{\dot{Q}^{n}_{RadC} + \dot{Q}^{n}_{RadWC} + \dot{Q}^{n}_{RadFC} + \dot{Q}^{n}_{ConvC}}{A_{Floor}}$$

$$\dot{q}^{n}_{OnWalls} = \frac{\dot{Q}^{n}_{RadW} - \dot{Q}^{n}_{RadWF} - \dot{Q}^{n}_{RadWC} + \dot{Q}^{n}_{ConvW}}{A_{walls}}$$

Calculation of the temperature of the walls and the temperatures of the slab:

$$\theta^n_{Walls} = \frac{\left(\dot{Q}^n_{RadW} - \dot{Q}^n_{RadWF} - \dot{Q}^n_{RadWC} + \dot{Q}^n_{ConvW}\right)\cdot \Delta t}{C_{Walls}\cdot A_{Walls}} + \theta^{n-1}_{Walls}$$

$$\theta^n_{D,1} = \frac{\left(\dot{q}^n_{OnFloor} + \frac{\left(\theta^{n-1}_{D,2} - \theta^{n-1}_{D,1}\right)}{RL_{D,1} + RU_{D,2}}\right)\cdot \Delta t}{C_{D,1}} + \theta^{n-1}_{D,1}$$

$$\theta^n_{D,i} = \frac{\left(\frac{\left(\theta^{n-1}_{D,i-1} - \theta^{n-1}_{D,i}\right)}{RL_{D,i-1} + RU_{D,i}} + \frac{\left(\theta^{n-1}_{D,i+1} - \theta^{n-1}_{D,i}\right)}{RL_{D,i} + RU_{D,i+1}} + f^n_{rm}\left(\theta^n_W - \theta^{n-1}_{D,i}\right)\cdot HC_{D,i}\right)\cdot \Delta t}{C_{I,i}} + \theta^{n-1}_{D,i}$$

$$\theta^n_{D,i_L} = \frac{\left(\dot{q}^n_{OnCeiling} + \frac{\left(\theta^{n-1}_{D,i_L-1} - \theta^{n-1}_{D,i_L}\right)}{RL_{D,i_L-1} + RU_{D,i_L}}\right)\cdot \Delta t}{C_{D,i_L}} + \theta^{n-1}_{D,i_L}$$

With $2 \le i \le i_L - 1$

$$\theta^n_F = \dot{q}^n_{OnFloor}\cdot\left(R_{add\,Floor} + RU_{D,1}\right) + \theta^n_{D,1}$$

$$\theta^n_C = \dot{q}^n_{OnCeiling}\cdot\left(R_{add\,Ceiling} + RL_{D,i_L-1}\right) + \theta^n_{D,i_L}$$

$$\theta^n_W = \dot{q}^n_{OnWalls}\cdot\left(R_{Walls}\right) + \theta^n_{Walls}$$

Calculation of outlet water temperature:

$$\theta^n_{W\,exit} = \theta^n_W - \frac{\left(\theta^n_W - \theta^{n-1}_{D,i_P}\right)}{R_t}\cdot \dot{m}_{H,sp}\cdot c_w \cdot A_{Floor}\ ,\ \ if\ f^n_{rm} = 1$$

$$\theta^n_{W\,exit} = \theta^n_{D,i_P}\ ,\ \ if\ f^n_{rm} = 0$$

Calculation of operative temperature:

$$\theta^n_{Op} = \frac{\theta^n_{Air} + \frac{\theta^n_F\cdot A_{Floor} + \theta^n_C\cdot A_{Floor} + \theta^n_W\cdot A_{Walls}}{\left(2\cdot A_{Floor} + A_{Walls}\right)}}{2}$$

A.5. Sizing of the system

The allowed range for the operative temperature of the room is 20°C to 25.5°C, as the program underestimates the temperature of the room. If the operative temperature is always in this range, the system is well sized, otherwise the running strategy, the supply water temperature or the circuit characteristics have to be changed.

Annex B
(Informative)

Tutorial guide for assessing the model

The following values will be used:

Δt	60 s	Input
$\dot{m}_{H,sp}$	10 kg/(m^2 s)	Input
c_w	4187 J/(kg K)	Input
T	0.2 m	Input
d_a	0.025 m	Input
s_R	0.0025 m	Input
A_{Floor}	15 m^2	Input
L_R	15/0.2 = 75 m	Result
λ_R	0.35 W/(m K)	Input
$\mathrm{R_t}$	0.073 m^{2}K/W	Result
P_W^{Max}	1000 W	Input
θ_w^0	19 °C	Input
$\theta_w^{\lim}$	19 °C	Input
A_{Walls}	33 m^2	Input
$F_{v\ Floor\text{-}Ext\ Wall}$	0.23	Input
$F_{v\ Floor\text{-}Ceiling}$	0.3	Input
$F_{v\ Floor\text{-}Walls}$	0.47	Result
$R_{add\ Floor}$	0.1 (m^2 K)/W	Input
$R_{add\ Ceiling}$	0 (m^2 K)/W	Input
R_{Walls}	0.05 (m^2 K)/W	Input
$h_{Air\text{-}Floor}$	1.5 W/(m^2 K)	Input
$h_{Air\text{-}Ceiling}$	5.5 W/(m^2 K)	Input

$h_{Air\text{-}Walls}$	2.5 W/(m^2 K)	Input
$h_{Floor\text{-}Walls}$	2.88 W/(m^2 K)	Result
$h_{Floor\text{-}Ceiing}$	1.84 W/(m^2 K)	Result
C_{Walls}	10600 J/(m^2 K)	Input
$\mathrm{T_{comfort}}$	25.5 °C	Input
$\dot{Q}^{n}_{Sun}$	300 W	Input
$\dot{Q}^{n}_{Transm}$	90 W	Input
$\dot{Q}^{n}_{Air}$	0 W	Input
$\dot{Q}^{n}_{IntRad}$	400 W	Input
$\dot{Q}^{n}_{IntConv}$	600 W	Input
f_{rm}	1	Input
s_1	0.14 m	Input
s_2	0.1 m	Input
J_1	3	Input
J_2	1	Input
ρ_1	700 kg/m^3	Input
c_1	2300 J/(kg K)	Input
λ_1	0.17 W/(m K)	Input
δ_1	0.04 m	Input
m_1	2	Input
R_1	0 (m^2 K)/W	Input
ρ_1	0 kg/m^3	Input
c_2	0 J/(kg K)	Input
λ_2	0 W/(m K)	Input
δ_2	0 m	Input
m_2	0	Input
R_2	0.18 (m^2 K)/W	Input

