Annex 28 Low Energy Cooling July 2000

# Low Energy Cooling

# Detailed Design Tools

Edited by
Henk Roel

IEA Energy Conservation in Buildings and Community Systems Programm

This document is one of a series produced by Annex 28 to assist with the design of low energy cooling systems. The other documents are:

"Review of Low Energy Cooling Technologies"
"Selection Guidance for Low Energy Cooling Technologies"
"Early Design Guidance for Low Energy Cooling Technologies"
"Case Studies of Low Energy Cooling Technologies".

For further information contact:

Nick Barnard
Oscar Faber Applied Research
United Kingdom
Tel: +44 208 784 5784
Fax: +44 208 784 5700

Denice Jaunzens
Building Research Establishment
United Kingdom
Tel: +44 1923 664522
Fax: +44 1923 664095

# Detailed Design Tools for Low Energy Cooling Technologies

# IEA Annex 28 - Subtask 2 Final report

# ACKNOWLEDGEMENTS

Thanks are due to the following organisations for their assistance in the management of IEA Annex 28:

The United Kingdom's Building Research Establishment Ltd (BRE) and the Department of Environment, Transport and the Regions (DETR) for their funding under the Sustainable Construction business plan of the Construction Research and Innovation programme, and Oscar Faber Applied Research as sub-contractors in support of the Operating Agent's role.

## SUMMARY

The aim of Annex 28 is to investigate the feasibility of, and provide design tools/guidance on the application of alternative cooling strategies to buildings. Outputs from the Annex include a review of the technologies, design tools and case study descriptions. This report is a compilation of tools developed for use during detailed design. The tools have been contributed by the individual member countries participating in the Annex.

There are a number of different types of tool including component models, air temperature/flow models and control algorithms. The majority are intended to be used as part of, or in conjunction with, simulation software. In general they have been developed using experimental data and/or theoretical relationships. A summary of the tools is given in the Table overleaf.

A common structure (based on the ASHRAE toolkit format) has been used for detailing the tools. Copies of the source code and executable files (where appropriate) are provided on the enclosed diskette.

Summary table of Detailed Design Tools

| Detailed Design Tool | N° | Summary | Source Code |
|---|---|---|---|
| Desiccant cooling | A | Model to investigate the performance of desiccant cooling systems. | Excel |
| Desiccant + evaporative cooling algorithms and control strategies | B | Model of a desiccant dehumidification wheel based on manufacturers data plus theoretical models of other system elements. | DOE Function |
| Evaporative cooling in office buildings | C | Strategy for control of direct and indirect evaporative coolers in conjunction with heating and cooling coils. | Turbo Pascal |
| Design tools for evaporative cooling | D | Models of direct and indirect evaporative coolers based on laboratory studies and published literature. | Fortran |
| Evaporative cooling | E | *Spreadsheet for processing outputs from* simulation to estimate the effect of introducing evaporative cooling on internal conditions, loads, energy and water consumptions. | Excel |
| Displacement ventilation and chilled ceiling multi-node model | F | Multi-node airflow model based on published data and simulation results. | HSLights |
| Night cooling control strategies in commercial buildings | G | Three rule based control strategies | N.A. |
| Night ventilation in residential buildings | H | Estimation of ventilation rate as a function of free opening area, outdoor noise, securety, occupation and solar shading using look-up tables based on monitored data. | Fortran77 |
| Seasonal groundwater cold water storage | I | System performance model based on theoretical rules and measured data. | N.A. |
| Programme for the simulation of air-earth heat exchangers | J | Theoretical model of a air-ground heat exchanger validated using monitored data. | Quick Basic |
| Slab cooling system, water cooled | K | Theoretical model of a water cooled slab validated using monitored data plus a theoretical model of a wet cooling tower. | Fortran |
| False floor slab, air cooled | L | Theoretical R/C network model of a hollow core slab. | Excel |
| Design tool for an absorption cooling machine | M | Model of an absorption chiller based on manufacturers data | Basic |

# TABLE OF CONTENTS

## PREFACE

The International Energy Agency (IEA) was established in 1974 within the framework of the Organisation for Economic Co-operation and Development (OECD) to implement an International Energy Programme. A basic aim of the IEA is to foster co-operation among the twenty-one IEA Participating Countries to: increase energy security through energy conservation, the development of alternative energy sources and energy research, development and demonstration (RD&D). This is achieved in part through a programme of collaborative RD&D consisting of forty-two Implementing Agreements, containing a total of over eighty separate energy RD&D projects. This publication forms one element of this programme.

### Energy Conservation in Buildings and Community Systems

The IEA sponsors research and development in a number of areas related to energy. In one of these areas, energy conservation in buildings, the IEA is sponsoring various exercises to predict more accurately the energy use of buildings, including comparison of existing computer programs, building monitoring, comparison of calculation methods, as well as air quality and studies of occupancy. Seventeen countries have elected to participate and have designated contracting parties to the Implementing Agreement covering collaborative research in this area. The designation by governments of a number of private organisations, as well as universities and government laboratories as contracting parties, has provided a broader range of expertise to tackle the projects in the different technology areas than would have been the case if participation was restricted to governments. The importance of associating industry with government sponsored energy research and development is recognised in the IEA, and every effort is made to encourage this trend.

### The Executive Committee

Overall control of the programme is maintained by the Executive Committee (ExCo) and the Implementation Agreement on Energy Conservation in Buildings and Community Systems (B&CS), which not only monitors existing projects but also identifies new areas where collaborative effort may be beneficial. The Executive Committee ensures that all projects fit into a pre-determined strategy, without unnecessary overlap or duplication but with effective liaison and communication. The Executive Committee has initiated the following projects to date (completed Annexes are identified by *):

| | |
|---|---|
| I | Load Energy Determination of Buildings* |
| II | Ekistics and Advanced Community Energy Systems* |
| III | Energy Conservation in Residential Buildings* |
| IV | Glasgow Commercial Building Monitoring* |
| V | Air Infiltration and Ventilation Centre |
| VI | Energy Systems and Design of Communities* |
| VII | Local Government Energy Planning* |
| VIII | Inhabitant Behaviour with Regard to Ventilation* |
| IX | Minimum Ventilation Rates* |
| X | Building HVAC Systems Simulation* |
| XI | Energy Auditing* |
| XII | Windows and Fenestration* |
| XIII | Energy Management in Hospitals* |
| XIV | Condensation* |
| XV | Energy Efficiency in Schools* |
| XVI | BEMS - 1: User Guidance* |
| XVII | BEMS - 2: Evaluation and Emulation Techniques* |
| XVIII | Demand Controlled Ventilating Systems* |
| XIX | Low Slope Roof Systems* |
| XX | Air Flow Patterns within Buildings* |
| XXI | Thermal Modelling* |
| XXII | Energy Efficient Communities* |
| XXIII | Multizone Air Flow Modelling (COMIS)* |
| XXIV | Heat Air and Moisture Transfer in Envelopes* |
| XXV | Real Time HVAC Simulation* |
| XXVI | Energy Efficient Ventilation of Large Enclosures* |
| XXVII | *Evaluation and Demonstration of Domestic Ventilation Systems* |
| XXVIII | Low Energy Cooling Systems |
| XXIX | Daylighting in Buildings |

| | |
|---|---|
| XXX | Bringing Simulation to Application |
| XXXI | Energy Related Environmental Impact of Buildings |
| XXXII | Integral Building Envelope Performance Assessment |
| XXXIII | Advanced Local Energy Planning |
| XXXIV | Computer Aided Fault Detection and Diagnosis |
| XXXV | HYBVENT |

**A. Introduction**

Cooling is a significant user of energy in buildings, and its impact as a contributor to greenhouse gas emissions is enhanced by the fact that these systems are usually electrically driven. Increasing use of information technology has led to an increasing demand for cooling in the commercial buildings sector, with consequent problems for utilities companies.

In response to these issues, the IEA's Future Building Forum workshop on Innovative Cooling (held in the United Kingdom in 1992) identified a number of technologies with the potential to reduce energy consumption in the field of alternative cooling strategies and systems, leading to the establishment of Annex 28. The emphasis for the project was on passive and hybrid cooling technologies and strategies. These require close integration of the dynamics of the building structure with the HVAC systems, and this is precisely the area in which the B&CS ExCo has established expertise.

**B. Objective**

Passive and hybrid cooling systems will only be taken up in practice if such systems can be shown to meet certain criteria:

a) the life cycle costs (including energy, maintenance etc.) of such systems are less than conventional systems;

b) the level of thermal comfort provided is acceptable to the occupants in the context of their task;

c) the systems are sufficiently robust to changes in building occupancy and use;

d) the design concepts for such systems are well defined, and appropriate levels of guidance are available at all stages of the design process, from sketch plan to detailed;

e) the necessary design tools are available in a form which designers can use in practice; and

f) the cooling system is shown to integrate with the other systems (e.g. heating and ventilation), as well as with the building and control strategy.

The objective of the Annex was to work towards fulfilling these requirements.

**C. Means**

The project was subdivided into three subtasks relating to the three phases of researching and documenting the various cooling strategies:

Subtask 1: Description of cooling strategies

The aim of this subtask was to establish the current state of the technologies in the participating countries. The findings are detailed in the report:

Review of Low Energy Cooling Technologies

The report also contains national data for climate, building standards, heat gains, comfort criteria, energy and water costs for each of the participating countries.

Subtask 2: Development of Design Tools

Different levels of tool are required throughout the design process. Initially little detailed data will be available and the emphasis will be on tools using rules-of-thumb. Having established suitable options, approximate performance data and practical guidance will be needed for early design and assessment. Finally, when the broad principles of the design have been established, such techniques as simulation modelling can be used for detailed design and optimisation. To reflect these requirements, three different levels of tool have been developed by the Annex:

Selection Guidance for Low Energy Cooling Technologies
This tool provides guidance on the initial selection of suitable low energy technologies. Paper and software (Visual Basic) versions of the tool have been produced.

Early Design Guidance for Low Energy Cooling Technologies
A collection of simplified tools based on design charts/tables and practical guidance to assist with early design development of a technology.

Detailed Design Tools for Low Energy Cooling Technologies (this document)
A collection of tools for use as part of, or in conjunction with, simulation software. Copies of source codes and executable files (where appropriate) are provided with the report on an enclosed diskette.

Subtask 3: Case studies

The third element of the work was to illustrate the various cooling technologies through demonstrated case studies. Approximately 20 case studies have been documented in the Annex report:

Case Studies of Low Energy Cooling Technologies

The case studies give feedback on performance and operation in practice and include design details and monitored performance data.

**D. Scope**

A number of different technologies have been considered by the Annex. The Table overleaf gives an overview of which of the Annex reports have information on which of the technologies.

Overview Table of Low Energy Cooling Technologies included in Annex Reports

| Technology | Review | Selection Guidance | Early Design Guidance | Detailed Design Tools | Case Studies |
|---|---|---|---|---|---|
| Night cooling (natural ventilation) | • | • | • | • | • |
| Night cooling (mechanical ventilation) | • | • | • | • | • |
| Slab cooling (air) | • | • | | • | |
| Slab cooling (water) | • | • | • | • | • |
| Evaporative cooling (direct and indirect) | • | • | • | • | • |
| Desiccant + evaporative cooling | • | • | | • | • |
| Chilled ceilings/beams | • | • | | | • |
| Displacement ventilation | • | • | | • | • |
| Ground cooling (air) | • | • | • | • | • |
| Aquifer | • | • | | • | • |
| Sea/river/lake water cooling | | • | | | • |

**E. Participation**

The participating countries in this task are Canada, Germany, Finland, France, Netherlands, Portugal, Sweden, Switzerland, United Kingdom and the United States of America. The funding groups for each country are given below.

Canada

Buildings Group
CANMET- Energy Technology Branch, NRCan
580 Booth St.
Ottawa, Ontario
K1A 0E4

| | |
|---|---|
| | Heat Management Technologies<br>Energy Diversification Research Laboratory<br>CANMET - Energy Technology Branch, NRCan<br>1615, Montée Ste-Julie<br>C.P. 4800<br>Varennes, Québec<br>J3X 1S6 |
| Germany | Bundesministerium für Bildung Technologie und Forschung (BMBE)<br>Postfach 200240<br>Bonn, Germany |
| Finland | Technology Development Centre<br>P.O. BOX 69<br>Fin - 00101 Helsinki |
| France | Agence de l'environnement et de la maitrise de l'énergie<br>Fédération nationale du bâtiment<br>Ministére de l'équipement - Plan Construction Architecture<br>Centre scientifique et technique du bâtiment<br>Ecole des mines de Paris<br>Gaz de France<br>Costic |
| Netherlands | Novem BV<br>Swentiboldstraat 21<br>P.O. BOX 17<br>6130 AA Sittard |
| Portugal | Center for Energy Conservation<br>Praceta à Estrada de Alfragide<br>Alfragide<br>2700 Amadora<br><br>Department of Mechanical Engineering<br>University of Porto<br>R. Bragas<br>4099 PORTO Codex |
| Sweden | Swedish Council for Building Research<br>P.O. BOX 12866<br>SE - 11298 Stockholm |
| Switzerland | Swiss Federal Office of Energy<br>CH - 3003 Berne |
| United Kingdom | British Gas<br>EA Technology<br>Gardiner & Theobald<br>Haden Young/Balfour Beatty Building<br>MEPC Investments<br>Oscar Faber<br>Ove Arup<br>Department of the Environment, Transport and the Regions<br>Building Research Establishment |
| United States of America | Office of Building Technologies<br>U.S. Department of Energy<br>1001 Independence Avenue<br>Washington DC 20585 |

## INTRODUCTION

This report is a compilation of tools for low energy cooling technologies intended for use during detailed design. It constitutes part of the output of Annex 28 in fulfilling its aim to provide design tools/guidance on the application of alternative cooling strategies to buildings. Design tools have also been developed by the Annex to assist with technology selection and early design. A review of the technologies and case study descriptions have also been produced (refer to Preface for an overview of Annex outputs).

The tools have been contributed by the individual member countries participating in the Annex and reviewed by another participant. There are a number of different types including component models, air temperature/flow models and control algorithms. The majority of the tools are intended to be used as part of or in conjunction with simulation software. For example the component models are intended for incorporation as modules in a simulation software package, whereas the control algorithms would generally be user-defined.

The tools have been documented where possible to a common structure (based on the ASHRAE toolkit format) comprising the following sections:

1. Technology Area
Specification of the technology to which the tool relates.

2. Developer
Contact address of tool provider.

3. General Description
An explanation of the purpose of the tool typically incorporating a schematic showing the system elements and their interaction plus an information flow diagram with algorithm inputs, outputs and parameters.

4. Nomenclature
Definition of the mathematical variables used in the mathematical description and the code variables used in the source code. (Note: Units and nomenclature are consistent within each tool but not necessarily between tools.)

5. Mathematical Description
Base equations for the algorithm, describing the relationships between the variables.

6. References

*The source/sources of empirical or non-standard mathematical equations and other data used.*

7. Algorithm

Definition of the structure of the algorithm as a step-by-step procedure detailing the order in which the base equations are calculated.

8. Flow Chart

Pictorial presentation of the calculation procedure defined by the algorithm.

9. Source Code

This is provided for most of the tools but is not appropriate in all cases. Software versions of the source codes and executable files are included on the enclosed diskette.

10. Sample Results

Input and output data is provided to give users an illustration of how tool is intended to be used and what results to expect.

*The tools have been grouped approximately* by technologies. In a few cases the tools can be used in combination, eg "Evaporative cooling control strategy" with "Direct and indirect evaporative coolers".

# IEA Annex 28

## Low Energy Cooling

### Subtask 2: Detailed Design Tools

**Desiccant Cooling**

**Excel oriented tools**

Ecole des Mines de Paris, Centre dÉnergétique
D. Marchio, M. Orphelin, S. Bech

# Contents Chapter A

# 1. Technology area

The purpose of this tool is to present a model to investigate performance of desiccant cooling systems.

The model could be used in conjunction with the following systems:

- a heat exchanger
- a direct evaporative cooler
- an indirect evaporative cooler
- a direct and indirect evaporative cooler
- a cooling coil.

Models for each of these components are available in Chapter E of this report.

# 2. Developed by

**Name** : *Matthieu Orphelin, Dominique Marchio*

**Organisation** : Ecole des Mines de Paris, Centre d'Energétique

**Address** : 60 Boulevard St. Michel, 75272 Paris Cedex 06, France

**Phone** : 33 01 40 51 91 80

**Fax** : 33 01 46 34 24 91

**E-mail** : orphelin@cenerg.ensmp.fr, marchio@cenerg.ensmp.fr

# 3. General description

The purpose of the model is to simulate the performance of desiccant cooling systems. The proposed model can be used in conjunction with a heat exchanger model, evaporative cooler (direct or/and indirect) models or a cooling coil model, as proposed Chapter E of this report.

Inputs are the process and regeneration air properties and flows. Parameters are wheel and desiccant material characteristics. The routine gives for outputs the obtained indoor air conditions.

This way, the calculation can proceed using Excel worksheets and macros.

**INPUTS**

Process air inlet temperature →
Process air inlet humidity ratio →
Process air mass flow rate →
Regeneration air inlet temperature →
*Regeneration air inlet humidity ratio* →
Regeneration air mass flow rate →

**OUTPUTS**

→ Process air outlet temperature
→ Process air inlet humidity ratio

→ Regeneration air outlet temperature
→ *Regeneration air inlet humidity ratio*

→ Energy consumed by heating element
→ Energy consumed by fans

**PARAMETERS**

Heating power
Wheel characteristics
Desiccant material characteristics

# 4. Nomenclature

| | | |
|---|---|---|
| m | Air mass flow rate | [kg/s] |
| d | Air volumic flow rate | [$m^3$/s] |
| T | Dry bulb temperature | [°C] |
| T' | Wet bulb temperature | [°C] |
| Tr | Dew point temperature | [°C] |
| ε,eps | Relative humidity | [%] |
| w | Humidity ratio | [kg water/kg dry air] |
| h,q' | Specific enthalpy | [kJ/kg] |
| c | Specific heat | [kJ/kg/°C] |
| v' | Specific volume | [$m^3$/kg] |
| P | Energy rate | [W] |

**Wheel characteristics**

| | | |
|---|---|---|
| K | Global coefficient of heat exchange | [W/(mK)] |
| D | Wheel diameter | [m] |
| L | Wheel length | [m] |
| f | fraction of the wheel section used for process (generally around 0.75) | |
| τ | Wheel revolution period | [h] |
| Md | Total mass of desiccant material | [kg] |

**Discretisation of the wheel in successive layers**

| | | |
|---|---|---|
| n | number of layers | |
| Mw,u | Unitary moisture transfer during one rotation | [kg] |
| Mi | Moisture transfer for all the $i^{th}$ layer during one rotation | [kg] |
| Mw,i | Total moisture transfer for the $i^{th}$ layer during one hour | [kg/h] |
| La | Load of the material during adsorption | [kg water/kg mater.] |
| Ld | Load of the material during desorption | [kg water/kg mater.] |
| x | a dimensional spatial coordinate (wheel axis direction) | |

**Indices**

| | |
|---|---|
| i | $i^{th}$ layer of the wheel (counted from the inlet of the process air to the outlet of the process air) |
| a1 | Process air entering the wheel |
| ai | Process air in the $i^{th}$ layer of the wheel |
| mai | Material in contact with the process air in the $i^{th}$ layer of the wheel, or air in equilibrium with the material in the $i^{th}$ layer of the wheel |
| a2 | Process air temperature leaving the wheel |
| d0 | Regeneration air entering the heater |
| d1 | Regeneration air entering the wheel |
| di | Regeneration air in the $i^{th}$ layer of the wheel |
| mdi | Material in contact with the regeneration air in the $i^{th}$ layer of the wheel, or air in equilibrium with the material in the $i^{th}$ layer of the wheel |
| d2 | Regeneration air leaving the wheel |

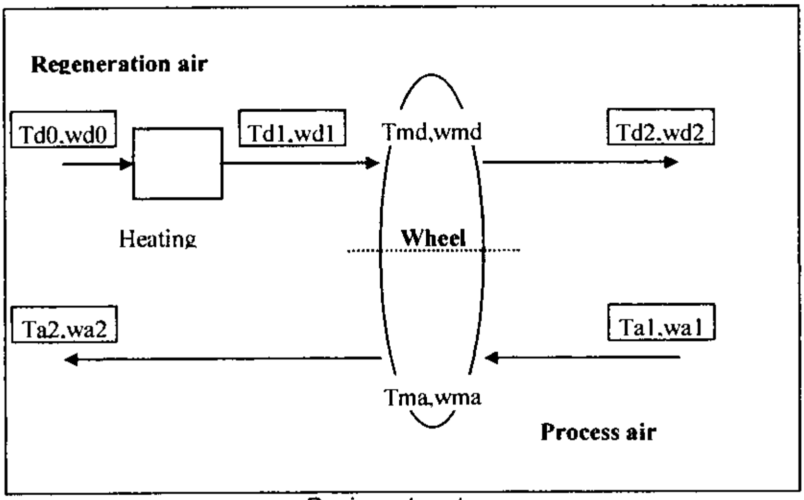
**Energy Balance**

| | | |
|---|---|---|
| P | Total energy rate | [W] |
| Pe | Heater energy rate | [W] |
| Pa | Energy rate of the process air fan | [W] |
| Pd | Energy rate of the regeneration air fan | [W] |
| Pw | Energy rate of the wheel motor | [W] |

**Desiccant material**

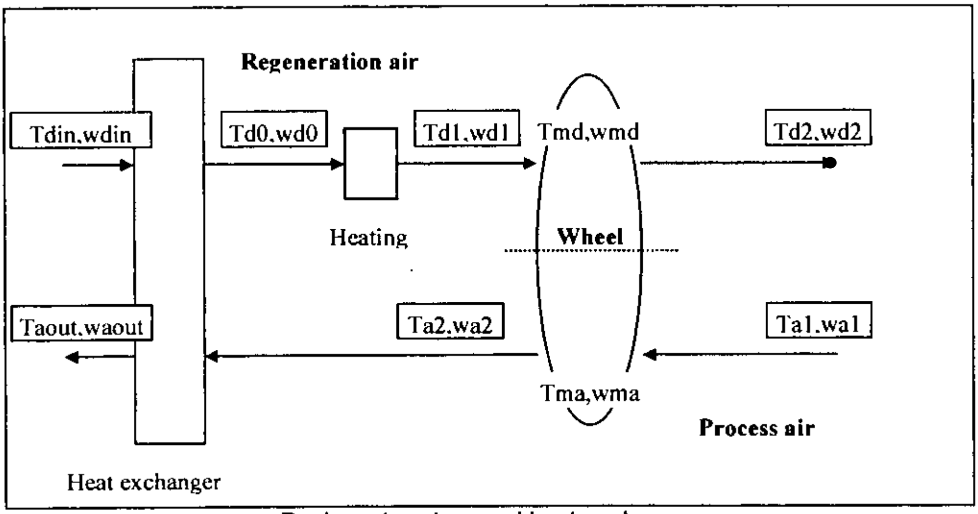
| | | |
|---|---|---|
| Cj | constant coefficients | |
| Ls | Heat of sorption | [kJ/kg sorbed water] |
| LF | Load factor | [kg water/kg matérial] |

The following diagram illustrates the system in its simplest form. The heat source can be electric, gas, solar, heat recovery, or any combination of two of these sources.

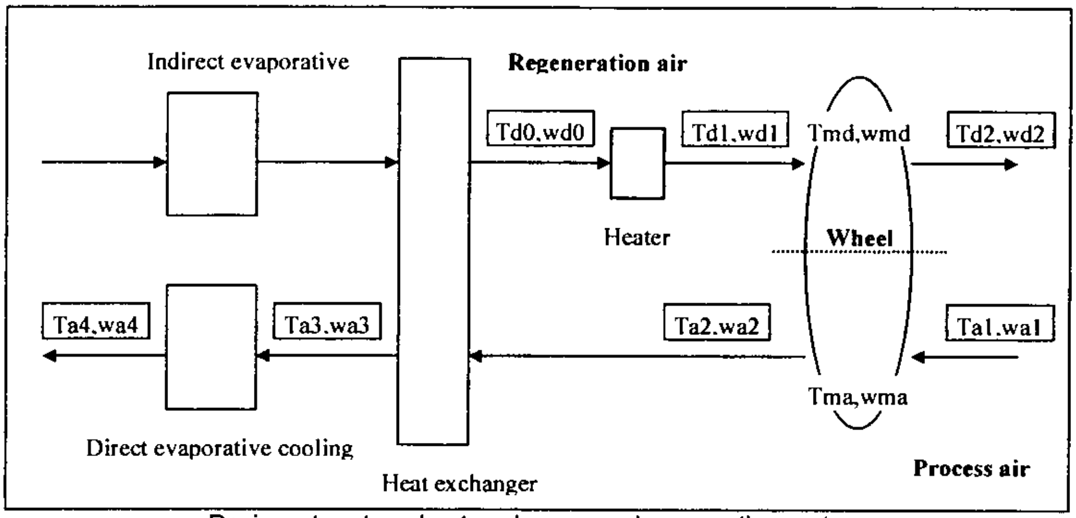



Desiccant system

As mentioned previously, the desiccant system can be used in conjunction with other air handling systems, such as an heat exchanger and an evaporative system. The following figures illustrate this point.




Desiccant system and heat exchanger




Desiccant system, heat exchanger and evaporative system

## 5. Mathematical description

Heat and mass transfers in a desiccant wheel are coupled by the four following differential equations :

1. mass conservation equation
2. mass transfer rate equation
3. energy conservation equation
4. energy transfer rate equation

Boundary conditions are periodical due to the wheel rotation.

In the model, the following assumptions have been made (see [VDBU92] and [MATH80] for more details):

- The desiccant wheel is under steady-state conditions (The resolution of coupled heat and mass transfer and conservation equations in [MATH80] shows that the steady-state conditions is quickly reached, justifying the steady-state assumption.)
- Heat and mass transfer coefficients are constant.
- The desiccant matrix can be considered as a succession of layers.
- Air properties are considered as constant on a surface perpendicular to the flow direction.
- Heat conduction and water diffusion in the axial direction are negligible.
- Flux coupling is neglected.
- The adsorption phenomenon is without hysteresis. In other words, the equilibrium relationships are the same for the adsorption and the desorption.
- No mixing or carry-over of streams occurs.
- There is no radial variation of fluid matrix states.
- Only the steady-state performance only of the dehumidifier is considered.

Air conditions (eg temperature and humidity ratio) vary across the wheel. As adsorption and desorption phenomena are strongly non-linear, assuming a balance between the inlet and the outlet of the wheel is not feasible. A spatial discretization of the wheel in successive layers has to be performed, in order to evaluate the condition of both air streams across the wheel. Ten successive layers have been used to achieve which sufficiently accurate results.

Two different air stream configurations may be analysed (parallel flow and counter flow) giving two different resolution methods, as expressed in the algorithms and equations sections. Use of parallel-flow rather than counter-flow enhances the numerical stability of the calculation but the process is inherently counter-flow.

The intrinsic characteristic of the desiccant materials need to be defined for the modelling.

**Modelling of desiccant materials**

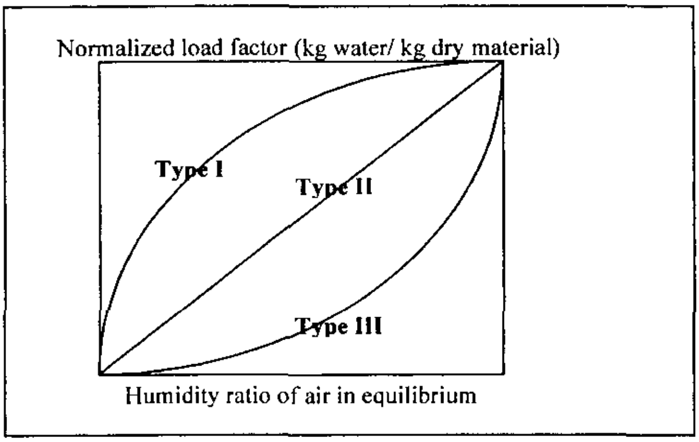
The following table from [CELE84] gives some indicative values for different material types.

| Material | Adsorption heat | Adsorption maximum load (saturated) |
|---|---|---|
| | (kJ/kg water) | kg water / kg material |
| Silica Gel microporous | 2900 | 0.38 |
| Silica Gel macroporous | 2900 | 0.64 |
| Activated alumina | 2900 | 0.40 |
| Molecular sieve | 4180 | 0.26 |

Manufacturers often give desiccant material performances as an isotherm, which gives the water load of the material (LF = load factor, expressed in kg of water per kg of dry material) in equilibrum with air at a given humidity ratio, for a fixed air temperature (eg 25°C).

The assumption of no hysteresis (equilibrium relations are the same for adsorption and desorption) means that the material may be described by this isotherm.

The shape of the isotherm is a characteristic of the material. The International Union of Pure and Applied Chemistry ([STIE95], [COLL92]) distinguishes three isotherms shapes, as represented on the next figure.




Many practical expressions of the load factors have been defined and used in past studies. Some examples are described below. (NB Any expression can be implemented in the model.)

- Equation using an adsorption potential [STIE95]

*(a)* $LF = 0.0385.\exp[-(A/620)^{0.5}] + 0.0460.\exp[-(A/620)^{1.5}]$
where $A = RT.\ln(ps/pv)$ defined as the adsorption potential
$ps$ = saturation vapor pressure at T [K] of material [kpa]
$pv$ = partial steam pressure [kpa]

- Equations using a multi-variable polynomial expression

The following equation is given for a silicagel in [DUPO94] :

*(b)* $eps = S1.T.LF^2 + S2.T.LF + S3.LF^4 + S4.LF^3 + S5.LF^2 + S6.LF$
where $S1 = -0.04031298$
$S2 = +0.02170245$
$S3 = +125.470047$
$S4 = -72.651229$
$S5 = +15.5223665$

S6 = +0.0084266

- Brunauer types [COLL92]

(c) SC = FC / (R + FC - R.FC)
where SC = solid relative concentration
FC = fluid relative concentration
R = separation factor (<1 type I, =1 type II, >1 type III)

- Second-degree polynominal expression

Such a regression is presented in [MATH80]. Some of the silical gel manufacturers give materials performance under the following form:

*(d)* $eps = C1.LF + C2.LF^2$
where eps = relative humidity of air in equilibrium with the material
LF = load factor of the material (in kg water per kg of dry desiccant)
C1 & C2 = Constant coefficients depending on the nature of the desiccant

This equation assumes that temperature does not influence the load factor. In other words, knowing the humidity ratio of the air in equilibrium with the material, the load factor is the root of the second-degree equation.

Any of these expressions (*(a)*, *(b)*, *(c)*, or *(d)*) , or any other, can be implemented in the model, depending on available information. Equation *(d)* has been implemented in the example given.

**Study of the isenthalpic assumption**

References listed in bibliography index indicate that desiccant systems may (or may not) be considered as isenthalpic. First of all we must understand the meaning of this assumption and see what role the isenthalpic efficiency plays.

In this example, the system under consideration is the moist air.

Initial state entering the wheel
Final state leaving the wheel

ha1 = ha1(Ta1,wa1) = ca.Ta1 + wa1(L + cv.Ta1)
ha2 = ha2(Ta2,wa2) = ca.Ta2 + wa2(L + cv.Ta2)

The **isenthalpic** hypothesis concerns the **phenomenon of adsorption** and so therefore we must take into account the heat of sorption (desiccant characteristic) noted Ls in our model. This hypothesis means that all the heat of sorption is transferred to the moist air, which gives :

ha2 = ha1 + Ls.(wa1-wa2) - L.(wa1-wa2)

ha2 - ha1 = (Ls - L).(wa1-wa2)

The heat of vaporization of water (2500 kJ/kg water) and the heat of sorption of the material are close to each other (at least for the particular mediums actually used) which makes the term (Ls - L) small. In this particular case, **the isenthalpic hypothesis, used for the phenomenon of adsorption can be applied to air treatment**. But on the other hand, if the value of heat of sorption of the material is higher (as it is for molecular sieves) we must consider the term (Ls - L).(wa1 - wa2) and calculate the variation of enthalpy.

As mentioned above, heat is transferred from the regeneration air stream to the supply air stream so that in practice the process is not enthalpic. The technologies employed, in particular those to assure good airtightness between the two air streams, enable isenthalpic efficiencies in the order of 0.95 to be achieved (see for example [STIE95] and [BURN85]).

In the model, this efficiency value is accounted for by the following relationship:

$\Delta ha2 = \Delta ha1 / \eta$

This efficiency takes into account the heat losses. The following table, constructed using manufacturers' data, shows that the efficiency defined in this manner varies only slightly with the intial state conditions. In the case study presented below the efficiency is held constant and is equal to 0.95.

| Wheel S | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| nominal air flow rate | | | | | | | | |
| $T_{a1}$ | $w_{a1}$ | $h_{a1}$ | $DT_a$ | $T_{a2}$ | $w_{a2}$ | $h_{a2}$ | $h_{a2} - h_{a1}$ | $\eta$ |
| °C | kg/kg | J/kg | °C | °C | kg/kg | J/kg | J/kg | - |
| 40 | 0,025 | 104646,00 | 23,0 | 63,0 | 0,0175 | 109240,22 | 4594,22 | 1,044 |
| 40 | 0,020 | 91772,80 | 22,5 | 62,5 | 0,0134 | 97994,25 | 6221,44 | 1,068 |
| 40 | 0,015 | 78899,60 | 21,2 | 61,2 | 0,0093 | 85936,51 | 7036,91 | 1,089 |
| 30 | 0,025 | 94114,50 | 23,0 | 53,0 | 0,0189 | 102485,26 | 8370,76 | 1,089 |
| 30 | 0,020 | 81333,60 | 22,5 | 52,5 | 0,0124 | 85079,17 | 3745,57 | 1,046 |
| 30 | 0,015 | 68552,70 | 21,2 | 51,2 | 0,0080 | 72320,92 | 3768,22 | 1,055 |

The standard AICVF equations for moist air calculations are applied. These equations are represented by the following numbers :

*1* w(T,eps)
*2* T(h,w)
*3* h(T,w)
*4* v(T,w)
*5* eps(T,h)

**Regeneration air outlet heater conditions**

*6* $$w_{d1} = w_{d0}$$

7 $$h_{d1} = h_{d0} + \frac{P_e}{m_d}$$

using equation *2* :

$$T_{d1} = T(h_{d1}, w_{d1})$$

**Discretisation of the wheel in n successive layers**

*8* $$M_{d,i} = \frac{M_d}{n} \qquad (n=10)$$

**For each layer of the wheel**

Inlet conditions of process air

with equations *3, 5, 4*:

$$h_{a,i} = h(T_{a,i}, w_{a,i})$$

$$eps_{a,i} = h(T_{a,i}, h_{a,i})$$

$$v_{a,i} = v(T_{a,i}, w_{a,i})$$

*9* $$m_{a,i} = \frac{d_a}{v_{a,i}}$$

Inlet conditions of regeneration air

with equations *3, 5, 4* :

$$h_{d,i} = h(T_{d,i}, w_{d,i})$$

$$eps_{d,i} = eps(T_{d,i}, h_{d,i})$$

$$v_{d,i} = v(T_{d,i}, w_{d,i})$$

*10* $$m_{d,i} = \frac{d_d}{v_{d,i}}$$

Average temperature of desiccant material in equilibrium with process air

*11* $$T_{ma,i} \approx T_{a,i}$$

Average temperature of desiccant material in equilibrium with regeneration air

*12* $$T_{md,i} \approx T_{d,i}$$

Average humidity ratios of air in equilibrium with material in each layer

Process air

*13* $$w_{ma,i} \approx w_{a,i}$$

with equation *5* :

$$eps_{ma,i} \approx eps(T_{a,i}, w_{a,i})$$

Regeneration air

*14* $$w_{md,i} \approx w_{d,i}$$

with equation *5* :

$$eps_{md,i} \approx eps(T_{d,i}, w_{d,i})$$

Average water load factor of the material in each layer i

La, the factor load of the material during the adsorption can be defined by the average quantity of water contained in the material, expressed in kg of water per kg of material. Ld, the factor load of the material during the adsorption, can be defined in the same way. Assuming that the equilibrium is effectively reached in both streams, f. (La - Ld) represents the unitary quantity of water transferred from the process air to the regeneration air.

For the wheel fraction in contact with the process air (adsorption):

15 $$L_a = \frac{-C_1 + \sqrt{(C_1{}^2 + 4.eps_{ma,i}.C_2)}}{2.C_2}$$

For the wheel fraction in contact with the regeneration air (desorption):

16 $$L_d = \frac{-C_1 + \sqrt{(C_1{}^2 + 4.eps_{md,i}.C_2)}}{2.C_2}$$

Equations 15 and 16 are based on material modelling using expression type (d) as described at the beginning of the section. When other expressions are used, these equations should be modified accordingly.

Moisture transfers

Unitary moisture transfer during one revolution of the wheel

17 $$Mw,u = f.(La,i - Ld,i)$$

Moisture transfer for all the $i^{th}$ layers during one revolution of the wheel

18 $$M_i = M_{w,u}.M_{d,i}$$

Total moisture transfer for the $i^{th}$ layer

19 $$M_{w,i} = M_i.\tau$$

Humidity ratios of both air streams leaving the $i^{th}$ layer

20 $$w_{a,i+1} = w_{a,i} - \frac{M_{w,i}}{m_{a,i}}$$

21 $$w_{d,i+1} = w_{d,i} + \frac{M_{w,i}}{m_{d,i}}$$

Conduction exchanges between process and regeneration parts of the material (non-isenthalpic process)

22 $$m_a.(h_{a,i+1} - h_{a,i}) = -K.S.(T_{a,i} - T_{d,i})$$

23 $$m_d.(h_{d,i+1} - h_{d,i}) = K.S.(T_{a,i} - T_{d,i})$$

As described above, one can here introduce an enthalpic efficiency $\eta$ to facilitate the calculation of equations 22 and 23. In the case of a material with a sorption heat which differs significantly from water vaporization heat, these equations should be modified as described above.

Outlet air conditions

with equations *22* and *23* :

$$h_{a,i+1} = h_{a,i} - \frac{K.S.(T_{a,i} - T_{d,i})}{m_{a,i}}$$

$$h_{d,i+1} = h_{d,i} + \frac{K.S.(T_{a,i} - T_{d,i})}{m_{d,i}}$$

Outlet air temperatures

with equation 2 :

$T_{a,i+1} = T(h_{a,i+1}, w_{a,i+1})$

$T_{d,i+1} = T(h_{d,i+1}, w_{d,i+1})$

Outlet relative humidities
with equation 5

$eps_{a,i+1} = eps(T_{a,i+1}, w_{a,i+1})$

$eps_{d,i+1} = eps(T_{d,i+1}, w_{d,i+1})$

**Resolution in the case of parallel flow**

On the first layer

25 $T_{a,i} = T_{a1}$

26 $w_{a,i} = w_{a,1}$

27 $T_{d,i} = T_{d1}$

28 $w_{d,i} = w_{d,1}$

Then, the problem can be solved in all successive layers up to layer n.

**Resolution in the case of counter flow**

On the first layer:

29 $T_{a,i} = T_{a1}$

30 $w_{a,i} = w_{a,1}$

On the last layer:

31 $T_{d,n+1} = T_{d1}$

32 $w_{d,n+1} = w_{d,1}$

A solver has to be used for the resolution of the successive layers:
- guess initial values for the regeneration air characteristics in the first layer.
- calculate all other layers
- compare obtained regeneration air temperature and humidity ratio to imposed initial values on layer n.
- guess new values for regeneration air characteristics in the first layer.

**Removed water**

33 ma.(wa,2 - wa,1)

**Process air temperature rise**

34 (Ta,2 - Ta,1)

**Energy balance**

35 P = Pe + Pa + Pd + Pw

According to manufacturers data, fans energy rates are about 10 % of the regeneration power.

# 6. References

[BURN85] 'Hybrid Dessicant Cooling Systems in Supermarket Application', P. R. Burns, J. W. Mitchell, W. A. Beckman, ASHRAE Transactions, CH-85-09 N°. 5

[CALT92] 'Application of a dessicant cooling system to supermarkets', D.S. Calton, Desiccant Cooling and Dehumidification, ASHRAE Special Publication, ISBN 0-910110-90-5, 1992

[COLL92] 'Desiccant Properties and their Effects on the Performance of Desiccant Cooling Systems', R. K. Collier, B. M. Cohen, R. B. Slosberg, Desiccant Cooling and Dehumidification, ASHRAE Special Publication, ISBN 0-910110-90-5, 1992

[CROO96] 'Controlling Rotary Desiccant Wheels for Dehumidification and Cooling', K. W. Crooks, N. J. Banks, ASRAE *Transactions, SA-96-10-5, 1996*

[DUPO94] 'Desiccant solar air conditioning in tropical climates', M. Dupont & al., Solar Energy, Vol. 52, N°. 6, pp. 509-517

[MATH80] 'Performance Predictions for Adiabatic Desiccant Dehumidifiers Using Linear Solutions', B. Mathiprakasam, Z. Lavan, Journal of Solar Energy Engineering, Vol. 102, pp. 73-79, 1980

[MUNT92] Munters deshydratation, Roues Déshydratantes, documentation CF-0005A-92.06, 1992

[SCHU89] 'Comparison of the DESSIM Model With a Finite Difference Solution for Rotary Desiccant Dehumidifiers', K. J. Schultz, J. W. Mitchell, Journal of Solar Energy Engineering, Vol. 111, pp. 286-291, 1989

[STIE95] 'Performance of Rotary Heat and Mass Exchangers', G. Stiesch, S.A. Klein, J.W. Mitchell, HVAC&R Research, 1995

[VDBU92] 'The design of deshumidifiers for use in desiccant cooling and dehumidification systems', E. Van den Bulck, J. W. Mitchell, S. A. Klein, Desiccant Cooling and Dehumidification, ASHRAE Special Publication, ISBN 0-910110-90-5, 1992

# 7. Algorithms

### 7.1 Parallel flow

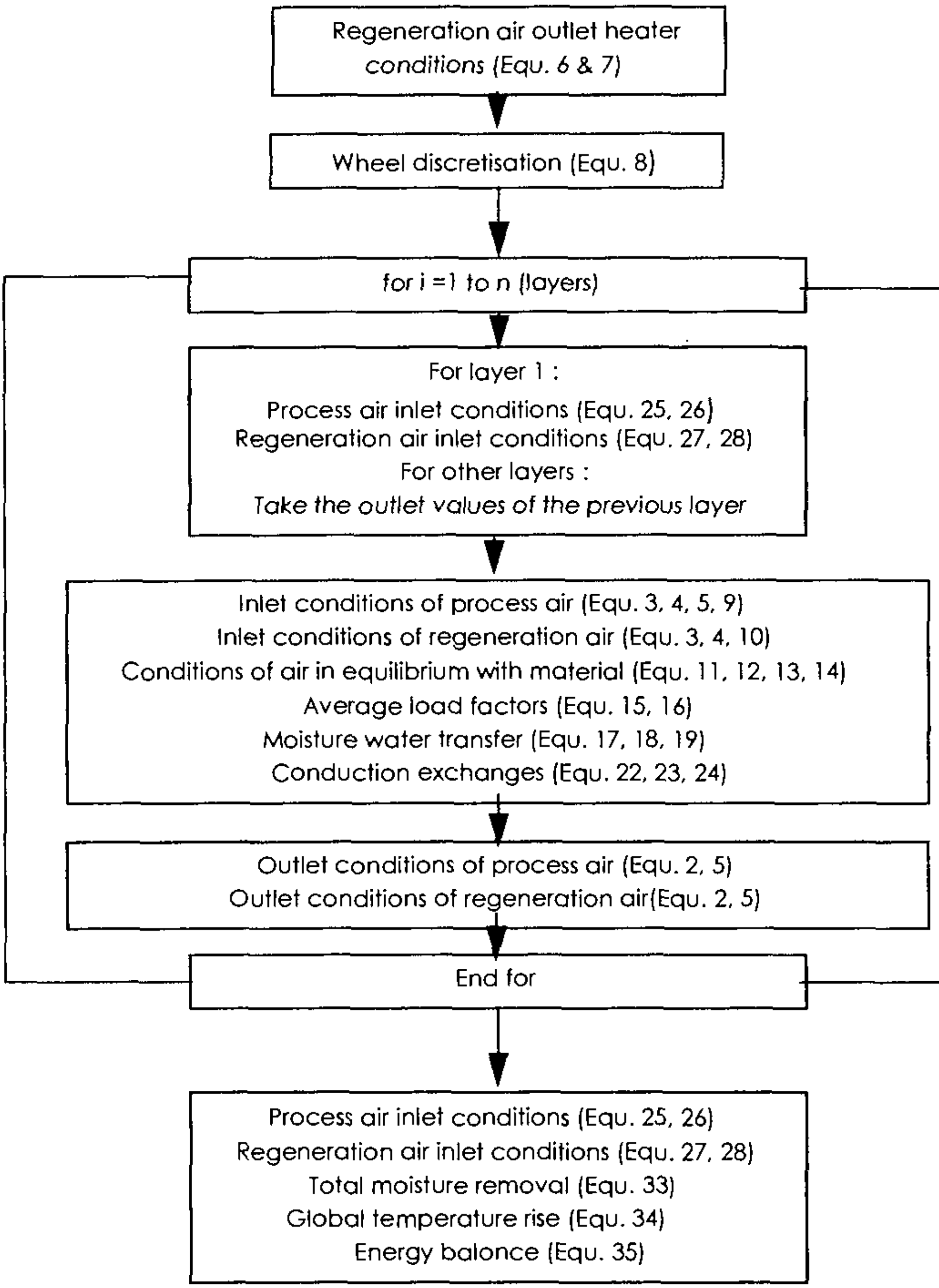
- Regeneration air outlet heater conditions
- Discretisation of wheel in n successive layers
- Initialisation of input values for first layer
- For each successive layer, for process and regeneration air streams
  - Inlet conditions
  - Average temperature of desiccant material
  - Average water load factor of material
  - Moisture transfer
  - Conduction exchanges
  - Outlet conditions
- Output process and regeneration air conditions
- Total moisture removed
- Process air temperature rise
- Energy balance

### 7.2 Counter flow

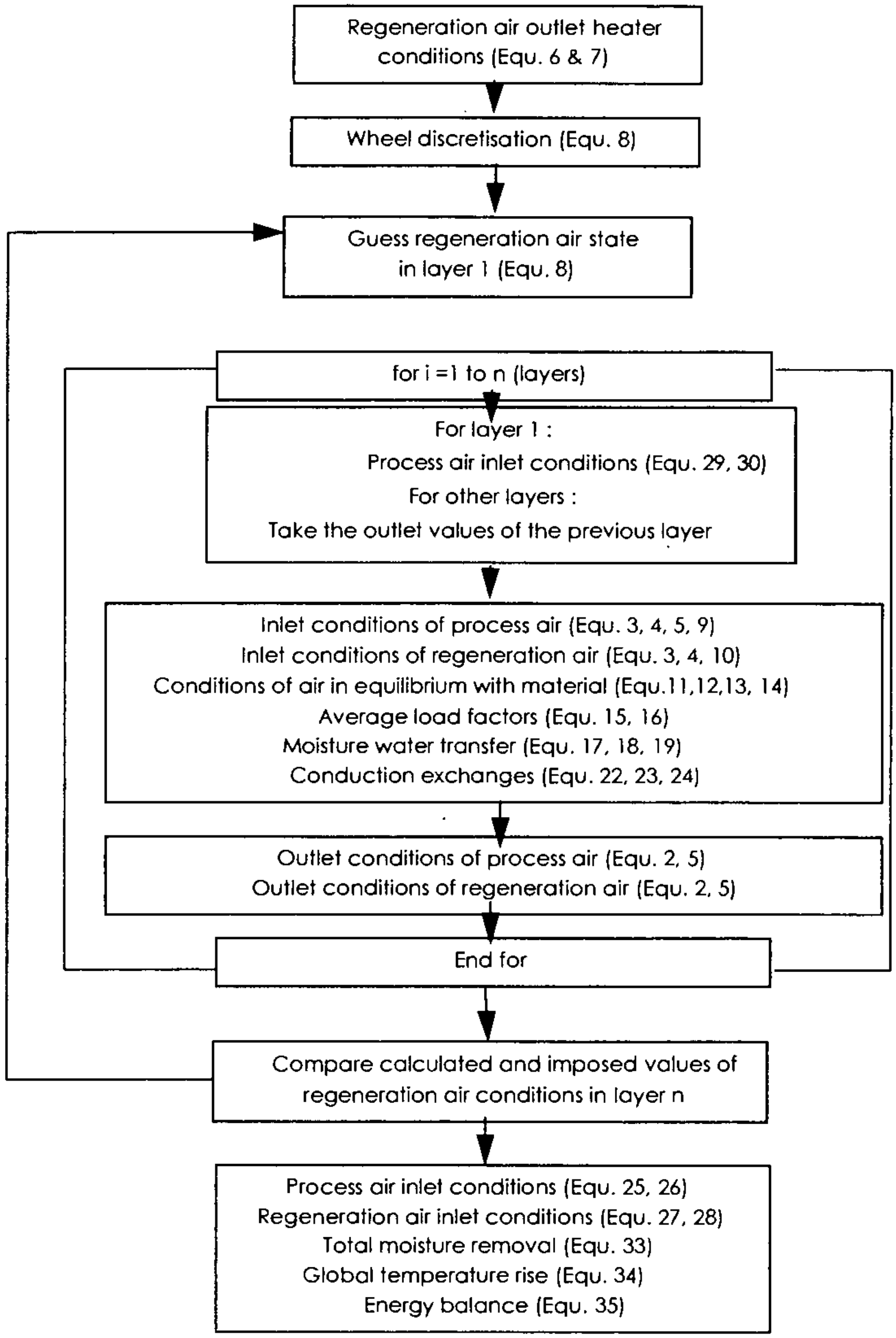
- Regeneration air outlet heater conditions
- Discretisation of wheel in n successive layers
- Initialisation of process air input values for first layer
- Guess output values for process air at first layer
- For each successive layer, for process and regeneration air streams
  - Inlet conditions
  - Average temperature of desiccant material
  - Average water load factor of material
  - Moisture transfer
  - Conduction exchanges
  - Outlet conditions
- Compare initial regeneration air guessed conditions and calculated values in layer n
- Guess new values and recalculate until convergence
- Output process and regeneration air conditions
- Total moisture removed
- Process air temperature rise
- Energy balance

# 8. Flowcharts

## 8.1 Parallel flow



**8.2 Counter flow**



# 9. Source code

These Excel Macros have been developed on a French version of Excel.

They can be used on an English (or other language) version with some minor modifications, which can be automatically done using International Macros (see the Microsoft Excel User's Guide), or done by the user via about 10 instructions (select 'replace all' in the Edition menu).

If the translation is not automatically done, here is the translation of relevant terms from French to English.

| French | English |
|---|---|
| RESULTAT() | RESULT() |
| ARGUMENT() | ARGUMENT() |
| RETOUR() | RETURN() |
| POSER.VALEUR() | PASTE.VALUE() |
| SI() | IF() |
| SINON() | ELSE.IF() |
| FIN.SI() | END.IF() |

**ε(q' ; w)**

```
eps_q_w

=RESULTAT(1)
=ARGUMENT("q";9)
=ARGUMENT("w";9)
=pv_w(w)
=t_q_w(q;w)
=pvs_temp(A8)
=A7*100/A9

=RETOUR(A10)
```

**$p_v$(w)**

```
pv_w

=RESULTAT(1)
=ARGUMENT("w";9)
=101325*w/(0,622+w)

=RETOUR(A20)
```

**$p_{vs}$(t)**

```
pvs_temp

=RESULTAT(1)
=ARGUMENT("temp";9)
=-5800,2206
=1,3914993
=-0,04860239
=0,000041764768
=-0,000000014452093
=6,5459673
=temp+273,15
=A30/A36+A31+A32*A36+A33*A36^2
=A37+A34*A36^3+A35*LN(A36)
=EXP(A38)

=RETOUR(A39)
```

**q'(t ; w)**

```
q_t_w

=RESULTAT(1)
=ARGUMENT("t";9)
=ARGUMENT("w";9)
=2500800*w+(1007+w*1846)*t

=RETOUR(A50)
```

**t(q' ; w)**

```
t_q_w

=RESULTAT(1)
=ARGUMENT("q";9)
=ARGUMENT("w";9)
=(q-2500800*w)/(1007+w*1846)

=RETOUR(C7)
```

**v'(t ; ε)**

```
v_t_eps

=RESULTAT(1)
=ARGUMENT("t";9)
=ARGUMENT("eps";9)
=pvs_temp(t)
=0,01*eps*C18
=t+273,15
=0,622*461,51*C20/(101325-C19)

=RETOUR(C21)
```

**v'(t ; w)**

```
v_t_w

=RESULTAT(1)
=ARGUMENT("t";9)
=ARGUMENT("w";9)
=t+273,15
=461,51*(0,622+w)*(t+273,15)/101325

=RETOUR(C33)
```

**w(t ; ε)**

```
w_t_eps

=RESULTAT(1)
=ARGUMENT("t";9)
=ARGUMENT("eps";9)
=pvs_temp(t)
=0,01*eps*C44
=0,622*C45/(101325-C45)

=RETOUR(C46)
```

**t(w ; v' )**

```
t_w_v

=RESULTAT(1)
=ARGUMENT("w";9)
=ARGUMENT("v";9)
=101325*v/461,24/(0,622+w)-

=RETOUR(E7)
```

**w(q' ; t)**

```
w_q_t

=RESULTAT(1)
=ARGUMENT("q";9)
=ARGUMENT("t";9)
=(q-t*1007)/(2500800+1846*t)

=RETOUR(E18)
```

**t' : First estimation**

```
thum_cad

= ARGUMENT("thum0";9)
= ARGUMENT("x";9)
= ARGUMENT("y";9)
= fonc
=w_t_eps(thum0;100)
=q_t_w(thum0;E30)
=x
=y
=2500,8*(E30-y)/(1,006+1,83*y)
=x-E34
= POSER.VALEUR(E29;E35)
= ATTEINDRE(E39)

= RETOUR(E29)
```

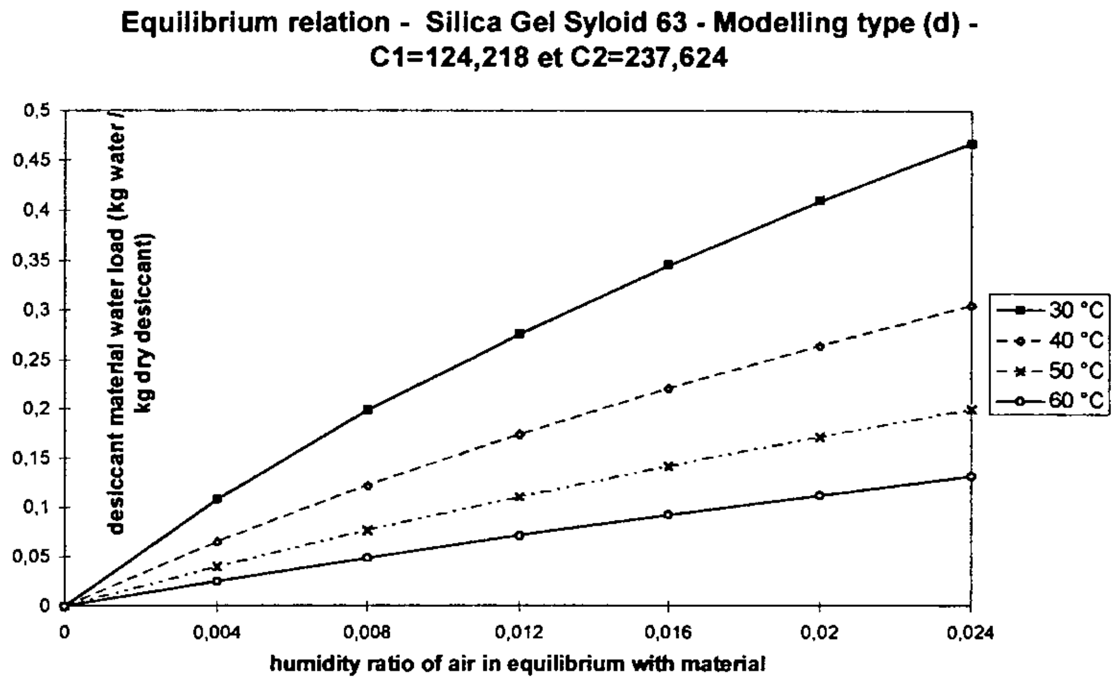
**t' :Iteration**

```
thum_dicho

= ARGUMENT("a";9)
= ARGUMENT("b";9)
= ARGUMENT("e";9)
= ARGUMENT("x";9)
= ARGUMENT("y";9)

= thum

fa=thum_cad(a;x;y)
=POSER.VALEUR(G14;G12)
=fa
fb=thum_cad(b;x;y)
=POSER.VALEUR(G17;G15)
=fb
= SI(fa=a)
= POSER.VALEUR(G10;a)
= ATTEINDRE(G53)
= FIN.SI()
= SI(fb=b)
= POSER.VALEUR(G10;b)
= ATTEINDRE(G53)
= FIN.SI()
d=0,5*(a+b)
=POSER.VALEUR(G28;G26)
=d
= SI(ABS(a-b)<e)
= POSER.VALEUR(G10;d)
=ATTEINDRE(G53)
= FIN.SI()
fd=thum_cad(d;x;y)
=POSER.VALEUR(G35;G33)
=fd
= SI(fd=d)
= POSER.VALEUR(G10;d)
= ATTEINDRE(G53)
= FIN.SI()
= SI((fa-a)*(fd-d)<0)
fb=fd
=fb
b=d
=b
= SINON()
fa=fd
=fa
a=d
=a
= FIN.SI()
=ATTEINDRE(G53)

= RETOUR(G10)
```

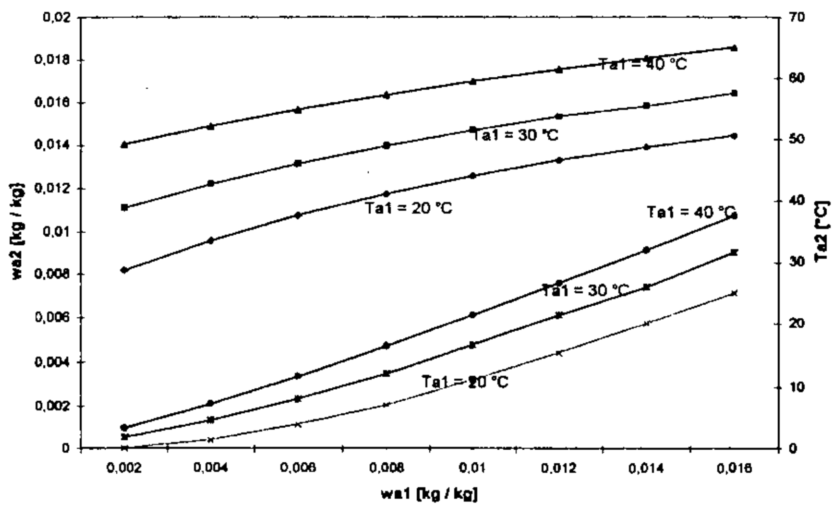
# 10. Sample results

The following pages give an idea of outputs which can be obtained with the model.

**Material performance**



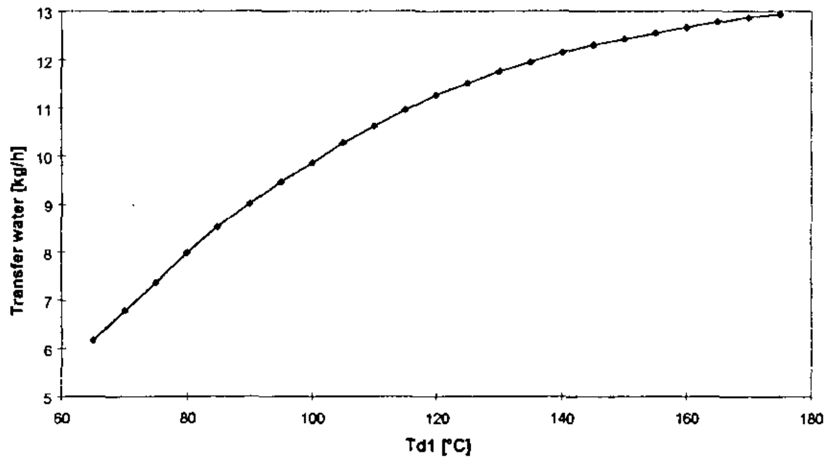

**Reconstruction of performance curves**




The model permits reconstruction of the performance curves close to those given by manufacturers. The above graph, validated by the model's results, illustrates the usual presentation of performance.

**Influence of regeneration temperature**

The studied wheel works with a high regeneration temperature (120 °C), which can be regulated. This value is high partly because the regeneration air flow rate of is only 1/3 that of the treated air and partly because the primary function of this machine is not to optimise the exiting temperature but to assure high dehumidification. The study of machines with equal air flow rates lets one obtain a better 'energy efficiency' (from our point of view).




*The figure above displays the influence* of the regeneration temperature on the dehydration capacity of the machine. We can see that for very high temperatures the capacity ascends asymptotically (at infinity, Ld = 0).

For temperatures which are too low, the capacity decreases rapidly. One can see that at an optimum temperature Td1 there would be a significant drop in the energy consumption of the wheel.

# Case Study

### Description

The application considered is a meeting room situated in an intermediate story of an office building. The west facing wall is in contact with the exterior. The meeting room is occupied between 8:00 a.m. to 6:00 p.m., Monday to Friday. The internal gains (occupants, lighting, ...) are 30 W/m² with a constant occupation of 30 people. Infiltration is estimated to be constant during the daytime at 0.3 AC/h. One window (U=3.3 W/(m².K)) of 5 m² is situated on the exterior wall.

The room has the following dimensions:

| | |
|---|---|
| h = 3 m | floor to ceiling height |
| L = 12.5 m | lenth of the room |
| l = 10 m | width of the room |

The area of the ceiling is 125 m² and the volume of the room is 375 $m^3$.

The wall composition (outside to inside) is given in the following table. The floor and ceiling are made of concrete slabs.

| | thickness | conductivity | density | thermal capacity |
|---|---|---|---|---|
| | cm | W/(m.K) | kg/m3 | Wh/(kg.K) |
| Concrete | 10 | 1.75 | 2300 | 0.27 |
| Polysterene | 8 | 0.04 | 40 | 0.33 |
| Plaster | 1.3 | 0.35 | 900 | 0.22 |

Set-points :

cooling setpoint temperature during occupation : 25°C
humidity setpoint during occupation : 60%

The dynamic simulations are produced by using the software COMFIE for typical summer weeks (SRY weather file) for the Carpentras weather station.

### Air treatment system

*Performance characterization for the desiccant wheel used*

The nominal wheel velocity is 8 revolutions per hour. The nominal air flow rates are :

| | |
|---|---|
| treated air flow rate | da = 0,75 m3/s |
| regeneration air flow rate | dd = 0,25 m3/s |

The wheel opening (defined by the ratio of the regeneration air section to the total size of the wheel) is 0,25.
The desiccant material used is made of High Perfomance Silicagel, for which we used the equation given by [MATH80] for the isotherms. At nominal conditions the mass of the wheel is estimated to be 26.7 kg.

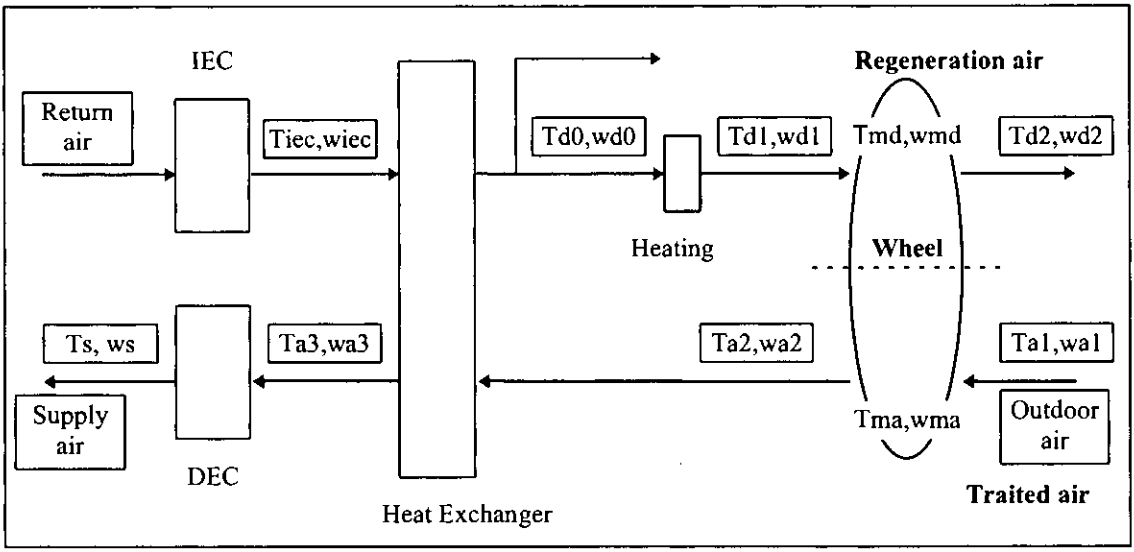
Power consumptions are as follows:

| | | |
|---|---|---|
| maximum regeneration power | 27000 | W |
| treated air renewal fan power | 2200 | W |
| regeneration air fan power | 1500 | W |
| wheel driving power | 500 | W |
| supply air fan power | 2200 | W |

The regeneration energy source can be electric, direct gas or by water vapor.

*Schematic of system principals*

The following figure is a schematic of the system. The system will operate at full fresh air. Direct and indirect evaporation cools the air in the system. Modelling equations for this equipment can be found in Chapter E of this report.




Schematic of installation principals

Only a part of the return air flow rate is used for heating of the wheel. In reality the wheel operates with an opening of 25% which imposes a regeneration air flow rate equal to one third of the supply air flow rate.

The humidifiers have efficiencies of 0.85 and the heat exchanger efficiency is 0.80.

*System control*

The wheel posseses a thermostat that allows the user to regulate the intake air temperature for the regeneration air stream. The recommended temperature is 120 °C (this value is very high,and is not an optimal temperature, as we explained previously). For evaporative systems, the evaporation can be limited if the maximum temperature obtained after hum idification is too low.

Because it is tricky to optimize the air flow rates across the wheel, the wheel operates with constant air flow rates. The evaporative systems therefore also operate at constant air flow rates.

For a correctly sized system the return air condtitions are at the space conditions and are considered to be constant over the one hour time step. Similarly, the exterior conditions are considered constant over the one hour time step.

For the hour considered we will assume continuous function :
$\phi$ max = Πs max . (q'a - q's min)
Πs max corresponds to the nominal air flow rate produced by the fan at the entrance of the treated air flow stream in the wheel.

The sensible loads, $\phi$ real, provide air flow conditions at the needed rate :
$\phi$ real = Πs real . (q'a - q's needed)

We can therefore calculate the running time of the system (expressed in %), and furthermore the supply air flow rate required to meet the sensible loads.

Remarks :

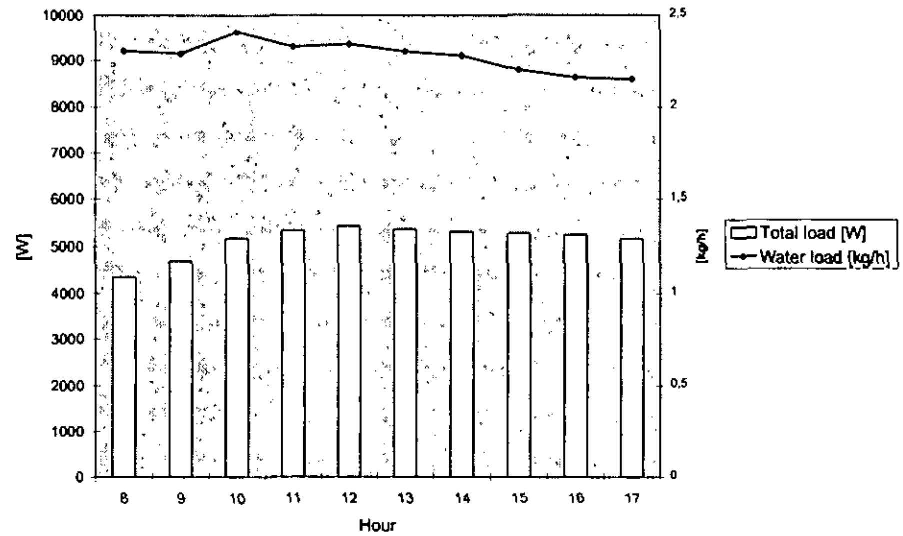
1. We always assure a minimum air change of 2 vol/h
2. The operation corresponds to the regulation of the space temperatures using a thermostat or nothing at all.

3. If (Πs real > Πs max), the working time of the system has to be greater than 100%, therefore the obtained temperature and humidity are greater than the set-points.

**Results of simulations**

*Calculated loads*

The following graph show the evolution of loads for a typical day. The ratio (latent loads/total loads) is usually less than 30% on average. Its therefore not a favorable case for desiccant air-conditioning.

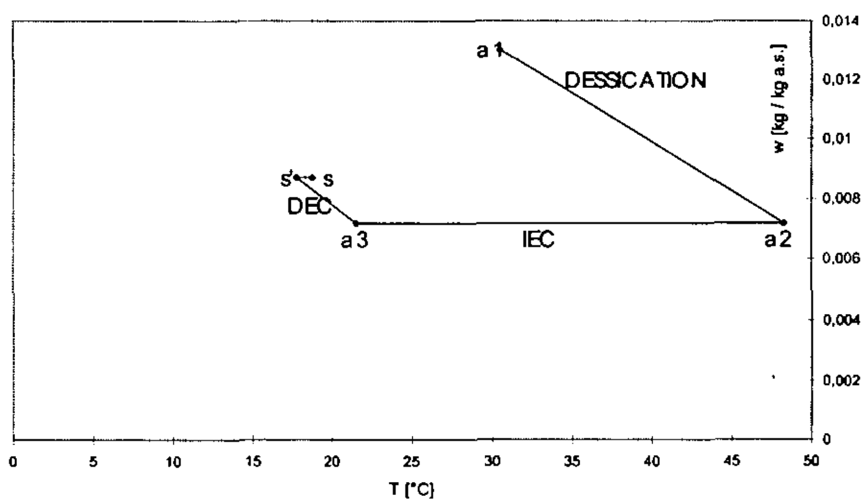



Evolution of loads during a typical day

*Schematic representation of treated air transformations*

The models of the evaporative systems are presented in detail in Chapter E of this report.

The following graph shows the evolution of treated air that sucessively crosses the desiccant wheel, the heat exchanger and the direct evaporative cooler. Note that the indirect evaporation is very efficient, because of the entry temperature for the treated air after the wheel (between 35 and 45 °C).




*Characterization of the system performance*

We define two COPs which correspond to the two extreme conditions (all electric regeneration, all 'free heat' regeneration).

**COP MIN = P cooling / (Pregeneration + P auxiliaries)**

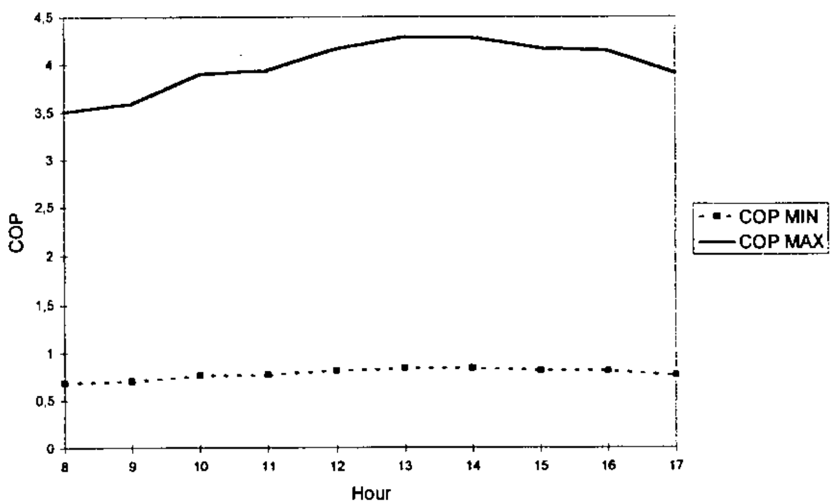
The cooling power is given by Πs . (q'a - q's)
The auxiliaries are composed of all the fans and the motor that powers the wheel.
This COP corresponds to a total electric regeneration without any recuperation of heat (the air is still at 40 - 50 °C at the point d2 in before being released into the atmosphere).

**COP MAX = P cooling / (P auxiliaries)**

This version of COP brings us back to the ideal case where all of the heat necessary for the regeneration is free.

The following figure demonstrates the evolution of the two COPs in the course of a typical day. The COP MIN is always in this case inferior to 1. On the other hand, the COP MAX can reach 4.5.

Recall that the studied system functions with a regeneration air temperature of 120 °C, which is very taxing on the COP MIN. A temperature of 70 up to 80 °C is sufficient with equal air flow rates on each side of the wheel.




**Synthesis of results**

The synthesis of obtained results for a simulation week is presented in the following table.

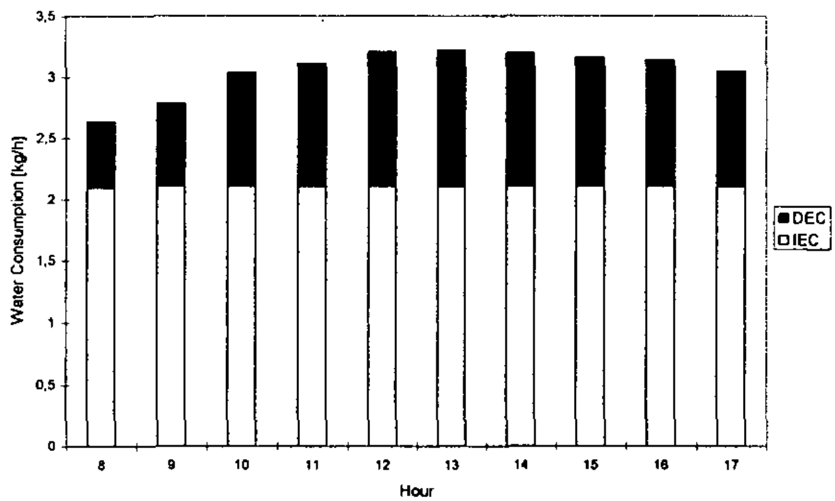
| | | Average | Minimum | Maximum |
|---|---|---|---|---|
| Texterior | °C | **26,65** | 18,80 | 34,00 |
| w, exterior | kg / kg a.s. | **0,0117** | 0,0089 | 0,0160 |
| Total load | W | **5375** | 4351 | 6233 |
| Hydraulic load | kg / h | **2,26** | 2,00 | 2,63 |
| Tsupply | °C | **18,84** | 17,76 | 19,88 |
| w, supply | | **0,0070** | 0,0047 | 0,0113 |
| Water consumption IEC | kg / h | **2,11** | 2,09 | 2,13 |
| Water consumption DEC | kg / h | **1,07** | 0,54 | 1,61 |
| Duration of operation | % | **67** | 37 | 100 |
| Supply air flow rate | vol / h | **4,48** | 2,66 | 7,51 |
| COP MIN | | **0,80** | 0,64 | 0,97 |
| COP MAX | | **4,11** | 3,25 | 4,95 |

For the week studied, there were only three hours for which the set-point temperature was not attained On average, the system operated 67% of the time with a supply air flow rate of 4.48 ACl/h, which may be more acceptable for the occupants' comfort than the air flow rates common to evaporative systems (around 8 AC/h).

The COP MIN is obtained by electrically driven regeneration and is not at an optimal condition (there is no heat regain after the wheel, so therefore the temperature levels at this point are still around 40 or 50 °C). The average COP MIN value obtained, 0.80, agrees with those values found in litterature. The installation of the heat exchanger-recuperator considerably improves this value. Researchers presently working on new desiccant materials claim that an average performace of 1.7 can be achieved, as reported IEA Annex 28 report "Review of Low Energy Technologies".

The COP maximum corresponds to the case where the heat of regeneration can be supplied by a free source.

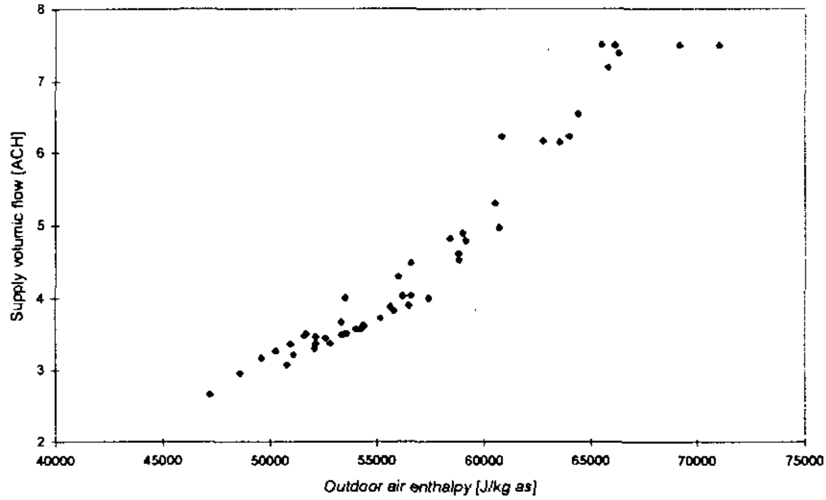
*Water consumption*




Water consumption attributed to indirect evaporation is practically constant during the day. In effect, the secondary air that is humidified is taken from the room and therefore is at the set-point temperature. The consumption of water by direct evaporative cooling varies.

*Required supply air flow rates*

By the study of the evaporative systems, we have shown that for the southern French climate, these systems are viable only with high fresh air flow rates (enabling increased the supply air temperatures to be used). The following figure represents the air flow rates necessary for a system with a wheel. It is noted that for the majority of the time an air flow rate of 5 vol/h is sufficient.




## Conclusion

For electric regeneration the COP today is slightly inferior to one. The amelioration of material performance and optimization of cycles in the next few years should permit COP values to reach 1.7, as reported IEA Annex 28 report "Review of Low Energy Technologies".

In cases where a free heat source is available, the COP could reach very high values (around 5 in our case of study, and up to 10 in most favorible situations).

# IEA Annex 28

## Low Energy Cooling

### Subtask 2: Detailed Design Tools

### Desiccant + Evaporative Cooling Algorithms and Control Strategies

Technical University of Nova Scotia
S.Kemp, N.Ben-Abdallah

CANMET Energy Diversification ResearchLaboratory
S.Hosatte, M. Stylianou

# Contents Chapter B

# 1. Technology area

This paper describes the development of computer routines and control strategies for combined solid desiccant and evaporative cooling systems and their implementation into the DOE-2.1E building energy simulation model.

# 2. Developed by


| | | |
|---|---|---|
| **Name:** | S.Kemp, N Ben Abdallah | S.Hosatte, MStylianou |
| **Organisation:** | Technical University of Nova Scotia | CANMET- Energy Diversification Research Laboratory |
| **Address** | PO Box 1000<br>Halifax, Nova Scotia<br>Canada<br>B3J2X4 | 1615 Lionel-Boulet Bvd.<br>Varennes, Quebec<br>Canada<br>J3X1S6 |
| **Phone** | (902) 420-7584 | (514) 652-5331 |
| **Fax** | (902) 423 2423 | (514) 652-5177 |


Acknowledgments

We thank Dr. Yu Joe Huang at the Lawrence Berkeley Laboratory for technical assistance regarding the programming of DOE-2.1E functions.

This work has been supported by the Department of Natural Resources, Canada and the Canadian Program on Energy Research and Development (PERD).

# 3. General Description

This algorithm has been developed to allow DOE-2.1E users to investigate the feasibility of combined desiccant and evaporative cooling air-conditioning systems.

In order to use the DOE-2.1E simulation software for desiccant and evaporative cooling systems described in Figure 1 the systems sub-program was modified through the use of DOE-2.1E functions. DOE-2.1E functions allow FORTRAN like sub-routines to be inserted at specified points in the DOE-2.1E algorithm. The functions have access to DOE-2.1E variables at the function call location and may return values for specified DOE-2.1E variables back to the DOE-2.1E algorithm. To model the desiccant and evaporative equipment the functions DKTEMP-2 and SDSF-1 were used. The FORTRAN code entered for these functions are the programs DKTEMPDES and SDSFDES.

A diagram showing the system components of the simulated desiccant/evaporative cooling system and the corresponding psychrometric processes is given in Figure 2.

As depicted in Figure 2, the simulated desiccant/evaporative cooling system has five main components as follows:

1. Air mixing chamber/plenum
2. Desiccant wheel/economizer
3. Indirect evaporative cooler
4. Direct evaporative cooler
5. Electric humidifier

The following sections describe the algorithm of the implemented functions themselves (section 5) and the algorithm of the DOE-2.1E system subroutines in which the functions are implemented (section 7).

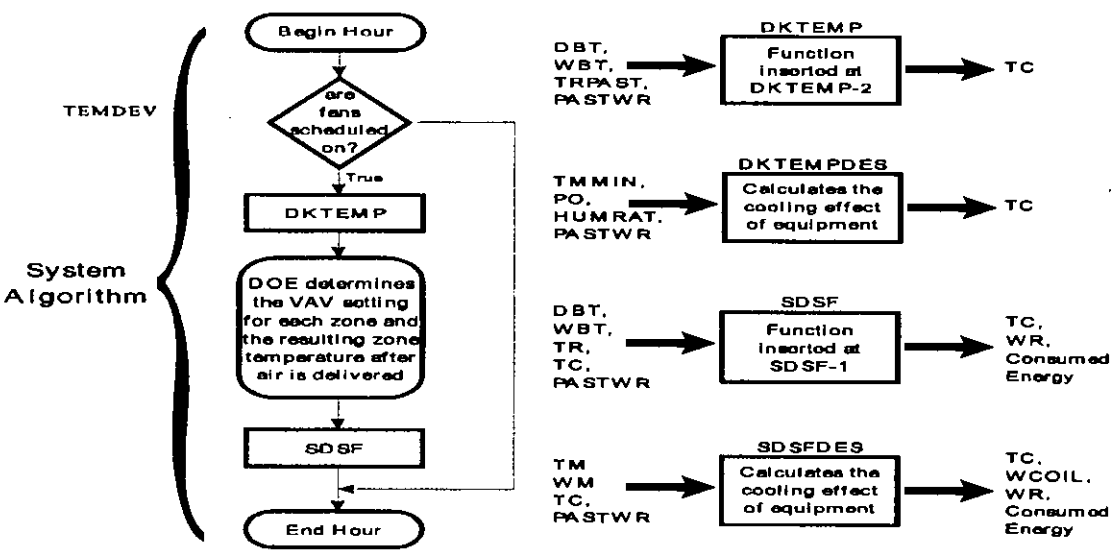
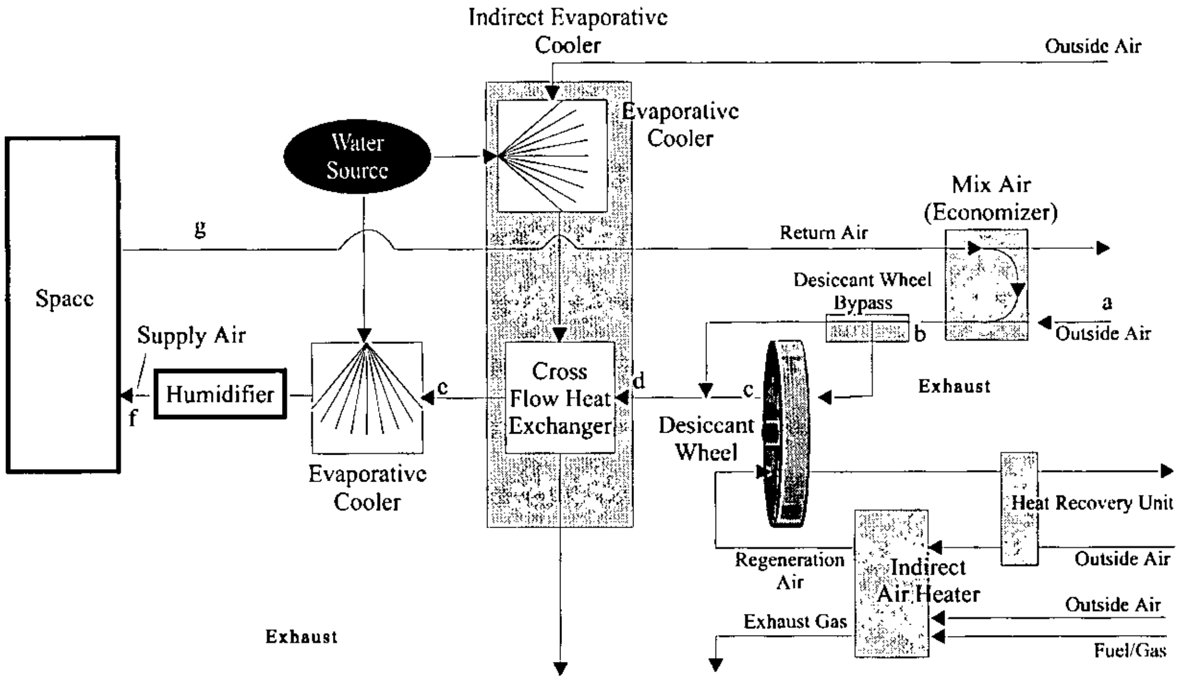
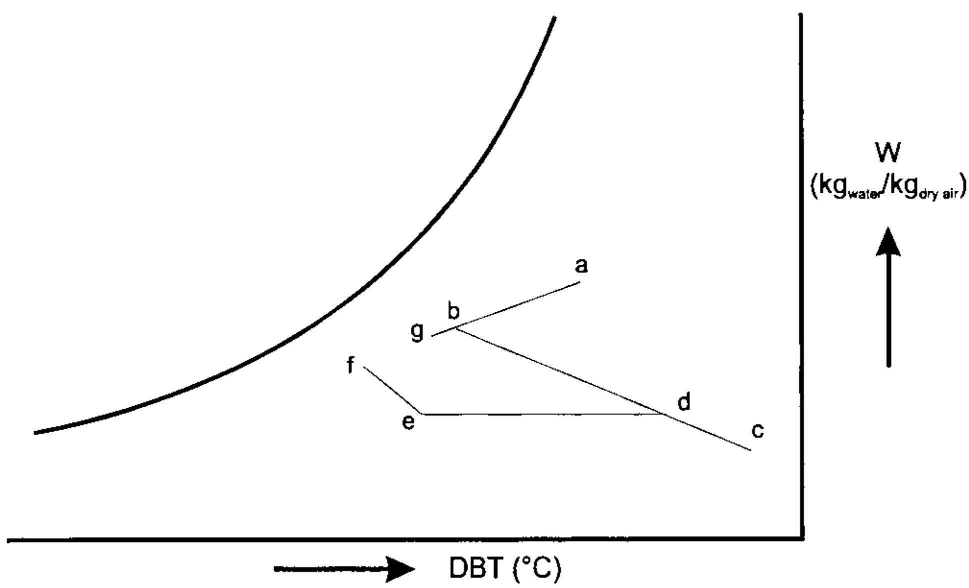



**Figure 1** DOE-2.1E algorithm diagram and the new functions (see Source Code Nomenclature for Variable Definitions)

## Air Flow Diagram




## Psychrometric Process



# 4. Nomenclature

| Variable | Description | English Units | SI Units |
|---|---|---|---|
| Area | nominal face area of desiccant wheel | $ft^2$ | $m^2$ |
| V | air flow rate | $ft^3/min$ | $m^3/s$ |
| $\Delta W$ | change in absolute humidity | $lb_{water}/lb_{air}$ | $kg_{water}/kg_{air}$ |
| $\varepsilon$ | effectiveness of the air-air heat exchanger for the indirect evaporative cooler | | |
| e | effectiveness of direct evaporative cooler (both for the stand-alone and the indirect) | | |
| h | specific enthalpy of the air | BTU/lb | kJ/kg |
| N | number of desiccant wheels | | |
| R | ratio of air flow for a particular process to total air flow | | |
| Q | heat energy | Btu | kJ |
| T | temperature | °F | °C |
| TRP | mixed air temperature | °F | °C |
| W | absolute humidity | $lb_{water}/lb_{air}$ | $kg_{water}/kg_{air}$ |
| WB | wet-bulb air temperature | °F | °C |
| Water | amount of water (water usage is sent directly to the output file in units of liters) | liters | |
| desweight | fraction of hour that desiccant wheel is active | hour | hour |
| K | K factor used to determine the heart carry-over from regeneration to dehumidification process | | |
| D | intermediate dummy variable for desiccant process | | |
| $v_{face}$ | face velocity entering the desiccant wheel dehumidification process | ft/min | m/s |

| Subscripts | Description |
|---|---|
| active | desiccant wheels being used during the modelled hour |
| c | cold side value for indirect evaporative cooler heat exchanger |
| d | desiccant dehumidification process |
| design | desiccant wheels needed for design air flow rate |
| e | direct evaporative cooling process |
| i | value entering process |
| installed | desiccant wheels actually installed in the system |
| ma | value for mixed air after the outdoor air economizer |
| max | maximum value |
| min | minimum value |
| gas | fuel/gas |
| o | value exiting process |
| oa | value for outside air |
| ra | value for return air |

| Subscripts | Description |
|---|---|
| sa | value for supply air |
| slice | time slice increment |
| tot | total fraction of time |
| x | indirect evaporative cooling process |
| p | immediately leaving the desiccant wheel, before mixing with non-dehumidified air stream |
| design | number of desiccant wheels needed for design airflow |
| wheel | desiccant wheel property |
| grain | absolute humidity in grains/lb |
| prg | purge air temperature used to cool desiccant material after regeneration |
| H | electric humidifier process |
| h | heat recovery process |
| g | regeneration process |

# 5. Mathematical Description

Internally, DOE-2.1E uses English units exclusively, thus the mathematical description is presented in English units with SI conversions. All relevant conversions have been confirmed through the use of debugging print statements which have since been removed.

**Mixing the Air - Economizer**

The economizer routine is used as is from DOE-2.1E, DOE-2.1E's economizer's routine may select the fresh air ratio (R) based on one of three control schemes selected by the user. These are:

1. Constant, OA-CONTROL=CONSTANT
2. Temperature, OA-CONTROL=TEMPERATURE
3. Enthalpy, OA-CONTROL=ENTHALPY

The constant control scheme specifies that DOE-2.1E will always mix return air and fresh air at a constant mixing ratio. This ratio is specified by the user by directly defining the fresh air ratio, or specifying the fresh airflow. DOE-2.1E will then calculate the fresh air ratio based on the design airflow and fresh airflow.

The temperature and enthalpy control schemes allow DOE-2.1E to determine each hour the optimum fresh air ratio based on the relative temperature or enthalpy of the return and fresh air flows. Depending on which is chosen, DOE-2.1E will choose a fresh air ratio that produces a mixed air condition with the lowest temperature or enthalpy. Concurrently, DOE-2.1E must satisfy the minimum fresh air requirements of the building. The minimum fresh air requirement of the building is entered into the DOE-2.1E system input file through one of a number of methods [1]. The user may specify an actual minimum fresh air ratio, or a fresh airflow value in $m^3$/hr or CFM. If the fresh air ratio is specified, this becomes $R_{min}$ ; if the user specifies a fresh airflow value then DOE-2.1E calculates $R_{min}$ each hour based on the airflow rate demand for that hour, ensuring that the minimum fresh airflow is supplied to the building. The two possible values for R are $R_{min}$ and $R_{max}$ ,this last value corresponding to 100% fresh air.

At the DKTEMP-2 function insertion point, DOE-2.1E has already determined the mixed air temperature using the economizer control scheme defined by the user. The mixed air temperature is available for the DKTEMPDES function however, the mixed air humidity is not, nor is the ratio that DOE-2.1E used to calculate the mixed air temperature available. However, the two possible values for the fresh air ratio are available, $R_{min}$ and $R_{max}$. To determine which value was used the DKTEMPDES function uses a temporary variable from the DOE-2.1E algorithm to determine the fresh air ratio. This variable is TRP. TRP has been used in DOE-2.1-E in the temperature or enthalpy optimization step, to calculate the temperature corresponding to the minimum fresh air value, $R_{min}$.

The DKTEMPDES subroutine determines the fresh air ratio used by comparing the mixed air temperature, $T_{ma}$, the calculated economizer outlet temperature, to the temporary variable used in the DOE mixing algorithm TRP.

If TRP=$T_{ma}$ then R=$R_{min}$ (the minimum fresh air value)

else R=$R_{max}$ (100% fresh air)

The mixed air absolute humidity ratio is then determined by:

$$W_{ma} = W_{oa} \bullet R + W_{ra} \bullet (1 - R)$$

Where the SDSFDES routine is implemented (the SDSF-1 function call) mixing of the air is not required as DOE-2.1E has stored the mixed air ratio (R) and the mixed air absolute humidity ratio ($W_{ma}$) has been calculated and is available to the SDSFDES routine without the need for extra calculations.

**Desiccant Wheel Dehumidification Process**

The equations goveming the desiccant dehumidification process come from regression analysis of manufacturer's performance curves. The algorithm is capable of simulating one or more desiccant wheels of the volumetric capacity specified by the user. The desiccant wheel module will dehumidify the fraction of the total supply air specified by the user defined parameter $R_d$, or desiccant fraction. The user enters values for $N_{design}$ and $N_{installed}$, from which DOE-2.1E calculates the resulting desiccant fraction, $R_d$. The selection of $N_{design}$, and $N_{installed}$ is described in the Algorithm section. The value for $R_d$ is constant for all airflow rates calculated by the variable air volume system.

As the airflow demand of the system varies, the number of wheels used to dehumidify the air is optimized so that the desiccant wheels are operating near their design airflow rate. The number of wheels to be used for the present hour is then:

$$N_{active} = R_d \bullet \frac{V_{tot}}{V_{wheel}} + 0.9 \text{ truncated to the integer}$$

This will allow 10% over the design airflow per wheel before an additional wheel is requested.

The algorithm is designed to calculate the desiccant dehumidifier exiting conditions based on the process air inlet temperature, humidity and face velocity. The entering temperature and humidity correspond to point b in Figure 2. The face velocity for each wheel is then:

$$v_{face} = \frac{V_{tot} \bullet R_d}{N_{active} \bullet Area}$$

The following equations are for the specific wheel modelled in the example code supplied in the Source Code section. The reactivation air temperature for the modelled desiccant wheel is fixed at 250°F(121°C). The user may replace these equations with their own for the wheel to be modelled. The process exiting temperature and absolute humidity must be returned to avoid errors.