ρ_3	2000 kg/m^3	Input
c_3	880 J/(kg K)	Input
λ_3	1.9 W/(m K)	Input
δ_3	0.1 m	Input
m_3	3	Input
R_3	0 (m^2 K)/W	Input
ρ_4	2000 kg/m^3	Input
c_4	880 J/(kg K)	Input
λ_4	1.9 W/(m K)	Input
δ_4	0.1 m	Input
m_4	3	Input
R_4	0 (m^2 K)/W	Input
θ_{Walls}^{n-1}	24 °C	Result of calculations at the previous time step
$\theta_{D,1}^{n-1}$	22.5 °C	Result of calculations at the previous time step
$\theta_{D,2}^{n-1}$	22.3 °C	Result of calculations at the previous time step
$\theta_{D,3}^{n-1}$	21.5 °C	Result of calculations at the previous time step
$\theta_{D,4}^{n-1}$	21.4 °C	Result of calculations at the previous time step
$\theta_{D,5}^{n-1}$	21.3 °C	Result of calculations at the previous time step
$\theta_{D,6}^{n-1}$	21.4 °C	Result of calculations at the previous time step
$\theta_{D,7}^{n-1}$	21.5 °C	Result of calculations at the previous time step
θ_{Walls}^{n}	24.074 °C	Result
$\theta_{D,1}^{n}$	22.521 °C	Result
$\theta_{D,2}^{n}$	22.297 °C	Result
$\theta_{D,3}^{n}$	21.5027 °C	Result

$\theta^{n}_{D,4}$	21.4019 °C	Result
$\theta^{n}_{D,5}$	21.287 °C	Result
$\theta^{n}_{D,6}$	21.4019 °C	Result
$\theta^{n}_{D,7}$	21.54 °C	Result
θ^{n}_{F}	24.58 °C	Result
θ^{n}_{C}	21.94 °C	Result
θ^{n}_{W}	24.73 °C	Result

Annex C (Informative)

Computer program

```
Program TC228_R5_RES_EL_OK

USE DFLIB

implicit none




! Definition of the Types in the main


Type      Layer                                  ! Definition of each layer

          Character*1    Kind                    ! "M" if it is a material layer; "R" if it is a pure resistance
                                                 layer; Every "R" layer must be bounded by two "M"
                                                 layers

          Integer        NElements               ! Number of parts into which the layer must be divided
                                                 in order to perform the calculations

          Real           Thickness               ! Thickness of the layer [m]

          Real           Lambda                  ! Conductivity of the material constituting the layer
                                                 [W/(m K)]

          Real           SpecHeat                ! Specific heat of the material constituting the layer
                                                 [J/(kg K)]

          Real           Rho                     ! Density of the material constituting the layer [Kg/m3]

          Real           Resistance              ! Resistance of the layer: to be compiled only if
                                                 Kind="R", otherwise its value is 0 [(m2 K)/W]

          Integer        InitialElement          ! Upper element belonging to the layer

          Integer        FinalElement            ! Lower element belonging to the layer

          Real           ElThickness             ! Thickness of each element of the layer [m]

End Type  Layer
```

```
Type      El                                    ! Definition of each element constituting the slab

          Real        Capacity                  ! Thermal capacity assigned to the present element
                                                [J/K]

          Real        ResistanceUp              ! Resistance connecting the present element with the
                                                upper one [(m2 K)/W]

          Real        ResistanceDown            ! Resistance connecting the present element with the
                                                lower one [(m2 K)/W]

          Integer     ExtH                      ! Possible connection of the present element with the
                                                circuit [(m2 K)/W]: 1 if the present element is at the
                                                pipes level, otherwise 0

End Type  El



Type      HeatLoadsAndCircuit                   ! Definition of the boundary conditions for loads, water
                                                temperature and running mode

          Integer     Time                      ! Final time of the present time step [s]

          Integer     RunningMode               ! Hydronic circuit running mode in the present time step
                                                [1/0]

          Real        Twater                    ! Inlet water temperature in the present time step [°C]

          Real        RadiantHeatFlux           ! Radiant heat flux imposed in the room in the present
                                                time step [W]

          Real        ConvectiveHeatFlux        ! Convective heat flux imposed in the room in the
                                                present time step [W]

          Real        QAir                      ! Convective heat flux extracted by the primary air
                                                circuit [W]

EndType HeatLoadsAndCircuit



! Definition of the variables involved by the main


Type      (Layer)::   Layers(1:20)              ! Maximum number of layers constituting the slab = 20

Type      (El)::      Element(1:50)             ! Maximum number of interfaces dividing the slab = 50

Type (HeatLoadsAndCircuit):: Boundary(0:320000) ! Maximum number of time steps for input of heat loads
                                                and other boundary conditions = 320000
```