The regression equations for the modelled wheel require that the absolute humidity be in units of grains/lb, therefore:

$$W_{grain} = W_{di} \bullet 7000$$

The exiting absolute humidity is calculated from the following pair of equations:

$$D = 2.072 \bullet 10^{-3} \bullet W_{grain}^2 + 5.814 \bullet 10^{-3} \bullet T_{di}^2 + 3.193 \bullet 10^{-3} \bullet W_{grain} \bullet T_{di} - 0.088 \bullet W_{grain} - 0.50283 \bullet T_{di} + 9.94773 \bullet$$

$$W_{po} = \frac{4.104 \bullet 10^{-6} \bullet D^2 - 5.313 \bullet 10^{-5} \bullet v_{face}^2 - 4.644 \bullet 10^{-4} \bullet D \bullet v_{face} + 0.69491 \bullet D + 0.07842 \bullet v_{face} - 24.8076}{7000}$$

for temperature and humidity in SI units:

$$D = 1.015 \bullet 10^5 \bullet W_{di}^2 + 0.0188 \bullet T_{di}^2 + 40.236 \bullet W_{di} \bullet T_{di} - 102.219 \bullet W_{di} - 0.235 \bullet T_{di} - 0.189$$

$$W_{po} = 5.862 \bullet 10^{-9} \bullet D^2 - 2.941 \bullet 10^{-4} \bullet v_{face}^2 - 1.306 \bullet 10^{-5} \bullet D \bullet v_{face}$$
$$+ 9.927 \bullet 10^{-5} \bullet D + 2.205 \bullet 10^{-3} \bullet v_{face} - 3.544 \bullet 10^{-3}$$

The exiting temperature is then determined with a series of three equations:

$$T_{prg} = 204.75 + 0.0758 \bullet W_{grain}$$

$$K = -2.1964 \bullet 10^{-6} \bullet W_{grain}^2 + 1.5714 \bullet 10^{-7} \bullet V_{face}^2 - 3.8982 \bullet 10^{-7} \bullet W_{grain} \bullet V_{face}$$
$$+ 8.6032 \bullet 10^{-4} \bullet W_{grain} - 2.3875 \bullet 10^{-4} \bullet V_{face} + 0.1104$$

$$T_{po} = T_{di} + 0.625 \bullet (W_{grain} - 7000 \bullet W_{po}) + K \bullet (T_{prg} - T_{di})$$

Or in SI units:

$$T_{prg} = 95.99 + 294.66 \bullet W_{di}$$

$$K = -107.621 \bullet W_{di}^2 + 6.089 \bullet 10^{-3} \bullet v_{face}^2 - 0.0537 \bullet W_{di} \bullet v_{face}$$
$$+ 6.022 \bullet W_{di} - 0.047 \bullet v_{face} + 0.1104$$

$$T_{po} = T_{di} + 2430.55 \bullet (W_{di} - W_{po}) + K \bullet \frac{5}{9} \bullet (T_{prg} - T_{di})$$

At this point the process air exiting temperature and humidity have been calculated. This corresponds to point c in Figure 2. If the user has decided to model a different desiccant wheel they must return the temperature and humidity values in °F to the algorithm at this point.

After the process air has been dehumidified it is reintroduced to the non-dehumidified air. The mixing ratio is defined by the desiccant fraction $R_d$. The properties of the air at point d, Figure 2 are then:

$$W_{do} = W_{po} \bullet R_d + W_{di} \bullet (1 - R_d)$$

$$T_{do} = T_{po} \bullet R_d + T_{di} \bullet (1 - R_d)$$

**Indirect Evaporative Cooler**

The indirect evaporative cooler is modelled as two separate components. The first component is a direct evaporative cooler which acts on the ambient outdoor air. This cooled air is then used as the cold side of an air to air heat exchanger which sensibly cools the process air. The two components are modelled using effectiveness ratios, thus the process air temperature exiting the indirect evaporative cooler is:

$$T_{xo} = T_{xi} - \varepsilon \bullet (T_{xi} - T_{oa}) - \varepsilon \bullet e \bullet (T_{oa} - WB_{oa})$$

The indirect evaporative cooling process is sensible only, therefore the exiting humidity is the same as the entering humidity.

$$W_{xo} = W_{xi}$$

The change in the absolute humidity of the cooling air through the direct evaporative cooler component is calculated for determination of the water usage by the evaporative cooler. To accomplish this, the cold side temperature is calculated and the process is assumed to be isenthalpic. Thus the humidity of the cold side air exiting the direct evaporative cooler and entering the heat exchanger is found using a root solver.

The enthalpy of the air entering the indirect evaporative cooler on the cold side is:

$$h_x = 0.24 \bullet T_{oa} + (1061.0 + 0.444 \bullet T_{oa}) \bullet W_{oa}$$

or in SI units:

$$h_x = 1.006 \bullet T_{oa} + (2501 + 1.775 \bullet T_{oa}) \bullet W_{oa}$$

The temperature on the cold side of the heat exchanger is then:

$$T_{xc} = T_{oa} - e \bullet (T_{oa} - WB_{oa})$$

The root solver must then be used for the value $W_{co}$ that yields the root of:

$$h_x - 0.24 \bullet T_{xc} + (1061.0 + 0.444 \bullet T_{xc}) \bullet W_{co} = 0$$

in SI units:

$$h_x - 1.006 \bullet T_{xc} + (2501 + 1.775 \bullet T_{xc}) \bullet W_{co}$$

The change in humidity through the direct evaporative cooler component of the indirect evaporative cooler is then:

$$\Delta W_x = W_{co} - W_{oa}$$

This value is later used in the calculation for water usage.

### Direct Evaporative Cooler

The direct evaporative cooler is modeled similarly to the direct evaporative cooler component of the indirect evaporative cooler. The difference is that the process air is being cooled instead of the ambient outdoor air. The temperature leaving the direct evaporative cooler is:

$$T_{eo} = T_{ei} - e \bullet (T_{ei} - WB_{ei})$$

Again the root solver is used to determine the exiting absolute humidity of the process air. The change in absolute humidity is stored to be used later to determine the water usage.

$$\Delta W_e = W_{eo} - W_{ei}$$

### Electric Humidifier

The electric humidifier model assumes that the unit has the capacity to satisfy the latent heating load for any air flow rate. If it is calculated that the return air relative humidity value is below the minimum humidity set point, then the return humidity is reassigned the value of the minimum humidity set-point. The humidity exiting the humidifier needed to achieve this is then calculated using:

$$W_{Ho} = W_{r,\min} - W_{load}$$

The mixed air value using the new return air humidity value is then:

$$W_m = W_{oa} \bullet R + (1 - R) \bullet W_{r,\min}$$

The energy consumed by the electric humidifier is then calculated as the latent heat of vaporization needed to raise the humidity value of the process air for the air flow rate:

$$\Delta W_H = W_{Ho} - W_{Hi}$$

$$Q_H = \Delta W_H \bullet 1061 \bullet \frac{v_{total}}{v} \bullet 60$$

or in SI units:

$$Q_H = \Delta W_H \bullet 2468 \bullet \frac{V_{total}}{v}$$

**Determination of the final return air (building) humidity**

This is only used in the SDSFDES routine (at the DOE-2.1E function insertion point SDSF-1) where the final building humidity level (i.e. the return air humidity) is calculated.

If the fans were on for the hour and cooling was performed then the return air is:

$$W_{ra} = W_{sa} + W_{load}$$

If the fans were not on:

$$W_{ra} = W_{ra} + W_{load}$$

**Desiccant Wheel Regeneration Process**

To avoid recalculation from the iterative loops, the regeneration process and energy calculations are not performed until just prior to exiting the SDSFDES function. This calculation models the heat recovery from the regeneration air exhaust as well as the gas consumption of the indirect air heater and the electrical consumption of the regeneration fan.

The air flow required for regeneration is determined by an equation supplied by the manufacturer. If the user has modelled another desiccant wheel, this equation must be replaced with the appropriate relationship.

$$V_g = \frac{0.04 \bullet V_{wheel} \bullet (T_{prg} - T_{di}) + V \bullet (T_{do} - T_{di})}{T_{gi} - T_{go}} \bullet N_{active}$$

For the presented model the regeneration temperature is fixed at $T_{gi}$=250°F (121.1°C) and the manufacturer's design procedure specifies that the exiting temperature from the regeneration side of the wheel is $T_{go}$=120°F (49°C) for all conditions. This assumption agrees well with the manufacturer's own data within ±1°C for all reasonable conditions.

The heat recovery unit, an air to air heat exchanger, recovers heat from the regeneration air stream leaving the desiccant wheels. The air temperature leaving the heat recovery unit is therefore:

$$T_{ho} = \varepsilon_h \bullet (T_{go} - T_{oa}) + T_{oa}$$

Again, if the user specifies another desiccant wheel, then they must determine their own value for $T_{go}$.

The regeneration air stream is then heated in the regeneration air heater. The air heater is a gas fired, indirect air heater that heats the air to the regeneration air temperature. For the desiccant wheel modelled here, the regeneration temperature is fixed at 120°C (250°F). The heat energy of regeneration for the modeled hour is then:

$$Q_g = \frac{V_g}{v_{stand}} \bullet (h_{gi} - h_{ho}) \bullet 60 \bullet desweight$$

or in SI units:

$$Q_g = \frac{V_g}{v_{stand}} \bullet (h_{gi} - h_{ho}) \bullet desweight$$

where: $V_{stand}$=13.3 ft$^3$/lb (0.830 m$^3$/kg), the specific volume of standard air.

and the enthalpy values are determined by the function:

$$h(T,W) = 0.24 \bullet T + (1061 + 0.44 \bullet T) \bullet W$$

or in SI units:

$$h(T,W) = 1.006 \bullet T + (2501 + 1.775 \bullet T) \bullet W$$

The amount of gas energy consumed by the air heater is then:

$$Q_{gas} = \frac{Q_g}{\eta_{burner}}$$

The conversion of gas energy to utility units is left to DOE-2.1E and user inputs.

**Water Consumption of the Evaporative Coolers**

The water usage is calculated using the change in absolute humidity in the evaporative coolers that were calculated earlier. For the indirect evaporative cooler the consumption of water is:

$$Water_x = \Delta W_x \bullet V_{tot}$$

and for the direct evaporative cooler:

$$Water_e = \Delta W_x \bullet V_{tot}$$

# 6. References


[1] DOE-2.1E manuals (User/Reference/Supplement)

[2] Munters DryCool Bulletin 400

[3] Performance of Desiccant/Evaporative Cooling in Canadian Office Buildings using the Functions of DOE-2.1E, S. Kemp, N. Ben-Abdallah, M. Stylianou and S. Hosatte, Ab-Sorption '96 Conference Proceedings.

[4] Numerical Calculation of Psychrometric Properties, Luther R. Wilheim, Transactions of the ASHRAE, 1976, Vol , Page 318 .


# 7. DOE-2.1E System Program Algorithm

DOE-2.1E simulates the installed system equipment hourly. It uses the outdoor weather conditions and the load results for each hour to determine the action of the cooling equipment. It then simulates the subsequent effect on the space and the energy consumed by the modelled equipment. In single supply air duct systems (e.g., *system-type VAVS and PVAVS*), DOE-2.1E models the effect of the cooling equipment in two locations in the algorithm, DKTEMP and SDSF. The DKTEMP subroutine determines the supply air temperature that is achievable using the cooling equipment. It models the cooling equipment from the mixing of return and fresh air, to the delivery of the process air to the supply fan. The temperature value of the supply air is passed on to the TEMDEV routine where the flow rates to each zone of the VAV system are determined. TEMDEV also determines the zone temperatures achieved for the hour. The flow rate is determined by proportional thermostatic control (*control input variable, thermostatic proportional* ). The return air temperature is calculated as the average weighted value of the return air from all the zones. The supply air flow is then summed from the flow rate demand from each of the zones. The values for the air flow rate and the return air temperature are then used by the SDSF routine to calculate the system cooling equipment again. In the SDSF routine the building latent load and cooling effect is calculated as well as the energy performance of the equipment.

## 7.1 Input File Parameters Needed to Use the Inserted Functions

Some system file variables must be specified as follows for the computer model to perform as expected. Failure to set these variables can have unexpected results.

The system specified must be of the single deck type. The single deck system types call on the SDSF subroutine where the SDSFDES function is inserted. Single deck types that have been tested by the author are the Packaged Variable Air Volume System (PVAVS) and the Variable Air Volume System (VAVS). Although other single air deck systems may be specified, the results should be carefully checked by the user to determine that the model has performed as expected. Inserting a function into SDSF modifies all the systems using it. Therefore, multiple single deck systems should not be modelled unless the user desires that all the systems make use of the DKTEMPDES and SDSFDES functions.

The cooling coil size (*system input variable, system-type*) must be specified in the DOE-2.1E system input file to a negligible value. If no cooling coil is specified, or if its size is specified to be zero, DOE-2.1E will automatically size the cooling coil and employ it in the DKTEMP subroutine. This would seemingly be an asset, as a combination desiccant/evaporative, direct expansion coil system could be modelled. However, the SDSFDES function is inserted in the DOE-2.1E algorithm at the SDSF-1 function call, after the cooling coils are calculated. The SDSF subroutine will calculate the cooling effect of a cooling coil regardless of the specified size of the coil. Therefore the input file must also specify that the cooling equipment is scheduled to be off. This is set in the *cooling-schedule* of the DOE-2.1E input file.

## 7.2 Control Variables for the Inserted Functions

The two functions, DKTEMPDES and SDSFDES model the effects of the cooling equipment as dictated by the control variables that are set by the user. These control variables allow the user to set several parameters to control the cooling process.

**Humidistat Control of the Desiccant Dehumidifier and Electric Humidifier**

The maximum humidity level set by the user in the DOE-2.1E system input file controls the activation of desiccant dehumidification of the supply air. Similarly the minimum humidity level set by the user will activate the electric humidifier. The desiccant wheel will also become active if the return air humidity is above the user specified maximum humidity allowed in the building. This value is set in the DOE-2.1E system load file. The minimum allowable humidity may also be set, and the electric humidifier will control the minimum humidity according to the value set in the system input file. The system input file variables are *MAX-HUMIDITY* and *MIN-HUMIDITY* respectively.

**Supply Air Temperature Control**

The user must specify the minimum and maximum temperatures for the supply air that the system will allow. $T_{min}$ is the minimum supply air temperature that the evaporative coolers will cool the supply air to. It is equivalent, and should be set to the same value as the MIN-SUPPLY-T variable in the DOE-2.1E system input file. $T_{max}$ represents the maximum temperature that will be allowed without the desiccant dehumidifier becoming active. If, after the evaporative cooling the air temperature is above $T_{max}$, then desiccant dehumidification will be increased until the air temperature is below $T_{max}$. By increasing the amount of dehumidification, the wet bulb depression is increased so that the evaporative coolers may achieve a lower exiting temperature.

***Ratio of dehumidified air***

The desiccant wheel routine will model the dehumidification of the fraction of total supply air specified by the user. This desiccant fraction, or $R_d$ is calculated by the algorithm from the user entered parameters $N_{design}$ and $N_{installed}$. To obtain values for $N_{design}$ and $N_{installed}$ the user must select a size for the desiccant wheel(s). The face area of the wheel is dependent on the size of the wheel selected and therefore must be entered into the algorithm by the user so that the face velocity may be calculated. For the desiccant wheel modelled in the Source Code section, the design face velocity is 800 ft/min (4.1 m/s), therefore the design airflow capacity of the wheel and face area are related by the equation:

$$V_{wheel} = Area \bullet 800 \tfrac{ft}{min}$$

or in SI units:

$$V_{wheel} = Area \bullet 4.1 \tfrac{m}{s}$$

Once the user selects the size of the wheel the number of wheels needed to accommodate the design air flow rate of the system, $N_{design}$ must be entered. The value of $N_{design}$ calculated by the user is:

$$N_{design} = \frac{V_{design}}{V_{wheel}}$$

The resulting $N_{design}$ may not be an integer value. However, the actual number of wheels installed in the system is defined by the desiccant fraction, or $R_d$. The optimum value for $R_d$ may be determined through parametric studies. The actual number of wheels that are present in the system is then:

$$N_{installed} = R_d \bullet N_{design}$$

As $N_{installed}$ is the actual number of wheels that is modelled in the algorithm, it must be an integer value. The user may therefore:

1. Round the value calculated for $N_{active}$ to the nearest integer. As $N_{active}$ and $N_{design}$ are the values that are entered into the algorithm, then the resulting $R_d$ value may be different than the user's original intention.
2. Judiciously size the desiccant wheel to obtain $N_{active}$ and $N_{design}$ values that produce the desired desiccant fraction of $R_d$. If an economic study is to be performed, it is recommended that the user choose a desiccant wheel so that a reasonable estimate can be made of its capital cost.

***Note:*** ***The number of wheels that are active for the airflow required for the modelled hour is $N_{active}$. The calculation of $N_{active}$ is described in the Mathematical Description.***

**Time slice for activation of desiccant wheels**

To better model the desiccant wheel the model is capable of working within the modelled hour interval that DOE-2.1E uses. To do this the model will calculate two supply air temperature and humidity values, one with the desiccant wheels active and one without. The supply air delivered to the building for the modelled hour will then be a weighted average of these two conditions. The weighted average is arrived at by using the time step entered by the user. The weighted average will be a even multiple of the time step which must be an integer division of an hour. The algorithm determines the minimum amount of time the desiccant is needed to be active for the hour so that the maximum return humidity and $T_{max}$ values are satisfied. The time step variable is entered in units of hours and the user should not enter a value less than the time required to complete one air change of the modelled building.

**Evaporative cooler and heat exchanger parameters**

Finally the effectiveness parameters for the direct evaporative coolers and the air to air heat exchanger must be specified by the user in the DKTEMPDES and SDSFDES function code.

**Notes about DOE-2.1E system input file**

In addition, the cooling system (COOLING-SCHEDULE) must be scheduled to be OFF for the entire simulation run. If COOLING-ON is scheduled then DOE will attempt to simulate the cooling coils before the desiccant/evaporative cooling system. Also the cooling coil capacity (COOLING-CAPACITY) must be given a nominal value of 1, assigning a value of zero will cause DOE to automatically size the cooling coil, and make use of it.

### 7.3 The DKTEMPDES Function

In the DOE-2.1E algorithm, the DKTEMP subroutine serves to determine the supply air temperature that may be achieved with the air handling equipment. This temperature is then used for the calculations of heat addition and subtraction in the zones.

Therefore the purpose of the DKTEMPDES function is to calculate the supply air temperature that is achievable by the desiccant and evaporative cooling equipment. The DKTEMPDES function receives from DOE-2.1E the outdoor air conditions, and the mixed air temperature exiting the fresh air economiser for the present hour being modelled. In addition the values for the humidity, flow rate and the absolute humidity increase of the building due to latent load are taken from the previously modelled hour.

The DKTEMPDES function simulates the air handling equipment twice. The first time with desiccant dehumidification inactive; the second time with desiccant dehumidification active. Each loop stores the resulting supply air temperature and humidity. It also calculates the return air humidity based on the previous hour's values for latent heat load and volume flow rate. Then the weighted average of these two conditions of supply air is calculated that satisfies the time step, the supply air temperature and return air humidity parameters. The calculated supply air temperature is returned to the DKTEMP subroutine as *TMMIN* and will be selected by DKTEMP as the supply air temperature (*TC*), provided that the cooling coil's size is defined in the system input file as 1 watt (see section 7.1).

### 7.4 Assigning Variables to DKTEMPDES from DOE-2.1E DKTEMP Subroutine

The DKTEMPDES function is implemented before DOE-2.1E calculates the cooling equipment specified in the input file and after the economiser calculates the mixed air temperature. As the input file must specify that the cooling coil size has a negligible value (see section 7.1) the DOE-2.1E cooling calculations will have a negligible effect. In the DKTEMPDES function the *TMMIN* variable is taken from the DKTEMP subroutine. This variable represents the coldest temperature that is exiting the economiser equipment. The *TMMIN* variable must be returned to the DKTEMP subroutine. If *TMMIN* is less than the *COOL-SET-T* specified in the system input file then the heating coils will be employed to heat the supply air to the *COOL-SET-T*. The variables assigned from the DKTEMP subroutine to the DKTEMPDES function are tabulated below.

### 7.5 The SDSFDES Function

The SDSF subroutine serves the DOE-2.1E algorithm by modelling the equipment, determining the building humidity level, and the energy consumption of the various system components of the air handling equipment. Before the SDSF subroutine is called on, DOE-2.1E has already modelled the heat extraction rates from the zones based on the supply air temperature calculated by DKTEMP. The return air temperature from the zones is therefore known for the current hour being modelled and is used by the subroutine. The SDSFDES function accepts from the SDSF subroutine values for the mixed air temperature and humidity leaving the fresh air economizer, the total latent heat gain for the zones serviced by the system, and the air flow demanded.

The SDSFDES function first models the air handling equipment without the desiccant wheel being active. If the calculated return air humidity is above the specified maximum then the calculation is repeated with the desiccant wheel active. The function repeats the air handling calculations seeking convergence of two variables.

The first convergence sought is the mixed air humidity. The supply air humidity is dependant on the entering mixed air humidity that is in turn dependant on the return air humidity and thus the supply air humidity. The function iterates the equipment performance until the value for $W_{ma}$ converges.

The second convergence sought involves the new value for CFM. The temperature of the supply air obtained by the equipment in the SDSFDES function, may not be the same as that calculated by DKTEMPDES. The air flow rate is therefore adjusted to deliver the same quantity of sensible cooling to the space. As the performance of the desiccant wheel is dependant on the face velocity, the loop is iterated until the air flow rate value converges.

After the supply air temperature and absolute humidity are calculated, the gas consumption by the desiccant wheel and the water used by the evaporative coolers are determined.

### 7.6 Assigning Variables to SDSFDES function from DOE-2.1E SDSF Subroutine

The SDSFDES function is implemented at the end of the SDSF subroutine in the DOE-2.1E algorithm. Thus DOE-2.1E has already calculated the system (e.g. PVAV) cooling equipment. The input file must specify that the *COOL-SCHEDULE* is set to off (see section 7.1) to ensure that no cooling will occur in the SDSF subroutine. If this is not set then the SDSF subroutine will have already used the system cooling coils to cool the supply air temperature to the value set by the DKTEMP subroutine, *TC*.

The variables received from DOE-2.1E into the SDSFDES function are tabulated in section 9. Any variable that is taken from the DOE-2.1E algorithm using the assign statement may be modified and will be returned to the DOE-2.1E algorithm with the new value.

## 8. Flowchart

N.A.

# 9. Source code

**Specifying the DOE-2.1E system type variables**

```
INPUT SYSTEMS  INPUT-UNITS  =METRIC  OUTPUT-UNITS=METRIC ..
$
-----------ZONES INFORMATION REMOVED FOR BREVITY-------------
$
SUBR-FUNCTIONS DKTEMP-2  =*DKTEMPDES*
          SDSF-1   =*SDSFDES* ..
$ This statement (above) allows the inclusion of the functions
$ Of note in the system is that the PVAV system is specified with
$ the cooling coil sized at 1W, (0 watt causes DOE to automatically
$ size a cooling coil) The COOLING-SCHEDULE is set to never allow cooling
MAIN =SYSTEM   SYSTEM-TYPE   =PVAVS
                ZONE-NAMES    =(B-INT1,B-INT2,HALLB,STORAGE,GD-N1,GD-WNW,
                             GD-W,GD-SW,GD-S1,GD-INT1,GD-N2,GD-E,GD-S2,
                             GD-INT2,HALLG,FOYER,L2-N1,L2-WNW,L2-W,L2-SW,
                             L2-S1,L2-INT1,L2-N2,L2-E,L2-S2,L2-INT2,HALL2,
                             L2-BR,LUNCH,RECEPTION,PLENUM-1,PLENUM-2,
                             PLENUM-3,SPRINKLER,STAIR1,STAIR2)
                PLENUM-NAMES  =(PLENUM-1,PLENUM-2,PLENUM-3)
                COOLING-CAPACITY  =1
                HEATING-CAPACITY  =-270000
                MIN-SUPPLY-T  =13
                COOL-SET-T    =13
                MAX-HUMIDITY  =60
                MIN-HUMIDITY  =30
                COOL-CONTROL  =CONSTANT
                COOLING-SCHEDULE   =COOL-ON-SCHED
                HEATING-SCHEDULE   =HEAT-ON-SCHED
                OA-CONTROL    =ENTHALPY
                FAN-SCHEDULE  =FANS-1
                FAN-CONTROL   =SPEED
           SUPPLY-STATIC =102
           SUPPLY-EFF    =0.675
           SUPPLY-MECH-EFF   =0.75
           RETURN-STATIC =10.2
           RETURN-EFF    =0.675
                NIGHT-CYCLE-CTRL   =STAY-OFF
           HEAT-SOURCE   =HOT-WATER
                COOL-FUEL-METER  =M3
                BASEBOARD-SOURCE   =HOT-WATER
```

## The DKTEMPDES Function

```
FUNCTION NAME =DKTEMPDES ..
$ This function was written for DOE-2.1E by:
$ Stephen L. Kemp
$ Technical University of Nova Scotia
$ Halifax, Nova Scotia
$ Canada
$ PO Box 1000
$ B3J 2X4
$ (902) 420-2602
$ kempsl@tuns.ca
$
$ This function must be inserted at function call DKTEMP-2
$
$ This function will estimate the supply air temperature that will be supplied
$ by the air handling equipment. As the return air temperature and return humidity
$ are not yet determined the values from the previous hour are used. The equipment
$ is modeled so that the desiccant wheel is only active in increments of 15 minutes.
$ The total amount of desiccant cooling will be that which will just satisfy the
$ minimum supply humidity condition. The supply air temperature for the desiccant active
$ versus no active desiccant wheel will be averaged for the hour. This should allow
$ for a minimal amount of CFM adjustment between the DKTEMP and SDSF routines.
$
$ If heating is needed then it will be called upon in the SDSF routine.
$
$ The ASSIGN statements link the Function variables (left side) to the DOE-2.1E
$ algorithm variables (right side). When the function is completed any changes to
$ these variables will be reflected in the DOE-2.1E algorithm.
ASSIGN TMMIN=TMMIN IMO=IMO IDAY=IDAY IHR=IHR
    PASTWR=PASTWR DW=DW
    DBT=DBT WBT=WBT HUMRAT=HUMRAT PATM=PATM
    POMIN=POMIN POMAX=POMAX CFM=CFM
    WCOIL=WCOIL WRMAX=WRMAX
    TRP=TRP ..
CALCULATE ..
C-- The MODE parameter defines the type of system to be modeled.
C-- Mode 1    Desiccant/Indirect/Direct
C-- Mode 2    Desiccant/Indirect
C-- Mode 3    Indirect/Direct
C-- Mode 4    Indirect
C-- Mode 5    Direct
C-- TMAX,TMIN -- the supply temperature control parameters degC (will be
                  converted to degF)
C-- WHEELS -- the DESIGN number of wheel for the design airflow (user
```

```
C--            calculated)
C-- INSTALLW -- the number of wheels installed, NOTE: INSTALLW/WHEELS is
C--              is the user defined parameter for the desiccant fraction
C--              of the air to be dehumidified. IMSTALLW must be a
C--              positive integer.
C-- WHEELAREA -- The face area of the selected desiccant wheel, ft^2
C-- CFMWHEEL -- The design capacity of the selected wheel. Must
C--               correspond to WHEELAREA, cuft/min
C-- TIME -- Time step selected by the user, hour
C-- AUTO -- BOOLEAN control variable, if set to 1 than model will
C--         number of wheel to be used will vary with the demanded
C--         airflow rate. If set to 0 the number of wheels used
C--         will always be INSTALLW.
C-- EVAPEFF-- The effectiveness of the evaporative coolers
C-- EXEFF -- The effectivess of the indirect evaporative components
C--          heat exchanger.
   MODE=1
   TMAX=18
   TMIN=15
   WHEELS=5.333
   INSTALLW=4
   WHEELAREA=12.38
   CFMWHEEL=9903
   TIME=0.167
   AUTO=1
   EVAPEFF=0.9
   EXEFF=0.8
C-- Converts the temperature control parameters into degF from degC
   TMAX=(TMAX+273.15)*9/5-459.67
   TMIN=(TMIN+273.15)*9/5-459.67
C-- Initialize for calculating the two conditions, desiccant and no
C-- desiccant
   DESON=0
C-- The maximum allowable supply air humidity to meet the latent load
   WCMAX=WRMAX-DW
C-- If TRP = TMMIN then we know that 100% O/A is being used and
C-- therefore the maximum outside air (POMAX) should be used else the
C-- calculated minimum air flow ratio is used based on the previous hour
5  CONTINUE
   TDES=0
   TDIR=0
   TIND=0
```

```
   TCOIL=0
   WCOIL=0
   IF ( TRP .EQ. TMMIN ) GO TO 6
      RATIO=POMAX
      GOTO 7
6  CONTINUE
   RATIO=POMIN
7  CONTINUE
   WAIR=RATIO*HUMRAT+(1-RATIO)*PASTWR
   WENTER=WAIR
   TAIR=TMMIN
   TENTER=TMMIN
C-- Calculate the supply air for no desiccant active.
   IF ( DESON .EQ. 0 ) GO TO 20
C-- THE DESICCANT WHEEL
10  CONTINUE
   DESON=1
C-- The humidity value is converted into grains/lb
   Wgrain=WENTER*7000
   TAIR=TMMIN
C-- Dummy variable from regression analysis
   D=0.002072*Wgrain*Wgrain+0.005814*TAIR*TAIR
   1+0.003193*Wgrain*TAIR-0.08758*Wgrain-0.50283*TAIR+9.94773
C-- The face velocity is determined for the S-30 wheel, where the nominal
C-- area is 12.5 ft^2
   IF ( AUTO .EQ. 1 ) ACTIVE=CFM*(INSTALLW/WHEELS)/CFMWHEEL
   IF ( AUTO .EQ. 0 ) ACTIVE=INSTALLW
   ACTIVE=INT(ACTIVE+0.9)
   IF ( ACTIVE .GT. INSTALLW ) ACTIVE=INSTALLW
   FV=CFM*(INSTALLW/WHEELS)/(ACTIVE*WHEELAREA)
C-- The face velocity and dummy variable determine the exiting humidity
   WPO=(0.00004104*D*D-0.00005313*FV*FV+0.0004644*D*FV
   1+0.69491*D+0.07842*FV-24.8076)/7000
C-- Leaving Purge air temperature
   TPRG=204.75+0.0758*Wgrain
C-- K-Factor
   K=-2.1964E-6*Wgrain*Wgrain+1.5714E-7*FV*FV-3.8982E-7*Wgrain*FV
   1+8.6032E-4*Wgrain-2.3875E-4*FV+0.11045
C-- The outlet temperature
   TPO=TAIR+0.625*(Wgrain-WPO*7000)+K*(TPRG-TAIR)
   TAIR=TPO*INSTALLW/WHEELS+TAIR*(1-INSTALLW/WHEELS)
```

```
      WAIR=WPO*INSTALLWWHEELS+WAIR*(1-INSTALLWWHEELS)
C-- The required purge air volume
C-- End of the desiccant routine and beginning of the evaporative
C-- coolers.
C-- The indirect evaporative cooler.
C-- The indirect evaporative cooler cools the air using
C-- outdoor ambient air and therefore the wetbulb depression
C-- is with respect to the outdoor wetbulb temperature.
20    CONTINUE
      IF ( MODE .EQ. 5 ) GO TO 30
      IF ( TAIR .LT. TMIN ) GO TO 70
      T=TAIR-EXEFF*(TAIR-DBT)-EXEFF*EVAPEFF*(DBT-WBT)
      IF ( T .GE. TMIN ) GO TO 22
         TAIR=TMIN
         TIND=TMIN
         WIND=WAIR
         GO TO 70
22    CONTINUE
      TAIR=T
      TIND=TAIR
      WIND=WAIR
      IF ( MODE .EQ. 2 ) GO TO 70
      IF ( MODE .EQ. 4 ) GO TO 70
      IF ( TIND .LT. TMIN ) GO TO 70
C-- THE DIRECT EVAPORATIVE COOLER
30    CONTINUE
      ENTHALPY=H(TAIR,WAIR)
      TWBAIR=WBFS(TAIR,WAIR,PATM)
      TAIR=TAIR-EVAPEFF*(TAIR-TWBAIR)
      TDIR=TAIR
      IF ( TDIR .GT. TMIN ) GO TO 31
         TAIR=TMIN
         TDIR=TAIR
C-- The root solver
31    CONTINUE
      HTEST=H(TAIR,WAIR)
32    IF (HTEST.GT.ENTHALPY) GO TO 33
      WAIR=WAIR+0.001
      HTEST=H(TAIR,WAIR)
      GO TO 32
33    CONTINUE
```

```
34  IF (HTEST.LT.ENTHALPY) GO TO 35
      WAIR=WAIR-0.0005
      HTEST=H(TAIR,WAIR)
      GO TO 34
35  CONTINUE
36  IF (HTEST.GT.ENTHALPY) GO TO 37
      WAIR=WAIR+0.00001
      HTEST=H(TAIR,WAIR)
      GO TO 36
37  CONTINUE
    WDIR=WAIR
    GO TO 70
70  CONTINUE
    IF ( DESON .EQ. 1 ) GO TO 71
      TAIRNODES=TAIR
      WNODES=WAIR
      DESON=1
      GO TO 10
71  CONTINUE
    TAIRDES=TAIR
    WDES=WAIR
    DESWEIGHT=0
C-- Calculate the fraction of hour desiccant dehumidification is needed
C-- and store the process air temperature TAIR, in the supply air temperature
C-- TC variable for the DKTEMP subroutine.
72  CONTINUE
    WAIR=WDES*DESWEIGHT+WNODES*(1-DESWEIGHT)
    TAIR=TAIRDES*DESWEIGHT+TAIRNODES*(1-DESWEIGHT)
    IF ( WAIR .LE. WCMAX .and. TAIR .LE. TMAX ) GO TO 75
      IF ( DESWEIGHT .GE. 1 ) GO TO 75
      DESWEIGHT=DESWEIGHT+TIME
      GO TO 72
75  CONTINUE
    IF ( DESWEIGHT .GE. 1 ) DESWEIGHT=1
    IF ( MODE .GE. 3 ) DESWEIGHT=0
    TAIR=TAIRDES*DESWEIGHT+TAIRNODES*(1-DESWEIGHT)
    WAIR=WDES*DESWEIGHT+WNODES*(1-DESWEIGHT)
80  CONTINUE
90  CONTINUE
    WCOIL=WAIR
    IF ( TAIR .LE. TMIN ) TAIR=TMIN
    TCOIL=(TAIR+459.67)/1.8-273.15
    TMMIN=TAIR
```

## 9.1 Nomenclature for DKTEMPDES

| Variable | Description | Units |
|---|---|---|
| ACTIVE | Number of desiccant wheels active for the hour | |
| AUTO | Boolean control variable: 1 for automatically choosing the number of active desiccant wheel for the hour, 0 for constant. | |
| CFM | The air flow rate for the hour. Taken from DOE, modified if the supply air temperature changes. | $ft^3/min$ |
| CFMWHEEL | The design air flow capacity of the desiccant wheel | $ft^3/min$ |
| D | Intermediate variable used to calculate the dehumidification process | |
| DBT | Outdoor dry bulb temperature taken from DOE-2.1E | °F |
| DESON | Boolean variable: 1 for desiccant wheels active 0 for inactive | |
| DESWEIGHT | The fraction of an hour the desiccant wheels are active | |
| DW | The increase in the building absolute humidity due to latent loads | $lb_{water}/lb_{air}$ |
| ENTHALPY | Enthalpy value for the air | Btu/lb |
| EVAPEFF | Effectiveness of the evaporative coolers | |
| EXEFF | Indirect component, heat exchanger effectiveness | |
| FV | Face velocity of air on desiccant wheel | ft/min |
| HTEST | Enthalpy value, used in root solver | Btu/lb |
| HUMRAT | Outdoor absolute humidity value | $lb_{water}/lb_{air}$ |
| INSTALLW | Number of desiccant wheels installed | |
| K | K factor, used to determine the heat carry over from the regeneration to dehumidification process | |
| MODE | Mode of operation for the algorithms | |
| PASTWR | Previous hour return air absolute humidity | $lb_{water}/lb_{air}$ |
| PATM | Ambient air pressure, taken from DOE-2.1E | in-$H_2O$ |
| POMAX | Maximum value for fresh air ratio, taken from DOE-2.1E | |
| POMIN | Minimum value for fresh air ratio, taken from DOE-2.1E | |
| RATIO | *Actual fresh air ratio* | |
| T | Temperature value used in root solvers | °F |
| TAIR | Process air temperature | °F |
| TAIRDES | Process air temperature for no desiccant dehumidification used | °F |

| Variable | Description | Units |
|---|---|---|
| TAIRNODES | Process air temperature for desiccant dehumidification used | °F |
| TDES | Air temperature after mixing with the dehumidified and non dehumidified air | °F |
| TDIR | Air temperature after the direct evaporative cooler | °F |
| TENTER | Saved value of the DOE-2.1E air temperature input into the function | °F |
| TIME | Time step parameter | hour |
| TIND | Air temperature after the direct evaporative cooler | °F |
| TMAX | Supply temperature control parameter, maximum temperature | °F |
| TMIN | Supply temperature control parameter, minimum temperature | °F |
| TMMIN | Air temperature received from DOE-2.1E DKTEMP sub-routine, returned as the new temperature after desiccant evaporative cooling | °F |
| TPO | Air temperature leaving individual desiccant wheel and before mixing with non dehumidified air. | °F |
| TPRG | Purge air temperature for the desiccant wheel | °F |
| TRP | Temporary variable used by DOE-2.1E to determine the fresh air ratio | °F |
| TWBAIR | Process wet bulb temperature | °F |
| WAIR | Process air absolute humidity | $lb_{water}/lb_{air}$ |
| WBT | Outdoor wet bulb temperature | °F |
| WCMAX | Maximum value that the supply air may be and still satisfy the latent cooling load | $lb_{water}/lb_{air}$ |
| WDES | Absolute humidity leaving the desiccant component | $lb_{water}/lb_{air}$ |
| WDIR | Absolute humidity leaving the direct evaporative component | $lb_{water}/lb_{air}$ |
| WENTER | Saved value for the absolute humidity entering the cooling equipment | $lb_{water}/lb_{air}$ |
| WGRAIN | Absolute humidity in grains/lb for the desiccant process regression equations | $lb_{water}/lb_{air}$ |
| WHEELAREA | Face area of the desiccant wheel's process side | $ft^2$ |
| WHEELS | Number of desiccant wheels needed for the design air flow | |
| WIND | Absolute humidity leaving the indirect evaporative component | $lb_{water}/lb_{air}$ |
| WNODES | Process air humidity for no desiccant wheel dehumidification | $lb_{water}/lb_{air}$ |
| WPO | Absolute humidity leaving the desiccant wheel before re-mixing with the non dehumidified air stream | $lb_{water}/lb_{air}$ |
| WRMAX | Maximum return air absolute humidity allowed, taken from DOE-2.1E | $lb_{water}/lb_{air}$ |

## 9.2 The SDSFDES Function

```
FUNCTION NAME =SDSFDES ..
$ This function was written for DOE-2.1E by:
$ Stephen L. Kemp
$ Technical University of Nova Scotia
$ Halifax, Nova Scotia
$ Canada
$ PO Box 1000
$ B3J 2X4
$ (902) 420-2602
$ kempsl@tuns.ca
$
$ This function must be inserted at function call SDSF-1
$
$ This function will determine the humidity ratio leaving the air handling equipment
$ as well as the amount of energy expended in the process. It does this by calculating
$ the exiting conditions of the supply air after being processed by the cooling
$ equipment. A second loop of the air handling equipment is then calculated using
$ the newly calculated return air humidity value for the present hour. This is repeated
$ until the humidity value difference is less than 0.0001 kg/kg or 0.7 grains/lb
$
$ The system attempts to cool the air to at least TMIN. Should a temperature of at
$ least TMAX not be achieved then the desiccant unit is used for a longer period of
$ time. Humidity is controlled by a simple two position humidistat that triggers the
$ desiccant wheel when it is needed as determined by the return humidity.
$ The desiccant wheel may be reported to be active for fractions of an hour.
$ Currently the algorithm allows for these hourly fractions to be 15 minutes, however
$ the manufacturer will need to be consulted before a final value is determined.
$
$ As this function bypasses the electric humidifier in DOE one is simulated here.
$ After the first loop, should the return air be calculated as below the minimum
$ requirements than the humidifier is activated.
$
$ The logic of the algorithm is to use indirect/direct evaporative cooling to reduce
$ the supply air temperature. If the calculated return air humidity is above the max
$ value then the calculation is repeated using the desiccant unit. A weighted value
$ of the supply air condition is used to simulate partial hour activation of the
$ desiccant unit. The weighted value uses both the result for desiccant/indirect/direct
$ cooling and indirect/direct cooling. The weighted values are simulated, as mentioned,
$ to allow for the desiccant wheel to be active for multiple values if 1/4 hour.
$ The ASSIGN statements link the Function variables (left side) to the DOE-2.1E
$ algorithm variable (right side). When the function is completed any changes to
```

```
$ these variables will be reflected in the DOE-2.1E algorithm.
ASSIGN IMO=IMO IDAY=IDAY IHR=IHR
     PO=PO FON=FON HON=HON
     TR=TR WR=WR WRMAX=WRMAX WRMIN=WRMIN
     TM=TM WM=WM PATM=PATM DBT=DBT WBT=WBT
     PASTWR=PASTWR HUMRAT=HUMRAT
     GW=GW F=F G=G DW=DW
     WCOIL=WCOIL CFM=CFM, RCFM=RCFM
     TC=TC WW=WW
     SKW=SKW FANKW=FANKW VENTKW=VENTKW SKWQC=SKWQC
     SCGAS=SCGAS QREG=QREG DGAS=DGAS COOLFL=COOLFL
     QHUM=QHUM COOLKW=COOLKW SKWQH=SKWQH AUXKW=AUXKW
     QREGP=QREGP ..
CALCULATE ..
C-- The MODE parameter defines the type of system to be modeled.
C-- Mode 1    Desiccant/Indirect/Direct
C-- Mode 2    Desiccant/Indirect
C-- Mode 3    Indirect/Direct
C-- Mode 4    Indirect
C-- Mode 5    Direct
C-- TMAX,TMIN -- the supply temperature control parameters degC (will be
                 converted to degF)
C-- WHEELS -- the DESIGN number of wheel for the design airflow (user
C--               calculated)
C-- INSTALLW -- the number of wheels installed, NOTE: INSTALLW/WHEELS is
C--                 is the user defined parameter for the desiccant fraction
C--                 of the air to be dehumidified. IMSTALLW must be a
C--                 positive integer.
C-- WHEELAREA -- The face area of the selected desiccant wheel, ft^2
C-- CFMWHEEL -- The design capacity of the selected wheel. Must
C--              correspond to WHEELAREA, cuft/min
C-- TIME -- Time step selected by the user, hour
C-- AUTO -- BOOLEAN control variable, if set to 1 than model will
C--           number of wheel to be used will vary with the demanded
C--           airflow rate. If set to 0 the number of wheels used
C--           will always be INSTALLW.
C-- EVAPEFF -- The effectiveness of the evaporative coolers
C-- EXEFF-- The effectiveness of the indirect evaporative components
C--            heat exchanger.
    MODE=1
    TMAX=18
    TMIN=15
    WHEELS=5.333
```

```
      INSTALLW=4
      WHEELAREA=12.38
      CFMWHEEL=9903
      TIME=0.167
      AUTO=1
      EVAPEFF=0.9
      EXEFF=0.8
C-- The above parameters are to be identical to those found in DKTEMPDES
C-- REXEFF -- The effectiveness of the heat recovery unit
C-- BEFF -- The efficiency of the gas fired air heater used in the
C--          regeneration process.
C-- EFREG -- Regeneration fan static efficiency
C-- RGHCPD -- Regeneration Heat Recovery Pressure Drop (in-Water)
C-- EFEVP-- Indirect Evaporative Cooler Fan Static Efficiency
C-- EXPD-- Pressure Drop (in-Water) across indirect heat exchanger
      REXEFF=0.77
      BEFF=0.85
      EFREG=0.75
      RGHCPD=0.75
      EFEVP=0.75
      EXPD=0.67
C-- Converts the temperature control parameters into degF from degC
      TMAX=(TMAX+273.15)*9/5-459.67
      TMIN=(TMIN+273.15)*9/5-459.67
C-- Initialization variables
      FINT=F
      FINT=0
      DESON=0
      DESWEIGHT=0
C-- The supply air humidity must fall between these two values.
      WCMAX=WRMAX-DW
      WCMIN=WRMIN-DW
C-- Save values for repeating the convergence loops
      WENTER=WM
      CFMOLD=CFM
      CFMNEW=CFM
C-- Iteration count for debugging if convergence does not occur
      ITER=0
      ITERCH=0
5     CONTINUE
C-- Initialize variable for repeating loop
      TDES=0
      WDES=0
```

```
      TDIR=0
      TIND=0
      WATDIR=0
      WATIND=0
      QREG=0
      TAIR=TC
      WAIR=WM
      LOOP=0
      QHUM=0
      ACTIVE=0
      REGPWR=0
      DESON=0
      DESWEIGHT=0
C-- If the fans are scheduled off then no cooling takes place and calculate
C-- the new humidity value in the space at line 100
      IF ( FON .EQ. 0 ) GO TO 100
C-- If a previous loop determined that desiccant would be needed then
C-- ensure desiccant is used.
      IF ( DESON .NE. 0 ) GO TO 6
C-- If no sensible cooling needed (i.e. mixed air temp below TMIN
C-- then continue and calculate the new return humidity value and determine if
C-- humidity needs to be addressed
      IF ( TM .LE. TMIN ) GO TO 70
C-- Check to see if sensible cooling is needed, if after performing evaporative
C-- cooling the humidity is too high then desiccant wheel will become active.
      IF ( TM .GT. TMIN ) GO TO 20
6     CONTINUE
C-- Desiccant wheel is to be active. First loop calculates the conditions
C-- for 1/4 hour, then 1/2 etc. until dehumidification is adequate.
      DESON=1
      DESWEIGHT=DESWEIGHT+TIME
      IF ( DESWEIGHT .GT. 1.0 ) DESWEIGHT=1.0
7     CONTINUE
C-- If not a Desiccant Dehumidification Mode got to indirect evap
      IF ( MODE .GE. 3) GO TO 20
C-- If returning from evaporative cooling only then return to top of evaporative
C-- cooling only
      IF ( DESON .NE. 1 ) GO TO 20
C-- Dehumidification has been determined to be needed.
C-- Calculate the return air humidity if the supply air has been dried to the required humidity
C-- level to meet the latent cooling requirements of the building.
C-- Desiccant Equations for commercial Wheel
C-- The number of wheels to be active for the modeled hour are determined
```

```
    IF ( AUTO .EQ. 1 ) ACTIVE=CFMNEW*(INSTALLW/WHEELS)/CFMWHEEL
    IF ( AUTO .EQ. 0 ) ACTIVE=INSTALLW
    ACTIVE=INT(ACTIVE+0.9)
    IF ( ACTIVE .GT. INSTALLW ) ACTIVE=INSTALLW
    FV=CFMNEW*(INSTALLW/WHEELS)/(ACTIVE*WHEELAREA)
C-- The humidity value is converted into grains/lb
    Wgrain=WM*7000
C-- Dummy variable from regression analysis
    D=0.002072*Wgrain*Wgrain+0.005814*TM*TM
   1+0.003193*Wgrain*TM-0.08758*Wgrain-0.50283*TM+9.94773
C-- The face velocity and dummy variable determine the exiting humidity
    WPO=(0.00004104*D*D-0.00005313*FV*FV+0.0004644*D*FV
   1+0.69491*D+0.07842*FV-24.8076)/7000
C-- Leaving Purge air temperature
    TPRG=204.75+0.0758*Wgrain
C-- K-Factor
    K=-2.1964E-6*Wgrain*Wgrain+1.5714E-7*FV*FV-3.8982E-7*Wgrain*FV
   1+8.6032E-4*Wgrain-2.3875E-4*FV+0.11045
C-- Water removed per wheel, to be added to the regeneration air (kg/hr)
    WDREM=(WM-WPO)*CFMNEW*(INSTALLW/WHEELS)/13.3*60/2.2
C-- The outlet temperature
    TPO=TM+0.625*(Wgrain-WPO*7000)+K*(TPRG-TM)
    WDES=WPO*INSTALLW/WHEELS+WM*(1-INSTALLW/WHEELS)
    WNODES=WM
    TAIRDES=TPO*INSTALLW/WHEELS+TM*(1-INSTALLW/WHEELS)
    TAIRNODES=TM
    TPI=TM
C-- value for output file
    TDES=DESWEIGHT*TAIRDES+(1-DESWEIGHT)*TAIRNODES
C-- End of the desiccant routine and the beginning of the evaporative
C-- coolers.
C-- THE INDIRECT EVAPORATIVE COOLER
C-- The indirect evaporative cooler is a combination of direct evaporative cooler
C-- and heat exchanger
C-- The indirect evaporative cooler cools the air using
C-- outdoor ambient air and therefore the wetbulb depression
C-- is with respect to the outdoor wetbulb temperature.
C-- There are two calculations performed, with and without the desiccant
C-- so that a weighted value may be found later. If the desiccant system is
C-- off then the two conditions are the same.
20  CONTINUE
    IF ( DESON .EQ. 1 ) GO TO 22
       TAIRDES=TM
```

```
      TAIRNODES=TM
      WDES=WM
      WNODES=WM
   IF ( MODE .EQ. 5 ) GO TO 40
22  CONTINUE
C-- Cold side direct evaporative cooler exiting temperature
   TINDC=DBT-EVAPEFF*(DBT-WBT)
   THIN=TAIRDES
C-- Heat exchanger effectiveness is 0.8 and dir evap cooler is 0.9 (0.8*0.9=0.72)
   TAIRNODES=TAIRNODES-EXEFF*(TAIRNODES-DBT)-EXEFF*EVAPEFF*(DBT-WBT)
   TAIRDES=TAIRDES-EXEFF*(TAIRDES-DBT)-EXEFF*EVAPEFF*(DBT-WBT)
C-- If indirect cooler cools air to below TMIN then only cool to TMIN
   IF ( TAIRNODES .LT. TMIN ) TAIRNODES=TMIN
   IF ( TAIRDES .LT. TMIN ) TAIRDES=TMIN
C-- Root Solver for Indirect Cooler
C-- This root solver finds the absolute humidity value for the isenthalpic
C-- process of the integral direct evaporative cooler in the unit. Thus the
C-- water usage may be determined.
   INDW=HUMRAT
   ENTHALPY=H(DBT,HUMRAT)
   HTEST=H(TINDC,INDW)
23  CONTINUE
   IF (HTEST.GT.ENTHALPY) GO TO 24
      INDW=INDW+0.001
      HTEST=H(TINDC,INDW)
      GO TO 23
24  CONTINUE
   IF (HTEST.LT.ENTHALPY) GO TO 25
      INDW=INDW-0.0005
      HTEST=H(TINDC,INDW)
      GO TO 24
25  CONTINUE
   IF (HTEST.GT.ENTHALPY) GO TO 26
      INDW=INDW+0.00001
      HTEST=H(TINDC,INDW)
      GO TO 25
26  CONTINUE
C-- Calculate the temperature exiting the heat exchanger
   TCOUT=EXEFF*(THIN-TINDC)+TINDC
C-- Caclulate the change in absolute humidity
   WATIND=(INDW-HUMRAT)
   TIND=DESWEIGHT*TAIRDES+(1-DESWEIGHT)*TAIRNODES
   WIND=DESWEIGHT*WDES+(1-DESWEIGHT)*WNODES
```

```
C-- If air temp is above TMIN then go on to direct evaporative cooling
C-- condition
    IF ( MODE .EQ. 2 ) GO TO 30
    IF ( MODE .EQ. 4 ) GO TO 30
    IF ( TIND .GT. TMIN ) GO TO 40
C-- The indirect cooler has reduced the air temperature enough. If DESON=1 then
C-- what is the weighted average supply air condition.
30  CONTINUE
    IF ( MODE .GE. 3 ) GO TO 35
    IF ( DESON .NE. 1 ) GO TO 35
    WAIR=WIND
    TAIR=TIND
    WR=(WAIR+DW+FINT*HUMRAT+GW)/(1.0+FINT+G)
C-- Calculate the new mixed air condition
    WMNEW=HUMRAT*PO+(1-PO)*WR
C-- Check for convergence
    CONVERGE=ABS(WMNEW-WM)
    IF ( CONVERGE .LE. 0.0001 ) GO TO 33
       WM=WMNEW
       WAIR=WMNEW
       GO TO 7
C-- If return air is still too humid and wheel is not active for the entire
C-- hour then repeat with wheel active for longer
33  CONTINUE
    IF ( WR .GT. WRMAX .and. DESWEIGHT .LT. 1.0 ) GO TO 6
C-- If the supply air isn't cooled to at least TMAX then repeat with wheel
C-- active for a longer time
    IF ( TAIR .GT. TMAX .and. DESWEIGHT .LT. 1.0 ) GO TO 6
C-- Desiccant wheel is active and loads have been satisfied. Go to calculate
C-- the energy and water usage.
    GO TO 80
35  CONTINUE
C-- Indirect Evaporative Cooling only. Calculate the Supply air condition.
    TIND=TAIRNODES
    WIND=WNODES
    TAIR=TIND
    WAIR=WIND
    WR=(WAIR+DW+FINT*HUMRAT+GW)/(1.0+FINT+G)
C-- Calculate the new mixed air condition
    WMNEW=HUMRAT*PO+(1-PO)*WR
C-- Check for convergence
    CONVERGE=ABS(WMNEW-WM)
    IF ( CONVERGE .LE. 0.0001 ) GO TO 37
```

```
      WM=WMNEW
      WAIR=WMNEW
      GO TO 7
C-- If return air is too humid then activate the desiccant wheel
37  CONTINUE
    IF ( MODE .GE. 3 ) GO TO 39
    IF ( WR .LE. WRMAX ) GO TO 38
      DESON=1
      WAIR=WENTER
      WM=WENTER
      GO TO 6
38  CONTINUE
    IF ( TAIR .LE. TMAX ) GO TO 39
      DESON=1
      WAIR=WENTER
      WM=WENTER
      GO TO 6
39  CONTINUE
    WM=WMNEW
    WAIR=WMNEW
C-- Indirect cooling only, go to humidifier
    GO TO 70
40  CONTINUE
C-- THE DIRECT EVAPORATIVE COOLER
C-- Direct evaporative Cooling for desiccant wheel active
C-- If temperature is above TMIN then go to direct evap cooling,
C-- else go to possible humidification of the supply air.
    LOOP=0
C-- Save for water useage calculation
    WDIRINDES=WDES
    WDIRINNODES=WNODES
    T=TAIRDES
    W=WDES
    LOOP=1
41  CONTINUE
    GO TO 50
42  CONTINUE
    TAIRDES=T
    WDES=W
    T=TAIRNODES
    W=WNODES
    LOOP=2
    GO TO 50
```

```
47  CONTINUE
    TAIRNODES=T
    WNODES=W
    GO TO 60
50  CONTINUE
C-- If air is already below TMIN then no cooling
    IF ( T .LE. TMIN ) GO TO 57
    ENTHALPY=H(T,W)
    TWET=WBFS(T,W,PATM)
    T=T-EVAPEFF*(T-TWET)
C-- Do not allow air to be cooled below TMIN
    IF ( T .GE. TMIN ) GO TO 51
       T=TMIN
C-- The root solver
51  CONTINUE
    HTEST=H(T,W)
52  IF (HTEST.GT.ENTHALPY) GO TO 53
       W=W+0.001
       HTEST=H(T,W)
       GO TO 52
53  CONTINUE
54  IF (HTEST.LT.ENTHALPY) GO TO 55
       W=W-0.0005
       HTEST=H(T,W)
       GO TO 54
55  CONTINUE
56  IF (HTEST.GT.ENTHALPY) GO TO 57
       W=W+0.00001
       HTEST=H(T,W)
       GO TO 56
57  CONTINUE
    IF ( LOOP .EQ. 1 ) GO TO 42
    IF ( LOOP .EQ. 2 ) GO TO 47
C-- The outlet temperatures are to be determined. If DESON=1 then what
C-- is the weighted average supply air condition.
60  CONTINUE
    WATDIR=(WDES-WDIRINDES)*DESWEIGHT+
    1(WNODES-WDIRINNODES)*(1-DESWEIGHT)
    IF ( MODE .GE. 3) GO TO 65
    IF ( DESON .NE. 1 ) GO TO 65
    TDIR=TAIRDES*DESWEIGHT+TAIRNODES*(1-DESWEIGHT)
    WDIR=WDES*DESWEIGHT+WNODES*(1-DESWEIGHT)
    WAIR=WDIR
```

```
    TAIR=TDIR
    WR=(WAIR+DW+FINT*HUMRAT+GW)/(1.0+FINT+G)
C-- Calculate the new mixed air condition
    WMNEW=HUMRAT*PO+(1-PO)*WR
C-- Check for convergence
    CONVERGE=ABS(WMNEW-WM)
    IF ( CONVERGE .LE. 0.0001 ) GO TO 63
        WM=WMNEW
        WAIR=WMNEW
        GO TO 7
C-- If return air is still too humid and wheel is not active for the entire
C-- hour then repeat with wheel active for longer
63  IF ( WR .GT. WRMAX .and. DESWEIGHT .LT. 1.0 ) GO TO 6
C-- If the supply air isn't cooled to at least TMAX then repeat with wheel
C-- active for a longer time
    IF ( TAIR .GT. TMAX .and. DESWEIGHT .LT. 1.0 ) GO TO 6
C-- Desiccant wheel is active and loads have been satisfied. Go to calculate
C-- the energy and water usage.
    WM=WMNEW
    GO TO 80
65  CONTINUE
C-- Evaporative Cooling only. Calculate the Supply air condition.
    TDIR=TAIRNODES
    WDIR=WNODES
    TAIR=TDIR
    WAIR=WDIR
    WR=(WAIR+DW+FINT*HUMRAT+GW)/(1.0+FINT+G)
C-- Calculate the new mixed air condition
    WMNEW=HUMRAT*PO+(1-PO)*WR
C-- Check for convergence
    CONVERGE=ABS(WMNEW-WM)
    IF ( CONVERGE .LE. 0.0001 ) GO TO 67
        WM=WMNEW
        WAIR=WMNEW
        GO TO 7
C-- If return air is too humid then activate the desiccant wheel
67  CONTINUE
    IF ( MODE .GE. 3 ) GO TO 69
    IF ( WR .LE. WRMAX ) GO TO 68
      DESON=1
      WAIR=WENTER
      WM=WENTER
      GO TO 6
```

```
68  CONTINUE
   IF ( TAIR .LE. TMAX ) GO TO 69
      DESON=1
      WAIR=WENTER
      WM=WENTER
      GO TO 6
69  CONTINUE
   WM=WMNEW
   GO TO 80
C-- HUMIDIFIER FOR NO COOLING DONE OR INDIRECT ONLY
70  CONTINUE
   IF ( HON .EQ. 0 ) GO TO 75
C-- No cooling was needed check to see if humidification is needed
C-- Calculate the return air at these conditions
   WR=(WAIR+DW+FINT*HUMRAT+GW)/(1.0+FINT+G)
   IF ( WR .GT. WRMIN ) GO TO 80
C-- Humidifier will add humidity to the air until the minimum humidity
C-- is reached, thus the return air humidity is WRMIN.
C-- The humidifier type is specified in the DOE-2.1E input file.
C-- The supply air needed to meet WRMIN
   WAIRMIN=(1+FINT+G)*WRMIN-DW-FINT*HUMRAT-GW
C-- Make sure this is not greater than saturation
   WAIRMIN=AMIN(WAIRMIN,WFUNC(TAIR,100.0,PATM))
C-- Calc mix air condition at this supply cond
   WM=HUMRAT*PO+(1-PO)*WRMIN
C-- Calc Moisture addition
   WW=WAIRMIN-WM
C-- Calc Humidification energy
   QHUM=WW*1061*CFM/V(TAIR,WAIRMIN,PATM)*60

C-- Set leaving condition
   WAIR=WAIRMIN
75  CONTINUE
   GO TO 80
80  CONTINUE
90  CONTINUE
C-- Adjust air flow for new TC value also an iteration. When the change
C-- in CFM is greater than 5% then recalculate with new value.
   CFMOLD=CFMNEW
   CFMNEW=(TR-TC)/(TR-TAIR)*CFM
   IF ( CFMNEW .LE. 0.4*CFM ) CFMNEW=0.4*CFM
   RCFM=(TR-TC)/(TR-TAIR)*RCFM
   CHANGECFM=(CFMNEW-CFMOLD)/CFMOLD
```

```
      ITERCH=ABS(CHANGECFM)
      ITER=ITER+1
      IF ( ITER .GE. 50 ) GO TO 91
      IF ( ITERCH .GT. 0.025 ) GO TO 5
C-- Changes in CFM and TC from DKTEMP
91    CONTINUE
      CFMOLD=CFM
      CFMNEW=(TR-TC)/(TR-TAIR)*CFM
      CFM=CFMNEW
      CHANGECFM=(CFMNEW-CFMOLD)/CFMOLD
      TCOLD=TC
      TC=TAIR
C-- Adjust the fanpower for new air flow rate
      SKW=SKW-((CFMOLD-CFMNEW)/CFMOLD)*(SFKW+RFKW)
      FANKW=FANKW-((CFMOLD-CFMNEW)/CFMOLD)*(SFKW+RFKW)
      VENTKW=VENTKW-((CFMOLD-CFMNEW)/CFMOLD)*(SFKW+RFKW)
C-- Adjust Humidification energy
      QHUM=QHUM*CFMNEW/CFMOLD
      SKWQH=SKWQH+QHUM*0.000293
      AUXKW=AUXKW+QHUM*0.000293
C-- Calculate the GAS used for Regeneration if applicable
      IF ( DESON .NE. 1 ) GO TO 95
      CFMREGEN=(0.04*CFMWHEEL*(TPRG-TPI)+CFM*INSTALLW/
     1(ACTIVE*WHEELS)*(TPO-TPI))/(250-120)
C-- The Desiccant wheel exiting condition is not calculated, however
C-- the manufacturer's documentation specifies that the exiting temperature is always
C-- 120F. When compared to the results supplied by the manufacturer this is true within 2F
C-- for all reasonable conditions
C-- A heat recovery unit uses the regeneration air exiting the wheel
C-- to preheat the incoming regeneration air
      THI=REXEFF*(120-DBT)+DBT
      HHI=H(THI,HUMRAT)
      HDREG=H(250,HUMRAT)
      CFMREGEN=CFMREGEN*ACTIVE
      QREG=(CFMREGEN/13.1)*(HDREG-HHI)*60*DESWEIGHT
      QREGP=QREGP+QREG
      DGAS=QREG/BEFF
      SCGAS=SCGAS+DGAS
      COOLFL=COOLFL+DGAS
C-- Regeneration Fan Energy
      REGPRESS=2.025E-5*HUMRAT+3.72152E-4*CFMREGEN+0.01026+RGHCPD
      REGPWR=(CFMREGEN*REGPRESS)/(8524*EFREG)*DESWEIGHT
      COOLKW=COOLKW+REGPWR
```

```
      SKW=SKW+REGPWR
      SKWQC=REGPWR
C-- Desiccant Motor Power - not implemented, assumed negligible
C-- Calculate the water used in evaporative coolers
95    CONTINUE
      VAIR=V(TM,WM,PATM)
      WATDIR=(60/2.2)*WATDIR*CFM/VAIR
      VAIR=V(DBT,HUMRAT,PATM)
      WATIND=(60/2.2)*WATIND*CFM/VAIR
      WCOIL=WAIR
      WR=(WCOIL+DW+FINT*HUMRAT+GW)/(1.0+FINT+G)
      GO TO 110
C-- Calculate the fan energy of the evaporative cooling fan
      EVAPWR=(CFM*EXPD)/(8524*EFEVP)
      COOLKW=COOLKW+EVAPWR
      SKW=SKW+EVAPWR
      SKWQC=SKWQC+EVAPWR
C-- Only executed if fans are not on.
100   CONTINUE
      WR=(PASTWR+DW+FINT*HUMRAT+GW)/(1.0+FINT+G)
110   CONTINUE
      PASTWR=WR
      IF ( FON .EQ. 0) GO TO 120
      PRINT 1, IMO, IDAY, IHR, TC, TCOLD, CHANGECFM, WATIND,
     1WATDIR
    1 FORMAT ( ' S- ',3F3.0,' ',2F5.1,' ',F7.3,' ',2F6.1 )
120   CONTINUE
      END
END-FUNCTION .
```

**Nomenclature for SDSFDES**

| Variable | Description | Units |
|---|---|---|
| ACTIVE | Number of desiccant wheels active for the hour | |
| AUTO | Boolean control variable: 1 for automatically choosing the number of active desiccant wheel for the hour, 0 for constant. | |
| AUXKW | Electrical energy sent to DOE-2.1E classified as auxiliary, returned to DOE-2.1E | kW*H |
| BEFF | Gas fired air heater efficiency | |
| CFM | Air flow rate for the hour. Taken from DOE, modified if the supply air temperature changes, returned to DOE-2.1E | $ft^3$/min |

| Variable | Description | Units |
|---|---|---|
| CFMNEW | Air flow rate variable used in convergence loop | $ft^3/min$ |
| CFMOLD | Air flow rate variable used in convergence loop | $ft^3/min$ |
| CFMREGEN | Regeneration air flow rate | $ft^3/min$ |
| CFMWHEEL | Design air flow capacity of the desiccant wheel | $ft^3/min$ |
| CHANGECFM | Change in air flow rate from that taken from SDSF | $ft^3/min$ |
| CONVERGE | Value of convergence criteria | |
| COOLFL | Gas energy used that is categorized as cooling, returned to DOE-2.1E | Btu |
| COOLKW | Electrical energy categorized as cooling, returned to DOE-2.1E | kW*H |
| D | Intermediate variable used to calculate the dehumidification process | |
| DBT | Outdoor dry bulb temperature taken from DOE-2.1E | °F |
| DESON | Boolean variable: 1 for desiccant wheels active 0 for inactive | |
| DESWEIGHT | Fraction of an hour the desiccant wheels are active | |
| DGAS | Gas energy used by desiccant unit, (DOE-2.1E has a variable for this, at the present time it is not known how to take advantage of it) | Btu |
| DW | Increase in the building absolute humidity due to latent loads | $lb_{water}/lb_{air}$ |
| EFEVP | Indirect evaporative cooler static fan efficiency | |
| EFREG | Regeneration fan static efficiency | |
| ENTHALPY | Enthalpy value for the air | |
| EVAPEFF | Effectiveness of the evaporative coolers | |
| EVAPWR | Electrical power consumed by the indirect cooling fan | kW*H |
| EXEFF | Indirect component, heat exchanger effectiveness | |
| EXPD | Pressure drop across the indirect heat exchanger | in-$H_2 0$ |
| F | Infiltration factor taken from DOE-2.1E | |
| FANKW | Total electrical power categorized to the fans, returned to DOE-2.1E | kW*H |
| FINT | Infiltration factor taken from DOE-2.1E | |
| FON | Boolean variable for the fans being active, taken from DOE-2.1E | |
| FV | Face velocity of air on desiccant wheel | ft/min |
| G | Process latent load factor taken from DOE-2.1E | |
| GW | Process latent load factor taken from DOE-2.1E | |
| HDREG | Regeneration air enthalpy value | Btu/lb |

| *Variable* | *Description* | *Units* |
|---|---|---|
| HHI | Enthalpy of the air entering the hot side of the heat recovery unit | |
| HON | Boolean variable for heating on, taken from DOE-2.1E | |
| HTEST | Enthalpy value, used in root solver | Btu/lb |
| HUMRAT | Outdoor absolute humidity value | $lb_{water}/lb_{air}$ |
| IDAY | Day of the simulation, taken from DOE-2.1E | |
| IHR | Hour of the simulation (adjusted for daylight savings time), taken from DOE-2.1E | |
| IMO | Month of the simulation, taken from DOE-2.1E | |
| INDW | Absolute humidity exiting the indirect evaporative cooler | |
| INSTALLW | Number of desiccant wheels installed | |
| ITER | Iteration number, used for debugging if convergence does not occur | |
| ITERCH | Change of value between iterations | |
| K | K factor, used to determine the heat carry over from the regeneration to *dehumidification process* | |
| LOOP | Boolean value: 0 non desiccant loop; 1 desiccant loop | |
| MODE | Mode of operation for the algorithms | |
| PASTWR | Previous hour return air absolute humidity, taken from DOE-2.1E | $lb_{water}/lb_{air}$ |
| PATM | Ambient air pressure, taken from DOE-2.1E | in-$H_20$ |
| PO | Fresh air ratio | |
| QHUM | Energy needed for humidifier, returned to DOE-2.1E | Btu |
| QREG | Regeneration energy used in the air heater, returned to DOE-2.1E (DOE-2.1E has a variable available for this value, however no further use is made of it), returned to DOE-2.1E | Btu |
| QREGP | Sum of regeneration energy, returned to DOE-2.1E same comment as QREG | Btu |
| RCFM | Air flow rate of the return air, modified as per the supply air flow is modified, returned to DOE-2.1E | $ft^3$/min |
| REGPRESS | Pressure drop across the regeneration side of the desiccant wheel | |
| REGPWR | Electrical energy consumed by the regeneration fan | kW*H |
| REXEFF | Heat recovery unit effectiveness | |
| RFKW | Return fan electrical consumption | kW*H |
| RGHCPD | Static pressure for the regeneration fan pressure drop | in-$H_20$ |

| Variable | Description | Units |
|---|---|---|
| SCGAS | Amount of gas consumed for cooling, returned to DOE-2,1E | Btu |
| SFKW | Sum of the electrical energy consumption by the fans, returned to DOE-2.1E | kW*H |
| SKW | Sum of the electrical energy consumption by all equipment at the system level, returned to DOE-2.1E | kW*H |
| SKWQC | Sum of the electrical energy apportioned to cooling at the system level, returned to DOE-2.1E | kW*H |
| SKWQH | Sum of the electrical energy apportioned to cooling at the system level, returned to DOE-2.1E | kW*H |
| T | Temperature value used in root solvers | °F |
| TAIR | Process air temperature | °F |
| TAIRDES | Process air temperature for no desiccant dehumidification used | °F |
| TAIRNODES | Process air temperature for desiccant dehumidification used | °F |
| TC | Supply air temperature calculated by SDSFDES and returned to DOE-2.1E | °F |
| TCOLD | Original supply air temperature taken from DOE-2.1E, this value was the value calculated in DKTEMP | °F |
| TCOUT | Air temperature leaving the cold side of the indirect evaporative cooler heat exchanger | °F |
| TDES | Air temperature after mixing with the dehumidified and non dehumidified air | °F |
| TDIR | Air temperature after the direct evaporative cooler | °F |
| THI | Air temperature entering air heater | °F |
| THIN | Air temperature of the process air entering the indirect heat exchanger | °F |
| TIME | Time step parameter | hour |
| TIND | Air temperature after the direct evaporative cooler | °F |
| TINDC | Air temperature leaving the evaporative cooler of the indirect component | °F |
| TM | Air temperature of the mixed air before any cooling occurs | °F |
| TMAX | Supply temperature control parameter, maximum temperature | °F |
| TMIN | Supply temperature control parameter, minimum temperature | °F |
| TPI | Air temperature entering the desiccant wheel | °F |

| Variable | Description | Units |
|---|---|---|
| TPO | Air temperature leaving individual desiccant wheel and before mixing with non dehumidified air. | °F |
| TPRG | Purge air temperature for the desiccant wheel | °F |
| TR | *Return air temperature* | °F |
| TWET | Outside wet bulb temperature | °F |
| VAIR | Specific volume of the air | $ft^3/lb$ |
| VENTKW | Electrical consumption of the ventilation fans | kW*H |
| W | Absolute humidity value for the root solvers | $lb_{water}/lb_{air}$ |
| WAIR | Process air absolute humidity | $lb_{water}/lb_{air}$ |
| WAIRMIN | Minimum supply air humidity to satisfy the latent load for the humidifier | $lb_{water}/lb_{air}$ |
| WATDIR | Water used by the direct evaporative cooler | liter |
| WATIND | Water used by the indirect evaporative cooler | liter |
| WBT | Outdoor wet bulb temperature | °F |
| WCMAX | Maximum value that the supply air may be and still satisfy the latent cooling load | $lb_{water}/lb_{air}$ |
| WCMIN | | |
| WCOIL | Supply air resulting humidity | $lb_{water}/lb_{air}$ |
| WDES | Absolute humidity leaving the desiccant component, desiccant humidification active | $lb_{water}/lb_{air}$ |
| WDIR | Absolute humidity leaving the direct evaporative component | $lb_{water}/lb_{air}$ |
| WDIRINDES | Absolute humidity before direct evaporative cooler, with desiccant dehumidification | $lb_{water}/lb_{air}$ |
| WDIRINNODES | Absolute humidity before direct evaporative cooler, with no desiccant dehumidification | $lb_{water}/lb_{air}$ |
| WENTER | Saved value for the absolute humidity entering the cooling equipment | $lb_{water}/lb_{air}$ |
| WGRAIN | Absolute humidity in grains/lb for the desiccant process regression equations | $lb_{water}/lb_{air}$ |
| WHEELAREA | Face area of the desiccant wheel's process side | $ft^2$ |
| WHEELS | Number of desiccant wheels needed for the design air flow | |
| WIND | Absolute humidity leaving the indirect evaporative component | $lb_{water}/lb_{air}$ |
| WM | Humidity of the mixed air before any cooling occurs, new value returned to DOE-2.1E | $lb_{water}/lb_{air}$ |

| Variable | Description | Units |
|---|---|---|
| WMNEW | Humidity of the mixed air before any cooling occurs, new value of iteration loop used for convergence | $lb_{water}/lb_{air}$ |
| WNODES | Absolute humidity leaving the desiccant component, desiccant humidification not active | $lb_{water}/lb_{air}$ |
| WPO | Absolute humidity leaving the desiccant wheel before re-mixing with the non dehumidified air stream | $lb_{water}/lb_{air}$ |
| WR | Return air humidity, new value returned to DOE-2.1E | $lb_{water}/lb_{air}$ |
| WRMAX | Maximum return air absolute humidity allowed, taken from DOE-2.1E | $lb_{water}/lb_{air}$ |
| WRMIN | Minimum return air absolute humidity allowed, taken from DOE-2.1E | $lb_{water}/lb_{air}$ |
| WW | Change in humidity through the humidifier | $lb_{water}/lb_{air}$ |

# 10. Sample results

Results are for the cooling season in Ottawa, Ontario for a medium sized office building. The cooling season is defined as May 1 to September 30.

These results are for the following inputs:

- The building has a UA value of 1978 W/°C;
- The weather file is for Ottawa, Ontario, a hot and humid cooling season, WBAN number 04772CWEC file name W04772W.CW2;
- The thermostats are set at 21°C for heating and 23.5°C for cooling;
- The throttling range is set at 1.5°C;
- The humidity settings are max 60% RH and min 35% RH;
- The air handling flow rate at design conditions is: 76,900 $m^3$/hr (45,300 SCFM), 6.0 air changes/hour

The function input variables are:

MODE=1

TMAX=18

TMIN=15

WHEELS=5.333

INSTALLW=4

TIME=0.167

AUTO=1

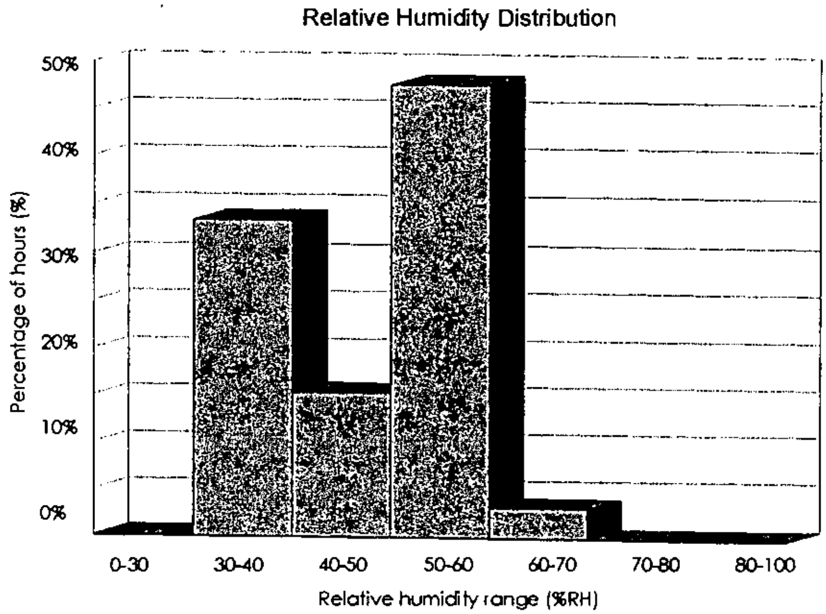
WHEELAREA=12.38

CFMWHEEL=9903

EVAPEFF=0.9

EXEFF=0.8

Relative Humidity Values Taken from DOE-2.1E output form SS-N:




Number of Hours above the Proportional Range of the Thermostat (°C)
or Undercooled Hours:
Taken from DOE-2.1E output form SS-F for each zone

| | May | June | July | Aug | Sept | Total |
|---|---|---|---|---|---|---|
| **GD-N1** | 0 | 1 | 36 | 2 | 0 | **39** |
| **GD-WNW** | 0 | 0 | 21 | 0 | 0 | **21** |
| **GD-W** | 0 | 0 | 19 | 0 | 0 | **19** |
| **GD-SW** | 0 | 0 | 18 | 0 | 0 | **18** |
| **GD-S1** | 0 | 1 | 32 | 2 | 0 | **35** |
| **GD-INT1** | 0 | 1 | 34 | 2 | 0 | **37** |
| **GD-N2** | 0 | 1 | 41 | 1 | 0 | **43** |
| **GD-E** | 0 | 1 | 38 | 3 | 0 | **42** |
| **GD-S2** | 0 | 0 | 28 | 0 | 0 | **28** |
| **GD-INT2** | 0 | 4 | 52 | 15 | 0 | **71** |
| **HALLG** | 0 | 2 | 27 | 2 | 0 | **31** |
| **FOYER** | 0 | 2 | 45 | 9 | 0 | **56** |
| **L2-N1** | 0 | 2 | 40 | 4 | 0 | **46** |
| **L2-WNW** | 0 | 0 | 23 | 0 | 0 | **23** |
| **L2-W** | 0 | 0 | 22 | 0 | 0 | **22** |

| | | | | | | |
|---|---|---|---|---|---|---|
| L2-SW | 0 | 0 | 19 | 0 | 0 | **19** |
| L2-S1 | 0 | 1 | 31 | 3 | 0 | **35** |
| L2-INT1 | 0 | 1 | 32 | 2 | 0 | **35** |
| L2-N2 | 0 | 2 | 43 | 7 | 0 | **52** |
| L2-E | 0 | 2 | 37 | 7 | 0 | **46** |
| L2-S2 | 0 | 0 | 26 | 0 | 0 | **26** |
| L2-INT2 | 0 | 4 | 50 | 17 | 0 | **71** |
| HALL2 | 0 | 2 | 30 | 6 | 0 | **38** |
| L2-BR | 0 | 0 | 21 | 0 | 0 | **21** |
| LUNCH | 0 | 0 | 10 | 0 | 0 | **10** |
| | | | | | | |
| **Min** | **0** | **1** | **31** | **3** | **0** | **35** |
| **Median** | **0** | **1** | **31** | **2** | **0** | **35** |
| **Mean** | **0** | **1** | **31** | **4** | **0** | **35** |
| **Max** | **0** | **4** | **52** | **17** | **0** | **71** |

Energy Usage (MWH) taken from DOE-2.1E output form BEPS:

| | Lights | Misc *Equip* | Space *Heat* | Space *Cool* | Pump *Misc* | Vent *Fan* | DHW | Elec *Tot* |
|---|---|---|---|---|---|---|---|---|
| Electricity | 97.3 | 36.9 | 0.1 | 1.4 | 11.0 | 18.4 | 1.2 | 166.4 |
| Gas | 0.0 | 0.0 | 0.0 | 114.0 | 0.0 | 0.0 | 0.0 | 114.0 |
| Oil | 0.0 | 0.0 | 2.7 | 0.0 | 0.0 | 0.0 | 0.0 | 2.7 |

Energy Usage (utility units) taken from DOE-2.1E output form BEPU:

| | | Lights | Misc Equip | Space Heat | Space Cool | Pump Misc | Vent Fan | DHW | Elec Tot |
|---|---|---|---|---|---|---|---|---|---|
| Electricity | (kWh) | 97307.0 | 36902.0 | 143.0 | 1423.0 | 11037.0 | 18360.0 | 1209.0 | 166380.0 |
| Gas | (Therms) | 0.0 | 0.0 | 0.0 | 3892.0 | 0.0 | 0.0 | 0.0 | 3892.0 |
| Oil | (Gallons) | 0.0 | 0.0 | 65.0 | 0.0 | 0.0 | 0.0 | 0.0 | 65.0 |

Water Usage in Liters (from function output to file columns 6 & 7):

**Indirect** **Direct**

# IEA Annex 28

## Low Energy Cooling

### Subtask 2: Detailed Design Tools

## EVAPORATIVE COOLING
## IN OFFICE BUILDINGS

### MODEL OF DIRECT AND INDIRECT EVAPORATIVE CENTRAL UNIT WITH COOLING AND HEATING COILS

CSTB, Marne La Vallée, France
J.R.Millet

# Contents Chapter C

## 1. Technology area

This tool is a model of a direct and indirect evaporative central unit with cooling and heating coils appropriate for analysing evaporative cooling in office buildings.

## 2. Developed by


**Name** : J.R. MILLET

**Organisation** : CSTB

**Address** : BP 02 - F-77421 Marne-la-Vallée Cedex 02

**Phone** : 33 01 64688300

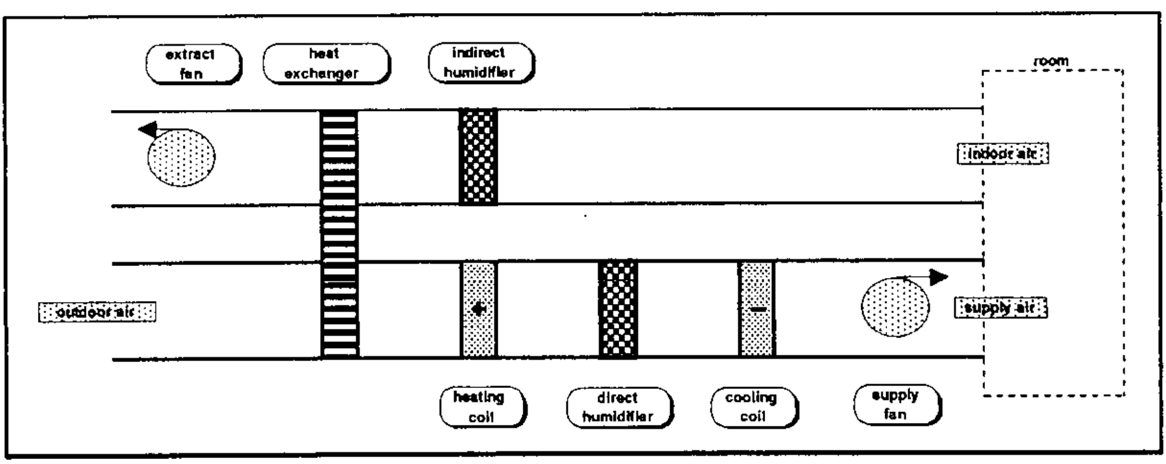
**Fax** : 33 01 64688350

**E-mail** : millet@cstb.fr


## 3. General description

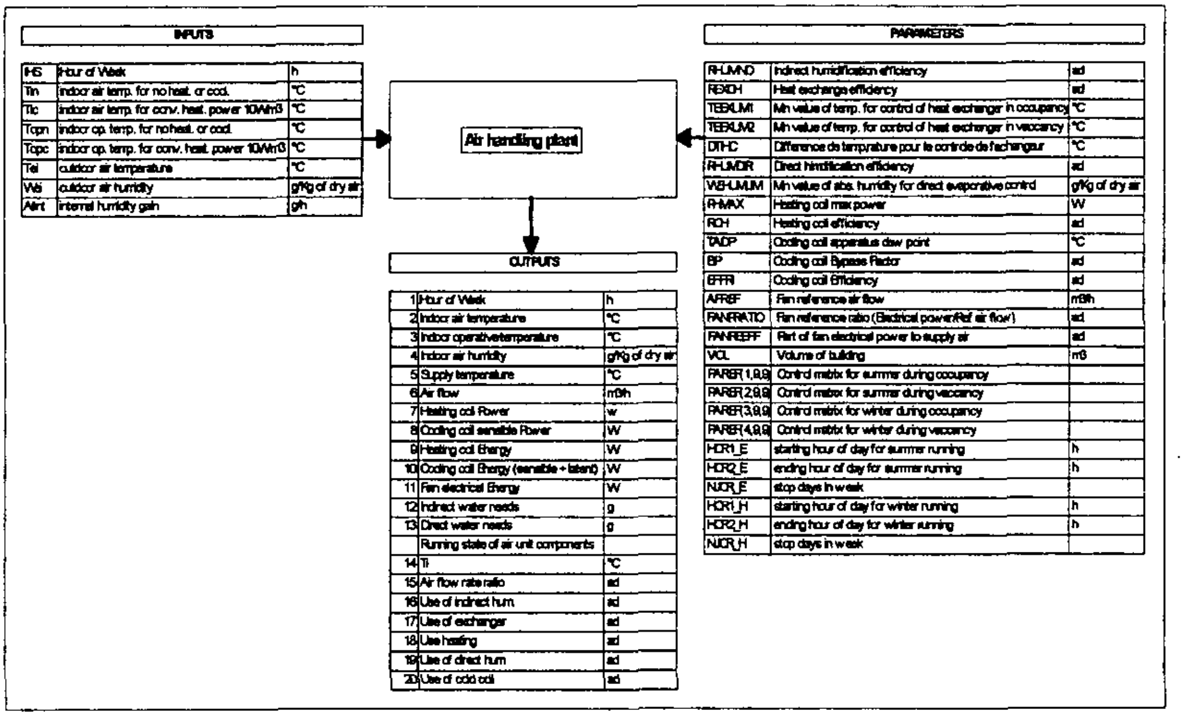
The model provides the state of running of an air handling unit controlled to indoor air temperature. The air handling plant consists of the following :

- a rotating exchanger
- a humidifier for the return air (indirect evaporative cooler)
- a heating coil
- a humidifier for the supply air (direct evaporative cooler)
- a cooling coil
- matched supply and extract multi-speed fans.

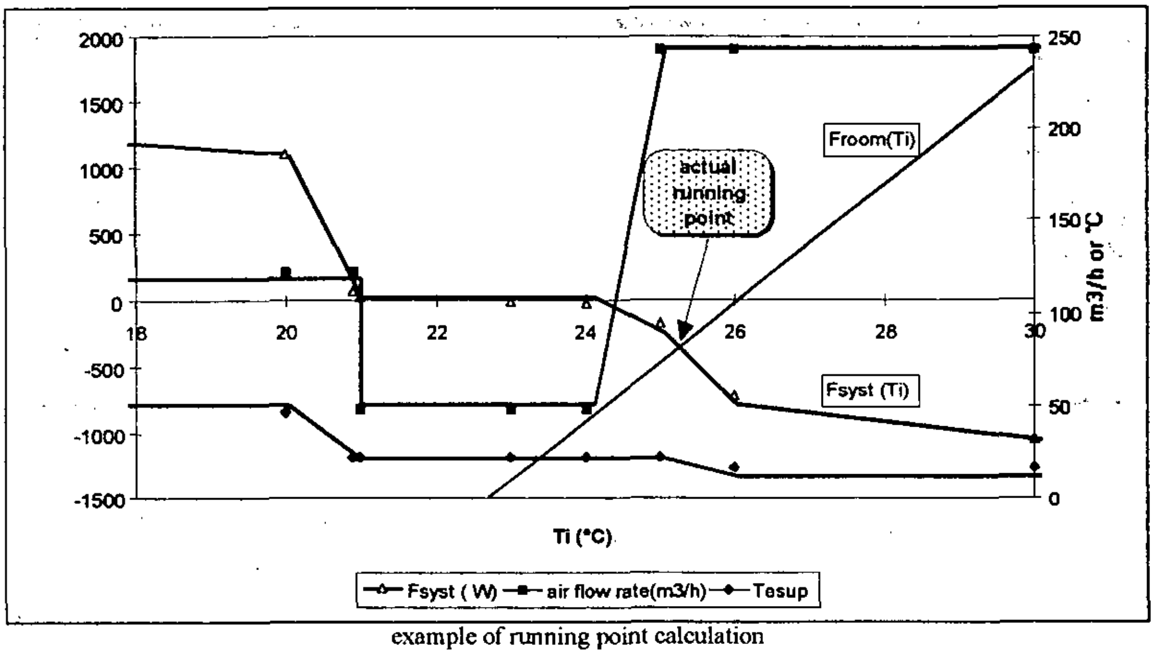


The system control is based on set point temperatures related to the operating modes of the different components of the system. The operation of a component is on/off or modulated within a temperature band (where the control is assumed to be linear) for the calculation timestep. Additional control can be taken into account to ensure correct operation.

The system heating or cooling power is related only to the indoor air temperature Ti by the function Fsyst (Ti). The room behaviour for a given timestep can be described by the function Froom (Ti), which describes the indoor temperature resulting from a given heating (or cooling) power. It is assumed that this relationship is linear. The running point is obtained by solving the two equations (for Fsist(Ti) and Froom(Ti)) at each timestep.




The solution for the running point is illustrated graphically in the following figure.



example of running point calculation

# 4. Nomenclature

| | | |
|---|---|---|
| Af | supply air flow | $m^3/h$ |
| Afref | reference supply air flow (maximum value) | $m^3/h$ |
| Ail | internal humidity gains | g/h |
| BP | cooling coil Bypass factor | ad |
| dTihc | temp diff for heat exchange control | °C |
| fanelp | fans electrical power | W |
| fanpratio | fan ref ratio = electrical power of fans at maximum speed / air flow of supply air | $Wh/m^3$ |
| fanrefaf | fan reference air flow (supply air value at maximum speed) | $m^3/h$ |
| fanrefeff | fan efficiency | ad |
| fanrefelp | fan ref electrical power | W |
| Froom (Ti) | heating or cooling power required by the room to obtain Ti | W |
| Fsys (Ti) | heating or cooling power delivered to the room by the air system versus Ti | W |
| Pheatref | reference power of the heating coil | W |
| Phmax | heating coil max power | W |
| Rexch | heat exchange efficiency | ad |
| Rhumdir | direct humidification efficiency | ad |
| Rhumind | indirect humidification efficiency | ad |
| Tadp | cooling coil dew point | °C |
| Tcool | supply air temperature after cooling coil | °C |
| Te | outdoor air temperature | °C |
| Teex | supply air temperature after heat exchanger | °C |
| Teexlim | min value of temp for control of heat exch. | °C |
| Tehe | supply air temperature after heating coil | °C |
| Tehum | supply air temperature after direct humidification | °C |
| Tesup | supply air to the room temperature | °C |
| Ti | indoor air temp | °C |
| Tic | Ti room for convective heating of 10 $W/m^3$ | °C |
| Tinat | Ti room for no heating or cooling | °C |
| Tiprev | Ti at the previous time step | °C |
| Top | indoor operative temperature | °C |
| Topnat | Top for no heating or cooling | °C |
| wadp | absolute air humidity at saturation for Tadp | g/kg of dry air |
| wcool | supply air humidity after cooling coil | g/kg of dry air |
| we | outdoor air humidity | g/kg of dry air |
| wehum | supply air humidity after direct humidification | g/kg of dry air |
| wehumlim | max value of absolute humidity for direct evap. control | g/kg of dry air |
| wesup | supply air to the room humidity | °C |
| wi | indoor air humidity | g/kg of dry air |
| wiprev | wi at the previous time step | g/kg of dry air |

# 5. Mathematical description

## 5.1. *Calculation of air temperatures and humidities*

### 5.1.1. After humidifier

For the purpose of this study, a FORTRAN function **FADIAB(T,w)** gives the increase of moisture in the air required to reach saturation from the dry bulb temperature T (°C) and the absolute humidity of the air w (g/kg of dry air).

The saturation curve is given by:

wsat=EXP(18.8161-4110.34/(T+235))

The effect of humidification is approximated by lines of 2.5 K/(g/kg).

Knowing the humidification efficiency **Rhumnom** at nominal air flow **Afnom**, the characteristics of humidified air Tihum and whum (full running of humidifier) are, for a given air flow Af:

**Rhum = 1- ( 1- Rhumnom )** $^{(Afnom/Af)^{0.2}}$
**whum = wi + (Rhum.) * FADIAB(T,W)**
**Thum = Ti - 2.5 * (Rhum.) * FADIAB(T,W)**

where:

T, w : dry bulb temperature and humidity of entering air
Thum, whum : dry bulb temperature and humidity of humidified air

The Rhum variation takes into account the fact that the humidification efficiency increases when the air flow is reduced. The formula is based on work undertaken by NLBL [2].

### 5.1.2. After heat exchanger

The heat exchanger is characterised by its efficiency. The supply temperature after heat exchanger is given by:

**Teex = Te (1- Rexch) + Rexch * Ti**

where:

Teex : temperature after heat exchanger
Te : outdoor air temperature
Ti : extract air temperature before heat exchanger

There is no change in air absolute humidity for this process.

### 5.1.3. After heating coil

The heating coil is defined by its reference maximum heating power Pheatref (W). The increase of temperature dT is given by:

**dT = Pheatref / (0.34 * AF)**

where:

AF : air flow in m3/h

There is no change in air absolute humidity for this process.

### 5.1.4. After cooling coil

The cooling coil is characterised by its dew point Tadp and its Bypass factor BP. At full running, the characteristics of leaving air, Tcool and wcool, are calculated by:

**Tcool = min ( Tent ; BP*Tent+(1-BP)*Tadp)**
wadp = exp.(18.8161-4110.34/(Tadp+235)
**wcool = BP*we+(1-BP)*wadp** **for wadp < we**
**wcool = we** **for wadp ≥ we**

where:

Tcool, wcool : characteristics of cooled air

### 5.1.5. After a fan

The increase of air temperature is given by:

**dT_fan = fanelp*fanrefeff/(0.34*Af)**

where:

fanelp : electrical power of the fan
fanrefeff : part of the electrical power heating the air

There is no change in air absolute humidity for this process.