Real	FvFloorToCeiling	
Real	hFloorToCeiling	! Radiant coefficient Floor-Ceiling [W/(m2 K)]
Real	hAirToFloor	! Convective coefficient Air-Floor [W/(m2 K)]
Real	hAirToCeiling	! Convective coefficient Air-Ceiling [W/(m2 K)]
Real	UpperResistance	! Additional resistance on the floor (such as carpets or moquette) [(m2 K)/W]
Real	LowerResistance	! Additional resistance covering the ceiling (such as suspended ceiling) [(m2 K)/W]
Real	WallsResistance	! Resistance related to the walls node
Real	Rtot	! Resistance concerning the circuit and connecting the average pipes level temperature with the inlet water temperature [(m2 K)/W]
Real	hAirToWalls	! Convective coefficient Air-Walls [W/(m2 K)]
Real	hSlabToWalls	! Radiant coefficient Walls-Slab [W/(m2 K)]
Real	FvSlabToExtWall	! Radiant coefficient Walls-Slab [W/(m2 K)]
Integer	NLayersUp	! Number of layers constituting the upper part of the slab
Integer	NLayersDown	! Number of layers constituting the lower part of the slab
Integer	UpperElement	! Ordinal number characterizing the upper element: imposed value = 1
Integer	PipesLevelElement	! Ordinal number characterizing the pipes level element
Integer	LowerElement	! Ordinal number characterizing the lower element
Real	FloorArea	! Area of the floor [m2]
Real	AreaWalls	! Area of the walls [m2]
Integer	TimeStep	! Time step for the imposition of boundary conditions [s]
Real	WallsInertia	! Walls thermal inertia per square meter [J/(m2 K)]
Integer NSteps		! Number of time steps used for the input of boundary conditions
Integer	NTimes	! Number of repetitions of the input loads cycle
Integer	TimeCycle	! Time of a single input loads cycle [s]
Integer	TotalTime	! Total time of the performed simulation [s]
Real	Temperatures(1:50,0:640000)	! Temperatures of the elements constituting the slab [°C] (maximum number of calculation time steps = 640000)

```
Real       Tair(0:640000)                ! Temperatures of the room air [°C] (maximum number
                                         of calculation time steps = 640000)

Real       TWalls(0:640000)              ! Temperatures of the walls [°C] (maximum number of
                                         calculation time steps = 640000)

Real       qsOnFloor(0:640000)           ! Global heat fluxes acting onto the floor [W/m2]
                                         (maximum number of calculation time steps = 640000)

Real       qsOnCeiling(0:640000)         ! Global heat fluxes acting onto the ceiling [W/m2]
                                         (maximum number of calculation time steps = 640000)

Real       qsOnWalls(0:640000)           ! Heat fluxes acting onto the walls [W/m2] (maximum
                                         number of calculation time steps = 640000)

Real       qsToCircuit(0:640000)         ! Heat fluxes extracted by the circuit [W/m2] (maximum
                                         number of calculation time steps = 640000)

Real       TOp(0:640000)                 ! Operative temperatures in the room (maximum
                                         number of calculation time steps = 640000)

Integer CalcTimeStep                     ! Calculation time step [s]

Integer NCalcTimeSteps                   ! Number of calculation steps executed in the whole
                                         simulation

Real       TimeInHours                   ! Support value for the output of the results [h]

Real       TsurfW(0:640000)

Real       TsurfF(0:640000)

Real       TsurfC(0:640000)

CHARACTER*32           OutputFile



Integer i                                ! Support counter

Integer j                                ! Support counter

Integer Deleted                          ! Support variable
```

```
! Main ________________________________________________________________________

! Removal of the previous simulation output and creation of the new one (the output file name is ".ris")

! Formats for input and output commands

1 Format(x,f6.3,3x,f6.3,3x,f6.3,3x,f6.3,3x,f6.3,3x,f6.3,3x,f6.3,x,f6.3,x,f6.3)

2 Format(x,i2,3x,a1,3x,i2,3x,f6.4,3x,f6.4,3x,f7.2,3x,f7.2,3x,f7.4,3x,i2,3x,i2,3x,f6.4)

3 Format(x,i9,xx,f13.1,xx,f7.4,xx,f7.4,xx,i1)

4 Format(x,f9.3,x,i1,x,f4.1,x,f6.2,x,f6.2,x,f6.2)

5 Format(x,f9.3,x,f8.3,x,f8.3,x,f8.3,x,f8.3,x,f8.3,x,f8.3)

6 Format(x,f9.3,3x,f6.2,3x,f6.2,12x,f10.2,11x,f10.2)

7 Format(x,a6,6x,i3,6x,i3,6x,i3,6x,i3,6x,i3,6x,i3,6x,i3,6x,i3,6x,i3,6x,i3,6x,i3,6x,i3,6x,i3,6x,i3,6x,i3,6x,i3,6x,i3,6x,
i3,6x,i3,6x,i3,6x,i3,6x,i3,6x,i3,6x,a20)

8 Format(x,i3,6x,i3,6x, f4.1,6x, f4.1,6x,f10.1)

! Call of the three subroutines for reading data, creating interfaces and calculating the thermal behaviour of the
room

Call ReadSlabAndLoads (hFloorToCeiling, hAirToFloor, hAirToCeiling, UpperResistance, LowerResistance,
WallsResistance, Rtot, hAirToWalls, hSlabToWalls, NLayersUp, NLayersDown, Layers, UpperElement,
PipesLevelElement, LowerElement, FloorArea, AreaWalls, TimeStep, WallsInertia, NSteps, NTimes,
TimeCycle, TotalTime, Boundary, FvSlabToExtWall, FvFloorToCeiling, OutputFile, CalcTimeStep)

Deleted = DELFILESQQ (OutputFile)

Open(unit=2, file=OutputFile, status='new')

Call CreateInterfaces (UpperResistance, LowerResistance, NLayersUp, NLayersDown, Layers,
UpperElement, PipesLevelElement, LowerElement, Element)

Call CreateTempAndFluxesTables (hFloorToCeiling, hAirToFloor, hAirToCeiling, UpperResistance,
LowerResistance, WallsResistance, Rtot, hAirToWalls, hSlabToWalls, NLayersUp, NLayersDown, Element,
UpperElement, PipesLevelElement, LowerElement, FloorArea, AreaWalls, TimeStep, WallsInertia, NSteps,
NTimes, TimeCycle, TotalTime, Boundary, Temperatures, Tair, TWalls, QsOnFloor, QsOnCeiling, QsOnWalls,
QsToCircuit, CalcTimeStep, NCalcTimeSteps, TOp, TsurfW, TsurfC, TSurfF)
```

```
! Output printing____________________________________________________________________

! Initial data printing

write(2,*) 'hF2C    hA2F    hA2C     UpRes     LowRes     Rtot   hA2W  hW2S WallsResistance'

write(2,1) hFloorToCeiling, hAirToFloor, hAirToCeiling, UpperResistance, LowerResistance, Rtot, hAirToWalls, hSlabToWalls, WallsResistance

write(2,*)

write(2,*) 'NLayUp        NLayDown     AFloor    AWalls    WallsInertia'

write(2,8) NLayersUp, NLayersDown, FloorArea, AreaWalls, WallsInertia

write(2,*)

! Main interfaces numbers and slab divisions printing

write(2,*) 'UpInterf       PLevelInterf     LowInterf'

write(2,*) UpperElement, PipesLevelElement, LowerElement

write(2,*)

write(2,*) 'NLay  Kind  NParts  LThick  LLamb  LSpecHeat  LRho  LRes  InSurf  FinSurf  ElThick'

do i=1,20

    write(2,2) i, Layers(i).Kind, Layers(i).NElements, Layers(i).Thickness, Layers(i).Lambda, Layers(i).SpecHeat, Layers(i).Rho, Layers(i).Resistance, Layers(i).InitialElement, Layers(i).FinalElement, Layers(i).ElThickness

enddo

write(2,*)