### 5.1.6. Indoor humidity calculation

The average indoor absolute humidity for the timestep is calculated from:

**wiact = [ wiprev + Af * wesup / vol + Ail / (1.2 * vol ) ] / [ 1 + Af / vol ]**

This assumes that there is no hygroscopic buffer effect in the room.

### 5.1.7. Air unit cooling or heating power to the room Fsyst

This is calculated from:

Fsyst = 0.34 * (Tsup - Ti ) * AF

*where:*

| | | |
|---|---|---|
| Tsup | : | supply air temperature |
| Ti | : | indoor air temperature |
| Af | : | air flow in $m^3/h$ |

## 5.2. Required electrical power for fans

The fan is assumed to be controlled in order to deliver a given amount of air which is a ratio of the nominal flow. If the fan efficiency was constant, the required power would be a cubic function of the ratio actual air flow / nominal air flow. *In practice however, this is not the case and the efficiency is reduced at* low air flows. Therefore the following formula has been used:

$$\text{fanelp} = \text{fanrefelp} * \max(0.1; (Af / Afref)^2)$$

where:

fanrefelp : reference electrical power.

## 5.3. System control

### 5.3.1. Controls based on set point temperature

Each component is on/off controlled or modulated according to a temperature control band. It is important to note that the control description must be based on its equivalent behaviour for the calculation timestep.

As a general rule, it is assumed that the behaviour is linear within the control band. For example, if the heating band control is 20 °C - 21 °C, it is assumed that the heating power will be at its maximum value for Ti < 20°C, equal to 0 for Ti > 21 °C and vary linearly between 20 °C and 21 °C. This does not necessarily mean that the control system must be a proportional one - a simple on/off control can lead to the same equivalent behaviour. A second assumption is that sequential (rather than overlapping) control bands are used to control the supply air temperature and air flow (so that cooling/heating power varies linearly).

The set point temperatures can be constant or vary with time (the set point for heating can for example be reduced at night in winter).

**Control matrixes**

For each system, we have defined control matrixes for summer and for winter conditions, during occupancy and inoccupancy (24 control matrixes). These matrixes are shown in Chapter 10.

**Transition**

When performing a calculation for a whole year, it is necessary to define transitions between winter control matrix and summer control matrix. When the calculation is done with winter control matrix, the indoor air temperature between $7^h$ and $8^h$ is checked. If this temperature is higher than 23°C, the transition with summer control matrix is made. When the calculation is done with summer control matrix, the indoor air temperature between $8^h$ and $9^h$ is checked. If this temperature is lower than 19°C, the transition with winter control matrix is made.

### 5.3.2. Additional controls

**Indirect humidification**

Indirect humidification is used if permitted by the set point control and if the humidified air Thum has a temperature lower than the outdoor air Te. The control is as follows:

**Te > Thum + dThic ⇒ control by set point**

$Te \leq Thum + dThic$ $\Rightarrow$ **no humidification**

The dThic value is used to avoid humidification if it is of low efficiency regarding outdoor temperature. It can be for example fixed to 2 K.

**Heat exchange**

When the room requires cooling, the heat exchanger is prevented from operating if outdoor air has a temperature lower than extract air prior to entry to the heat exchanger.
The heat exchanger is controlled to avoid temperatures lower than a limit value **Teexlim** (16°C during occupancy and 11°C during non occupancy). If this limit is reached, direct humidification is prevented from operating.

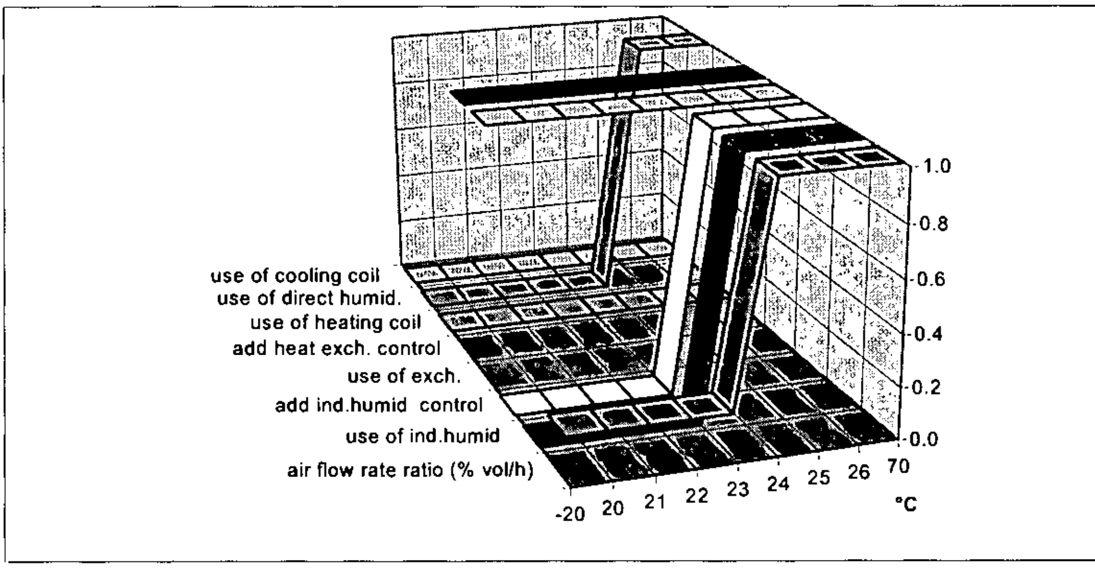
**Direct humidification**

The direct humidification is controlled to avoid air absolute humidities higher than a limit value **wehumlim**.

**Example of control**

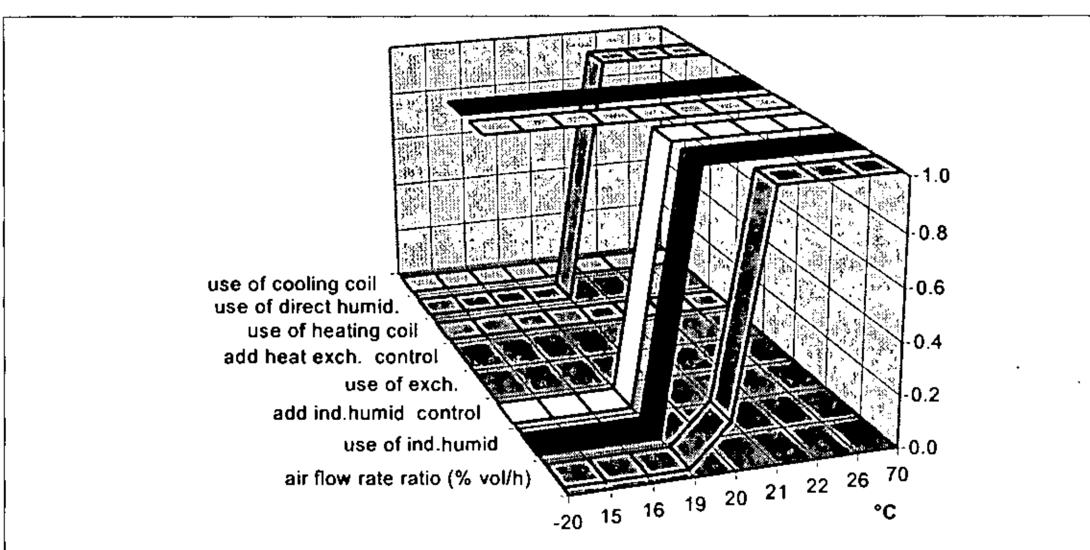
An example of a control scheme for summer and winter operation for an evaporative indirect / direct system with a maximum air flow rate of 6 AC/h is presented overleaf.

| ti-occ | -20 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 70 |
|---|---|---|---|---|---|---|---|---|---|
| air flow rate ratio (% vol/h) | 0.17 | 0.17 | 0.17 | 0.17 | 0.17 | 1.0 | 1.0 | 1.0 | 1.0 |
| use of ind.humid | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 |
| add ind.humid control | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 |
| use of exch. | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| add heat exch. control | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| use of heating coil | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| use of direct humid. | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 |
| use of cooling coil | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |




during occupancy time

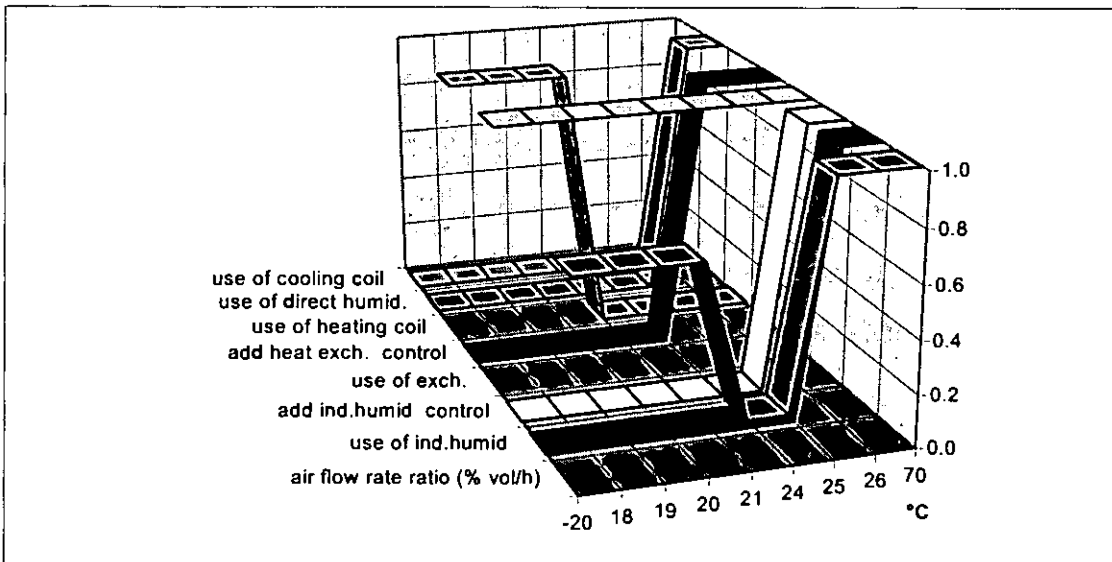
| ti-inoc | -20 | 15 | 16 | 19 | 20 | 21 | 22 | 26 | 70 |
|---|---|---|---|---|---|---|---|---|---|
| air flow rate ratio (% vol/h) | 0.01 | 0.01 | 0.01 | 0.01 | 0.17 | 1.0 | 1.0 | 1.0 | 1.0 |
| use of ind.humid | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 |
| add ind.humid control | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 |
| use of exch. | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| add heat exch. control | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| use of heating coil | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| use of direct humid. | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 |
| use of cooling coil | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |




during non occupancy time

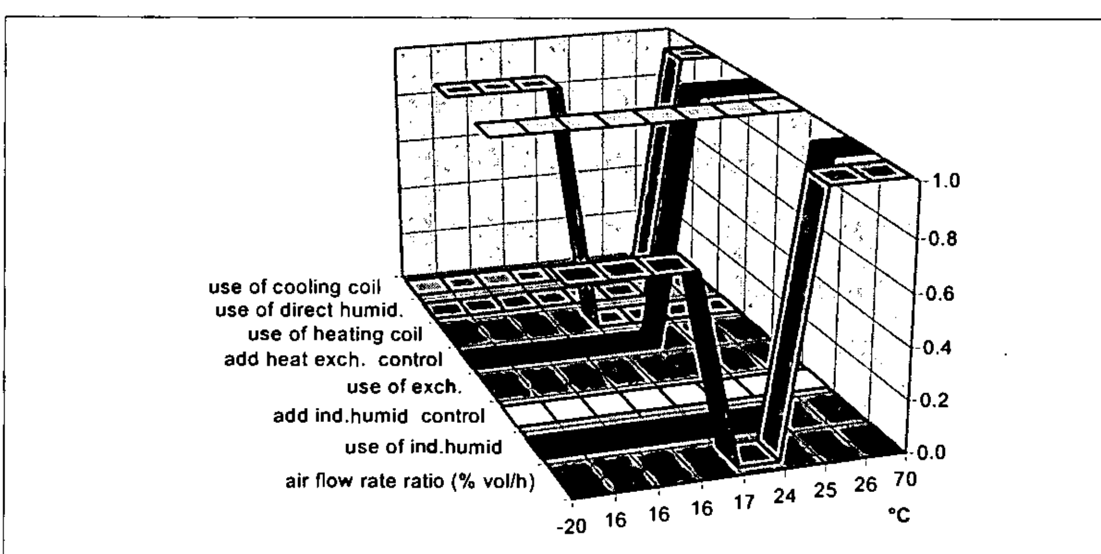
**Example of a control scheme for an evaporative indirect/direct system in summer**

| ti-occ | -20 | 18 | 19 | 20 | 21 | 24 | 25 | 26 | 70 |
|---|---|---|---|---|---|---|---|---|---|
| air flow rate ratio (% vol/h) | 0.75 | 0.75 | 0.75 | 0.75 | 0.17 | 0.17 | 1.0 | 1.0 | 1.0 |
| use of ind.humid | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 |
| add ind.humid control | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 |
| use of exch. | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| add heat exch. control | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 |
| use of heating coil | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 |
| use of direct humid. | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 |
| use of cooling coil | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |




during occupancy time

| ti-inoc | -20 | 16 | 16 | 16 | 17 | 24 | 25 | 26 | 70 |
|---|---|---|---|---|---|---|---|---|---|
| air flow rate ratio (% vol/h) | 0.75 | 0.75 | 0.75 | 0.75 | 0.01 | 0.01 | 1.0 | 1.0 | 1.0 |
| use of ind.humid | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 |
| add ind.humid control | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| use of exch. | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| add heat exch. control | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 |
| use of heating coil | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 |
| use of direct humid. | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 |
| use of cooling coil | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |




during non occupancy time

**Example of a control scheme for an evaporative indirect / direct system in winter**

## 5.4. End calculation

### 5.4.1. Indoor air temperature calculation

The basis of the model is to define for each set point the corresponding supply air temperature and the air flow. Each set point can then be related to a given value of cooling or heating power delivered to the room. Assumptions are made so that this power varies linearly for the supply air temperature and the air flow between two consecutive set points.

For each timestep, the Fsys(Ti) can be calculated. The Froom(Ti) is then calculated using a simplified or detailed specific tool. The assumption made here is that the Froom is linear. It is defined by two points with convective powers of 0 and 10 $W/m^3$ and corresponding temperatures of Tinat and Tic (this choice has no effect on the results).

Considering two consecutive air set points, Tsp1 and Tsp2, and the corresponding values of supply air Tsup and air flow Af. The corresponding Froom values are Froom1 and Froom2. The actual point of running Tiact is within Tsp1 and Tsp2 where:

Froom2 > Fsyst2 and Froom1 < Fsyst1

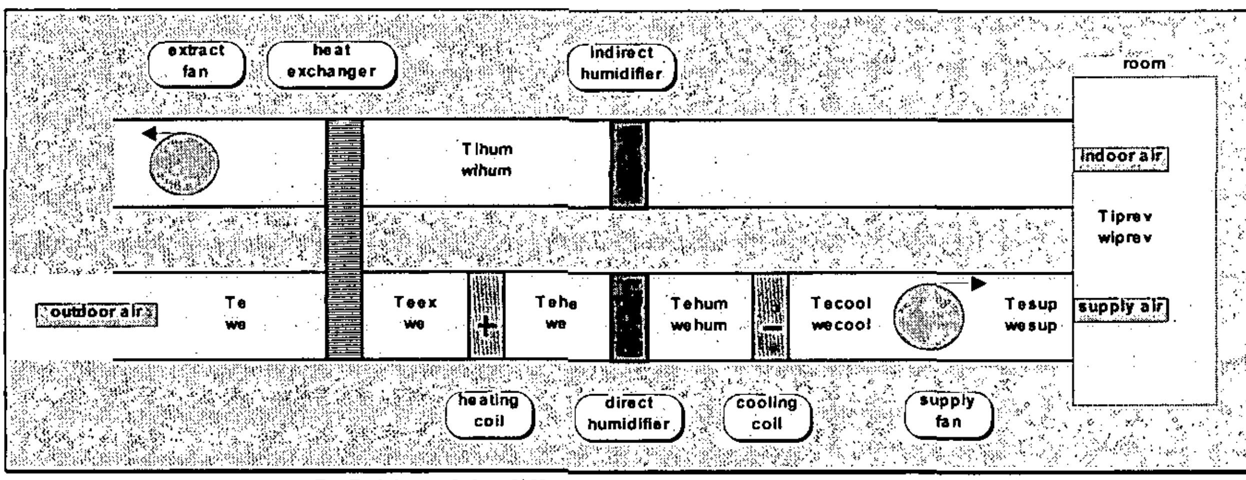
As Froom increases with Ti and Fsyst decreases, it is always possible to identify the Tsp1 Tsp2 band. *The Tiact value can be characterized by using the variable fpt:*

Tiact = fpt*Tsp2 + (1-fpt)*Tsp1

It is then possible to write the set of equations which can be solved:

Froom = fpt*Froom2 +(1-fpt)*Froom1

Fyst = 0.34 * (fpt*Af2 + (1-fpt)*Af1 ) * (fpt*(Tsup2 - Tsp2) + (1-fpt)*(Tsup1-Tsp1))




**Definition of the different temperature and humidities**

## 5.5. Other parameters calculation

Knowing the indoor air temperature, it is possible to go back easily to the state of the system for each parameter as their values are known for each set point value and linear variations between two consecutive set points have been assumed. It should be noted that the assumption of a linear variation is not theoretically required and the approach could be extended to non-linear relationships.

# 6. References

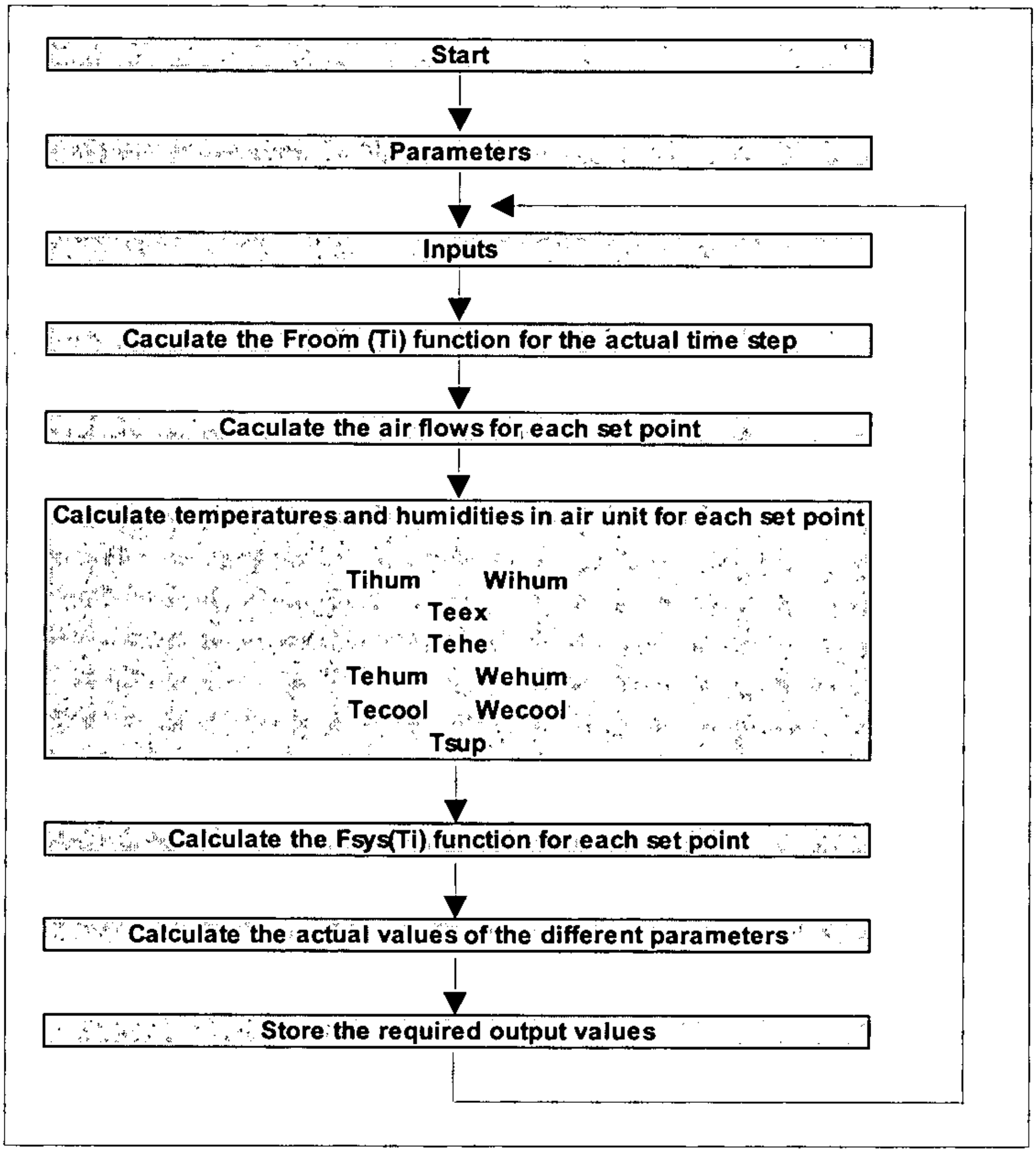
[1] Mathematical equations developed at CSTB on the basis of basic physical laws.

[2] Variation of humidification efficiency is based on the work presented by Joe HUANG (NLBL) in Chapter D of this report.

# 7. Algorithm

1. read the air handling plant parameters
2. read the inputs
3. calculate the air flows for each set point
4. calculate the Fsys (Ti) function for each set point
5. calculate the temperatures and humidities in the unit for each set point
6. calculate the Froom (Ti) function for the actual time step (thermal room model)
7. calculate the actual Ti values by solving Fsys(Ti) = Froom (Ti)
8. calculate the actual values of the different parameters
9. calculate the indoor air humidity
10. store the required output values

# 8. Flowchart



# 9. Source code

```
***********************************************************************
      Program cta
***********************************************************************
c     EVAPORATIVE AIR CENTRAL HANDLING PLANT
***********************************************************************
c NOMENCLATURE
```

| | |
|---|---|
| AF | Supply air flow (m3/h) |
| AFREF | Fan reference air flow maximum value (m3/h) |
| ALINT | internal humidity gain (g/h) |
| BP | Cooling coil Bypass Factor |
| DTIHC | Temperature difference for heat exchange control (°C) |
| EFFRI | Cooling coil Efficiency |
| FANELP | Fan electrical power (W) |
| FANPRATIO | Fan reference ratio (Electrical power/Ref air flow) |
| FANREEFF | Part of fan electrical power to supply air |
| FANREFELP | Fan reference electrical power (W) |
| FPT | Function point |
| FROOM(Ti) | Heataing or cooling power required by room to obtain Ti value |
| FSYS(Ti) | Heataing or cooling power delivred to the room by the air system versus Ti |
| PCOOLING | Cooling coil sensible power (W) |
| PFAN_ELEC | Electrical power of fan (W) |
| PHEATING | Heating coil power (W) |
| PHMAX | Heating coil maxpower (W) |
| PLAT | Plat latent power of the cooling coil (W) |
| RCH | Heating coil efficiency |
| REXCH | Heat exchange efficiency |
| RHUMDIR | Direct himdification efficiency |
| RHUMIND | Indirect humidification efficiency |
| TADP | Cooling coil apparatus dew point (oC) |
| TECOOL | Supply air temperature after cooling coil (°C) |
| TEEX | Supply air temperature after heat exchange (°C) |
| TEEXLIM | Min value of temp. for control of heat exchanger (°C) |
| TEHE | Supply air temperature after heating coil (°C) |
| TEHUM | Supply air temperature after direct humidification (°C) |
| TEI | outdoor air temperature (°C) |
| TIACT | indoor air temperature (°C) |
| TIC | indoor air temperature for Fsys convective Power = 10 W/m3 (°C) |
| TIHUM | Supply air temperature after indirect humidification (°C) |
| TIN | *indoor air temperature for no heating or cooling (°C)* |
| TIPREV | Ti at the previous time step (°C) |
| TOPACT | indoor operative temperature (°C) |
| TOPC | indoor operative temperature for Fsys convective Power = 10 W/m3 (°C) |
| TOPN | indoor operative temperature for no heating or cooling (°C) |
| TSUP | Supply air to the room temperature (°C) |
| VOL | Volume of building (m3) |
| WAT_DIR | Water needs for direct Humidifier (g) |
| WAT_IND | Water needs for indirect Humidifier (g) |
| WECOOL | Supply air humidity after cooling coil |
| WEHUM | Supply air humidity after direct humidification (g/Kg of dry air) |
| WEHUMLIM | Min value of abs. humidity for direct evaporative control (g/Kg of dry air) |
| WEI | outdoor humidity (g/Kg of dry air) |
| WESUP | Supply air humidity to the room (g/Kg of dry air) |
| WI | Indoor air humidity (g/Kg of dry air) |
| WIHUM | Supply air humidity after indirect humidification (g/Kg of dry air) |
| WIPREV | WI at the previous time step (g/Kg of dry air) |

```
c INPUTS

c       IHS     Hour of week
c       ------------------------------------------------
c       TIN     indoor air temperature for Convective Power =0 W/m3
c       TIC     indoor air temperature for Fsys convective Power = 10 W/m3
c       ------------------------------------------------
c       TOPN    indoor operative temperature for Convective Power =0 W/m3
c       TOPC    indoor operative temperature for Fsys convective Power = 10 W/m3
c       ------------------------------------------------
c       TEI     outdoor temperature (°C)
c       WEI     outdoor humidity (g/Kg of dry air)
c       ------------------------------------------------
c       ALINT   internal humidity gain (g/h)
c       ------------------------------------------------
c
c PARAMETERS
c
c       RHUMIND         Indirect humidification efficiency
c       ------------------------------------------------
c       REXCH Heat exchange efficiency
c       TEEXLIM1 Min value of temp. for control of heat exchanger in occupancy (°C)
c       TEEXLIM2 Min value of temp. for control of heat exchanger in vaccancy (°C)
c       DTIHC  Difference de temp,rature pour le controle de l'echangeur (°C)
c       ------------------------------------------------
c       RHUMDIR         Direct himdification efficiency
c       WEHUMLIM Min value of abs. humidity for direct evaporative control (g/Kg of dry air )
c       ------------------------------------------------
c       PHMAX           Heating coil max power (W)
c       RCH     Heating coil efficiency
c       ------------------------------------------------
c       TADP    Cooling coil apparatus dew point (øC)
c       BP      Cooling coil Bypass Factor
c       EFFRI   Cooling coil Efficiency
c       ------------------------------------------------
c       AFREF   Fan reference air flow maximum value (m3/h)
c       FANPRATIO Fan reference ratio (Electrical power/Ref air flow)
c       FANREEFF  Part of fan electrical power to supply air
c       ------------------------------------------------
c       VOL     Volume of building (m3)
c       Free line
c       ------------------------------------------------
c       PAREF(1,9,9) Control matrix for summer during occupancy
c       Free line
c       ------------------------------------------------
c       PAREF(2,9,9) Control matrix for summer during vaccancy
c       Free line
c       ------------------------------------------------
c       PAREF(3,9,9) Control matrix for winter during occupancy
c       Free line
c       ------------------------------------------------
c       PAREF(4,9,9) Control matrix for winter during vaccancy
c       Free line
c       ------------------------------------------------
c       Summer running profile
c       HCR1_E          starting hour of day for summer running
c       HCR2_E          ending hour of day for summer running
c       NJCR_E          stop days in week
c       ------------------------------------------------
c       Winter running profile
c       HCR1_H          starting hour of day for winter running
c       HCR2_H          ending hour of day for winter running
c       NJCR_H          stop days in week
c       ------------------------------------------------
```

```
c
c OUTPUTS
c
c       1-Hour of Week
c       2-Indoor air temperature (øC)
c       3-Indoor operative temperature (øC)
c       4-Indoor air humidity (g/kg of dry air)
c       5-Supply temperature (øC)
c       ------------------------------------------------
c       6-Air flow (m3/h)
c       ------------------------------------------------
c       7-Heating coil Power (W)
c       8-Cooling coil sensible Power (W)
c       ------------------------------------------------
c       9-Heating coil Energy (W)
c       10-Cooling coil Energy (sensible + latent) (W)
c       11-Fan electrical Energy (W)
c       ------------------------------------------------
c       12-Indirect water needs (g)
c       13-Direct water needs (g)
c       ------------------------------------------------
c       14 to 23 Running state of air unit components
c       ------------------------------------------------
c       14-Ti
c       15-Air flow rate ratio
c       16-Use of indirect hum.
c       17-Use of exchangere
c       18-Use heating
c       19-Use of direct hum.
c       20-Use of cold coil
c       ------------------------------------------------

c    DECLARATION OF VARIABLES

      REAL TIN,TIC,TOPN,TOPC,TEI,WEI,TIPREV,WIPREV,ALINT

      INTEGER IHS

      REAL RHUMIND,
     1 REXCH,TEEXLIM1,TEEXLIM2,DTIHC,
     1 RHUMDIR,WEHUMLIM,
     1 PHMAX,RCH,
     1 TADP,BP,EFFRI,
     1 Afref,FANPRATIO,FANREFEFF,
     1 PARREF(4,10,10),
     1 HCR1_E,HCR2_E,NJCR_E,HCR1_H,HCR2_H,NJCR_H,
     1 VOL

      REAL PLOCAL(10),PELEC_FAN(10)

      REAL PAR(24,10),
     1 TSUP,TEEX,TEHE,TEHUM,WEHUM,TECOOL,WECOOL,
     1 TIHUM,WIHUM,AF,
     1 PAIR_EXT,PEXCHANG,PHEATING,PDIRCOOL,PCOOLING,PFAN,PFAN_ELEC,
     1 PEXCH_TIPREV,PINDIRECT,PTOT_EXCHANGE,WAT_IND,WAT_DIR,
     1 PLAT,WAT_IND0,FPT

      INTEGER IFTIK,JN,IRUNSIH,IA,IHJ,IJS

      REAL FSYSACT,TIACT,TOPACT,WIACT,PARACT(9)

      INTEGER IRUNSYS_E(168),IRUNSYS_H(168)

      REAL XOUT(20)
```

```
      CHARACTER*20 XLBL(20)

      open (unit=10,file='CTA.DAT',status='unknown')
      open (unit=11,file='INPUTS.DAT',status='unknown')
      open (unit=12,file='OUTPUTS.DAT',status='unknown')

c**** Declaration of labels for outputs
      Xlbl(1) = 'Hour of week '
      Xlbl(2) = 'Internal temp. (øC) '
      Xlbl(3) = 'Operative temp.(øC) '
      Xlbl(4) = 'Internal hum. (g)'
      Xlbl(5) = 'Supply temp. (øC)   '
      Xlbl(6) = 'Air flow (m3/h)     '
      Xlbl(7) = 'Heat.coil power (W)'
      Xlbl(8) = 'Cool.coil power (W)'
      Xlbl(9) = 'Heat. coil energy(W)'
      Xlbl(10) = 'Cool. coil energy(W)'
      Xlbl(11) = 'Fan elect. energy(W)'
      Xlbl(12) = 'Needs Wat_ind(g) '
      Xlbl(13) = 'Needs Wat_dir(g) '
c     Running state of air unit components
      Xlbl(14)='Ti'
      Xlbl(15)='Air flow rate ratio'
      Xlbl(16)='Use of indirect hum.'
      Xlbl(17)='Use of exchangerc'
      Xlbl(18)='Use heating'
      Xlbl(19)='Use of direct hum.'
      Xlbl(20)='Use of cold coil'

      write (12,2000) (Xlbl(i),char(9),i=1,20)
2000  FORMAT(20(A20,A1))

c     Read titles in inputs file
      Read (11,*)
```

```
c***** Evaporative Air Central Handling Plant

c***** Description of matrix control with additionalcontrols
c     for indirect humidifier and for the exchanger :
c     if use =1 and control=1: runnig under control;
c     if use = 1 and control =0 : forced running ;
c     0   : running not allowed ;
c     jn=1 summer-day matrix ; jn=2 summer-night matrix ;
c     jn=3 winter-day matrix ; jn=4 winter-night matrix;
c     parref(j,1,ik) : Ti
c     parref(j,2,ik) : air flow rate ratio (vol/h)
c     parref(j,3,ik) : use of indirect hum.
c     parref(j,4,ik) : add indirect hum. control
c     parref(j,5,ik) : use of exchanger
c     parref(j,6,ik) : add exchanger control
c     parref(j,7,ik) : use heating
c     parref(j,8,ik) : use of direct hum.
c     parref(j,9,ik) : use of cold coil

c***** Matrix for running under control
c     par(2,ik) : Ti
c     par(3,ik) : air flow rate ratio (vol/h)
c     par(4,ik) : use of indirect hum.
c     par(5,ik) : use of exchangerc
c     par(6,ik) : use heating
c     par(7,ik) : use of direct hum.
c     par(8,ik) : use of cold coil
c     par(9,ik) : libre
c
c***** definition of unit parameters
c     par(10,ik) : flow rate (m3/h)
c     par(11,ik) : Actual temperature after indirect humidifier
c     par(12,ik) : Actual humidity after indirect humidifier
c     par(13,ik) : Temperature after the exchanger
c     par(14,ik) : Temperature after the heating coil
c     par(15,ik) : Actual temperature after direct humidifier
c     par(16,ik) : Actual humidity after direct humidifier
c     par(17,ik) : Temperature after the cooling coil
c     par(18,ik) : Actual humidity after the cooling coil
c     par(19,ik) : Supply Temperature
c     par(20,ik) : Power of the system
c     par(21,ik) : Efficiency of the indirect humidifier (RendHi)
c     par(22,ik) : Efficiency of the direct humidifier (RendHd)
c     Par(23,ik) : Temperature after indirect humidifier (Tihum)
c     Par(24,ik) : Humidity after indirect humidifier (Wihum)

c***** Reading control parameters of cta
      read (10,*) Rhumind
      read (10,*) REXCH,Teexlim1,Teexlim2,DTihc
      read (10,*) Rhumdir,Wehumlim
      read (10,*) PHMAX,RCH
      read (10,*) TADP,BP,EFFRI
      Read (10,*) Afref,Fanpratio,Fanrefeff
      Read (10,*) VOL
      Read (10,*)
      do j=1,4
      do i=1,9
      read (10,*) (PARref(J,I,IK),IK=1,9)
      end do
      read (10,*)
      end do
      read (10,*) HCR1_E,HCR2_E,NJCR_E,HCR1_H,HCR2_H,NJCR_H
```

```
c***** Initialisation
      Tiprev=20
      Wiprev=10

      DO WHILE (.NOT. EOF(11))

c***** Reading inputs
      Read (11,*) IHS,TIN,TIC,TOPN,TOPC,TEI,WEI,ALINT

c***** initialisation of the control matrix

c     Summer profile
      call profile (IHS,HCR1_E,HCR2_E,NJCR_E,IA,IJS,IHJ)
      Irunsys_E(IHS) = IA-1

c     Winter profile
      call profile (IHS,HCR1_H,HCR2_H,NJCR_H,IA,IJS,IHJ)
      Irunsys_H(IHS) = IA-1

c     Summer running
      if ((ihj.eq.9).and.(Tiact.GT.23.)) jn=1

c     Winter running
      if ((ihj.eq.10).and.(Tiact.LE.19.)) jn=3

c***** Choice of the profile
      if (jn.LE.2) then
           Irunsih= Irunsys_E(ihs)
      else
           Irunsih= Irunsys_H(ihs)
      end if

c***** Choice of the day-night matrix
      if ((Irunsih.EQ.1).and.(jn.LE.2)) ij=1
      if ((Irunsih.EQ.0).and.(jn.LE.2)) ij=2
      if ((Irunsih.EQ.1).and.(jn.GE.3)) ij=3
      if ((Irunsih.EQ.0).and.(jn.GE.3)) ij=4

c***** Assignment of the used matrix
      do ik=1,9
         par(2,ik) = parref(ij,1,ik)
         par(3,ik) = parref(ij,2,ik)
         if ((parref(ij,3,ik).EQ.1).and.(parref(ij,4,ik).EQ.1)) then
           par(4,ik) = 0.5
         else if ((parref(ij,3,ik).EQ.1).and.(parref(ij,4,ik).EQ.0)) then
           par(4,ik) = 1
         else
           par(4,ik) = 0
         end if
         if ((parref(ij,5,ik).EQ.1).and.(parref(ij,6,ik).EQ.1)) then
           par(5,ik) = 0.5
         else if ((parref(ij,5,ik).EQ.1).and.(parref(ij,6,ik).EQ.0)) then
           par(5,ik) = 1
         else
           par(5,ik) = 0
         end if
         par(6,ik) = parref(ij,7,ik)
         par(7,ik) = parref(ij,8,ik)
         par(8,ik) = parref(ij,9,ik)
      end do
      jn = ij

c***** Calculation of the reference electrical power of the fan
      Fanrefelp = Afref*fanpratio
```

```
c***** Parameters of the Froom(Ti) function
      Alocal= 10*Vol/(Tic-Tin)
      Blocal= -10*vol*Tin/(Tic-Tin)

c***** Calculation of the air flow for each set point (m3/h, par(10,ik))
      do ik=1,9
            Par(10,ik)=Afref*par(3,ik)
      end do

c***** Calculation of the efficiency for the direct and
c     the indirect humidifiers for each set point
      do ik=1,9
            Par(21,ik)=1-(1-Rhumind)**(Afref/par(10,ik))**0.2
            Par(22,ik)=1-(1-Rhumdir)**(Afref/par(10,ik))**0.2
      end do

c***** Calculation of temperature et humidity
c     after indirect humidifier for each set point (Tihum,Wihum)
      do ik=1,9
            if (par(4,ik).LT.0.5) par(21,ik)=0
            Par(23,ik) = TIprev - 2.5*par(21,ik)*FADIAB(TIprev,WIprev)
            Par(24,ik) = WIprev + par(21,ik)*FADIAB(TIprev,WIprev)
      end do

c***** running of indirect humidifier for each set point
c     0.5 : under control, 1 : forced running, 0 : not allowed
      Do ik=1,9
            If (par(4,ik).EQ.0.5) then
             If (TEI.GT.(Par(23,ik) + DTihc)) then
               Par(4,ik) = 1
             else
               Par(4,ik)=0
             end if
            else
             Par(4,ik) = par(4,ik)
            end if
           end do

c***** running of the exchanger for each set point
c     0.5 : under control, 1 : forced running, 0 : not allowed
           Do IK=1,9
            If (par(5,ik).EQ.0.5) then
             If (TEI.GT.(Par(23,ik) + DTihc)) then
               Par(5,IK) = 1
             else
               Par(5,IK)=0
             end if
            else
             Par(5,IK) = par(5,ik)
            end if
           end do
```

```
c***** Actual temperature (par(11,ik)) and actual humidity (par(12,ik))
c     after indirect humidifier for each set point
      do ik = 1,9
            if (PAR(4,ik).LT.0.5) then
              PAR(11,IK)=TIprev
              PAR(12,IK)=WIprev
            else
              PAR(11,IK)=Par(23,ik)
              PAR(12,IK)=Par(24,ik)
            endif
      end do

c***** Temperature after the exchanger for each set point (par(13,ik))
      do ik =1,9
            Teex_avec = Tei*(1-REXCH) + par(11,ik)*REXCH
            Teex_control = Teex_avec*par(5,ik) + Tei*(1-par(5,ik))

c***** limitation of supply air temperature in occupancy and vaccancy
            if (irunsih.eq.1) then
              Teexlim=Teexlim1
            else
              Teexlim=Teexlim2
            end if

c***** Teex with control and limitation
            if (Teex_control.LT.Teexlim) then
              par(13,ik) = MIN(Teex_avec,Teexlim)
c     When supply air temperature is limited, direct humidifier
c     doesn't run
              par(7,ik) = 0
            else
              par(13,ik) = Teex_control
            end if
      end do

c***** Temperature after the heating coil for each set point (par(14,ik))
      do ik=1,9
            PAR(14,IK) = PAR(13,IK)+PHMAX/0.34/PAR(10,IK)*par(6,ik)
      end do

c***** Running of the direct humidifier or
c     cooling coil
c     Temperature (par(15,ik)) and humidity (par(16,ik))
c     after direct humidifier if allowed for each set point (par(7,ik)=1)
      do ik=1,9
            if (par(7,ik).LT.0.5) par(22,ik)=0.
            Wehum = Wei + Par(22,ik)*FADIAB(par(14,ik),wei)

            if (wehumlim.LT.Wehum) Wehum=wehumlim
            if (wehumlim.LT.Wei) Wehum=wei
            DWehum = (Wehum-Wei)

            if (DWehum.LT.0.001) then
              par(7,ik)=0
            else
              par(7,ik)=par(7,ik)
            end if

            DTehum = (-2.5 * Dwehum)*par(7,ik)
            par(15,ik) = par(14,ik) +DTehum
            par(16,ik) = wei + DWehum*par(7,ik)

      end do
```

```
c***** Cooling coil
      A=18.8161
      B=4110.34
      C=235.
      WADPmin = EXP(A-B/(Tadp+C))

c***** For each set point :
c     Temperature if the cooling coil is running (Tecool1)
c     Temperature with limitation to Tadp
c     Humidity after cooling coil
      do ik=1,9
         Tecool1 = BP*par(15,ik) + (1-BP)*Tadp
         if(Tecool1.GT.Tadp) then
          Tecool2 = Tecool1
         else
          Tecool2 = par(15,ik)
         end if
         IF (Wadpmin.GE.Wei) then
          Wecool2= Wei
         else
          Wecool2= BP*Wei+ (1-BP)*Wadpmin
         end if
      end do

c***** Actual temperature after cooling coil for each set point(par(17,ik))
c     Actual humidity after cooling for each set point(par(18,ik))
      do ik=1,9
         par(17,ik) = par(8,ik)*Tecool2 + (1-par(8,ik))*par(15,ik)
         par(18,ik) = par(8,ik)*Wecool2 + (1-par(8,ik))*par(16,ik)
      end do

c***** DT fan and supply air temperature for each set point (par(19,ik))
      Do ik=1,9
         Pelec_Fan(ik)=MAX((0.1*Fanrefelp),
     1   (Fanrefelp*(par(10,ik)/Afref)**2))
         if (par(3,ik).LT.0.02) Pelec_fan(ik)=0
         DT_Fan = Pelec_Fan(ik) * Fanrefeff/(0.34*par(10,ik))
         par(19,ik) = par(17,ik)+ DT_fan
      end do

c***** Power of the system for each set point
      do ik=1,9
         PAR(20,ik)= 0.34*par(10,ik)*(par(19,ik) - par(2,ik))
      end do

c***** Determination of actual running point Fpt
      do ik =1,9
         Plocal(ik)= Alocal*Par(2,Ik) + blocal
         If (Plocal(ik).GT.Par(20,Ik)) goto 31
      end do

31    continue

      IFtik = ik
```

```
      Z1 = Plocal(ik) - plocal(ik-1)
      Z2 = Plocal(ik-1)
      Z3 = 0.34 * (par(10,ik) - par(10,ik-1))
      Z4 = 0.34 * par(10,ik-1)
      Z5 = (par(19,ik)-par(2,ik))-(par(19,ik-1)-par(2,ik-1))
      Z6 = par(19,ik-1)-par(2,ik-1)

      ZA = Z3 * Z5
      ZB = (Z4 * Z5) + (Z3 * Z6)-Z1
      ZC = Z4*Z6 - Z2
      Fpt=1/(1-(Par(20,ik)-plocal(ik))/(par(20,ik-1)-plocal(ik-1)))
      if (ZA.NE.0) Fpt = (-ZB - (ZB**2 - 4*ZA*ZC)**0.5)/(2*ZA)
      if ((fpt.lt.0).or.(fpt.gt.1))
     1 Fpt = (-ZB + (ZB**2 - 4*ZA*ZC)**0.5)/(2*ZA)

c***** calculation of actual system power and temperatures
c     at running point
      Tiact  = Fpt*par(2,IFtik)+(1-Fpt)*par(2,IFtik-1)
      Topact  = Tiact*(Topc-Topn)/(tic-Tin)+
     1              Topn-Tin*(Topc-Topn)/(Tic-Tin)

      Af  = Fpt*par(10,IFtik)+(1-Fpt)*par(10,IFtik-1)
      Tihum      = Fpt*par(11,IFtik)+(1-Fpt)*par(11,IFtik-1)
      Wihum      = Fpt*par(12,IFtik)+(1-Fpt)*par(12,IFtik-1)
      Teex       = Fpt*par(13,IFtik)+(1-Fpt)*par(13,IFtik-1)
      Tehe       = Fpt*par(14,IFtik)+(1-Fpt)*par(14,IFtik-1)
      Tehum      = Fpt*par(15,IFtik)+(1-Fpt)*par(15,IFtik-1)
      Wehum      = Fpt*par(16,IFtik)+(1-Fpt)*par(16,IFtik-1)
      Tecool     = Fpt*par(17,IFtik)+(1-Fpt)*par(17,IFtik-1)
      Wecool     = Fpt*par(18,IFtik)+(1-Fpt)*par(18,IFtik-1)
      TSup       = Fpt*par(19,IFtik)+(1-Fpt)*par(19,IFtik-1)
      Fsysact    = Af * 0.34 * (Tsup - Tiact)

c***** Power delivered by each component

c***** Outdoot air
      Pair_ext = 0.34 * Af * (Tei-Tiact)

c***** Heat exchanger
      Pexchang = 0.34 * Af * (Teex-Tei)

c***** Heating coil
      Pheating = 0.34 * Af * (Tehe-Teex)

c***** Direct humidifier
      Pdircool = 0.34 * Af * (Tehum-Tehe)

c***** Cooling coil
      Pcooling = 0.34 * Af * (Tecool-Tehum)

c***** Fan heating power
      Pfan = 0.34 * Af * (Tsup-Tecool)

c***** Electrical power of fan

      Pfan_elec= Pfan/fanrefeff

c***** Balance
      Ptot= Pair_ext + Pexchang + Pheating +
     1 Pdircool + Pcooling + Pfan

c***** Extract air
      Pexch_Tiprev = REXCH*0.34*Af*(Tiprev-Tei)
```

```
c***** Indirect humidifier
      Pindirect = REXCH*0.34*Af*(Tihum-Tiprev)

c***** Pexchange balance
      Ptot_exchange = Pexch_Tiprev + Pindirect

c***** Water needs for indirect humidifier(g/h)
      Wat_ind0 = Af * (Wihum-Wiprev)
c***** Needs of water if the exchanger is running
      if (par(5,iFtik).GT.0) then
          Wat_ind = Af * (Wihum-Wiprev)
      else
          Wat_ind = 0
      end if

c***** Needs of water for direct humidifier (g/h)
      Wat_dir = Af * (Wehum-Wei)

c***** Calculation of indoor humidity

c***** doesn't take into account of the deshumidification
c      due to the cooling coil
      wiact = (1/(1+Af/vol))*(wiprev+Af/vol*wecool+Alint/(1.2*Vol))

c***** Latent power of the cooling coil
      Plat=0.81*(Wecool-Wehum)*Af

c***** Running state of air unit components
      do is=1,9
          Paract(is) = Fpt*par(is,IFtik)+(1-Fpt)*par(is,IFtik-1)
      end do

c****  Saving previous values
      TIPREV = Tiact
      WIPREV = Wiact

c***** Output of the module under Xoutputs form ******
      Xout(1) = IHS

c***** Temperatures (øC) and humidity (g)
      Xout(2) = Tiact
      Xout(3) = Topact
      Xout(4) = Wiact
      Xout(5) = Tsup

c***** Air flow (m3/h)
      Xout(6) = Af

c***** Sensible power (W)
      Xout(7) = Pheating
      Xout(8) = - Pcooling

c***** System Energy (sensible + latent)(W)
      Xout(9)     = Pheating/Rch
      Xout(10) = -(Pcooling + Plat)/effri
      Xout(11) = Pfan_elec

c***** Water needs (g)
      Xout(12) = Wat_ind0
      Xout(13) = Wat_dir
```

```
c***** Running state of air unit components
     do is=2,8
        Xout (12+is) = Paract(is)
     end do

c***** Write outputs in outouts file
     write (12,2001) (Xout(i),char(9),i=1,20)
2001    FORMAT(20(F8.2,A1))

     END DO

c***** Closing file
     CLOSE (10)
     CLOSE (11)
     CLOSE (12)

     END

c ***********************************************************
c       profile Generator 1 ou 2 based on 168 h
c       ihs:hour of (1 ... 168) ,ia : index 1 or 2,
c       IJS:day of week, ihj : hour of day

        SUBROUTINE profile (IHS,H1,H2,NJA,IA,IJS,IHJ)
        REAL H1,H2,NJA

        IJS=1+(IHS-1)/24
        IHJ=IHS-24*(IJS-1)
        IA = 2
        IF ((IHJ.LT.(H1+1)).OR.(IHJ.GT.(H2+0.5))) IA=1
        IF (IJS.GT.(7-NJA)) IA=1
        IF (H1.EQ.H2) IA=2
        return
        end

c ***********************************************************
C ***** function FADIAB= possible increase of humidity

     FUNCTION FADIAB(T,W)
     FA=0.0
     A=18.8161
     B=4110.34
     C=235
     FXH=MAX(0,0.2545*T-0.3636*W)
50   FA1=FA
     FA=(T+C-2.5*FXH)*(A-LOG(W+FXH))-B
     IF(FA.LE.0) GO TO 100
     FXH=FXH+0.1
     GO TO 50
100  CONTINUE
     FADIAB=FXH-0.1*FA/(FA-FA1)
     RETURN
     END
```

note : the executable version given in the disquette must be run under DOS directly to obtain the screen displayed results

# 10. Sample results

Detailed results for the following four cases are presented below for a typical warm day in summer (system sizing) and for a reference typical year (location Trappes):

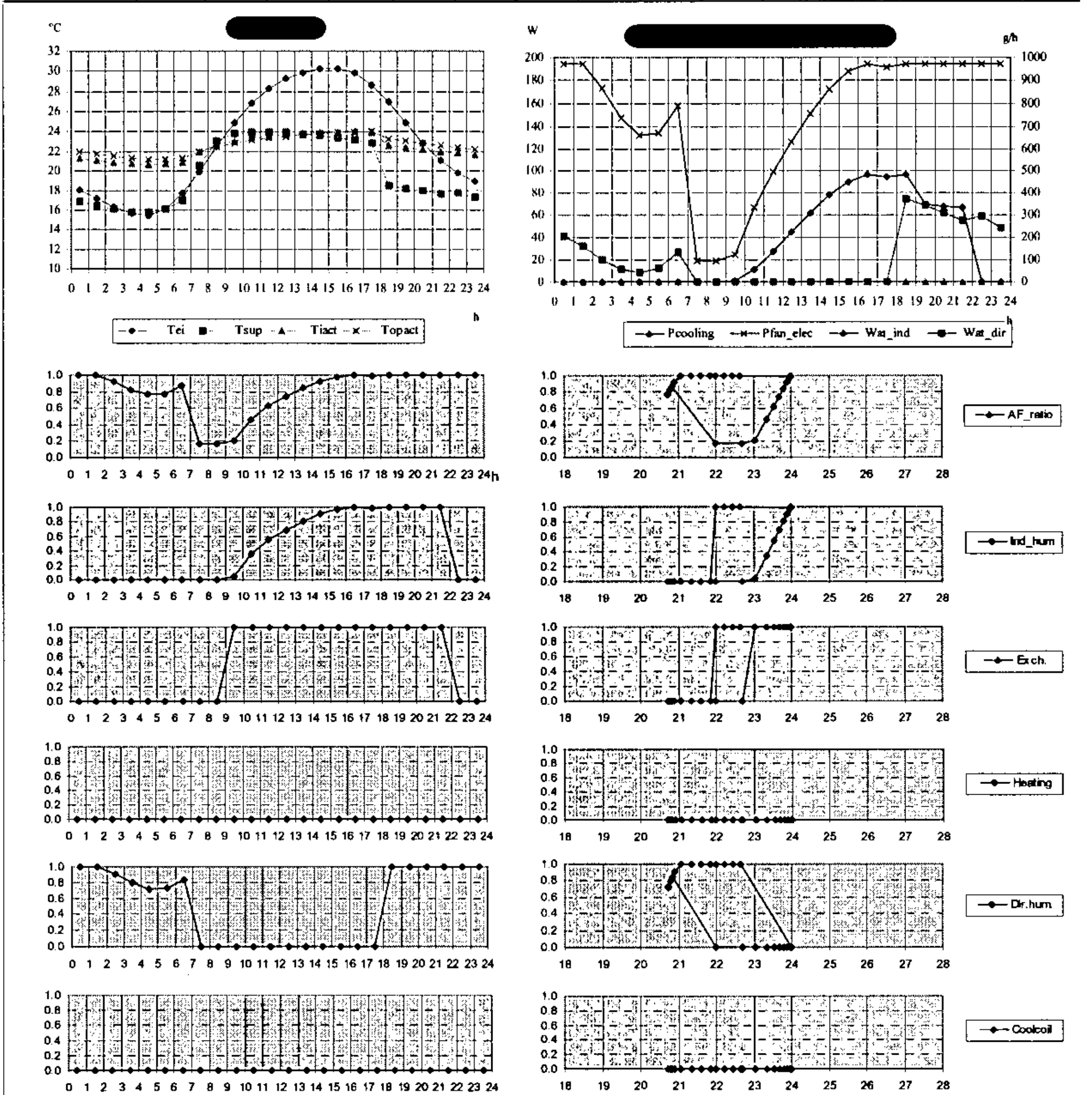
- two systems :direct + indirect evaporative system and indirect evaporative + cooling coil
- two building characteristics :high inertia with East orientation and low inertia with West orientation.