write(2,*) 'i IntCapacity  IntResUp IntResDown IntExtH'

do i=1,23

    write(2,3) i, Element(i).Capacity, Element(i).ResistanceUp, Element(i).ResistanceDown, Element(i).ExtH

enddo

write(2,*)
```

```
! Printing of the time values involved by the calculations


write(2,*) 'TimeStep   NSteps   NTimes  TimeCycle   TotalTime   NCalcTimeSteps'

write(2,*) TimeStep, NSteps, NTimes, TimeCycle, TotalTime, NCalcTimeSteps

write(2,*)


! Boundary loads printing


write(2,*) 'Time   Run  Twat  Rad   Conv  QAir'

do i=0,NSteps*NTimes

     IF (MOD(i*CalcTimeStep,600).eq.0) THEN

     TimeInHours=Boundary(i).Time/3600.0

     write(2,4) TimeInHours, Boundary(i).RunningMode, Boundary(i).Twater, Boundary(i).RadiantHeatFlux,
     Boundary(i).ConvectiveHeatFlux, Boundary(i).QAir

     ENDIF

enddo

write(2,*)


! Main temperatures printing


write(2,*) ' Time   TFloor    TCore    TCeiling    Tair    TOp     TWalls'

do i=0,NCalcTimeSteps

     IF (MOD(i*CalcTimeStep,600).eq.0) THEN

     TimeInHours=i*CalcTimeStep/3600.0

     write(2,5)      TimeInHours, TSurfF(i), Temperatures(PipesLevelElement,i), TSurfC(i), Tair(i), TOp(i),
     TSurfW(i)

     ENDIF

enddo

write(2,*)
```

```
! Main heat fluxes printing


write(2,*) 'Time     QsOnFloor         QsOnCeiling       QsOnWalls       QsToCircuit'

do i=0,NCalcTimeSteps

     IF (MOD(i*CalcTimeStep,600).eq.0) THEN

     TimeInHours=i*CalcTimeStep/3600.0

     write(2,6) TimeInHours, QsOnFloor(i), QsOnCeiling(i), QsOnWalls(i), QsToCircuit(i)

     ENDIF

enddo


write(6,*) 'Fine'

stop

end



! SUBROUTINES ____________________________________________________________________________


! Subroutine "ReadSlabAndLoads": it reads the values of materials, characteristics of the circuit and boundary
conditions, according with an external file named "InitialData.txt" and enclosed in the present Standard


Subroutine ReadSlabAndLoads (hFloorToCeiling, hAirToFloor, hAirToCeiling, UpperResistance,
LowerResistance, WallsResistance, Rtot, hAirToWalls, hSlabToWalls, NLayersUp, NLayersDown, Layers,
UpperElement, PipesLevelElement, LowerElement, FloorArea, AreaWalls, TimeStep, WallsInertia, NSteps,
NTimes, TimeCycle, TotalTime, Boundary, FvSlabToExtWall, FvFloorToCeiling, OutputFile, CalcTimeStep)


Implicit none


! Definition of the Types involved by "ReadSlabAndLoads"


Type          Layer

              Character*1       Kind
```

```
          Integer        NElements

          Real           Thickness

          Real           Lambda

          Real           SpecHeat

          Real           Rho

          Real           Resistance

          Integer        InitialElement

          Integer        FinalElement

          Real           ElThickness

End Type  Layer


Type      HeatLoadsAndCircuit

          Integer        Time

          Integer        RunningMode

          Real           Twater

          Real           RadiantHeatFlux

          Real           ConvectiveHeatFlux

          Real           QAir

EndType HeatLoadsAndCircuit


! Definition of the variables involved by "ReadSlabAndLoads"


          Type           (Layer)::          Layers(1:20)

          Type           (HeatLoadsAndCircuit):: Boundary(0:320000)


          Real           hFloorToCeiling

          Real           FvFloorToCeiling

          Real           hAirToFloor

          Real           hAirToCeiling
Real            UpperResistance
Real            LowerResistance
Real            WallsResistance
Real            Rtot
Real            hAirToWalls
Real            hSlabToWalls
Real            FvSlabToExtWall
Integer         NLayersUp
Integer         NLayersDown
Integer         UpperElement
Integer         PipesLevelElement
Integer         LowerElement
Real            FloorArea
Real            AreaWalls
Integer         TimeStep
Real            WallsInertia
Integer         NSteps
Integer         NTimes
Integer         TimeCycle
Integer CalcTimeStep
Integer         TotalTime
Character*32    OutputFile
Real            Trash                   ! Support variable
Character*100 TrashC                    ! Support variable
Integer i                               ! Support counter
```

```
! Subroutine "ReadSlabAndLoads"

! Opening the input file and inizialization of the variables

    Open(unit=1, file='ART_W_TS_CONT_KOS.txt', status='old')

    hFloorToCeiling = 0
    hAirToFloor = 0
    hAirToCeiling = 0
    UpperResistance = 0
    LowerResistance = 0
    FvFloorToCeiling = 0
    FvSlabToExtWall = 0
    Rtot = 0
    hAirToWalls = 0
    hSlabToWalls = 0
    NLayersUp = 0
    NLayersDown = 0
    do i = 1, 20
        Layers(i).Kind = 'N'
        Layers(i).NElements = 0
        Layers(i).Thickness = 0
        Layers(i).Lambda = 0
        Layers(i).SpecHeat = 0
        Layers(i).Rho = 0
        Layers(i).Resistance = 0
        Layers(i).InitialElement = 0
        Layers(i).FinalElement = 0
        Layers(i).ElThickness = 0
        enddo

        UpperElement = 0

        PipesLevelElement = 0

        LowerElement = 0

        FloorArea = 0

        AreaWalls = 0

        TimeStep = 0

        WallsInertia = 0

        NSteps = 0

        NTimes = 0

        TimeCycle = 0

        CalcTimeStep = 0

        TotalTime = 0

        do i = 0, 320000

                Boundary(i).Time = 0

                Boundary(i).RunningMode = 0

                Boundary(i).Twater = 0

                Boundary(i).RadiantHeatFlux = 0

                Boundary(i).ConvectiveHeatFlux = 0

                Boundary(i).QAir = 0

        enddo


! Reading the input data from the file "InitialData.txt"


        READ(1,*)

        READ(1,*)

        READ(1,*)

        Read(1,*) OutputFile

        READ(1,*)
READ(1,*) FvFloorToCeiling, FvSlabToExtWall, hAirToFloor, hAirToCeiling, hAirToWalls, Rtot,
UpperResistance, LowerResistance, WallsResistance

READ(1,*)

hSlabToWalls = (1-FvFloorToCeiling-FvSlabToExtWall)*4*300**3*5.67/10**8*0.9

hFloorToCeiling = FvFloorToCeiling*4*300**3*5.67/10**8*0.9

Read (1,*) TrashC, NLayersUp

Read (1,*)

Read (1,*)


do i = 1, NLayersUp

    Read (1,*) Layers(i).Kind, Layers(i).NElements, Layers(i).Thickness, Layers(i).Lambda,
    Layers(i).SpecHeat, Layers(i).Rho, Layers(i).Resistance

    if (Layers(i).Kind.eq.'R') then

        Layers(i).NElements = 0

        Layers(i).Thickness = 0

        Layers(i).Lambda = 0

        Layers(i).SpecHeat = 0

        Layers(i).Rho = 0

    endif

    if (i.eq.1) then

        Layers(i).InitialElement = 1

    else

        if (Layers(i).Kind.eq.'R') then

            Layers(i).InitialElement = Layers(i-1).FinalElement

        else

            Layers(i).InitialElement = Layers(i-1).FinalElement+1

        endif

    endif

    if (Layers(i).Kind.eq.'R') then

        Layers(i).FinalElement = Layers(i).InitialElement
        else