Dept : Trappes | Simulation of Air Handling Plant (dimensioning) | 31-Jan-97

Cooling plant : None | Evaporative system : direct+indirect | Air flow (m3/h) : 243 | N° simul: 177

Int . Gains (W/m2) : 10 | Inertia : high | Orientation : East | Fs : 0.25

| | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Climatic data | | | | Internal gains | | | Pconv: | (W/h) | (gr/h) | Ventilation | | | | (vol/h) | |
| Dept | idmer (km) | alt (m) | Lat | | from (h) | to (h) | 0.5 | sens. | lat | | from (h) | to (h) | | Air flow | |
| 78 | 400 | 0 | 48.8 | occ : | 8 | 18 | vacc | 0 | 0 | occ : | 8 | 18 | vacc | 0 | Teextim |
| | | | | Stop days /week : | | 2 | occ | 150 | 110 | Stop days /week : | | 2 | occ | 0 | occ |
| System | Effnhi | Rexch | EffnhD | Pch | Rch | Tadp | Bp | Effri | Fanrefaf | Fanpratio | Fanrefeff | Dtihc | Wehumlim | | 16 |
| | 0.8 | 0.7 | 0.8 | 2000 | 1 | 12 | 1 | 2 | 243 | 0.8 | 0.4 | 2 | 15 | | vacc |
| Running summer | from | 7 | to (h) | 18 | stop | 2 | Running winter | from | 5 | to (h) | 18 | stop | 2 | | 10 |



| Simulation of Air Handling Plant (consumption) | Annexe 28 | 14-jan-97 |
|---|---|---|

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| Cooling plant : | None | Evaporative system : | direct+indirect | Air flow (m3/h) : | 243 | N° simul: | 177 |
| Int. Gains (W/m2) | 10 | Inertia : high | | Orientation : | East | Fs : 0.25 | |

| Climatic data |
|---|
| Reference year |
| Trappes. |

| Internal gains | from (h) | to (h) | Pconv: | (Wh) | (gr/h) | Ventilation | from (h) | to (h) | | (vol/h) | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | 0.5 | sens. | lat | | | | | Air flow | |
| occ : | 8 | 18 | vacc | 0 | 0 | occ : | 8 | 18 | vacc | 0 | Teexlim |
| Stop days /week : | | 2 | occ | 150 | 110 | Stop days /week : | | 2 | occ | 0 | occ |

| System | Effnhi | Rexch | EffnhD | Pch | Rch | Tadp | Bp | Effri | Fanrefaf | Fanpratio | Fanrefeff | Dthc | Wehumlim | | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | 0.8 | 0.7 | 0.8 | 2000 | 1 | 12 | 1 | 2 | 243 | 0.8 | 0.4 | 2 | 15 | | vacc |
| Running summer | from | 7 | to (h) | 18 | stop | 2 | Running winter | from | 5 | to (h) | 18 | stop | 2 | | 10 |

control matrix occ - summer

| Ti | -20 | 20 | 21 | 22 | 23 | 24 | 25 |
|---|---|---|---|---|---|---|---|
| Act. AF / Max. AF | 0.167 | 0.167 | 0.167 | 0.167 | 0.167 | 1.000 | 1.000 |
| use ind.humid | 0 | 0 | 0 | 0 | 0 | 1 | 1 |
| ind.humid control | 0 | 0 | 0 | 0 | 0 | 1 | 1 |
| use of exch. | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| heat exch.control | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| use of heat. coil | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| use direct humid. | 0 | 0 | 0 | 0 | 0 | 0 | 1 |
| use cooling coil | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

control matrix occ - winter

| Ti | -20 | 18 | 19 | 20 | 21 | 24 | 25 |
|---|---|---|---|---|---|---|---|
| Act. AF / Max. AF | 0.667 | 0.667 | 0.667 | 0.667 | 0.167 | 0.167 | 1.000 |
| use ind.humid | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| ind.humid control | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| use of exch. | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| heat exch.control | 0 | 0 | 0 | 0 | 0 | 0 | 1 |
| use of heat. coil | 1 | 1 | 1 | 1 | 0 | 0 | 0 |
| use direct humid. | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| use cooling coil | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

control matrix vacc - summer

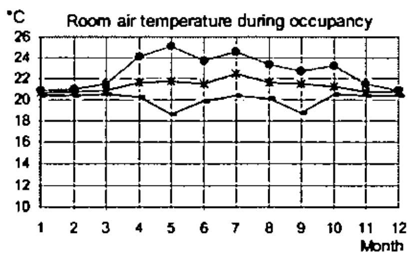
| Ti | -20 | 15 | 16 | 19 | 20 | 21 | 22 |
|---|---|---|---|---|---|---|---|
| Act. AF / Max. AF | 0.010 | 0.010 | 0.010 | 0.010 | 0.167 | 1.000 | 1.000 |
| use ind.humid | 0 | 0 | 0 | 0 | 1 | 1 | 1 |
| ind.humid control | 0 | 0 | 0 | 0 | 1 | 1 | 1 |
| use of exch. | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| heat exch.control | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| use of heat. coil | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| use direct humid. | 0 | 0 | 0 | 0 | 0 | 1 | 1 |
| use cooling coil | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

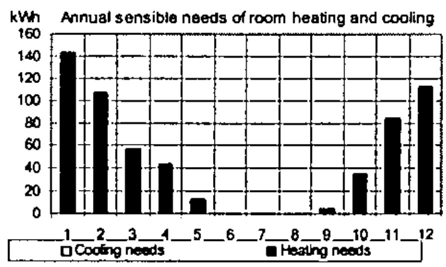
control matrix vacc - winter

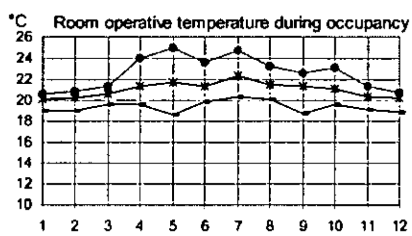
| Ti | -20 | 16 | 16 | 16 | 17 | 24 | 25 |
|---|---|---|---|---|---|---|---|
| Act. AF / Max. AF | 0.667 | 0.667 | 0.667 | 0.667 | 0.010 | 0.010 | 1.000 |
| use ind.humid | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| ind.humid control | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| use of exch. | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| heat exch.control | 0 | 0 | 0 | 0 | 0 | 0 | 1 |
| use of heat. coil | 1 | 1 | 1 | 1 | 0 | 0 | 0 |
| use direct humid. | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| use cooling coil | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

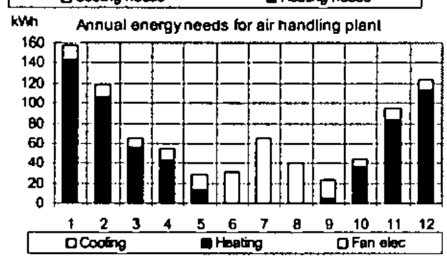
| Building | vol (m3) | Hmass (kg) | Am (m2) | At (m2) | he | hci | hri |
|---|---|---|---|---|---|---|---|
| | 40.5 | 8477 | 57 | 73 | 17 | 4 | 5.5 |

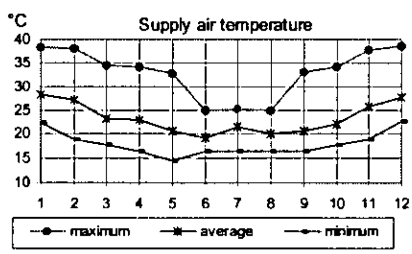
| ITYP | UP | SP | SPSW | SPLV | AP | IOR | INC | IDP | ROL | IBR | IPV | IJN | IPS |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 0.44 | 0.02 | 0 | 0 | 5.1 | 270 | 90 | 0 | 0 | 0 | 0 | 0 | 0 |
| 3 | 2.92 | 0.25 | 0.2 | 0 | 3 | 270 | 90 | 0 | 0 | 0 | 0 | 0 | 0 |

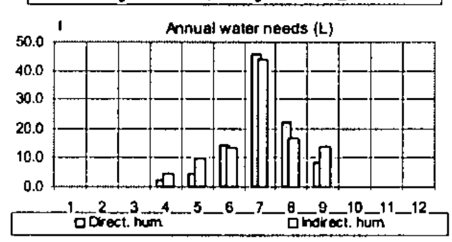



**Condensed results**

| Temperatures | | Max. | Min. | Ave. |
|---|---|---|---|---|
| external | Vac. | 29.4 | -6.5 | 9.2 |
| | occ. | 30.6 | -4.1 | 12.0 |
| supply | Vac. | 42.2 | 10.0 | 17.0 |
| | occ. | 38.5 | 14.6 | 23.2 |
| internal | Vac. | 24.8 | 16.6 | 19.9 |
| | Occ. | 25.1 | 18.6 | 21.3 |
| operative | Vac. | 24.8 | 16.3 | 20.0 |
| | Occ. | 25.0 | 18.6 | 21.1 |

| Sensible needs of room | | |
|---|---|---|
| (kWh) | cool | heat |
| Vac. | 0 | 276 |
| Occ. | 0 | 324 |
| Over all | 0 | 601 |

| Energy needs for air handling plant | | | |
|---|---|---|---|
| (kWh) | Fan elec. | Heat | cool |
| Vac. | 164 | 276 | 0 |
| Occ. | 84 | 324 | 0 |
| Over all | 248 | 601 | 0 |

| Water. needs.(l) | | | |
|---|---|---|---|
| (L) | Total | Ind.H. | Dir.H. |
| Vac. | 172 | 81 | 95 |
| Occ. | 23 | 21 | 2 |
| Over all | 195 | 102 | 98 |

Dept : Trappes

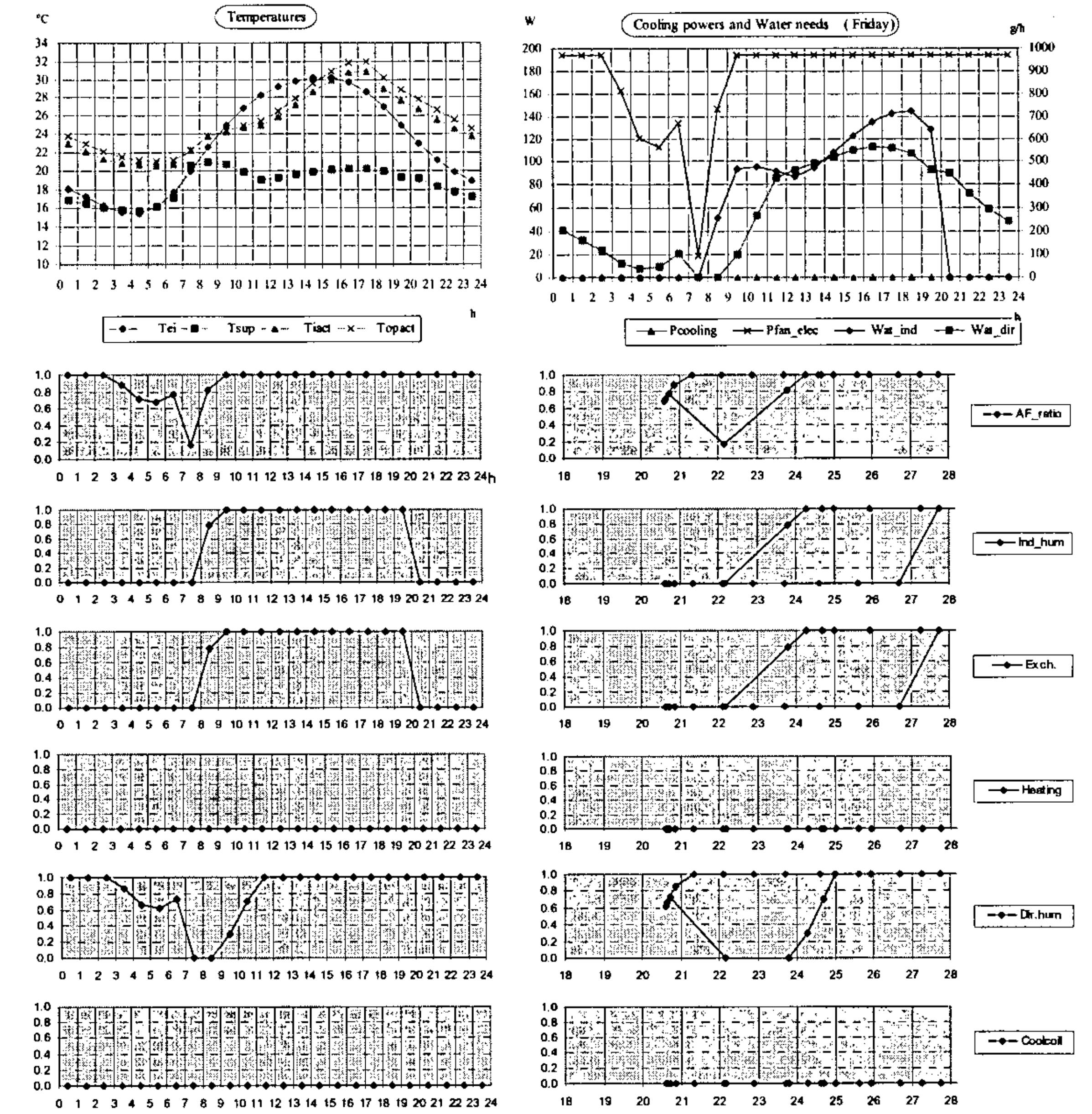
Simulation of Air Handling Plant (dimensioning)

31-Jan-97

Cooling plant : None | Evaporative system : direct+indirect | Air flow (m3/h) : 243 | N° simul: 192

Int . Gains (W/m2) : 30 | Inertia : low | Orientation : West | Fs : 0.75

| Climatic data | | | | Internal gains | | | Pconv: | (W/h) | (gr/h) | Ventilation | | | | (vol/h) | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Dept | idmer (km) | alt (m) | Lat | | from (h) | to (h) | 0.5 | sens. | lat | | from (h) | to (h) | | Air flow | |
| 78 | 400 | 0 | 48.8 | occ : | 8 | 18 | vacc | 0 | 0 | occ : | 8 | 18 | vacc | 0 | Teexlim |
| | | | | Stop days /week : | | 2 | occ | 450 | 110 | Stop days /week : | | 2 | occ | 0 | occ |
| System | Effnhi | Rexch | EffnhD | Pch | Rch | Tadp | Bp | Effri | Fanrefaf | Fanpratio | Fanrefeff | Dtihc | Wehumlim | | 16 |
| | 0.8 | 0.7 | 0.8 | 2000 | 1 | 12 | 1 | 2 | 243 | 0.8 | 0.4 | 2 | 15 | | vacc |
| Running summer | from | 7 | to (h) | 18 | stop | 2 | Running winter | | from | 5 | to (h) | 18 | stop | 2 | 10 |



Simulation of Air Handling Plant (consumption) Annexe 28 14-jan-97

Cooling plant : None | Evaporative system : direct+indirect | Air flow (m3/h) : 243 | N° simul: 192

Int. Gains (W/m 2) 30 | Inertia : low | Orientation : West | Fs : 0.75

| | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Climatic data | | | Internal gains | | | Pconv: | (Wh) | (gr/h) | Ventilation | | | | (vol/h) | |
| Reference year | | | | from (h) | to (h) | 0.5 | sens. | lat | | from (h) | to (h) | | Air flow | |
| Trappes | | | occ : | 8 | 18 | vacc | 0 | 0 | occ : | 8 | 18 | vacc | 0 | Teexdim |
| | | | Stop days /week : | | 2 | occ | 450 | 110 | Stop days /week : | | 2 | occ | 0 | occ |
| System | Effnhi | Rexch | EffnhD | Pch | Rch | Tadp | Bp | Effri | Fanrefaf | Fanpratio | Fanrefeff | Dtihc | Wehumlim | 16 |
| | 0.8 | 0.7 | 0.8 | 2000 | 1 | 12 | 1 | 2 | 243 | 0.8 | 0.4 | 2 | 15 | vacc |
| Running summer | from | 7 | to (h) | 18 | stop | 2 | Running winter | | from | 5 | to (h) | 18 | stop | 2 |

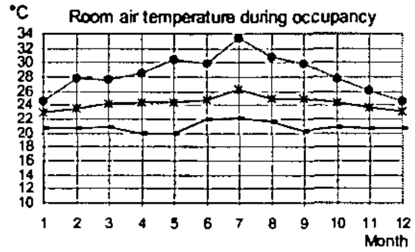
(last cell of the Running row: 10)

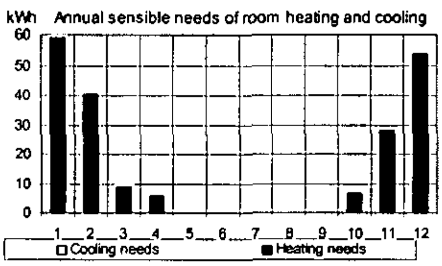
| control matrix occ - summer | | | | | | | | control matrix occ - winter | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Ti | -20 | 20 | 21 | 22 | 23 | 24 | 25 | -20 | 18 | 19 | 20 | 21 | 24 | 25 |
| Act. AF / Max. AF | 0.167 | 0.167 | 0.167 | 0.167 | 0.167 | 1.000 | 1.000 | 0.667 | 0.667 | 0.667 | 0.667 | 0.167 | 0.167 | 1.000 |
| use ind.humid | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| ind.humid control | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| use of exch. | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| heat exch.control | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |
| use of heat. coil | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 0 | 0 | 0 |
| use direct humid. | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| use cooling coil | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| control matrix vacc - summer | | | | | | | | control matrix vacc - winter | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Ti | -20 | 15 | 16 | 19 | 20 | 21 | 22 | -20 | 16 | 16 | 16 | 17 | 24 | 25 |
| Act. AF / Max. AF | 0.010 | 0.010 | 0.010 | 0.010 | 0.167 | 1.000 | 1.000 | 0.667 | 0.667 | 0.667 | 0.667 | 0.010 | 0.010 | 1.000 |
| use ind.humid | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| ind.humid control | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| use of exch. | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| heat exch.control | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |
| use of heat. coil | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 0 | 0 | 0 |
| use direct humid. | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| use cooling coil | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Building | vol (m3) | Hmass (kg) | Am (m2) | At (m2) | he | hci | hri |
|---|---|---|---|---|---|---|---|
| | 40.5 | 2402 | 28 | 73 | 17 | 4 | 5.5 |

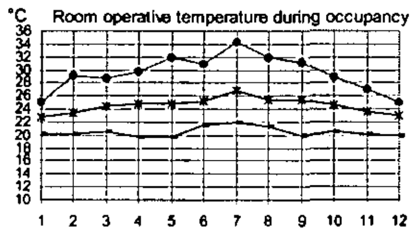
| ITYP | UP | SP | SPSW | SPLV | AP | IOR | INC | IDP | ROL | IBR | IPV | IJN | IPS |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 0.44 | 0.02 | 0 | 0 | 5.1 | 90 | 90 | 0 | 0 | 0 | 0 | 0 | 0 |
| 3 | 2.92 | 0.75 | 0.65 | 0 | 3 | 90 | 90 | 0 | 0 | 0 | 0 | 0 | 0 |

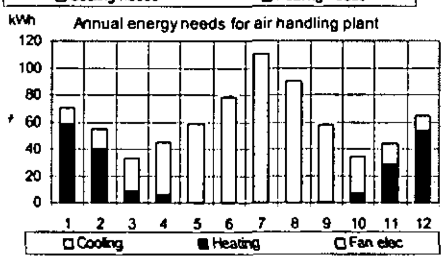



**Condensed results**

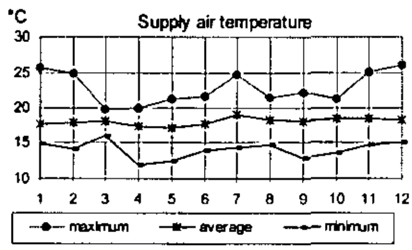
| Temperatures | | Max. | Min. | Ave. |
|---|---|---|---|---|
| external | Vac. | 29.4 | -6.5 | 9.2 |
| | occ. | 30.6 | -4.1 | 12.0 |
| supply | Vac. | 38.1 | 9.0 | 17.1 |
| | occ. | 26.0 | 11.8 | 18.0 |
| internal | Vac. | 30.9 | 16.6 | 21.1 |
| | Occ. | 33.4 | 20.0 | 24.3 |
| operative | Vac. | 32.0 | 16.2 | 21.3 |
| | Occ. | 34.3 | 19.7 | 24.5 |

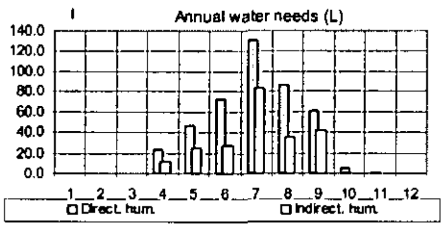



| Sensible needs of room | | |
|---|---|---|
| (kWh) | cool | heat |
| Vac. | 0 | 197 |
| Occ. | 0 | 7 |
| Over all | 0 | 203 |

| Energy needs for air handling plant | | | |
|---|---|---|---|
| (kWh) | Fan elec. | Heat | cool |
| Vac. | 263 | 197 | 0 |
| Occ. | 272 | 7 | 0 |
| Over all | 536 | 203 | 0 |



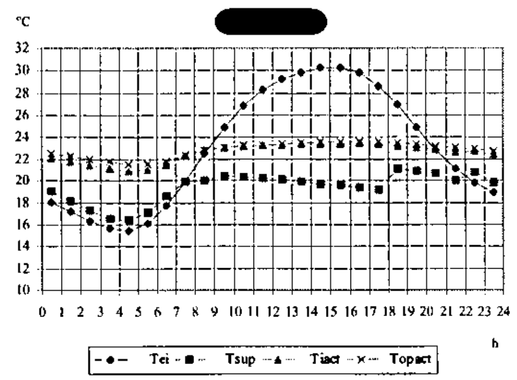

| Water. needs.(l) | | | |
|---|---|---|---|
| (L) | Total | Ind.H. | Dir.H. |
| Vac. | 319 | 95 | 230 |
| Occ. | 326 | 128 | 198 |
| Over all | 645 | 222 | 428 |

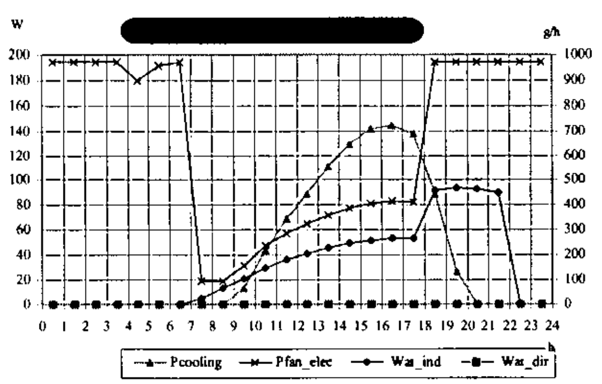
Dept : Trappes | Simulation of Air Handling Plant (dimensioning) | 31-jan-97

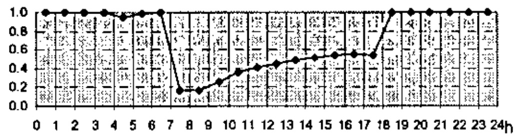
Cooling plant : With | Evaporative system : indirect + cool plant | Air flow (m3/h) : 243 | N° simul: 321

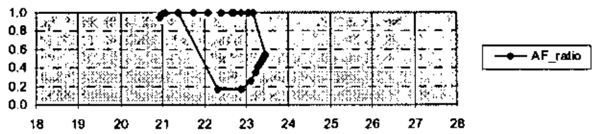
Int . Gains (W/m2) : 10 | Inertia : high | Orientation : East | Fs : 0.25

| | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Climatic data | | | | Internal gains | | | Pconv: | (W/h) | (gr/h) | Ventilation | | | | (vol/h) | |
| Dept | idmer (km) | alt (m) | Lat | | from (h) | to (h) | 0.5 | sens. | lat | | from (h) | to (h) | | Air flow | |
| 78 | 400 | 0 | 48.8 | occ : | 8 | 18 | vacc | 0 | 0 | occ : | 8 | 18 | vacc | 0 | Teexdim |
| | | | | Stop days /week : | | 2 | occ | 150 | 110 | Stop days /week : | | 2 | occ | 0 | occ |
| System | Effnhi | Rexch | EffnhD | Pch | Rch | Tadp | Bp | Effri | Fanrefaf | Fanpratio | Fanrefeff | Dtihc | Wehumlim | ☺ | 16 |
| | 0.8 | 0.7 | 0 | 2000 | 1 | 12 | 0.3 | 2 | 243 | 0.8 | 0.4 | 2 | 15 | | vacc |
| Running summer | from | 7 | to (h) | 18 | stop | 2 | Running winter | | from | 5 | to (h) | 18 | stop | 2 | 10 |

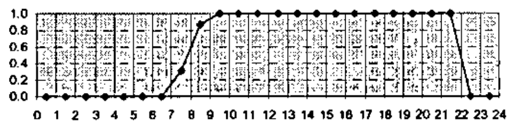


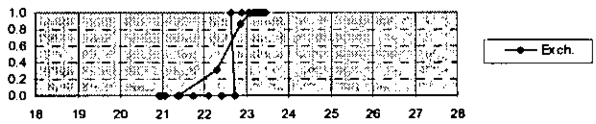



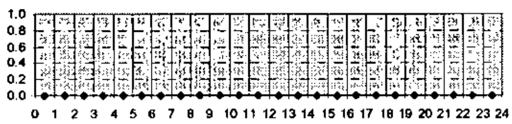



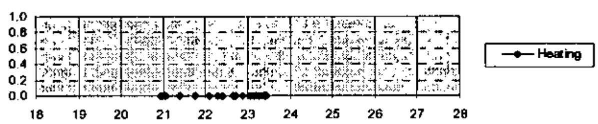



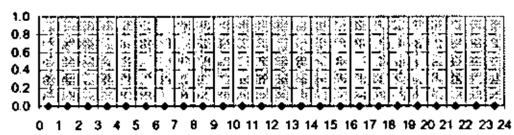


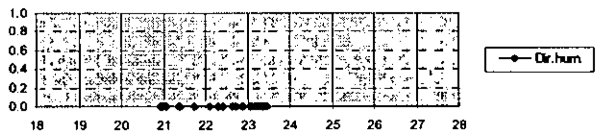



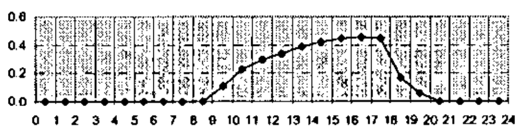



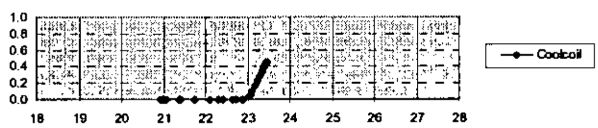



Simulation of Air Handling Plant (consumption) Annexe 28 14-jan-97

Cooling plant : With Evaporative system : indirect + cool plant Air flow (m3/h) : 243 N° simul: 321

Int . Gains (W/m 2) 10 Inertia : high Orientation : East Fs : 0.25

| Climatic data |
|---|
| Reference year |
| Trappes |

| Internal gains | from (h) | to (h) | Pconv: | (W/h) | (gr/h) | Ventilation | from (h) | to (h) | | (vol/h) Air flow | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | 0.5 | sens. | lat | | | | | | |
| occ : | 8 | 18 | vacc | 0 | 0 | occ : | 8 | 18 | vacc | 0 | Teexim |
| Stop days /week : | 2 | | occ | 150 | 110 | Stop days /week : | 2 | | occ | 0 | occ |

| System | Effnhi | Rexch | EffnhD | Pch | Rch | Tadp | Bp | Effri | Fanrefaf | Fanpratio | Fanrefeff | Dtihc | Wehumlim | | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | 0.8 | 0.7 | 0 | 2000 | 1 | 12 | 0.3 | 2 | 243 | 0.8 | 0.4 | 2 | 15 | | vacc |
| Running summer | from | 7 | to (h) | 18 | stop | 2 | Running winter | | from | 5 | to (h) | 18 | stop | 2 | 10 |

control matrix occ - summer

| Ti | -20 | 20 | 21 | 22 | 23 | 24 | 25 |
|---|---|---|---|---|---|---|---|
| Act. AF / Max. AF | 0.167 | 0.167 | 0.167 | 0.167 | 0.167 | 1.000 | 1.000 |
| use ind.humid | 0 | 0 | 0 | 0 | 1 | 1 | 1 |
| ind.humid control | 0 | 0 | 0 | 0 | 1 | 1 | 1 |
| use of exch. | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| heat exch.control | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| use of heat. coil | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| use direct humid. | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| use cooling coil | 0 | 0 | 0 | 0 | 0 | 1 | 1 |

control matrix occ - winter

| | | | | | | |
|---|---|---|---|---|---|---|
| -20 | 18 | 19 | 20 | 21 | 24 | 25 |
| 0.667 | 0.667 | 0.667 | 0.667 | 0.167 | 0.167 | 1.000 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| 0 | 0 | 0 | 0 | 0 | 0 | 1 |
| 1 | 1 | 1 | 1 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 |

control matrix vacc - summer

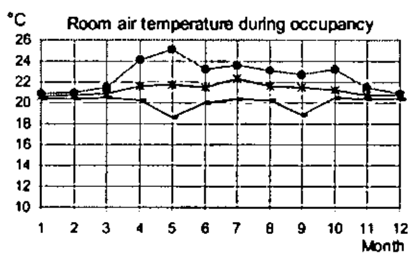
| Ti | -20 | 15 | 16 | 19 | 20 | 21 | 23 |
|---|---|---|---|---|---|---|---|
| Act. AF / Max. AF | 0.010 | 0.010 | 0.010 | 0.010 | 0.167 | 1.000 | 1.000 |
| use ind.humid | 0 | 0 | 0 | 0 | 1 | 1 | 1 |
| ind.humid control | 0 | 0 | 0 | 0 | 1 | 1 | 1 |
| use of exch. | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| heat exch.control | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| use of heat. coil | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| use direct humid. | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| use cooling coil | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

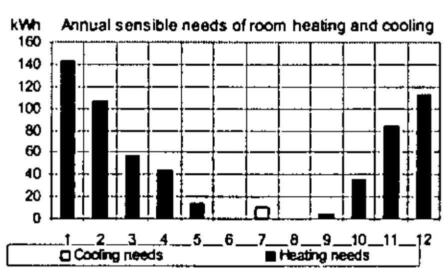
control matrix vacc - winter

| | | | | | | |
|---|---|---|---|---|---|---|
| -20 | 16 | 16 | 16 | 17 | 24 | 25 |
| 0.667 | 0.667 | 0.667 | 0.667 | 0.010 | 0.010 | 1.000 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| 0 | 0 | 0 | 0 | 0 | 0 | 1 |
| 1 | 1 | 1 | 1 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Building | vol (m3) | Hmass (kg) | Am (m2) | At (m2) | he | hci | hri |
|---|---|---|---|---|---|---|---|
| | 40.5 | 8477 | 57 | 73 | 17 | 4 | 5.5 |

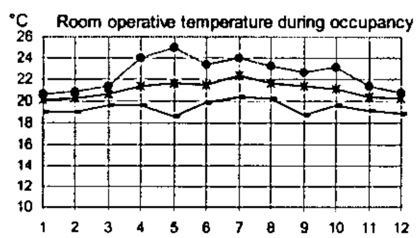
| ITYP | UP | SP | SPSW | SPLV | AP | IOR | INC | IDP | ROL | IBR | IPV | IJN | IPS |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 0.44 | 0.02 | 0 | 0 | 5.1 | 270 | 90 | 0 | 0 | 0 | 0 | 0 | 0 |
| 3 | 2.92 | 0.25 | 0.2 | 0 | 3 | 270 | 90 | 0 | 0 | 0 | 0 | 0 | 0 |

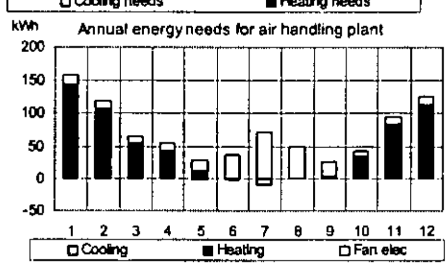



**Condensed results**

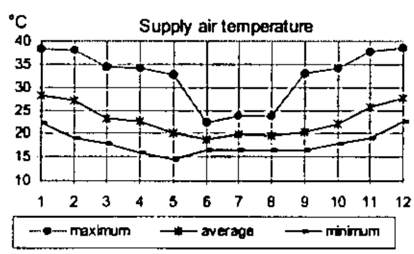
| Temperatures | | Max. | Min. | Ave. |
|---|---|---|---|---|
| external | Vac. | 29.4 | -6.5 | 9.2 |
| | occ. | 30.6 | -4.1 | 12.0 |
| supply | Vac. | 42.2 | 10.0 | 17.2 |
| | occ. | 38.5 | 14.6 | 22.9 |
| internal | Vac. | 24.8 | 16.6 | 20.0 |
| | Occ. | 25.1 | 18.7 | 21.3 |
| operative | Vac. | 24.8 | 16.3 | 20.0 |
| | Occ. | 25.0 | 18.6 | 21.1 |

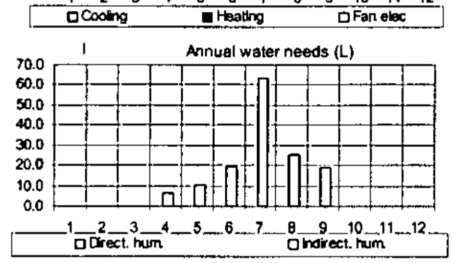



| Sensible needs of room | | |
|---|---|---|
| (kWh) | cool | heat |
| Vac. | 3 | 276 |
| Occ. | 10 | 324 |
| Over all | 14 | 601 |

| Energy needs for air handling plant | | | |
|---|---|---|---|
| (kWh) | Fan elec. | Heat | cool |
| Vac. | 194 | 276 | 3 |
| Occ. | 78 | 324 | 9 |
| Over all | 272 | 601 | 12 |



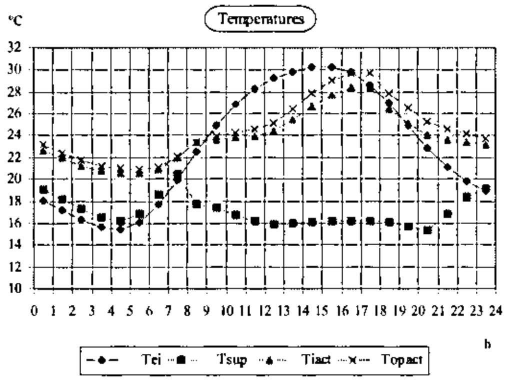

| Water. needs (l) | | | |
|---|---|---|---|
| (L) | Total | Ind.H. | Dir.H. |
| Vac. | 114 | 118 | 0 |
| Occ. | 25 | 26 | 0 |
| Over all | 140 | 144 | 0 |

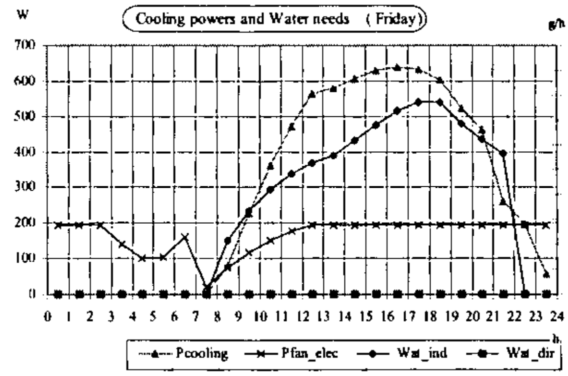
Dept : **Trappes** — **Simulation of Air Handling Plant (dimensioning)** — 31-jan-97

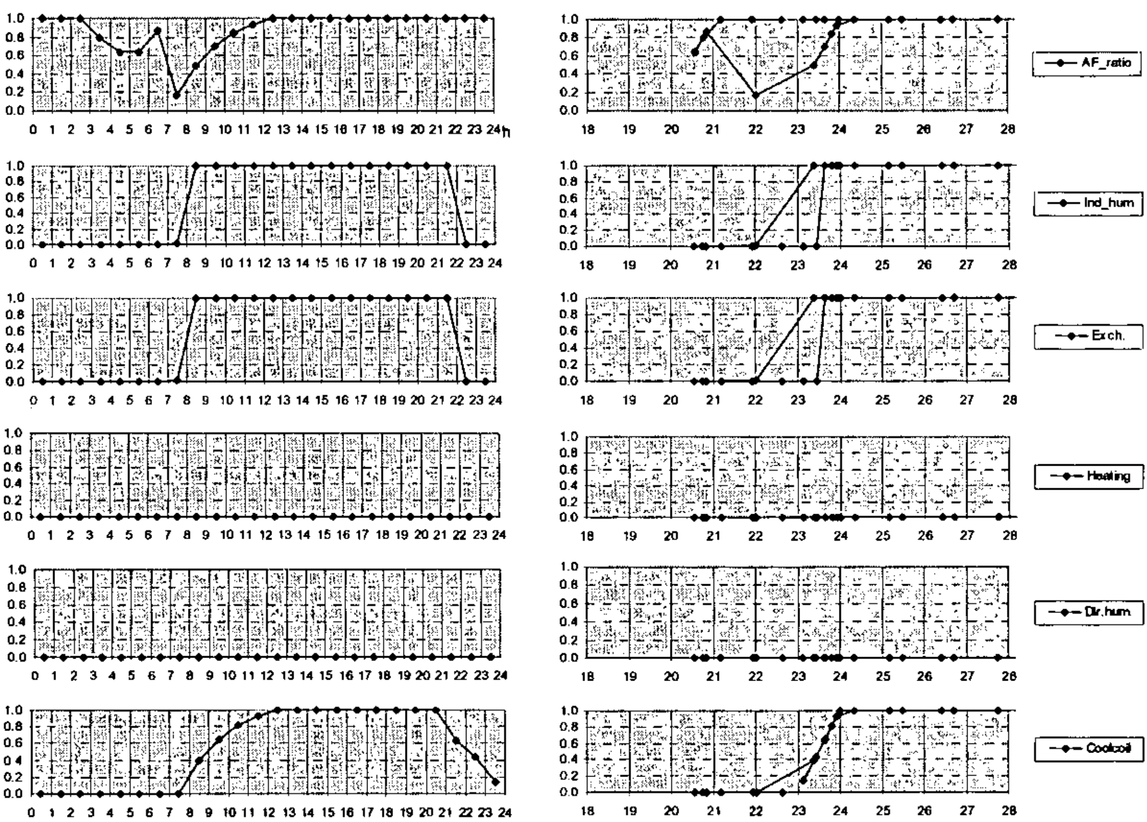
Cooling plant : With — Evaporative system : Indirect + cool plant — Air flow (m3/h) : 243 — N° simul: 336

Int . Gains (W/m2) : 30 — Inertia : low — Orientation : West — Fs : 0.75

| | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Climatic data | | | | Internal gains | | | Pconv: | (W/h) | (gr/h) | Ventilation | | | | (vol/h) | |
| Dept | Idmer (km) | alt (m) | Lat | | from (h) | to (h) | 0.5 | sens. | lat | | from (h) | to (h) | | Air flow | |
| 78 | 400 | 0 | 48.8 | occ : | 8 | 18 | vacc | 0 | 0 | occ : | 8 | 18 | vacc | 0 | Teexdim |
| | | | | Stop days /week : | | 2 | occ | 450 | 110 | Stop days /week : | | 2 | occ | 0 | occ |
| System | Effnhi | Rexch | EffnhD | Pch | Rch | Tadp | Bp | Effri | Fanrefaf | Fanpratio | Fanrefeff | Dthc | Wehumlim | | 16 |
| | 0.8 | 0.7 | 0 | 2000 | 1 | 12 | 0.3 | 2 | 243 | 0.8 | 0.4 | 2 | 15 | | vacc |
| Running summer | from | 7 | to (h) | 18 | stop | 2 | Running winter | from | 5 | to (h) | 18 | stop | 2 | 10 | |



| Simulation of Air Handling Plant (consumption) | Annexe 28 | 14-jan-97 |
|---|---|---|