                Layers(i).FinalElement = Layers(i).InitialElement + Layers(i).NElements-1
        endif
        if (Layers(i).Kind.eq.'R') then
                Layers(i).ElThickness = 0
        else
                Layers(i).ElThickness = Layers(i).Thickness/Layers(i).NElements
        endif
enddo

do i = 1, 6-NLayersUp
        Read (1,*)
enddo

Read (1,*) TrashC, NLayersDown
Read (1,*)
Read (1,*)

do i = NLayersUp+1, NLayersUp+NLayersDown
        Read (1,*) Layers(i).Kind, Layers(i).NElements, Layers(i).Thickness, Layers(i).Lambda,
        Layers(i).SpecHeat, Layers(i).Rho, Layers(i).Resistance
        if (Layers(i).Kind.eq.'R') then
                Layers(i).NElements = 0
                Layers(i).Thickness = 0
                Layers(i).Lambda = 0
                Layers(i).SpecHeat = 0
                Layers(i).Rho = 0
        endif
        if (i.eq.NLayersUp+1) then
                Layers(i).InitialElement = Layers(i-1).FinalElement
```

```
          else
                    if (Layers(i).Kind.eq.'R') then
                              Layers(i).InitialElement = Layers(i-1).FinalElement
                    else
                              Layers(i).InitialElement = Layers(i-1).FinalElement+1
                    endif
          endif
          if (Layers(i).Kind.eq.'R') then
                    Layers(i).FinalElement = Layers(i).InitialElement
          else
                    Layers(i).FinalElement = Layers(i).InitialElement + Layers(i).NElements - 1
          endif
          if (Layers(i).Kind.eq.'R') then
                    Layers(i).ElThickness = 0
          else
                    Layers(i).ElThickness = Layers(i).Thickness/Layers(i).NElements
          endif
enddo

UpperElement = 1
PipesLevelElement = Layers(NLayersUp).FinalElement
LowerElement = Layers(NLayersUp+NLayersDown).FinalElement

do i = 1, 5-NLayersDown
          Read (1,*)
enddo

Read(1,*) TrashC, FloorArea, TrashC, TrashC, AreaWalls
Read(1,*)
```

```
    Read(1,*) TrashC, TrashC, TimeStep, TrashC, TrashC, WallsInertia

    Read(1,*)

    Read(1,*) TrashC, NSteps

    Read(1,*) TrashC, CalcTimeStep

    Read(1,*) TrashC, NTimes

    Read(1,*)

    Read(1,*)


    do i=0,NSteps

        Boundary(i).Time=TimeStep*i

        Read(1,*) Trash, Trash, Trash, Trash, Trash, Trash, Boundary(i).ConvectiveHeatFlux,
        Boundary(i).RadiantHeatFlux, Boundary(i).Qair, Boundary(i).RunningMode, Boundary(i).Twater

    enddo



! Creation of the total list of boundary conditions, taking into account the number of times the boundary load
conditions must be repeated


    TimeCycle = NSteps*TimeStep

    TotalTime = TimeCycle * NTimes

    do i=NSteps+1,NTimes*NSteps

        Boundary(i).Time = i*TimeStep

        Boundary(i).RunningMode = Boundary(Mod(i,NSteps)).RunningMode

        Boundary(i).Twater = Boundary(Mod(i,NSteps)).Twater

        Boundary(i).RadiantHeatFlux = Boundary(Mod(i,NSteps)).RadiantHeatFlux

        Boundary(i).ConvectiveHeatFlux = Boundary(Mod(i,NSteps)).ConvectiveHeatFlux

        Boundary(i).QAir = Boundary(Mod(i,NSteps)).QAir

    enddo


return
```

```
EndSubroutine



! Subroutine "CreateInterfaces": it uses the input data concerning the slab in order to define the characteristics
of each interface dividing the slab


Subroutine CreateInterfaces (UpperResistance, LowerResistance, NLayersUp, NLayersDown, Layers,
UpperElement, PipesLevelElement, LowerElement, Element)


Implicit none


! Definition of the Types involved by "CreateInterfaces"


Type        Layer

            Character*1     Kind

            Integer         NElements

            Real            Thickness

            Real            Lambda

            Real            SpecHeat

            Real            Rho

            Real            Resistance

            Integer         InitialElement

            Integer         FinalElement

            Real            ElThickness

End Type    Layer


Type        El

            Real            Capacity

            Real            ResistanceUp

            Real            ResistanceDown
```

```
                Integer          ExtH

End Type        El


! Definition of the variables involved by "CreateInterfaces"


                Type             (Layer)::            Layers(1:20)

                Type             (El)::               Element(1:50)


                Real             UpperResistance

                Real             LowerResistance

                Integer          NLayersUp

                Integer          NLayersDown

                Integer UpperElement

                Integer PipesLevelElement

                Integer LowerElement

                Integer          NElementsUp

                Integer          NElementsDown

                Integer i                             ! Support counter

                Integer j                             ! Support counter

                Integer k                             ! Support counter



! Subroutine "CreateInterfaces"____________________________________________________________


                NElementsUp = Layers(NLayersUp).FinalElement

                NElementsDown = Layers(NLayersUp+NLayersDown).FinalElement-
                                                      Layers(NLayersUp).FinalElement


! Inizialization of the variables
```

```
          do  i = 1, 50

                    Element(i).Capacity = 0

                    Element(i).ResistanceUp = 0

                    Element(i).ResistanceDown = 0

                    Element(i).ExtH = 0

          enddo


! Definition of the characteristics of the first element (starting from the floor)


          Element(1).Capacity = Layers(1).ElThickness*Layers(1).SpecHeat*Layers(1).Rho

          Element(1).ResistanceUp = UpperResistance + (Layers(1).ElThickness/2)/Layers(1).Lambda

          if ((1.eq.Layers(1).FinalElement).and.(Layers(2).Kind.eq.'R')) then

                    Element(1).ResistanceDown =
                    Layers(2).Resistance/2+(Layers(1).ElThickness/2)/Layers(1).Lambda

          else

                    Element(1).ResistanceDown = (Layers(1).ElThickness/2)/Layers(1).Lambda

          endif

          Element(1).ExtH = 0


! Definition of the characteristics of the middle interfaces (starting from the floor)


          do  i = 2, NElementsUp-1

                    do j=1, NLayersUp

                    if (((i.ge.Layers(j).InitialElement).and.(i.le.Layers(j).FinalElement)).and.(Layers(j).Kind.ne.'R'))
                    then

                              Element(i).Capacity = Layers(j).ElThickness*Layers(j).SpecHeat*Layers(j).Rho

                              if ((i.eq.Layers(j).InitialElement).and.(Layers(j-1).Kind.eq.'R')) then

                                        Element(i).ResistanceUp = Layers(j-1).Resistance/2+(Layers(j).ElThickness/2) /