**Cooling plant:** With — **Evaporative system:** indirect + cool plant — **Air flow (m3/h):** 243 — **N° simul:** 336

Int. Gains (W/m2) 30 — **Inertia :** low — **Orientation :** West — **Fs :** 0.75

| Climatic data |
|---|
| Reference year |
| Trappes |

| Internal gains | from (h) | to (h) | Pconv: 0.5 | (Wh) sens. | (gr/h) lat |
|---|---|---|---|---|---|
| occ : | 8 | 18 | vacc | 0 | 0 |
| Stop days /week : | | 2 | occ | 450 | 110 |

| Ventilation | from (h) | to (h) | | (vol/h) Air flow | Teexdim |
|---|---|---|---|---|---|
| occ : | 8 | 18 | vacc | 0 | occ |
| Stop days /week : | | 2 | occ | 0 | 16 |
| | | | | | vacc |
| | | | | | 10 |

| System | Effnhi | Rexch | EffnhD | Pch | Rch | Tadp | Bp | Effri | Fanrefaf | Fanpratio | Fanrefeff | Dtihc | Wehumlim |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | 0.8 | 0.7 | 0 | 2000 | 1 | 12 | 0.3 | 2 | 243 | 0.8 | 0.4 | 2 | 15 |
| Running summer | from | 7 | to (h) | 18 | stop | 2 | Running winter | from | 5 | to (h) | 18 | stop | 2 |

control matrix occ - summer

| Ti | -20 | 20 | 21 | 22 | 23 | 24 | 25 |
|---|---|---|---|---|---|---|---|
| Act. AF / Max. AF | 0.167 | 0.167 | 0.167 | 0.167 | 0.167 | 1.000 | 1.000 |
| use ind.humid | 0 | 0 | 0 | 0 | 1 | 1 | 1 |
| ind.humid control | 0 | 0 | 0 | 0 | 1 | 1 | 1 |
| use of exch. | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| heat exch.control | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| use of heat. coil | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| use direct humid. | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| use cooling coil | 0 | 0 | 0 | 0 | 0 | 1 | 1 |

control matrix occ - winter

| Ti | -20 | 18 | 19 | 20 | 21 | 24 | 25 |
|---|---|---|---|---|---|---|---|
| Act. AF / Max. AF | 0.667 | 0.667 | 0.667 | 0.667 | 0.167 | 0.167 | 1.000 |
| use ind.humid | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| ind.humid control | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| use of exch. | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| heat exch.control | 0 | 0 | 0 | 0 | 0 | 0 | 1 |
| use of heat. coil | 1 | 1 | 1 | 1 | 0 | 0 | 0 |
| use direct humid. | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| use cooling coil | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

control matrix vacc - summer

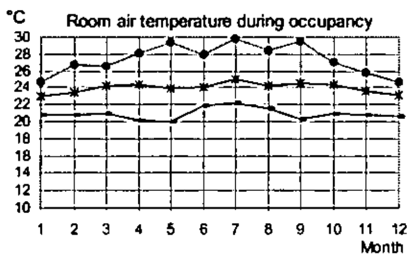
| Ti | -20 | 15 | 16 | 19 | 20 | 21 | 23 |
|---|---|---|---|---|---|---|---|
| Act. AF / Max. AF | 0.010 | 0.010 | 0.010 | 0.010 | 0.167 | 1.000 | 1.000 |
| use ind.humid | 0 | 0 | 0 | 0 | 1 | 1 | 1 |
| ind.humid control | 0 | 0 | 0 | 0 | 1 | 1 | 1 |
| use of exch. | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| heat exch.control | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| use of heat. coil | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| use direct humid. | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| use cooling coil | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

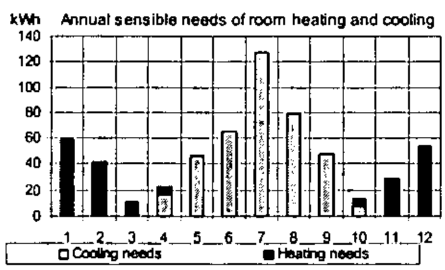
control matrix vacc - winter

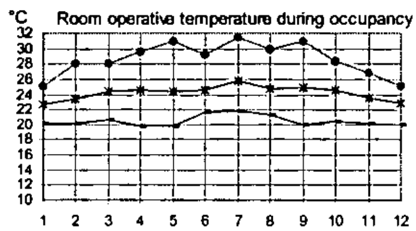
| Ti | -20 | 16 | 16 | 16 | 17 | 24 | 25 |
|---|---|---|---|---|---|---|---|
| Act. AF / Max. AF | 0.667 | 0.667 | 0.667 | 0.667 | 0.010 | 0.010 | 1.000 |
| use ind.humid | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| ind.humid control | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| use of exch. | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| heat exch.control | 0 | 0 | 0 | 0 | 0 | 0 | 1 |
| use of heat. coil | 1 | 1 | 1 | 1 | 0 | 0 | 0 |
| use direct humid. | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| use cooling coil | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

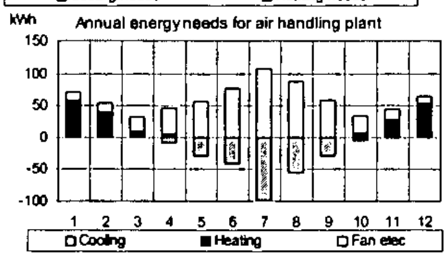
| Building | vol (m3) | Hmass (kg) | Am (m2) | At (m2) | he | hci | hri |
|---|---|---|---|---|---|---|---|
| | 40.5 | 2402 | 28 | 73 | 17 | 4 | 5.5 |

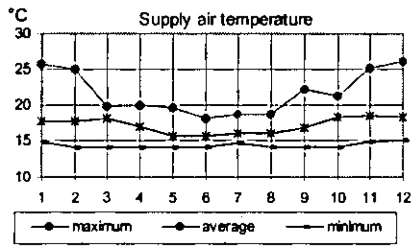
| ITYP | UP | SP | SPSW | SPLV | AP | IOR | INC | IDP | ROL | IBR | IPV | UN | IPS |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 0.44 | 0.02 | 0 | 0 | 5.1 | 90 | 90 | 0 | 0 | 0 | 0 | 0 | 0 |
| 3 | 2.92 | 0.75 | 0.65 | 0 | 3 | 90 | 90 | 0 | 0 | 0 | 0 | 0 | 0 |

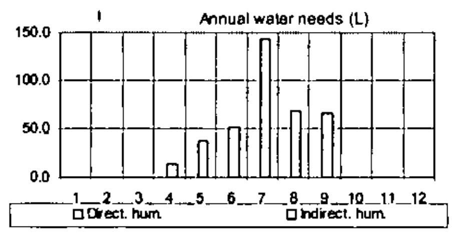



**Condensed results**

| Temperatures | | Max. | Min. | Ave. |
|---|---|---|---|---|
| external | Vac. | 29.4 | -6.5 | 9.2 |
| | occ. | 30.6 | -4.1 | 12.0 |
| supply | Vac. | 38.1 | 10.0 | 17.2 |
| | occ. | 26.0 | 14.1 | 17.1 |
| internal | Vac. | 27.8 | 16.6 | 21.1 |
| | Occ. | 29.8 | 20.0 | 24.0 |
| operative | Vac. | 29.4 | 16.2 | 21.2 |
| | Occ. | 31.5 | 19.7 | 24.2 |

| Sensible needs of room | | |
|---|---|---|
| (kWh) | cool | heat |
| Vac. | 104 | 197 |
| Occ. | 290 | 7 |
| Over all | 394 | 204 |

| Energy needs for air handling plant | | | |
|---|---|---|---|
| (kWh) | Fan elec. | Heat | cool |
| Vac. | 275 | 197 | 75 |
| Occ. | 253 | 7 | 193 |
| Over all | 528 | 204 | 267 |

| Water. needs (l) | | | |
|---|---|---|---|
| (L) | Total | Ind.H. | Dir.H. |
| Vac. | 166 | 168 | 0 |
| Occ. | 206 | 211 | 0 |
| Over all | 372 | 379 | 0 |

## Control Matrixes:

For each system, control matrixes have been defined which were used in all previous calculations, for summer and for winter conditions, during occupancy and inoccupancy (24 control matrixes):

**Control matrix during occupancy for summer**

System : none

| | ti-occ | -20.0 | 20.0 | 21.0 | 22.0 | 23.0 | 24.0 | 25.0 | 26.0 | 70.0 |
|---|---|---|---|---|---|---|---|---|---|---|
| air flow | AF max : 2 vol/h | 0.50 | 0.50 | 0.50 | 0.50 | 0.50 | 1.0 | 1.0 | 1.0 | 1.0 |
| rate ratio | AF max : 4 vol/h | 0.25 | 0.25 | 0.25 | 0.25 | 0.25 | 1.0 | 1.0 | 1.0 | 1.0 |
| actual Air flow / Maximal Air flow | AF max : 6 vol/h | 0.17 | 0.17 | 0.17 | 0.17 | 0.17 | 1.0 | 1.0 | 1.0 | 1.0 |
| | AF max : 8 vol/h | 0.13 | 0.13 | 0.13 | 0.13 | 0.13 | 1.0 | 1.0 | 1.0 | 1.0 |
| | use of ind.humid | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | add ind.humid control | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | use of exch. | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| | add heat exch. control | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| | use of heating coil | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | use of direct humid. | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | use of cooling coil | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

System : evaporative direct

| | ti-occ | -20.0 | 20.0 | 21.0 | 22.0 | 23.0 | 24.0 | 25.0 | 26.0 | 70.0 |
|---|---|---|---|---|---|---|---|---|---|---|
| air flow | AF max : 2 vol/h | 0.50 | 0.50 | 0.50 | 0.50 | 0.50 | 1.0 | 1.0 | 1.0 | 1.0 |
| rate ratio | AF max : 4 vol/h | 0.25 | 0.25 | 0.25 | 0.25 | 0.25 | 1.0 | 1.0 | 1.0 | 1.0 |
| actual Air flow / Maximal Air flow | AF max : 6 vol/h | 0.17 | 0.17 | 0.17 | 0.17 | 0.17 | 1.0 | 1.0 | 1.0 | 1.0 |
| | AF max : 8 vol/h | 0.13 | 0.13 | 0.13 | 0.13 | 0.13 | 1.0 | 1.0 | 1.0 | 1.0 |
| | use of ind.humid | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | add ind.humid control | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | use of exch. | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| | add heat exch. control | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| | use of heating coil | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | use of direct humid. | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 |
| | use of cooling coil | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

System : evaporative indirect

| | ti-occ | -20.0 | 20.0 | 21.0 | 22.0 | 23.0 | 24.0 | 25.0 | 26.0 | 70.0 |
|---|---|---|---|---|---|---|---|---|---|---|
| air flow | AF max : 2 vol/h | 0.50 | 0.50 | 0.50 | 0.50 | 0.50 | 1.0 | 1.0 | 1.0 | 1.0 |
| rate ratio | AF max : 4 vol/h | 0.25 | 0.25 | 0.25 | 0.25 | 0.25 | 1.0 | 1.0 | 1.0 | 1.0 |
| actual Air flow / Maximal Air flow | AF max : 6 vol/h | 0.17 | 0.17 | 0.17 | 0.17 | 0.17 | 1.0 | 1.0 | 1.0 | 1.0 |
| | AF max : 8 vol/h | 0.13 | 0.13 | 0.13 | 0.13 | 0.13 | 1.0 | 1.0 | 1.0 | 1.0 |
| | use of ind.humid | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 |
| | add ind.humid control | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 |
| | use of exch. | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| | add heat exch. control | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| | use of heating coil | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | use of direct humid. | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | use of cooling coil | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

System : evaporative indirect + direct

| | ti-occ | -20.0 | 20.0 | 21.0 | 22.0 | 23.0 | 24.0 | 25.0 | 26.0 | 70.0 |
|---|---|---|---|---|---|---|---|---|---|---|
| air flow | AF max : 2 vol/h | 0.50 | 0.50 | 0.50 | 0.50 | 0.50 | 1.0 | 1.0 | 1.0 | 1.0 |
| rate ratio | AF max : 4 vol/h | 0.25 | 0.25 | 0.25 | 0.25 | 0.25 | 1.0 | 1.0 | 1.0 | 1.0 |
| actual Air flow / Maximal Air flow | AF max : 6 vol/h | 0.17 | 0.17 | 0.17 | 0.17 | 0.17 | 1.0 | 1.0 | 1.0 | 1.0 |
| | AF max : 8 vol/h | 0.13 | 0.13 | 0.13 | 0.13 | 0.13 | 1.0 | 1.0 | 1.0 | 1.0 |
| | use of ind.humid | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 |
| | add ind.humid control | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 |
| | use of exch. | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| | add heat exch. control | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| | use of heating coil | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | use of direct humid. | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 |
| | use of cooling coil | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

System : evaporative indirect + cooling coil

| | ti-occ | -20.0 | 20.0 | 21.0 | 22.0 | 23.0 | 24.0 | 25.0 | 26.0 | 70.0 |
|---|---|---|---|---|---|---|---|---|---|---|
| air flow | AF max : 2 vol/h | 0.50 | 0.50 | 0.50 | 0.50 | 0.50 | 1.0 | 1.0 | 1.0 | 1.0 |
| rate ratio | AF max : 4 vol/h | 0.25 | 0.25 | 0.25 | 0.25 | 0.25 | 1.0 | 1.0 | 1.0 | 1.0 |
| actual Air flow / Maximal Air flow | AF max : 6 vol/h | 0.17 | 0.17 | 0.17 | 0.17 | 0.17 | 1.0 | 1.0 | 1.0 | 1.0 |
| | AF max : 8 vol/h | 0.13 | 0.13 | 0.13 | 0.13 | 0.13 | 1.0 | 1.0 | 1.0 | 1.0 |
| | use of ind.humid | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 |
| | add ind.humid control | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 |
| | use of exch. | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| | add heat exch. control | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| | use of heating coil | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | use of direct humid. | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | use of cooling coil | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 |

System : cooling coil only

| | ti-occ | -20.0 | 20.0 | 21.0 | 22.0 | 23.0 | 24.0 | 25.0 | 26.0 | 70.0 |
|---|---|---|---|---|---|---|---|---|---|---|
| air flow | AF max : 2 vol/h | 0.50 | 0.50 | 0.50 | 0.50 | 0.50 | 1.0 | 1.0 | 1.0 | 1.0 |
| rate ratio | AF max : 4 vol/h | 0.25 | 0.25 | 0.25 | 0.25 | 0.25 | 1.0 | 1.0 | 1.0 | 1.0 |
| actual Air flow / Maximal Air flow | AF max : 6 vol/h | 0.17 | 0.17 | 0.17 | 0.17 | 0.17 | 1.0 | 1.0 | 1.0 | 1.0 |
| | AF max : 8 vol/h | 0.13 | 0.13 | 0.13 | 0.13 | 0.13 | 1.0 | 1.0 | 1.0 | 1.0 |
| | use of ind.humid | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | add ind.humid control | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | use of exch. | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| | add heat exch. control | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| | use of heating coil | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | use of direct humid. | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | use of cooling coil | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 |

**Control matrix during non occupancy for summer**

System : none

air flow
rate ratio
Actual Air flow / Maximal Air flow

| ti-inoc | -20.0 | 15.0 | 16.0 | 19.0 | 20.0 | 21.0 | 22.0 | 26.0 | 70.0 |
|---|---|---|---|---|---|---|---|---|---|
| AF max : 2 vol/h | 0.01 | 0.01 | 0.01 | 0.01 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 |
| AF max : 4 vol/h | 0.01 | 0.01 | 0.01 | 0.01 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 |
| AF max : 6 vol/h | 0.01 | 0.01 | 0.01 | 0.01 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 |
| AF max : 8 vol/h | 0.01 | 0.01 | 0.01 | 0.01 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 |
| use of ind.humid | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| add ind.humid control | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| use of exch. | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| add heat exch. control | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| use of heating coil | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| use of direct humid. | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| use of cooling coil | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

System : evaporative direct

air flow
rate ratio
Actual Air flow / Maximal Air flow

| ti-inoc | -20.0 | 15.0 | 16.0 | 19.0 | 20.0 | 21.0 | 22.0 | 26.0 | 70.0 |
|---|---|---|---|---|---|---|---|---|---|
| AF max : 2 vol/h | 0.01 | 0.01 | 0.01 | 0.01 | 0.50 | 1.0 | 1.0 | 1.0 | 1.0 |
| AF max : 4 vol/h | 0.01 | 0.01 | 0.01 | 0.01 | 0.25 | 1.0 | 1.0 | 1.0 | 1.0 |
| AF max : 6 vol/h | 0.01 | 0.01 | 0.01 | 0.01 | 0.17 | 1.0 | 1.0 | 1.0 | 1.0 |
| AF max : 8 vol/h | 0.01 | 0.01 | 0.01 | 0.01 | 0.13 | 1.0 | 1.0 | 1.0 | 1.0 |
| use of ind.humid | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| add ind.humid control | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| use of exch. | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| add heat exch. control | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| use of heating coil | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| use of direct humid. | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 |
| use of cooling coil | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

System : evaporative indirect

air flow
rate ratio
Actual Air flow / Maximal Air flow

| ti-inoc | -20.0 | 15.0 | 16.0 | 19.0 | 20.0 | 21.0 | 22.0 | 26.0 | 70.0 |
|---|---|---|---|---|---|---|---|---|---|
| AF max : 2 vol/h | 0.01 | 0.01 | 0.01 | 0.01 | 0.50 | 1.0 | 1.0 | 1.0 | 1.0 |
| AF max : 4 vol/h | 0.01 | 0.01 | 0.01 | 0.01 | 0.25 | 1.0 | 1.0 | 1.0 | 1.0 |
| AF max : 6 vol/h | 0.01 | 0.01 | 0.01 | 0.01 | 0.17 | 1.0 | 1.0 | 1.0 | 1.0 |
| AF max : 8 vol/h | 0.01 | 0.01 | 0.01 | 0.01 | 0.13 | 1.0 | 1.0 | 1.0 | 1.0 |
| use of ind.humid | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 |
| add ind.humid control | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 |
| use of exch. | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| add heat exch. control | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| use of heating coil | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| use of direct humid. | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| use of cooling coil | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

System : evaporative indirect + direct

air flow
rate ratio
Actual Air flow / Maximal Air flow

| ti-inoc | -20.0 | 15.0 | 16.0 | 19.0 | 20.0 | 21.0 | 22.0 | 26.0 | 70.0 |
|---|---|---|---|---|---|---|---|---|---|
| AF max : 2 vol/h | 0.01 | 0.01 | 0.01 | 0.01 | 0.50 | 1.0 | 1.0 | 1.0 | 1.0 |
| AF max : 4 vol/h | 0.01 | 0.01 | 0.01 | 0.01 | 0.25 | 1.0 | 1.0 | 1.0 | 1.0 |
| AF max : 6 vol/h | 0.01 | 0.01 | 0.01 | 0.01 | 0.17 | 1.0 | 1.0 | 1.0 | 1.0 |
| AF max : 8 vol/h | 0.01 | 0.01 | 0.01 | 0.01 | 0.13 | 1.0 | 1.0 | 1.0 | 1.0 |
| use of ind.humid | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 |
| add ind.humid control | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 |
| use of exch. | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| add heat exch. control | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| use of heating coil | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| use of direct humid. | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 |
| use of cooling coil | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

System : evaporative indirect + cooling coil

air flow
rate ratio
Actual Air flow / Maximal Air flow

| ti-inoc | -20.0 | 15.0 | 16.0 | 19.0 | 20.0 | 21.0 | 23.0 | 24.0 | 70.0 |
|---|---|---|---|---|---|---|---|---|---|
| AF max : 2 vol/h | 0.01 | 0.01 | 0.01 | 0.01 | 0.50 | 1.0 | 1.0 | 1.0 | 1.0 |
| AF max : 4 vol/h | 0.01 | 0.01 | 0.01 | 0.01 | 0.25 | 1.0 | 1.0 | 1.0 | 1.0 |
| AF max : 6 vol/h | 0.01 | 0.01 | 0.01 | 0.01 | 0.17 | 1.0 | 1.0 | 1.0 | 1.0 |
| AF max : 8 vol/h | 0.01 | 0.01 | 0.01 | 0.01 | 0.13 | 1.0 | 1.0 | 1.0 | 1.0 |
| use of ind.humid | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 |
| add ind.humid control | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 |
| use of exch. | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| add heat exch. control | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| use of heating coil | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| use of direct humid. | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| use of cooling coil | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 |

System : cooling coil only

air flow
rate ratio
Actual Air flow / Maximal Air flow

| ti-inoc | -20.0 | 15.0 | 16.0 | 19.0 | 20.0 | 21.0 | 23.0 | 24.0 | 70.0 |
|---|---|---|---|---|---|---|---|---|---|
| AF max : 2 vol/h | 0.01 | 0.01 | 0.01 | 0.01 | 0.01 | 0.01 | 0.50 | 1.0 | 1.0 |
| AF max : 4 vol/h | 0.01 | 0.01 | 0.01 | 0.01 | 0.01 | 0.01 | 0.25 | 1.0 | 1.0 |
| AF max : 6 vol/h | 0.01 | 0.01 | 0.01 | 0.01 | 0.01 | 0.01 | 0.17 | 1.0 | 1.0 |
| AF max : 8 vol/h | 0.01 | 0.01 | 0.01 | 0.01 | 0.01 | 0.01 | 0.13 | 1.0 | 1.0 |
| use of ind.humid | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| add ind.humid control | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| use of exch. | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| add heat exch. control | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| use of heating coil | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| use of direct humid. | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| use of cooling coil | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 |

**Control matrix during occupancy for winter**

System : none

| | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | ti-occ | -20.0 | 18.0 | 19.0 | 20.0 | 21.0 | 24.0 | 25.0 | 26.0 | 70.0 |
| air flow | AF max : 2 vol/h | 1.00 | 1.00 | 1.00 | 1.00 | 0.50 | 0.50 | 1.0 | 1.0 | 1.0 |
| rate ratio | AF max : 4 vol/h | 1.00 | 1.00 | 1.00 | 1.00 | 0.25 | 0.25 | 1.0 | 1.0 | 1.0 |
| Actual Air flow / Maximal Air flow | AF max : 6 vol/h | 0.67 | 0.67 | 0.67 | 0.67 | 0.17 | 0.17 | 1.0 | 1.0 | 1.0 |
| | AF max : 8 vol/h | 0.50 | 0.50 | 0.50 | 0.50 | 0.13 | 0.13 | 1.0 | 1.0 | 1.0 |
| | use of ind.humid | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | add ind.humid control | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | use of exch. | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| | add heat exch. control | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 |
| | use of heating coil | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 |
| | use of direct humid. | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | use of cooling coil | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

System : evaporative direct

| | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | ti-occ | -20.0 | 18.0 | 19.0 | 20.0 | 21.0 | 24.0 | 25.0 | 26.0 | 70.0 |
| air flow | AF max : 2 vol/h | 1.00 | 1.00 | 1.00 | 1.00 | 0.50 | 0.50 | 1.0 | 1.0 | 1.0 |
| rate ratio | AF max : 4 vol/h | 1.00 | 1.00 | 1.00 | 1.00 | 0.25 | 0.25 | 1.0 | 1.0 | 1.0 |
| Actual Air flow / Maximal Air flow | AF max : 6 vol/h | 0.67 | 0.67 | 0.67 | 0.67 | 0.17 | 0.17 | 1.0 | 1.0 | 1.0 |
| | AF max : 8 vol/h | 0.50 | 0.50 | 0.50 | 0.50 | 0.13 | 0.13 | 1.0 | 1.0 | 1.0 |
| | use of ind.humid | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | add ind.humid control | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | use of exch. | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| | add heat exch. control | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 |
| | use of heating coil | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 |
| | use of direct humid. | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 |
| | use of cooling coil | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

System : evaporative indirect

| | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | ti-occ | -20.0 | 18.0 | 19.0 | 20.0 | 21.0 | 24.0 | 25.0 | 26.0 | 70.0 |
| air flow | AF max : 2 vol/h | 1.00 | 1.00 | 1.00 | 1.00 | 0.50 | 0.50 | 1.0 | 1.0 | 1.0 |
| rate ratio | AF max : 4 vol/h | 1.00 | 1.00 | 1.00 | 1.00 | 0.25 | 0.25 | 1.0 | 1.0 | 1.0 |
| Actual Air flow / Maximal Air flow | AF max : 6 vol/h | 0.67 | 0.67 | 0.67 | 0.67 | 0.17 | 0.17 | 1.0 | 1.0 | 1.0 |
| | AF max : 8 vol/h | 0.50 | 0.50 | 0.50 | 0.50 | 0.13 | 0.13 | 1.0 | 1.0 | 1.0 |
| à controler | use of ind.humid | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 |
| | add ind.humid control | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 |
| | use of exch. | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| | add heat exch. control | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 |
| | use of heating coil | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 |
| | use of direct humid. | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | use of cooling coil | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

System : evaporative indirect + direct

| | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | ti-occ | -20.0 | 18.0 | 19.0 | 20.0 | 21.0 | 24.0 | 25.0 | 26.0 | 70.0 |
| air flow | AF max : 2 vol/h | 1.00 | 1.00 | 1.00 | 1.00 | 0.50 | 0.50 | 1.0 | 1.0 | 1.0 |
| rate ratio | AF max : 4 vol/h | 1.00 | 1.00 | 1.00 | 1.00 | 0.25 | 0.25 | 1.0 | 1.0 | 1.0 |
| Actual Air flow / Maximal Air flow | AF max : 6 vol/h | 0.67 | 0.67 | 0.67 | 0.67 | 0.17 | 0.17 | 1.0 | 1.0 | 1.0 |
| | AF max : 8 vol/h | 0.50 | 0.50 | 0.50 | 0.50 | 0.13 | 0.13 | 1.0 | 1.0 | 1.0 |
| | use of ind.humid | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 |
| | add ind.humid control | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 |
| | use of exch. | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| | add heat exch. control | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 |
| | use of heating coil | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 |
| | use of direct humid. | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 |
| | use of cooling coil | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

System : evaporative indirect + cooling co

| | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | ti-occ | -20.0 | 18.0 | 19.0 | 20.0 | 21.0 | 24.0 | 25.0 | 26.0 | 70.0 |
| air flow | AF max : 2 vol/h | 1.00 | 1.00 | 1.00 | 1.00 | 0.50 | 0.50 | 1.0 | 1.0 | 1.0 |
| rate ratio | AF max : 4 vol/h | 1.00 | 1.00 | 1.00 | 1.00 | 0.25 | 0.25 | 1.0 | 1.0 | 1.0 |
| Actual Air flow / Maximal Air flow | AF max : 6 vol/h | 0.67 | 0.67 | 0.67 | 0.67 | 0.17 | 0.17 | 1.0 | 1.0 | 1.0 |
| | AF max : 8 vol/h | 0.50 | 0.50 | 0.50 | 0.50 | 0.13 | 0.13 | 1.0 | 1.0 | 1.0 |
| | use of ind.humid | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | add ind.humid control | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | use of exch. | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| | add heat exch. control | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 |
| | use of heating coil | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 |
| | use of direct humid. | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | use of cooling coil | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 |

System : cooling coil only

| | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | ti-occ | -20.0 | 18.0 | 19.0 | 20.0 | 21.0 | 24.0 | 25.0 | 26.0 | 70.0 |
| air flow | AF max : 2 vol/h | 1.00 | 1.00 | 1.00 | 1.00 | 0.50 | 0.50 | 1.0 | 1.0 | 1.0 |
| rate ratio | AF max : 4 vol/h | 1.00 | 1.00 | 1.00 | 1.00 | 0.25 | 0.25 | 1.0 | 1.0 | 1.0 |
| Actual Air flow / Maximal Air flow | AF max : 6 vol/h | 0.67 | 0.67 | 0.67 | 0.67 | 0.17 | 0.17 | 1.0 | 1.0 | 1.0 |
| | AF max : 8 vol/h | 0.50 | 0.50 | 0.50 | 0.50 | 0.13 | 0.13 | 1.0 | 1.0 | 1.0 |
| | use of ind.humid | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | add ind.humid control | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | use of exch. | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| | add heat exch. control | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 |
| | use of heating coil | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 |
| | use of direct humid. | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | use of cooling coil | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 |

**Control matrix during non occupancy for winter**

System : none

air flow
rate ratio
actual Air flow / Maximal Air flow

| ti-inoc | -20.0 | 16.0 | 16.0 | 16.0 | 17.0 | 24.0 | 25.0 | 26.0 | 70.0 |
|---|---|---|---|---|---|---|---|---|---|
| AF max : 2 vol/h | 1.00 | 1.00 | 1.00 | 1.00 | 0.01 | 0.01 | 1.0 | 1.0 | 1.0 |
| AF max : 4 vol/h | 1.00 | 1.00 | 1.00 | 1.00 | 0.01 | 0.01 | 1.0 | 1.0 | 1.0 |
| AF max : 6 vol/h | 0.67 | 0.67 | 0.67 | 0.67 | 0.01 | 0.01 | 1.0 | 1.0 | 1.0 |
| AF max : 8 vol/h | 0.50 | 0.50 | 0.50 | 0.50 | 0.01 | 0.01 | 1.0 | 1.0 | 1.0 |
| use of ind.humid | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| add ind.humid control | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| use of exch. | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| add heat exch. control | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 |
| use of heating coil | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 |
| use of direct humid. | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| use of cooling coil | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

System : evaporative direct

air flow
rate ratio
actual Air flow / Maximal Air flow

| ti-inoc | -20.0 | 16.0 | 16.0 | 16.0 | 17.0 | 24.0 | 25.0 | 26.0 | 70.0 |
|---|---|---|---|---|---|---|---|---|---|
| AF max : 2 vol/h | 1.00 | 1.00 | 1.00 | 1.00 | 0.01 | 0.01 | 1.0 | 1.0 | 1.0 |
| AF max : 4 vol/h | 1.00 | 1.00 | 1.00 | 1.00 | 0.01 | 0.01 | 1.0 | 1.0 | 1.0 |
| AF max : 6 vol/h | 0.67 | 0.67 | 0.67 | 0.67 | 0.01 | 0.01 | 1.0 | 1.0 | 1.0 |
| AF max : 8 vol/h | 0.50 | 0.50 | 0.50 | 0.50 | 0.01 | 0.01 | 1.0 | 1.0 | 1.0 |
| use of ind.humid | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| add ind.humid control | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| use of exch. | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| add heat exch. control | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 |
| use of heating coil | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 |
| use of direct humid. | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 |
| use of cooling coil | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

System : evaporative indirect

air flow
rate ratio
actual Air flow / Maximal Air flow

| ti-inoc | -20.0 | 16.0 | 16.0 | 16.0 | 17.0 | 24.0 | 25.0 | 26.0 | 70.0 |
|---|---|---|---|---|---|---|---|---|---|
| AF max : 2 vol/h | 1.00 | 1.00 | 1.00 | 1.00 | 0.01 | 0.01 | 1.0 | 1.0 | 1.0 |
| AF max : 4 vol/h | 1.00 | 1.00 | 1.00 | 1.00 | 0.01 | 0.01 | 1.0 | 1.0 | 1.0 |
| AF max : 6 vol/h | 0.67 | 0.67 | 0.67 | 0.67 | 0.01 | 0.01 | 1.0 | 1.0 | 1.0 |
| AF max : 8 vol/h | 0.50 | 0.50 | 0.50 | 0.50 | 0.01 | 0.01 | 1.0 | 1.0 | 1.0 |
| use of ind.humid | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 |
| add ind.humid control | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 |
| use of exch. | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| add heat exch. control | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 |
| use of heating coil | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 |
| use of direct humid. | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| use of cooling coil | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

System : evaporative indirect + direct

air flow
rate ratio
actual Air flow / Maximal Air flow

| ti-inoc | -20.0 | 16.0 | 16.0 | 16.0 | 17.0 | 24.0 | 25.0 | 26.0 | 70.0 |
|---|---|---|---|---|---|---|---|---|---|
| AF max : 2 vol/h | 1.00 | 1.00 | 1.00 | 1.00 | 0.01 | 0.01 | 1.0 | 1.0 | 1.0 |
| AF max : 4 vol/h | 1.00 | 1.00 | 1.00 | 1.00 | 0.01 | 0.01 | 1.0 | 1.0 | 1.0 |
| AF max : 6 vol/h | 0.67 | 0.67 | 0.67 | 0.67 | 0.01 | 0.01 | 1.0 | 1.0 | 1.0 |
| AF max : 8 vol/h | 0.50 | 0.50 | 0.50 | 0.50 | 0.01 | 0.01 | 1.0 | 1.0 | 1.0 |
| use of ind.humid | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 |
| add ind.humid control | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| use of exch. | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| add heat exch. control | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 |
| use of heating coil | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 |
| use of direct humid. | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 |
| use of cooling coil | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

System : evaporative indirect + cooling coil

air flow
rate ratio
actual Air flow / Maximal Air flow

| ti-inoc | -20.0 | 16.0 | 16.0 | 16.0 | 17.0 | 24.0 | 25.0 | 26.0 | 70.0 |
|---|---|---|---|---|---|---|---|---|---|
| AF max : 2 vol/h | 1.00 | 1.00 | 1.00 | 1.00 | 0.01 | 0.01 | 1.0 | 1.0 | 1.0 |
| AF max : 4 vol/h | 1.00 | 1.00 | 1.00 | 1.00 | 0.01 | 0.01 | 1.0 | 1.0 | 1.0 |
| AF max : 6 vol/h | 0.67 | 0.67 | 0.67 | 0.67 | 0.01 | 0.01 | 1.0 | 1.0 | 1.0 |
| AF max : 8 vol/h | 0.50 | 0.50 | 0.50 | 0.50 | 0.01 | 0.01 | 1.0 | 1.0 | 1.0 |
| use of ind.humid | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| add ind.humid control | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| use of exch. | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| add heat exch. control | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 |
| use of heating coil | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 |
| use of direct humid. | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| use of cooling coil | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 |

System : cooling coil only

air flow
rate ratio
actual Air flow / Maximal Air flow

| ti-inoc | -20.0 | 16.0 | 16.0 | 16.0 | 17.0 | 24.0 | 25.0 | 26.0 | 70.0 |
|---|---|---|---|---|---|---|---|---|---|
| AF max : 2 vol/h | 1.00 | 1.00 | 1.00 | 1.00 | 0.01 | 0.01 | 1.0 | 1.0 | 1.0 |
| AF max : 4 vol/h | 1.00 | 1.00 | 1.00 | 1.00 | 0.01 | 0.01 | 1.0 | 1.0 | 1.0 |
| AF max : 6 vol/h | 0.67 | 0.67 | 0.67 | 0.67 | 0.01 | 0.01 | 1.0 | 1.0 | 1.0 |
| AF max : 8 vol/h | 0.50 | 0.50 | 0.50 | 0.50 | 0.01 | 0.01 | 1.0 | 1.0 | 1.0 |
| use of ind.humid | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| add ind.humid control | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| use of exch. | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| add heat exch. control | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 |
| use of heating coil | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 |
| use of direct humid. | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| use of cooling coil | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 |

# IEA Annex 28

# Low Energy Cooling

## Subtask 2: Detailed Design Tools

## Design Tools for Evaporative Cooling

### Algorithms for direct and indirect evaporative coolers

Lawrence Berkeley National Laboratory, Berkeley (USA)

Peilin Chen, Yu Joe Huang

# Contents Chapter D

# 1. Technology area

Component models for indirect and direct evaporative cooling systems.

# 2. Developed by

**Name** : Peilin Chen and Yu Joe Huang

**Organisation** : Lawrence Berkeley National Laboratory

**Address** : Mail 90-2000, 1 Cyclotron Road, Berkeley CA 94720 U.S.A.

**Phone** : (510) 486-7082

**Fax** : (510) 486-4673

**E-mail** : YJHuang@lbl.gov

# 3. General description

The purpose of the models are to simulate the performance of direct and indirect evaporative cooling systems. Although the models rely on basic principles of heat and mass transfer, a number of simplifications have been made to reduce the computational effort. The models can be either incorporated directly into building energy simulation programs, or used to generate effectiveness curves that can then be utilized in the building simulations.

Two direct evaporative cooling models are included, one based on laboratory test data of rigid cellulose media and aspen pads, and the other based on regression analysis of manufacturer's data. The required inputs to *dec1.f* are the dry-bulb temperature and humidity ratio of the entering air, the dimensions and type of direct evaporative cooling media, air flow rate, and fan efficiency. The outputs are the psychrometric conditions of the leaving air, the effectiveness of the evaporative cooler, fan and pump power, and total power demand. *Figure 1 is an information flow diagram for dec1.f*.

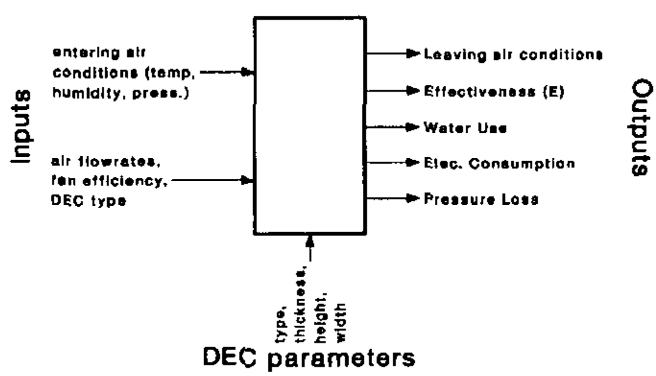



*fig. 1. inputs and outputs for direct evaporative cooler model dec1.f*

*dec2.1* is another simple regression model of direct cooling effectiveness based on manufacturer's data provided for reference. The required inputs are the dry-bulb and wet-bulb temperatures of the entering air, the thickness, height, and width of the rigid cellulose media, and the air flow rate. The outputs are the psychrometric conditions of the leaving air, the effectiveness of the direct evaporative cooler, and its water consumption. The information flow is similar to that shown in Fig. 1 for *dec1.f.*

The indirect evaporative cooling model *idec.f* considers three basic heat-exchanger designs - tube, plate, and plate-fin types. The model inputs are the physical dimensions and thermal properties of the heat exchanger, and the entering air parameters and flow rates. The outputs are the leaving primary and secondary air parameters, effectiveness, water evaporation rate, air pressure drops, cooling capacity, and power demand of the indirect evaporative cooler. Figure 2 is an information flow diagram for *idec.f*

Comparisons of this model to laboratory tests showed an average difference in effectiveness of less than 0.03, and pressure drop of less than 5 Pa. This level of error translates to a difference in predicted temperature of 0.3°C.

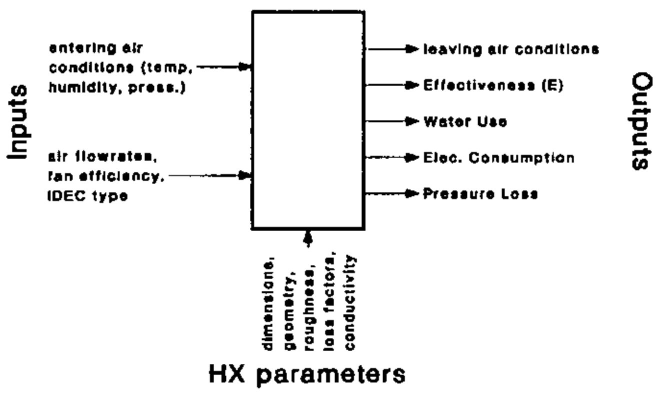



*fig. 2. inputs and outputs for indirect evaporative cooler model idec.f*

## 4. Nomenclature

| | |
|---|---|
| A | = area, $m^2$ |
| B | = barometric pressure, mbar |
| $C_o$ | = local loss coefficient, dimensionless |
| COP | = coefficient of performance |
| $c_p$ | = specific heat of air, kJ/kg°C |
| D | = mass transfer coefficient, g/$m^2$s |
| d | = diameter or hydraulic diameter, m |
| E | = effectiveness, dimensionless |
| f | = friction loss factor, dimensionless |
| g | = mass flowrate, kg/s |
| h | = enthalpy, kJ/kg |
| K | = conductivity, W/m°C |
| L | = length, m, or volumetric flowrate, $m^3$/s |
| Nu | = Nusselt Number, $Nu = \frac{\alpha d}{K}$ |
| Pr | = Prandtl Number, $Pr = \frac{1000 c_p \mu}{K}$ |
| Q | = cooling capacity, kW |
| Re | = Reynolds Number, $Re = \frac{v d \rho}{\mu}$ |
| RH | = relative humidity, dimensionless |
| s | = tube distance, m |
| t | = temperature, °C s |
| v | = velocity, m/s |
| w | = humidity ratio, kg/kg |
| W | = required fan power, W |
| $\alpha$ | = convective heat transfer coefficient, W/m2oC |
| P | = total pressure drop, Pa |
| Pt | = friction loss, Pa |
| Pl | = local loss, Pa |
| $\rho$ | = density, kg/$m^3$ |
| $\mu$ | = dynamic viscosity of air, kg/ms |

***Subscripts***

| | | | | | |
|---|---|---|---|---|---|
| db | = dry bulb | p | = primary | 1 | = entering |
| wb | = wet bulb | s | = secondary | 2 | = leaving |
| i | = inside | r | = room | w | = of effective surface |
| o | = outside | m | = average | | |

# 5. Mathematical description

The heat and mass transfer processes inside an indirect evaporative cooler are complicated. For practical purposes, the following simplifying assumptions are made :

1. The water film temperature over the tubes and plates in the secondary air is assumed to be uniform and represented by an effective surface temperature $t_{wo}$.
2. The heat and mass transfer effects between the water droplets and the air in the secondary air are assumed to be negligible and ignored in the calculations.
3. The surface temperature of the tubes and plates on the primary air side are also assumed to be uniform and represented by an effective surface temperature $t_{wi}$ .

### 5.1 Heat and mass transfer between the secondary air and the wet surface

For a differential heat transfer area $dA_o$, the following equation can be written :

$$g_s dh_s = D\,(h_{wo} - h_s)\,dA_o \tag{1}$$

where gs = secondary air flow rate, kg/s
$h_{wo}$ = enthalpy of saturation air with effective surface temperature $t_{wo}$, kJ/kg
D = mass transfer coefficient, $kg/m^2s$
$h_s$ = enthalpy of secondary air, kJ/kg

Substituting the Lewis relationship $D_2 = \alpha_1/1000C_p$ into Equation 1 gives :

$$g_s dh = \frac{\alpha_o}{1000c_p}(h_w - h_s)dA_o \tag{2}$$

where $c_p$ = specific heat of air,kJ/kg°C
$\alpha_o$ = convective heat transfer coefficient, $W/m^{2\circ}C$

Integrating Equation 2 results in :

$$\frac{h_w - h_{s1}}{h_w - h_{s2}} = \exp\left(\frac{\alpha_o A_o}{1000 g_s c_p}\right) \tag{3}$$

where $h_{s1}$ = enthalpy of entering secondary air, kJ/kg
$h_{s2}$ = enthalpy of leaving secondary air, kJ/kg
$A_o$ = heat transfer area at secondary air side, $m^2$

From Equation 3 the following equation can be obtained for calculating $h_{s2}$ :

$$h_{s2} = h_w - \frac{h_w - h_{s1}}{\exp\left(\frac{\alpha_o A_o}{1000 g_s c_p}\right)} \tag{4}$$

### 5.2 Heat balance of the primary air

When the cooling process of the primary air is sensible, its heat balance equation is :

$$1000g_p c_p (t_{dbp1} - t_{dbp2}) = \alpha_i \frac{t_{dbp1} - t_{dbp2}}{\ln(\frac{t_{dbp1} - t_w}{t_{dbp2} - t_w})} A_i \tag{5}$$

After rearranging Equation 5 the following form can be obtained :

$$t_{dbp2} = t_w + \frac{t_{dbp1} - t_w}{\exp(\frac{\alpha_i A_i}{1000 g_p c_p})} \tag{6}$$

where $t_{dbp1}$ = dry bulb temperature of entering primary air, °C
$t_{dbp2}$ = dry bulb temperature of leaving primary air, °C
$t_w$ = effective surface temperature, °C
$\alpha_i$ = convective heat transfer coefficient, $W/m^2\,°C$
$g_p$ = primary air flow rate, kg/s
$A_i$ = heat transfer area at the primary air side, $m^2$

For a plate-type heat exchanger, $A_o = A_i = A_o$.

### 5.3 Heat balance between primary and secondary air

The heat loss of the primary air must be equal to the heat gain of the secondary air. Therefore;

$$g_s (h_{s2} - h_{s1}) = g_p c_p (t_{dbp1} - t_{dbp2}) \tag{7}$$

From here,

$$h_{s2} = h_{s1} + \frac{g_p}{g_s} c_p (t_{dbp1} - t_{dbp2}) \tag{8}$$

Substituting Eq. 6 into Eq. 8 gives :

$$h_{s2} = h_{s1} + \frac{g_1}{g_2} c_p (t_{dbp1} - t_w) \left[ 1 - \frac{1}{\exp(\frac{\alpha_i A_i}{1000 g_i c_p})} \right] \tag{9}$$

### 5.4 Relationship between effective surface temperatures $t_{wo}$ and $t_{wi}$

For a tube-type indirect evaporative cooler,

$$1000 g_s (h_{s2} - h_{s1}) = n 2\pi K_m L \frac{t_{wi} - t_{wo}}{\ln \frac{d_o}{d_i}} \tag{10}$$

or

$$t_{wi} = \frac{1000 g_s (h_{s2} - h_{s1})}{n 2\pi K_m L} \ln \frac{d_o}{d_i} + t_{wo} \tag{11}$$

where n = number of tubes
L = length of tubes
$K_m$ = conductance of tube ($W/m\,°C$)
$d_o$ = outside diameter of tubes
$d_i$ = inside diameter of tubes

For a plate-type evaporative cooler,

$$1000g_s(h_{s2} - h_{s1}) = A\frac{K_m}{l}(t_{wi} - t_{wo}) \tag{12}$$

or

$$t_{wi} = \frac{1000g_s(h_{s2} - h_{s1})l}{AK_m} + t_{wo} \tag{13}$$

where $l$ = thickness of the plates, m.

### 5.5 Heat transfer effectiveness E

The heat transfer effectiveness, E, is defined as :

$$E = \frac{t_{dbp1} - t_{dbp2}}{t_{dbp1} - t_{wbs1}} \tag{14}$$

where $t_{sw1}$ = wet bulb temperature of entering primary and secondary air, °C.

In the above derivation, there are four unknown variables, $t_{wo}$, $t_{wi}$, $h_{sw}$, and $t_{dbp2}$. It would be mathematically difficult to solve for the four unknown variables theoretically. However, it is quite easy to do so using a trial-and-error method in a computer program :

1. make an initial guess of $t_{wo}$, and calculate the initial enthalpy $h_{wo}$.
2. use Eq. 4 to calculate $h_{s2}$
3. use Eq. 11 or Eq. 13 to calculate $t_{wi}$.
4. use Eq. 9 to calculate a new $h_{s2}$.
5. compare the two values of $h_{s2}$. If they are not identical, make a new guess of $t_{wo}$, and repeat the procedure until the two $h_{s2}$ values equal each other. This produces the correct values for $t_{wo}$, $t_{wi}$ and hs2.
6. use Eq. 6 to calculate $t_{dbp2}$. Other parameters for theleaving primary and secondary air streams can be calculated using standard psychrometric equations.
7. use Eq. 14 to calculate the heat transfer effectiveness, E.

This procedure has been programmed into the *idec0.for* and *idec.for* programs. The only difference between the two programs is that *idec0.for* has an interactive input stream, while *idec.for* is set up for batch processing with an input file *idec.inp*, and an output file *idec.out*.