                                        Layers(j).Lambda

                              else

                              Element(i).ResistanceUp = (Layers(j).ElThickness/2)/Layers(j).Lambda
```

```
            endif

            if ((i.eq.Layers(j).FinalElement).and.(Layers(j+1).Kind.eq.'R')) then

                Element(i).ResistanceDown = Layers(j+1).Resistance/2+(Layers(j).ElThickness/2) / Layers(j).Lambda

            else

            Element(i).ResistanceDown = (Layers(j).ElThickness/2)/Layers(j).Lambda

            endif

            Element(i).ExtH = 0

            goto 10

        endif

10      enddo

    enddo



    Element(NElementsUp).Capacity = Layers(NLayersUp).ElThickness*Layers(NLayersUp).SpecHeat*Layers(NLayersUp).Rho + Layers(NLayersUp+1).ElThickness*Layers(NLayersUp+1).SpecHeat*Layers(NLayersUp+1).Rho

    if ((NElementsUp.eq.Layers(NLayersUp).InitialElement).and.(Layers(NLayersUp-1).Kind.eq.'R')) then

        Element(NElementsUp).ResistanceUp = Layers(NLayersUp-1).Resistance/2+(Layers(NLayersUp).ElThickness)/Layers(NLayersUp).Lambda

    else

        Element(NElementsUp).ResistanceUp = (Layers(NLayersUp).ElThickness)/Layers(NLayersUp).Lambda

    endif

    if ((NElementsUp.eq.Layers(NLayersUp+1).FinalElement).and.(Layers(NLayersUp+2).Kind.eq.'R')) then

        Element(NElementsUp).ResistanceDown = Layers(NLayersUp+2).Resistance/2+(Layers(NLayersUp+1).ElThickness)/Layers(NLayersUp+1).Lambda

    else

        Element(NElementsUp).ResistanceDown = (Layers(NLayersUp+1).ElThickness)/Layers(NLayersUp+1).Lambda

    endif

    Element(NElementsUp).ExtH = 1
```

```
          do  i = NElementsUp+1, NElementsUp+NElementsDown

                do j=NLayersUp+1, NLayersUp+NLayersDown

                if (((i.ge.Layers(j).InitialElement).and.(i.le.Layers(j).FinalElement)).and.(Layers(j).Kind.ne.'R'))
                then

                      Element(i).Capacity = Layers(j).ElThickness*Layers(j).SpecHeat*Layers(j).Rho

                      if ((i.eq.Layers(j).InitialElement).and.(Layers(j-1).Kind.eq.'R')) then

                            Element(i).ResistanceUp = Layers(j-1).Resistance/2+(Layers(j).ElThickness/2) /
                            Layers(j).Lambda

                      else

                            Element(i).ResistanceUp = (Layers(j).ElThickness/2)/Layers(j).Lambda

                      endif

                      if ((i.eq.Layers(j).FinalElement).and.(Layers(j+1).Kind.eq.'R')) then

                            Element(i).ResistanceDown = Layers(j+1).Resistance/2+(Layers(j).ElThickness/2)
                            / Layers(j).Lambda

                      else

                            Element(i).ResistanceDown = (Layers(j).ElThickness/2)/Layers(j).Lambda

                      endif

                      Element(i).ExtH = 0

                      goto 11

                endif

11              enddo

          enddo


! Definition of the characteristics of the first element (starting from the ceiling)


          Element(LowerElement).Capacity =
          Layers(NLayersUp+NLayersDown).ElThickness*Layers(NLayersUp+NLayersDown).SpecHeat*Layers(
          NLayersUp+NLayersDown).Rho

          if ((LowerElement.eq.Layers(NLayersUp+NLayersDown).InitialElement) .and.
          (Layers(NLayersUp+NLayersDown-1).Kind.eq.'R')) then

                Element(LowerElement).ResistanceUp = Layers(NLayersUp+NLayersDown-
                1).Resistance/2+(Layers(NLayersUp+NLayersDown).ElThickness/2)/Layers(NLayersUp+NLayer
                sDown).Lambda
```

```
        else

                Element(LowerElement).ResistanceUp =
                (Layers(NLayersUp+NLayersDown).ElThickness/2)/Layers(NLayersUp+NLayersDown).Lambda

        endif

        Element(LowerElement).ResistanceDown = LowerResistance +
        (Layers(NLayersUp+NLayersDown).ElThickness/2)/Layers(NLayersUp+NLayersDown).Lambda

        Element(LowerElement).ExtH = 0



return

EndSubroutine




! Subroutine "CreateTempAndFluxesTables": it calculates the values of temperatures and heat fluxes of air,
slab and walls



Subroutine CreateTempAndFluxesTables (hFloorToCeiling, hAirToFloor, hAirToCeiling, UpperResistance,
LowerResistance, WallsResistance, Rtot, hAirToWalls, hSlabToWalls, NLayersUp, NLayersDown, Element,
UpperElement, PipesLevelElement, LowerElement, FloorArea, AreaWalls, TimeStep, WallsInertia, NSteps,
NTimes, TimeCycle, TotalTime, Boundary, Temperatures, Tair, TWalls, QsOnFloor, QsOnCeiling, QsOnWalls,
QsToCircuit, CalcTimeStep, NCalcTimeSteps, TOp, TsurfW, TsurfC, TSurfF)



Implicit none



! Definition of the Types involved by "CreateTempAndFluxesTables"



Type        El

            Real        Capacity

            Real        ResistanceUp

            Real        ResistanceDown

            Integer     ExtH

End Type    El



Type        HeatLoadsAndCircuit
```

```
          Integer       Time
          Integer       RunningMode
          Real          Twater
          Real          RadiantHeatFlux
          Real          ConvectiveHeatFlux
          Real          QAir
EndType HeatLoadsAndCircuit


! Definition of the variables involved by "CreateTempAndFluxesTables"


          Type(El)::    Element(1:50)
          Type (HeatLoadsAndCircuit)::      Boundary(0:320000)


          Real          hFloorToCeiling
          Real          hAirToFloor
          Real          hAirToCeiling
          Real          UpperResistance
          Real          LowerResistance
          Real          WallsResistance
          Real          Rtot
          Real          hAirToWalls
          Real          hSlabToWalls
          Integer       NLayersUp
          Integer       NLayersDown
          Integer UpperElement
          Integer PipesLevelElement
          Integer LowerElement
          Real          FloorArea
          Real          AreaWalls
```

```
Integer         TimeStep
Real            WallsInertia
Integer         NSteps
Integer         NTimes
Integer         TimeCycle
Integer TotalTime
Integer         CalcTimeStep
Integer NCalcTimeSteps
Real            Temperatures(1:50,0:640000)
Real            Tair(0:640000)
Real            TWalls(0:640000)
Real            qsOnFloor(0:640000)
Real            qsOnCeiling(0:640000)
Real            qsOnWalls(0:640000)
Real            qsToCircuit(0:640000)
Real            TOp(0:640000)
Integer         RunningMode          ! Support variable
Real            Twater               ! Support variable
Real            RadiantHeatFlux      ! Support variable
Real            ConvectiveHeatFlux   ! Support variable
Real            QAir                 ! Support variable
Real            qOnFloor             ! Support variable
Real            qOnCeiling           ! Support variable
Real            RCAC                 ! Convective thermal resistance Air-Ceiling
Real            RRWC                 ! Radiant thermal resistance Walls-Ceiling
Real            RRFC                 ! Radiant thermal resistance Floor-Ceiling
Real            RCAW                 ! Convective thermal resistance Air-Walls
Real            RRWF                 ! Radiant thermal resistance Walls-Floor
Real            RCAF                 ! Convective thermal resistance Air-Floor
```

```
        Real       QRadW              ! Radiant heat loads acting onto the walls
        Real       QRadF              ! Radiant heat loads acting onto the floor
        Real       QRadC              ! Radiant heat loads acting onto the ceiling
        Real       QRadWF             ! Radiant heat flux acting from the walls onto the floor
        Real       QRadWC             ! Radiant heat flux acting from the walls onto the ceiling
        Real       QRadFC             ! Radiant heat flux acting from the floor onto the ceiling
        Real       QConvW             ! Convective heat loads acting onto the walls
        Real       QConvF             ! Convective heat loads acting onto the floor
        Real       QConvC             ! Convective heat loads acting onto the ceiling
        Real       TSurfW(0:640000)
        Real       TSurfC(0:640000)
        Real       TSurfF(0:640000)

        Integer i                     ! Support counter
        Integer j                     ! Support counter
        Integer k                     ! Support counter


! Subroutine "CreateTempAndFluxesTables"____________________________________________

! Calculation of the number of times the calculation must be performed

    NCalcTimeSteps=TotalTime/CalcTimeStep

! Inizialization of the variables

    do i = 0,640000
        do j = 1,50
            Tair(i) = 0
```

```
                        TWalls(i)= 0

                        Temperatures(j,i)= 0

                enddo


                TOp(i) = 0

                TWalls(i) = 0

                qsOnFloor(i) = 0

                qsOnCeiling(i) = 0

                qsOnWalls(i) = 0

                qsToCircuit(i) = 0

                TSurfW(i) = 0

                TSurfF(i) = 0

                TSurfC(i) = 0

        enddo


        do j=1,50

                Temperatures(j,0)=22.

        enddo


        TOp(0) = 22.

        Tair(0) = 22.

        TWalls(0) = 22.

        TSurfW(0) = 22.

        TSurfF(0) = 22.

        TSurfC(0) = 22.


        do i=1, NCalcTimeSteps

                do j=1, NTimes*NSteps

                        if ((i*CalcTimeStep.gt.Boundary(j-1).Time).and.(i*CalcTimeStep.le.Boundary(j).Time))
                        then
```

```
                    RunningMode=Boundary(j).RunningMode

                    Twater=Boundary(j).Twater

                    RadiantHeatFlux=Boundary(j).RadiantHeatFlux

                    ConvectiveHeatFlux=Boundary(j).ConvectiveHeatFlux

                    QAir=Boundary(j).QAir

                    goto 12

               endif

12          enddo


! Calculation of the involved resistances


            RCAC = (1/hAirToCeiling + Element(LowerElement).ResistanceDown)

            RRWC = (1/hSlabToWalls + WallsResistance*FloorArea/AreaWalls +
            Element(LowerElement).ResistanceDown )

            RRFC = (1/hFloorToCeiling + Element(1).ResistanceUp +
            Element(LowerElement).ResistanceDown)

            RCAW = (1/hAirToWalls + WallsResistance)

            RRWF = (1/hSlabToWalls + Element(1).ResistanceUp + WallsResistance*FloorArea/AreaWalls)

            RCAF = (1/hAirToFloor + Element(1).ResistanceUp)


! Calculation of the air temperature


            Tair(i) = (ConvectiveHeatFlux + QAir + AreaWalls/RCAW * TWalls(i-1) + FloorArea/RCAF *
            Temperatures(1,i-1) + FloorArea/RCAC * Temperatures(LowerElement,i-1)) / (AreaWalls/RCAW
            + FloorArea/RCAF + FloorArea/RCAC)


! Calculation of the heat fluxes acting on the internal surfaces


            QRadW = RadiantHeatFlux * AreaWalls/(2*FloorArea + AreaWalls)

            QRadF = RadiantHeatFlux * FloorArea/(2*FloorArea + AreaWalls)

            QRadC = RadiantHeatFlux * FloorArea/(2*FloorArea + AreaWalls)
```

```
          QRadWF = (TWalls(i-1) - Temperatures(1,i-1)) / RRWF * FloorArea

          QRadWC = (TWalls(i-1) - Temperatures(LowerElement,i-1)) / RRWC * FloorArea

          QRadFC = (Temperatures(1,i-1) - Temperatures(LowerElement,i-1)) / RRFC * FloorArea


          QConvW = (Tair(i)-TWalls(i-1))/RCAW * AreaWalls

          QConvF = (Tair(i)-Temperatures(1,i-1))/RCAF * FloorArea

          QConvC = (Tair(i)-Temperatures(LowerElement,i-1))/RCAC * FloorArea


          qOnFloor = (QRadF + QRadWF - QRadFC + QConvF)/FloorArea

          qOnCeiling = (QRadC + QRadWC + QRadFC + QConvC)/FloorArea


! Calculation of the temperatures of the walls and the slab interfaces


          TWalls(i) = (QRadW - QRadWF - QRadWC + QConvW)*CalcTimeStep/ (WallsInertia *
          AreaWalls) +  TWalls(i-1)


          Temperatures(1,i)=(qOnFloor+(Temperatures(2,i-1)-Temperatures(1,i-1)) /
          (Element(1).ResistanceDown+Element(2).ResistanceUp))*CalcTimeStep/Element(1).Capacity +
          Temperatures(1,i-1)


          do k=2,LowerElement-1

                Temperatures(k,i)= ((Temperatures(k-1,i-1)-Temperatures(k,i-1)) /
                (Element(k).ResistanceUp+Element(k-1).ResistanceDown)+(Temperatures(k+1,i-1) -
                Temperatures(k,i-1))/(Element(k).ResistanceDown+Element(k+1).ResistanceUp) +
                (Twater-Temperatures(k,i-1))*RunningMode/Rtot*Element(k).ExtH)*CalcTimeStep /
                Element(k).Capacity+Temperatures(k,i-1)

          enddo


          Temperatures(LowerElement,i)=(qOnCeiling+(Temperatures(LowerElement-1,i-1) -
          Temperatures(LowerElement,i-1))/(Element(LowerElement).ResistanceUp +
          Element(LowerElement-1).ResistanceDown))*CalcTimeStep/Element(LowerElement).Capacity +
          Temperatures(LowerElement,i-1)
```

```
                    TSurfF(i)=QOnFloor*Element(1).ResistanceUp+Temperatures(1,i)

                    TSurfC(i)=QOnCeiling*Element(LowerElement).ResistanceDown+Temperatures(LowerElement,i)

                    TSurfW(i) = TWalls(i) + (QRadW - QRadWF - QRadWC + QConvW)/AreaWalls *
                    WallsResistance


! Last outputs definition


                    qsOnFloor(i)=qOnFloor

                    qsOnCeiling(i)=qOnCeiling

                    qsOnWalls(i) = (QRadW - QRadWF - QRadWC + QConvW) / FloorArea

                    QsToCircuit(i)=(Twater-Temperatures(PipesLevelElement,i))*RunningMode/Rtot

                    TOp(i)=((TSurfF(i)*FloorArea+TSurfC(i)*FloorArea+TSurfW(i)*AreaWalls) /
                    (FloorArea*2+AreaWalls)+Tair(i))/2


          enddo


return

EndSubroutine
```

Bibliography

Hauser, G., Kempkes, Ch., Olesen, B. W. (2000), Computer Simulation of the Performance of a Hydronic Heating and Cooling System with Pipes Embedded into the Concrete Slab between Each Floor. ASHRAE Trans. V. 106, pt.1

Koschenz, M., Lehman, B. (2000). „Thermoaktive Bauteilsysteme tabs", ISBN 3-905594-19-6

Meierhans, R.A. (1993), Slab cooling and earth coupling, ASHRAE Trans. V. 99, Pt 2.

Meierhans, R. A. (1996), Room air conditioning by means of overnight cooling of the

concrete ceiling. ASHRAE Trans. V. 102, Pt. 2.

Olesen, B. W., Sommer, K. and Düchting, B. (2002), "Control of slab heating and cooling systems studied by dynamic computer simulations., ASHRAE Trans. V.108, Pt.2.

Olesen, B.W., Dossi, F.C.(2004): Operation and Control of Activated Slab Heating and Cooling Systems, CIB World Building Congress 2004, May 2004.

Olesen, B. W., Koschenz, M., Johansson, C.(2003): New European Standard Proposal for Design and Dimensioning of Embedded Radiant Surface heating and Cooling Systems. ASHRAE Transactions, Volume 109, Part 2, 2003.