### 5.6 Calculation of convective heat transfer coefficients

To solve the above equations, it is necessary to calculate the convective heat transfer coefficients for the primary and secondary air streams ($\alpha_i$ and $\alpha_o$). The following equations are used to calculate K, $\mu$ , and $\rho$ :

$$K = 7.6916x10^{-5}t + 0.024178 \tag{15}$$

$$\mu = 9.80665x10^{-6}(1.712 + 0.0058t) \tag{16}$$

$$\rho = \frac{B(1+w)}{4.615(273.15 + t)(0.62198 + w)} \tag{17}$$

where t = air temperature, °C
B = barometric pressure, mbar
w = humidity ratio of air, kg/kg

#### 5.6.1 Tube-type indirect evaporative coolers

The following equations are used for forced convection inside tubes under turbulent flow :

$$Nu_i = \frac{\alpha_i d_i}{K_i} = 0.23\,Re_i^{0.8}\,Pr_i^{0.4}\left[1 + (\frac{d_i}{L})^{0.7}\right] \tag{18}$$

$$Re_i = \frac{v_i d_i \rho_i}{\mu_i} \tag{19}$$

$$Pr_i = \frac{1000 c_p \mu_i}{K_i} \tag{20}$$

$$K_i = 7.691 x 10^{-5} t_{pm} + 0.024178 \tag{21}$$

$$\mu_i = 9.80665 x 10^{-6} (1.712 + 0.0058 t_{pm}) \tag{22}$$

$$\rho_i = \frac{B(1 + w_{pm})}{4.615(273.15 + t_{pm})(0.62198 + w_{pm})} \tag{23}$$

where L = tube length, m
$d_i$ = inside tube diameter, m

From Equation 18,

$$\alpha_i = \frac{K_i}{d_i} 0.23\,Re_i^{0.8}\,Pr_i^{0.4}\left[1 + (\frac{d_i}{L})^{0.7}\right] \tag{24}$$

For forced convection when the secondary air flow is turbulent and normal to staggered tubes, and row number is greater than 10, the equation is :

$$Nu_o - \frac{\alpha_o d_o}{K_o} = 0.31 Re_o^{0.6} (\frac{S_1}{S_2})^{0.2} \tag{25)}$$

Here,

$$Re_o = \frac{v_o d_o \rho_o}{K_o} \tag{26}$$

$$Pr_o = \frac{1000 c_p \mu_o}{K_o} \tag{27}$$

$$K_o = 7.6916 x 10^{-5} t_{sm} + 0.024178 \tag{28}$$

$$\mu_o = 9.80665 x 10^{-6} (1.712 + 0.0058 t_{sm}) \tag{29}$$

$$\rho_o = \frac{B(1 + w_{sm})}{4.615(273.15 + t_{sm})(0.62198 + w_{sm})} \tag{30}$$

where $S_1$ = tube distance (center to center) in a row, m
$S_2$ = row distance (center to center), m
$d_o$ = outside tube diameter, m

From Equation 25,

$$\alpha_o = \frac{K_o}{d_o} 0.31 Re_o^{0.6} (\frac{S_1}{S_2})^{0.2} \tag{31}$$

### 5.6.2 Plate-type indirect evaporative coolers

For very narrow passages between plates, the following equations can be used :

a. When Re $\geq$ 1000,

$$Nu = \frac{\alpha d}{K} = 0.2 Re^{0.67} Pr^{0.4} (\frac{\mu}{\mu_w})^{0.1} \tag{32}$$

b. When Re $\leq$ 10,

$$Nu = \frac{\alpha d}{K} = 1.68 (\frac{Re Pr d}{L})^{0.4} (\frac{\mu}{\mu_w})^{0.1} \tag{33}$$

where d = hydraulic diameter of primary or secondary air passages, m
L = length of primary or secondary air passages, m
$\mu_w$ = dynamic viscosity of air with effective surface temperature $t_w$, kg/ms

In Eqs 32 and 33, $\mu$, K and $\rho$ (needed to calculate Re) should be calculated using the average air temperature.

c. When 10 < Re < 1000, an interpolative method can be used.

### 5.7 Calculation of friction loss factors

Colebrook's formula is used to calculate the friction loss factor :

$$\frac{1}{\sqrt{f}} = 1.74 - 2\log(\frac{2k}{d} + \frac{18.7}{Re\sqrt{f}}) \tag{34}$$

where f = friction loss factor, dimensionless
k = roughness of passage walls, m
d = hydraulic diameter, m
Re = Reynold's Number of the air flow, dimensionless

Since it is difficult to solve f directly by Eq, 34, an iterative method is used.

# 6. References

Chen, P.L., Huang, Y.J., Qin, H.M., and Wu, H.F. 1991, "A heat and mass transfer model for thermal and hydraulic calculations of indirect evaporative cooler performance", *ASHRAE Transactions*, American Society of Heating, Refrigeration, and Air-conditioning Engineers, Atlanta USA.

Chen, P.L., Qin, H.M., and Huang, Y.J. 1994. "Laboratory validation of the heat and mass transfer model for calculations of indirect evaporative cooler performance" (draft), Lawrence Berkeley National Laboratory, Berkeley USA.

Chen, P.L. 1992. "A modified model for thermal calculations of indirect evaporative air cooler performance and its laboratory validation", *Zhileng Xuebao (Journal of Refrigeration)*, No. 52 (in Chinese).

Dowdy, J.A. et al. 1987. "Experimental determination of heat and mass transfer coefficients in aspen pads", *ASHRAE Transactions*, Vol. 92, Part 2 , pp. 60-70. American Society of Heating, Refrigeration, and Air-conditioning Engineers, Atlanta USA

Dowdy, J.A. et al. 1988. "Experimental determination of heat and mass transfer coefficients in rigid impregnated cellulose evaporative media", *ASHRAE Transactions*, Vol. 93, Part 2 , pp. 382-395. American Society of Heating, Refrigeration, and Air-conditioning Engineers, Atlanta USA

# 7 Flowcharts

**Figure 3. Flowchart for DEC1.F**

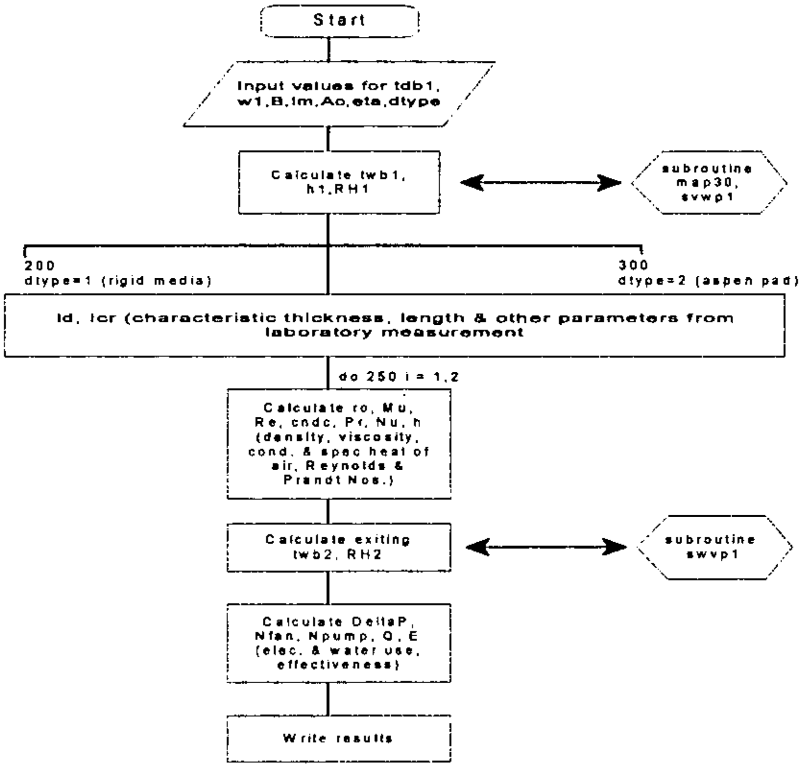



**Figure 4. Flowchart for DEC2.F**

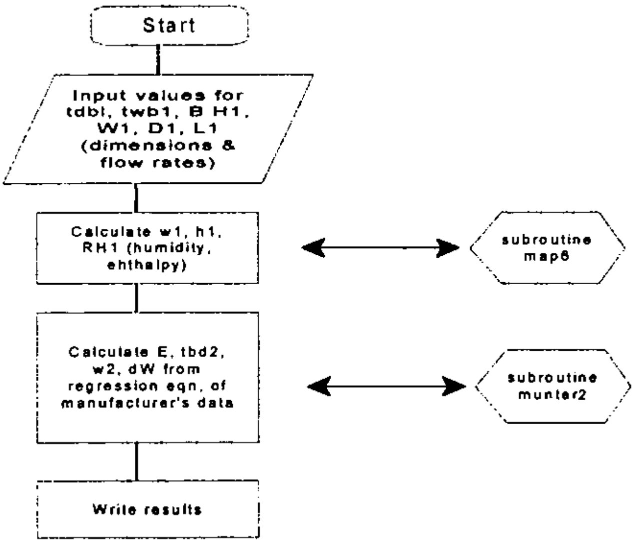
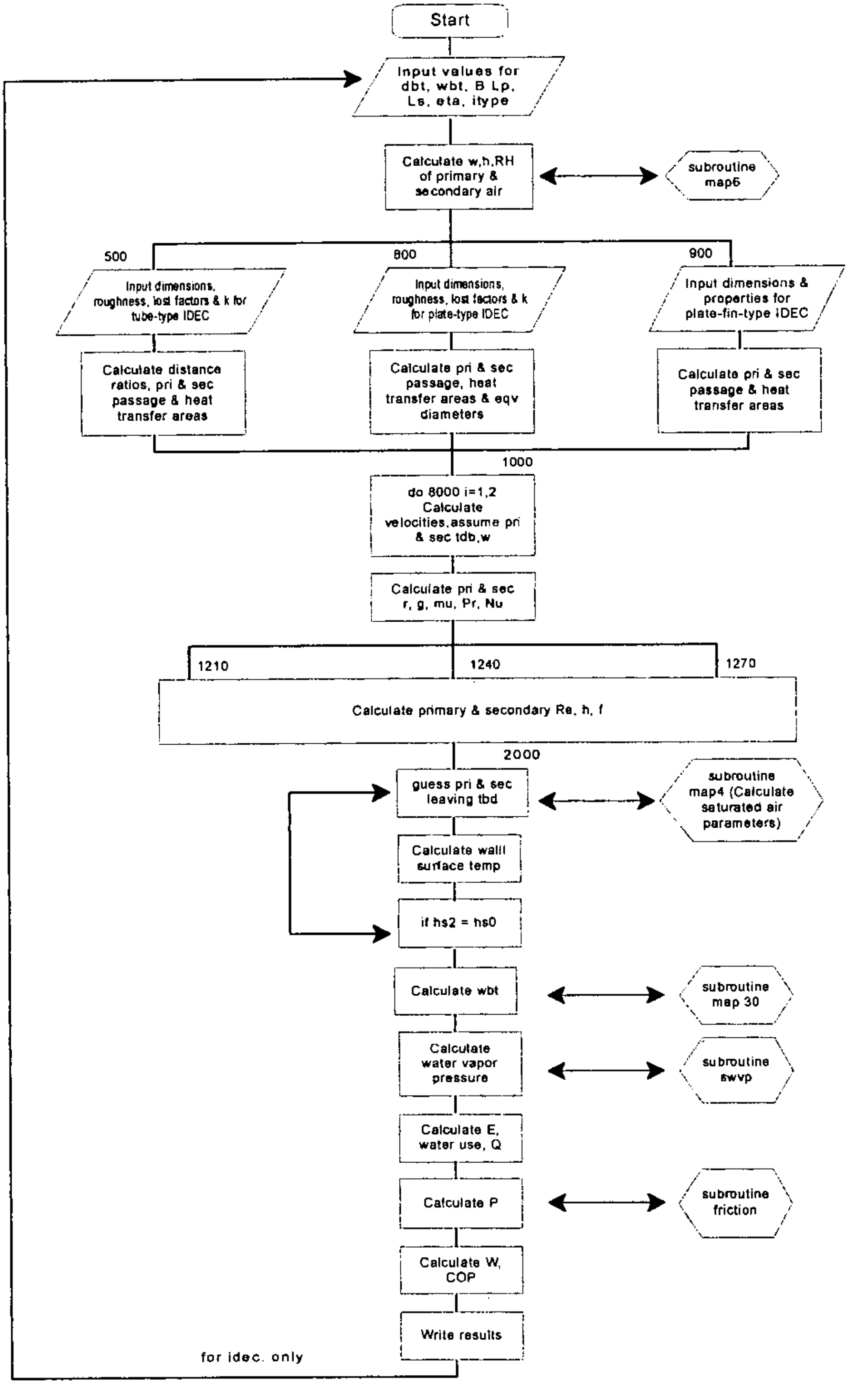


**Figure 5. Flowchart for IDEC.F**



# 8. Source code

### Program dec1.f

```
c      This program is used to calculate the effectveness, sensible
c  cooling capacity and energy and water consumption of direct eva-
c  porative coolers including the rigid impregnated cellulose evapo-
c  rative media and the aspen pad.

      program dec1

      implicit real(l,m,n)
      character*8 name
      open(5,file='dec1.inp')
      open(6,file='dec1.out')
C
C start batch mod by YJH
C
  5   read(5,99)
 99    format(3(/))
c
c tdb1  w1      B       L   lm    Ac   eta      idir
c  dbt humrat atmp   flow  len  area fan_eff (1=cell,
c (C)    [ ]   [100Pa](m3/s) [m]  [m2]  [ ]     2=aspen)  title
c--------------------------------------------------------------
c 40.0 .01161 132.5  1.33  0.12  0.50 0.85     1       TEST1
c
 50   read(5,*,end=999) tdb1,w1,B,L,lm,Ac,eta,idir,name

      h1=1.01*tdb1+w1*(2500.0+1.84*tdb1)
      call map30(tdb1,h1,w1,B,twb1)
      call swvp1(tdb1,et1)
      RH1=w1*B/((0.62198+w1)*et1)

c           h1=enthalpy of outdoor air, kJ/kg
c           twb1=outdoor air wet-bulb temperature, C
c           RH1=relative humidity of outdoor air
c - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
c       Guess the initial values of the average temperature and
c  humidity ratio of the air:
      tdbm=tdb1
      wm=w1

      v=L/Ac
c           v=face air velocity, m/s
c - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
      if(idir.eq.1) goto 200
      if(idir.eq.2) goto 300
c - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
200   write(6,210)name
210   format('MEDIA : RIGID IMPREGNATED CELLULOSE  CASE=',a8)

      le=0.00285
      lcr=le

c            le=characteristic thickness obtained from experiment
c            lcr=the length used in Re and Nu
      xa=0.1
      xb=0.12
      As=351.0*lm
      goto 240
c - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
300   write(6,310) name
310   format('MEDIA : ASPEN PAD                    CASE=',a8)

      le=0.002
      lcr=lm
      xa=1.77
      xb=0.63
      As=503.7*lm
c - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
240   continue

      do 250 i=1,2

      ro=B*(1.0+wm)/(4.615*(273.15+tdbm)*(0.62198+wm))
      Mu=9.80665e-6*(1.712+0.0058*tdbm)
      Re=v*lcr*ro/Mu
      cndc=7.6916e-5*tdbm+2.4178e-2
      c=1.01
      Pr=c*Mu*1000.0/cndc
```

```
c          ro=air density, kg/m**3


c          Mu=dynamic viscosity of air, kg/ms
c          Re=Reynolds Number
c          cndc=heat conductivity of air, W/mC
c          c=specific heat of air, kJ/kgC
c          Pr=Prandtl Number

c Because of Nu=hH*lcr/cndc=xa*(le/lm)**xb*Re**0.8*Pr**0.3333

      hH=cndc/lcr*xa*(le/lm)**xb*Re**0.8*Pr**0.3333

c          Nu=Nusselt Number
c          hH=convective heat transfer coefficient, W/m**2C

      NTU=hH*As/(ro*c*v*1000.0)

c          NTU=number of transfer units

      tdb2=twb1+(tdb1-twb1)*exp(-NTU)
      h2=h1
      w2=(h2-1.01*tdb2)/(2500.0+1.84*tdb2)

      tdbm=(tdb1+tdb2)/2.0
      wm=(w1+w2)/2.0

250   continue

      call swvp1(tdb2,et2)
      RH2=w2*B/((0.62198+w2)*et2)
      twb2=twb1

c          tdb2,twb2,h2,w2=dry-bulb temperature, wet-bulb temperature
c                          enthalpy and humidity ratio of supply air
c                          respectively.
      G=L*ro
      deltaW=G*(w2-w1)

      if(idir.eq.2) goto 260

      deltaP=0.0248*v**2*1000.0*lm
      Nfan=L*deltaP/eta
      Npump=20.0
      Ntotal=(Nfan+Npump)/1000.0

260   Q=1.01*G*(tdb1-tdb2)
      E=(tdb1-tdb2)/(tdb1-twb1)
      Ao=351.0*Ac*lm

c          deltaW=water consumption, kg/s
c          deltaP=air pressure drop, Pa
c          Nfan=fan power demand, W
c          Npump=pump power demand, W
c          Ntotal=total power demand, kW
c          Q=sensible coolin capacity, kW
c          E=effectiveness
c          Ao=heat transfer area, m**2
c - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
      write(6,400)tdb1,w1,twb1,RH1,tdb2,w2,twb2,RH2,B
400   format(' tdb1=',f6.2,1x,'w1=',f8.5,1x,'twb1=',f6.2,1x,'RH1=',
     1   f6.3/' tdb2=',f6.2,1x,'w2=',f8.5,1x,'twb2=',f6.2,1x,'RH2=',
     2   f6.3,1x,'B=',f8.2)


      write(6,420)L,G,lm,Ac
420   format(' Air Flowrate L, m**3/s=',f6.2,2x,
     1     ' G,kg/s=',f6.2,3x/
     2     ' Media Thickness lm, m= ',f6.4,3x,'Face Area Ac,m**2=',f6.2)

      write(6,440)E,Q
440   format(' Effectiveness E = ',6x,f5.3,3x,
     1        'Sensible Cooling Capacity Q,kW=',f6.2)

      write(6,460)deltaW,deltaP
460   format(' Water Consumption,kg/s= 'f5.3,3x,
     1        'Pressure Drop,Pa=',f6.2)

      write(6,480)v,Ao
480   format(' Air Velocity v,m/s= ',5x,f4.2,3x,
     1        'Heat Transfer Area Ao,m**2=',f5.1)

      write(6,490)Ntotal
490   format(' Power Demand, kW=',6x,f6.2)
c - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
      go to 50
```

```
999   close (5)
      end

      subroutine swvpl(t,p)
c  Calculation of Saturated Water Vapor Pressure
      a=(273.15+t)/273.16
      a1=10.79574*(1-1/a)
      a2=5.02800*alog10(a)
      a3=1.50475e-4*(1-10**(-8.2969*(a-1)))
      a4=0.42873e-3*(10**(4.76955*(1-1/a))-1)
      p=1013.25*10**(a1-a2+a3+a4-2.2195768)
      return
      end
```

***for listing of subroutine map30.f, see listing for program idec.f***

## Program dec2.f

```
c  This program calculates the effectiveness and water consumption of direct
c  evaporative coolers based on regression equations from Munters literature
c
      program dec2
      implicit real(l,m,n)
      open(5,file='dec2.inp')
      open(6,file='dec2.out')
C
C read data
C
   5  read(5,99)
  99   format(3(/))
c
c tdb1  twb1   B      HT    WD    D1     L1
c  dbt  wbt   atmp   heigh width depth  flow
c [C]   [C]   [Pa]   [m]   [m]    [m]   [m/s]
c-------------------------------------------------
c 40.0  25.0 101325. 0.33  0.50  0.102 0.165
c
  50  read(5,*,end=999) tdb1,twb1,Bs,HT,WD,D1,L1
c     write(*,*)tdb1,twb1,Bs,HT,WD,D1,L1
c
c
c  calculate humidity ratio, enthalpy, and relative humidity
C
      call map6(tdb1,twb1,Bs,w1,h1,RHT)

      call munters2(HT,D1,WD,L1,tdb1,twb1,h1,w1,Bs,tdb2,w2,dW,E1)

      write(6,120) HT,WD,D1,L1,tdb1,twb1,w1,Bs,tdb2,w2,dW,E1
120   format ('HT[m]=',f5.3,'  WD[m]=',f5.3,'  DP[mm]=',f5.1,
     1         '  L1[m3/s]=',f5.3,/
     1         'dbe[C]=',f5.1,'  wbe[C]=',f5.1,'  w1=',f7.4,
     1         '  B[Pa]=',f8.0,/
     1         'dbl[C]=',f5.1,'  w2=',f7.4,' h2o[g/s]=',f6.3,'  E=',f6.3)
      go to 50
999   close (5)
      end

c  This subroutine is used to calculate direct evaporative
c  air coolers of CELdek/GLASdek pads (MUNTERS HUMI-KOOL).

      subroutine munters2(HT,D1,WD,L1,tdb1,twb1,h1,w1,Bo,
     1tdb2,w2,dW,E1)
c
      real L1
c  - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
c  HT,WD-------height and width of the CELdek/GLASdek pads (m)
c  D1----------depth of the CELdek/GLASdek pads (m)
c              (standard depths 0.102 (12in),
c  L1----------air flowrate, (m3/s).
c  tdb1,twb1---dry and wet bulb temperatures of entering air, (C)
c  h1----------enthalpy of entering air (kJ/kg)
c  w1----------humidity ratio of entering air
c  tdb2--------dry bulb temperatures of leaving air, (C)
c  w2----------humidity ratio of leaving air
c  dW----------water consumption, (kg/s)
c  E1----------heat transfer effectiveness.
c  - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
c
c     convert to SI
c
      D1=D1*1000
      v=L1/(WD*HT)
c  - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
      a=-1.732+0.9361*D1-34.843e-4*D1**2+5.837e-6*D1**3
     1-0.36e-8*D1**4
```

```
      b=0.16424+0.00073*D1-0.08031e-4*D1**2
     1+0.01851e-6*D1**3-0.00131e-8*D1**4
      E1=(a/v**b)*0.01

c      write(*,*) D1,a,b,v,E1

c  The leaving air parameters of the CELdek/GLASdek are:
      tdb2=tdb1-E1*(tdb1-twb1)
      h2=h1
      w2=(h2-1.01*tdb2)/(2500.0+1.84*tdb2)
c  - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
c  The water consumption is:
      dW=L1*1.2*1000*(w2-w1)
      return
      end
```

***for listing of subroutines map6.f and swvp.f, see end of following program idec.f***

## Program idec.f

***(idec0.f similar)***

```
      program idec
c
c this is a modified version of evap2.f to do batch runs (YJH 6/24/97)
c
c This program is made for thermal and hydraulic calculations of
c tube type, plate type, or plate-fin type indirect evaporative
c coolers.  The primary air is outdoor air and the secondary air
c can be the room return air or outdoor air.
c ____________________________________________________________________

      implicit real(l,m,n)
      integer itype
      character*10 typenam(3)
      character*8 atype,name
      character*4 flag
      data typenam/'TUBE','PLATE','PLATE FIN'/
      open(5,file='idec.inp')
      open(6,file='idec.out')
C
C start batch mod by YJH
C
 105  read(5,98)flag
  98  format(a4)
      if (flag.eq.'LINE') then
       read(5,99)
  99   format(3(/))
           go to 105
      endif
c
cLINE 1 :
c tpdb1 tpwb1 tsdb1 tswb1  B    Lp    Ls    eta  itype
c  pri   pri   sec   sec  atm   pri   sec   fan
c  dbt   wbt   dbt   wbt  pres flow  flow   eff
c  [C]   [C]   [C]   [C]  [Pa][m3/s][m3/s]       [a8]
c
  50  read(5,100,end=999) tpdb1,tpwb1,tsdb1,tswb1,B,Lp,Ls,eta,atype,name
100   format(4f6.1,f8.0,2f6.3,f6.2,2(1x,a8))
      if (atype.eq.'TUBE    ') itype = 1
      if (atype.eq.'PLATE   ') itype = 2
      if (atype.eq.'PLATEFIN') itype = 3
      if (tsdb1.eq.0)tsdb1 = tpdb1
      if (tswb1.eq.0)tswb1 = tpwb1
      if (Ls.eq.0) Ls = Lp
C      write(6,100) tpdb1,tpwb1,tsdb1,tswb1,B,Lp,Ls,eta,atype,name
      write(6,1001) typenam(itype),name
1001  format('IDEC TYPE : ',a10,'(',a8,')')
      call map6(tpdb1,tpwb1,b,wp1,hp1,rhp1)
      call map6(tsdb1,tswb1,b,ws1,hs1,rhs1)
      if (itype.eq.1) goto 500
      if (itype.eq.2) goto 800
      if (itype.eq.3) goto 900

 500  continue
c
cLINE 2 if itype = 1 (tube type) :
c bs    L    di   do    n    n1    n2     s1   s2 rough coeff1coeff2coeffwcndcw
c inl  inl  tub  tub  tub  num          dis/ row ness  loss  loss  loss   k
c ht   len   id   od   no  /row rows   row  dis       to_t  fr_t  spray [kJ/
c [m]  [m] [mm] [mm]                  [mm] [mm] [mm]                      mC]
c
      read(5,*) bs,L,di,do,n,n1,n2,s1,s2,
     +  rough,coeff1,coeff2,coeffw,cndcw
      read(5,*)
```

```
 101  format(4f5.2,3f5.0,5f5.2,f5.1,f5.0)
      di=di/1000.
      do=do/1000.
      s1=s1/1000.
      s2=s2/1000.
      rough=rough/1000.
c      write(6,1011) bs,L,di,do,n,n1,n2,s1,s2,
c     +  rough,coeff1,coeff2,coeffw,cndcw
1011  format(2f5.2,2f7.4,3f5.0,3f7.4,3f5.1,f5.0)

c Calculate the distance ratios, the primary air passage area
c Ap (m**2), the secondary air passage area As (m**2), the outside heat
c transfer area Ao (m**2), and the inside heat transfer area Ai (m**2)

      s=s1/s2
      Ap=n*3.1416*di*di/4.
      As=(bs-n1*do)*L
      Ao=n*3.1416*do*L
      Ai=n*3.1416*di*L

      goto 1000
c
c
cLINE 2 if itype = 2 (plate type) :
c bl   hl   L1   dL1  np   cndcw coefp coefs coefw roughp roughs
c pri  pri  sec  plt  no.   k    pri   sec  spray  pri    sec
c wid  ht   wid  thk  plts [kJ/  loss  loss  loss
c [m]  [m]  [m]  [mm]       mC]                    [mm]   [mm]
c
 800  read(5,*) bl,hl,L1,dL1,np,cndcw,
     +   coeffp,coeffs,coeffw,roughp,roughs
 102  format(4f5.2,2f5.0,2f5.2,f5.0,2f5.2)
      read(5,*)
      dL1 = dL1/1000.
      roughp = roughp/1000.
      roughs = roughs/1000.
c      write(6,1021) bl,hl,L1,dL1,np,cndcw,
c     +   coeffp,coeffs,coeffw,roughp,roughs
1021  format(4f5.2,f9.6,f5.0,f5.1,2f5.2,f5.1,2f8.4)

c Calculate the primary air passage area Ap (m**2), the secondary air
c passage area As (m**2), and the heat transfer area A (m**2):

      Ap=(bl-np*dL1)/2.*hl
      As=(bl-np*dL1)/2.*L1
      A=np*hl*L1

c Equivalent diameters of the primary and secondary air passages (dpeq
c and dseq) are:

      width=(bl-np*dL1)/(np+1.)
      dpeq=(4.*width*hl)/(2.*(width+hl))
      dseq=(4.*width*L1)/(2.*(width+L1))
c Where width=distance between two ajacent plates (surface to surface),
c m.

      goto 1000
c
cLINE 2 if itype = 3 (plate fin type) :
c bl   hl   L1   dL1  np   Ap   As   dpeq  dseq cndcw coeffp coeffs coeffw roughp roughs
c pri  pri  sec  plt  no.  pri  sec  pri   sc    k     pri    sec    spray   pri    sec
c wid  ht   wid  thk  plts area area eqd   eqd  [kJ/   loss   loss    loss
c [m]  [m]  [m] [mm]       [m2] [m2] [mm] [mm]  mC]                            [mm]   [mm]
c
 900  read(5,*) bl,hl,L1,dL1,np,Ap,As,dpeq,dseq,cndcw,
     +          coeffp,coeffs,coeffw,roughp,roughs
      read(5,*)
 103  format(4f5.2,f5.0,4f5.2,f5.0,2f5.2,f5.1,2f5.2)
      dL1=dL1/1000.
      dpeq=dpeq/1000.
      dseq=dseq/1000.
      roughp = roughp/1000.
      roughs = roughs/1000.
C      write(6,1031) bl,hl,L1,dL1,np,Ap,As,dpeq,dseq,cndcw,
C     +          coeffp,coeffs,coeffw,roughp,roughs
C1031  format(3f5.2,f8.5,f5.0,2f5.2,2f8.5,f5.0,2f5.2,f5.1,2f5.2)

C  original interactive input deleted

c Calculate the primary air passage area Ap (m**2), the secondary air
c passage area As (m**2), and the heat transger area A (m**2):
      A=np*hl*L1
c where 0.5*(np+1.0)=number of primary or sceondary air passages.
c
c Calculate the primary and secondary air velocities (vp and vs):
c
```

```
 1000 vp=Lp/Ap
      vs=Ls/As

c Assume the average entering temperatures and himidity ratios of the
c primary and secondary air:
      tpm=tpdb1
      tsm=tsdb1
      wpm=wp1
      wsm=ws1
c_______________________________________________________________________
      do 8000 i=1,2

c Calculate the average density of the primary air (rop) and that of
c the secondary air (ros):
      rop=b*(1.+wpm)/(461.5*(273.15+tpm)*(.62198+wpm))
      ros=b*(1.+wsm)/(461.5*(273.15+tsm)*(.62198+wsm))

c Calculate the mass flowrates (gp and gs), kg/s:
      gp=Lp*rop
      gs=Ls*ros

c Calculate the dynamic viscosities (mu), kg/m s; Reynolds Numbers
c (Re); heat conductivities (cndc), W/m C; Prandtl Numbers (Pr);
c and the Nusselt Numbers (Nu):

      c=1.01
c where c=specific heat of air, kJ/kg C.

      cndcp=7.6916e-5*tpm+2.4178e-2
      mup=9.80665e-6*(1.712+.0058*tpm)
      Prp=c*mup*1000./cndcp

      cndcs=7.6916e-5*tsm+2.4178e-2
      mus=9.80665e-6*(1.712+.0058*tsm)
      Prs=c*mus*1000./cndcs
c_______________________________________________________________________
      if (itype.eq.1) goto 1210
      if (itype.eq.2) goto 1240
      if (itype.eq.3) goto 1270
c_______________________________________________________________________
c     Nup=hcp*di/cndcp=0.023*Rep**0.8*Prp**).4*(1.0+(di/L)**0.7)
c     Nus=hcs*do/cndcs=0.31*Res**0.6*s**0.2
c where hc=convective heat transfer coefficient, kJ/m**2 C.

 1210 Rep=vp*di*rop/mup
      Res=vs*do*ros/mus
      hcp=cndcp/di*.023*Rep**.8*Prp**.4*(1.+(di/L)**.7)
      hcs=cndcs/do*.31*Res**.6*s**.2

      f1=exp(hcs*Ao/(gs*c*1000.))
      f2=exp(hcp*Ai/(gp*c*1000.))
      goto 2000
c_______________________________________________________________________
c     Nup=hcp*dpeq/cndcp=0.036*Rep**0.8*Prp**0.33*(dpeq/L1)**0.055
c     Nus=hcs*dseq/cndcs=0.036*Res**0.8*Prs**0.33*(dseq/h1)**0.055
c where hc=convective heat transfer coefficient, kJ/m**2 C.

 1240 Rep=vp*dpeq*rop/mup
      Res=vs*dseq*ros/mus
      hcp=cndcp/dpeq*.036*Rep**.8*Prp**.33*(dpeq/L1)**.055
      hcs=cndcs/dseq*.036*Res**.8*Prs**.33*(dseq/h1)**.055
      f1=exp(hcs*A/(gs*c*1000.))
      f2=exp(hcp*A/(gp*c*1000.))
      goto 2000
c_______________________________________________________________________
c If Re is less than 10.0, Nu1=1.68*(Re*Pr*deq/length)**0.4
c If Re is greater than 1000.0, Nu2=0.2*Re**0.67*Pr**0.4
c If Re=10.0-1000.0, Nu3=Nu1+(Nu2-Nu1)/(1000.0-10.0)*(Re-10.0)
c in above equations, length=passage length,m.  For the primary air
c passages length=L1; for the secondary air passages length=h1.

 1270 Rep=vp*dpeq*rop/mup
      Res=vs*dseq*ros/mus

      hcp=cndcp/dpeq*.036*Rep**.8*Prp**.33*(dpeq/L1)**.055
      hcs=cndcs/dseq*.036*Res**.8*Prs**.33*(dseq/h1)**.055

      f1=exp(hcs*A/(gs*c*1000.))
      f2=exp(hcp*A/(gp*c*1000.))
c_______________________________________________________________________
 2000 t1=tpdb1
      t2=tswb1
c Guess an initial value of the water temperature (which equals
c the wall surface temperature at the secondary air side) tw=tw0:
 2100 tw0=.5*(t1+t2)
      call map4(tw0,b,ww0,hw0)
```

```
      hs2=hw0-(hw0-hs1)/f1
c Calculate the wall surface temperature at the primary air side twp:
      if (itype.eq.1) goto 2400
      twp=1000.*gs*(hs2-hs1)*dL1/(A*cndcw)+tw0
      goto 2410

 2400 twp=1000.*gs*(hs2-hs1)/(n*2.*3.1416*cndcw*l)*alog(do/di)+tw0
 2410 call map4(twp,b,wwp,hwp)
      call map0(tpdb1,b,wp1,RHp1,tpdp1)

      if (tpdp1.lt.twp) goto 2450
      hs20=hs1+gp/gs*c*(hp1-hwp)*(1.-1./f2)
      goto 2460

 2450 hs20=hs1+gp/gs*c*(tpdb1-twp)*(1.-1./f2)
c_____________________________________________________________________
 2460 if(abs(hs2-hs20).lt..15) goto 2500
      if(hs2.gt.hs20) goto 2510
      if(hs2.lt.hs20) goto 2520

 2510 if(tpdb1.lt.tswb1) goto 2515
      t1=tw0
      goto 2100
 2515 t2=tw0
      goto 2100

 2520 if(tpdb1.lt.tswb1) goto 2525
      t2=tw0
      goto 2100
 2525 t1=tw0
      goto 2100

 2500 tw=tw0
c_____________________________________________________________________
      if(tpdp1.lt.twp) goto 2530
      hp2=hwp+(hp1-hwp)/f2
      wp2=wwp+(wp1-wwp)/f2
      tpdb2=(hp2-2500.*wp2)/(1.01+1.84*wp2)
      goto 2540

 2530 wp2=wp1
      tpdb2=twp+(tpdb1-twp)/f2
      hp2=1.01*tpdb2+wp2*(2500.+1.84*tpdb2)

 2540 ws2=ww0-(ww0-ws1)/f1
      tsdb2=(hs2-2500.*ws2)/(1.01+1.84*ws2)

      tpm=.5*(tpdb1+tpdb2)
      tsm=.5*(tsdb1+tsdb2)
      wpm=.5*(wp1+wp2)
      wsm=.5*(ws1+ws2)

 8000 continue
c_____________________________________________________________________
      call map30(tpdb2,hp2,wp2,b,tpwb2)
      call swvp(tpdb2,etp2)
      rhp2=wp2*b/((wp2+.62198)*etp2)

      call map30(tsdb2,hs2,ws2,b,tswb2)
      call swvp(tsdb2,ets2)
      rhs2=ws2*b/((ws2+.62198)*ets2)
c_____________________________________________________________________
c The heat transfer effectiveness for the primary air (E) is:
      e=(tpdb1-tpdb2)/(tpdb1-tswb1)
c The water evaporation rate (dW), kg/s, is:
      dw=gs*(ws2-ws1)
c The evaporative cooling capacity (Q), kW, is:
      Q=gp*(hp1-hp2)
c_____________________________________________________________________
      if(itype.eq.1) goto 4000
      call friction(roughp,dpeq,Rep,fp)
      call friction(roughs,dseq,Res,fs)
c  Pressure drop of the primary air dpp, Pa:
      dpp=(coeffp+fp*L1/dpeq)*vp*vp*rop/2.0
c  Pressure drop of the secondary air dps, Pa:
      dps=(coeffw+coeffs+fs*h1/dseq)*vs*vs*ros/2.0
      goto 4200
c---------------------------------------------------------------------
 4000 call friction(rough,di,Rep,fp)
      coeffp=coeff1+coeff2
c  Pressure drop of the primary air dpp, Pa:
      dpp=(coeffp+fp*L/di)*vp*vp*rop/2.0
      call loss1(do,s1,s2,n2,Res,coeffs)
c  Pressure drop of the secondary air dps, Pa:
      dps=(coeffw+coeffs)*vs*vs*ros/2.0
c---------------------------------------------------------------------
```

```
 4200 powerp=dpp*Lp/eta
      powers=dps*Ls/eta
c  where powerp, powers----powers required to overcome the hydraulic
c  resistances to the primary air and the secondary air, respectively.
c  The coefficient of performance of the indirect evaporative cooler is
      COP=Q*1000.0/(powerp+powers)
c-----------------------------------------------------------------------
      write(6,3001) tpdb1,tpwb1,rhp1,tpdb2
c      write(6,3000) tpdb1,tpwb1,rhp1,wp1,hp1
c      write(6,3100) tpdb2,tpwb2,rhp2,wp2,hp2
c      write(6,3200) tsdb1,tswb1,rhs1,ws1,hs1
c      write(6,3300) tsdb2,tswb2,rhs2,ws2,hs2
      write(6,3400) b,lp,ls,ap,as
      write(6,3500) vp,vs,ao,ai,a
      write(6,3600) gp,gs,tw,twp
      write(6,3700) Rep,Res,hcp,hcs
      write(6,3800) e,dw,q
      write(6,3900) dpp,dps,powerp,powers,COP
      go to 50
 3001 format('DBTin=',f5.2,' WBTin=',f5.2,' RHin=',f5.3,' DBTout=',f5.2)
 3000 format('tpdb1=',f5.2,2x,'tpwb1=',f5.2,2x,'RHp1=',f5.3,3x,
     1'wp1=',f7.5,2x,'hp1=',f7.3)
 3100 format('tpdb2=',f5.2,2x,'tpwb2=',f5.2,2x,'RHp2=',f5.3,3x,
     1'wp2=',f7.5,2x,'hp2=',f7.3)
 3200 format('tsdb1=',f5.2,2x,'tswb1=',f5.2,2x,'RHs1=',f5.3,3x,
     1'ws1=',f7.5,2x,'hs1=',f7.3)
 3300 format('tsdb2=',f5.2,2x,'tswb2=',f5.2,2x,'RHs2=',f5.3,3x,
     1'ws2=',f7.5,2x,'hs2=',f7.3)
 3400 format('B=',f8.1,3x,'Lp=',f6.3,4x,'Ls=',f6.3,4x,
     1'Ap=',f7.4,3x,'As=',f7.4)
 3500 format('vp=',f5.2,5x,'vs=',f5.2,5x,'Ao=',f7.2,3x,
     1'Ai=',f7.2,3x,'A=',f7.2)
 3600 format('gp=',f6.3,4x,'gs=',f6.3,4x,'tw=',f5.2,5x,
     1'twp=',f5.2)
 3700 format('Rep=',f8.1,1x,'Res=',f8.1,1x,'hcp=',f6.2,3x,
     1'hcs=',f6.2)
 3800 format('E=',f5.3,6x,'dW=',f7.4,3x,'Q=',f7.3)
 3900 format('dpp=',f7.2,2x,'dps=',f6.0,3x,'powerp=',f6.1,1x,
     1        'powers=',f6.1,1x,'COP=',f7.2)
  999 continue
      close(1)
      end

      subroutine map6(tdbs,twbs,bs,ws,hs,rhs)

c This subroutine is used to calculate the humidity ratio(ws), enthalpy(hs)
c and the relative humidity(RS).  Given: dry bulb temperature(tdbs), wet
c bulb temperature(twbs) and barometric pressure(Bs,Pa).

      call swvp(twbs,ewbs)

c ewbs=water vapor pressure of saturated air with the temperature of twbs,
c      Pa.

      wwbs=.62198*ewbs/(bs-ewbs)

c wwbs=humidity ratio of saturated air with the temperature of twbs, kg/kg.

      ws=(wwbs*(2500.-2.35*twbs)-1.01*(tdbs-twbs))/(2500.+1.85*tdbs
     1-4.19*twbs)
      hs=1.01*tdbs+ws*(2500.+1.84*tdbs)

      call swvp(tdbs,edbs)

c edbs=water vapor pressure of the air, Pa.

      rhs=(ws*bs)/((.62198+ws)*edbs)
      return
      end

      subroutine map4(tws,bs,wws,hws)

c This subroutine can be used to calculate the parameters of the
c saturated air when its temperature is given.
c     tws=temperature of the saturated air, C
c     Bs =barometric pressure, Pa
c     wws=humidity ratio, kg/kg
c     hws=enthalpy, kJ/kg
c______________________________________________________________

      call swvp(tws,ews)
      wws=.62198*ews/(bs-ews)
      hws=1.01*tws+wws*(2500.+1.84*tws)
      return
      end
```

```
      subroutine map30(tdbs,hs,ws,bs,twbs)

c This subroutine is to calculate the wet-bulb temperature when other
c parameters of the air are known.
c     tdbs=dry-bulb temperature, C
c     hs  =enthalpy, kJ/kg
c     ws  =humidity ratio, kg/kg
c     Bs  =barometric pressure, Pa
c     twbs=wet-bulb temperature, C
c_______________________________________________________________________
      y=alog(hs/4.1868)
c_______________________________________________________________________
c Guess an initive value of the wet-bulb temperature: twbs=twbs00

      if(hs.gt.27.) goto 10
      twbs00=-3.171+2.5641*y+1.2776*y*y+.45779*y*y*y
      goto 20
c_______________________________________________________________________
 10   twbs00=6.0163-11.061*y+8.1178*y*y-.70713*y*y*y
c_______________________________________________________________________
 20   twbs0=tdbs-(tdbs-twbs00)*101325./bs
c_______________________________________________________________________
 30   call swvp(twbs0,ewbs0)
      wwbs0=.62198*ewbs0/(bs-ewbs0)
      ws0=(wwbs0*(2500.-2.35*twbs0)-1.01*(tdbs-twbs0))/
     1(2500.+1.84*tdbs-4.19*twbs0)
c_______________________________________________________________________
      if(abs(ws-ws0).lt..00003) goto 100
      if(ws.gt.ws0) goto 50
      if(ws.lt.ws0) goto 60
c_______________________________________________________________________
 50   twbs0=twbs0+.01
      goto 30
c_______________________________________________________________________
 60   twbs0=twbs0-.01
      goto 30
c_______________________________________________________________________
 100  twbs=twbs0
      return
      end

      subroutine swvp(t,p)
      a=(273.15+t)/273.16
      a1=10.79574*(1.-1./a)
      a2=5.028*alog10(a)
      a3=1.50475e-4*(1.-10.**(-8.2969*(a-1.)))
      a4=.42873e-3*(10.**(4.76955*(1.-1./a))-1)
      p=101325.*10.**(a1-a2+a3+a4-2.2195768)
      return
      end

c  This subroutine calculates friction loss factor
c  It reads the roughness (rough), hydralic diameter (d), Reynolds's
c  Number (Re), and returns the friction loss coefficient (f).
c-----------------------------------------------------------------------
      subroutine friction(r,d,Re,f0)
      f0=0.05
400   Y1=2.0*r/d+18.7/(Re*f0**0.5)
      Y2=1.74-2.0*alog10(Y1)
      f1=1.0/Y2**2.0
c      write(6,1323) f1,f0
C1323  format ('f1,f0 ',2f15.5)
      if (abs(f1-f0).lt.0.001) goto 410
      f0=f1
      goto 400
410   continue
      return
      end

      subroutine loss1(dob,s1b,s2b,n2b,Reb,coeffb)

c  This subroutine is made for calculating the pressure drop of the air
c  flow passing through a tube bundle.
c      dob---outside diameter of tubes, m
c      s1b---tube distance (center to center) in a row, m
c      s2b---row distance (center to center), m
c      n2b---number of rows
c      Reb---Reynolds Number
c      coeffb---local loss coefficient of the tube bundle
c-----------------------------------------------------------------------
      implicit real(n)
      s3=(s1b*s1b/4.0+s2b*s2b)**0.5
      s4=s1b/dob
      phi=(s1b-dob)/(s3-dob)

      if(phi.gt.1.7) goto 50
```

```
      phi1=1.7-phi
      if(s4.lt.1.44) goto 10
      cs=3.2+0.66*phi1**1.5
      goto 80

10    cs=3.2+0.66*phi1**1.5+(1.44-s4)*(0.8+0.2*phi1**1.5)/0.11
      goto 80

50    if(s4.gt.3.0) goto 70
      phi2=1.0+phi
      if(s4.ge.1.44) goto 60
      cs=(0.44+1.44-s4)*phi2**2
      goto 80

60    cs=0.44*phi2**2
      goto 80

70    xi=1.83/s4**1.46
      goto 90

80    xi=cs/Reb**0.27

90    coeffb=xi*(1.0+n2b)

      return
      end
c This subroutine is used to calculate the air dew point temperature.
c Bs is the barometric pressure, Pa.

      subroutine map0(tdbs,Bs,ws,RHs,tdps)

      call swvp(tdbs,ets)
      RHs=Bs*ws/((0.62198+ws)*ets)
c ----------------------------------------------------------
      ets1=RHs*ets
      ets2=ets1
40    ys=ALOG(ets2/133.332)
      tdps2=-18.807+11.005*ys+0.88887*ys**2
c ----------------------------------------------------------
50    call swvp(tdps2,ets20)

      if(abs(ets20-ets1).lt.3.0) goto 100
      if(ets20.gt.ets1) goto 60
      if(ets20.lt.ets1) goto 70

60    tdps2=tdps2-0.01
      goto 50

70    tdps2=tdps2+0.01
      goto 50

100   tdps=tdps2
      return
      end
```

# 10. Sample results

### Sample input file for dec1

```
tdbl   wl      B      L    lm    Ac    eta       idir
 dbt humrat atmp   flow  len  area fan_eff (1=cell,
[C]   [ ]  [100Pa][m3/s] [m]  [m2]  [ ]    2=aspen)  title
-----------------------------------------------------------
40.0 0.013 1013.25 0.33  0.308 0.20 0.85     1       CELL12
40.0 0.013 1013.25 0.66  0.308 0.20 0.85     1       CELL12
40.0 0.013 1013.25 1.00  0.308 0.20 0.85     1       CELL12
40.0 0.013 1013.25 0.33  0.153 0.20 0.85     2       ASPEN6
40.0 0.013 1013.25 0.66  0.153 0.20 0.85     2       ASPEN6
40.0 0.013 1013.25 1.00  0.153 0.20 0.85     2       ASPEN6
```

### Sample output file for dec1

```
MEDIA : RIGID IMPREGNATED CELLULOSE  CASE=CELL12
 tdb1= 40.00 w1= 0.01300 twb1= 24.40 RH1= 0.281
 tdb2= 25.68 w2= 0.01881 twb2= 24.40 RH2= 0.902 B= 1013.25
 Air Flowrate L, m**3/s=  0.33   G,kg/s=  0.38
 Media Thickness lm, m= 0.3080   Face Area Ac,m**2=  0.20
 Effectiveness E =       0.918   Sensible Cooling Capacity Q,kW=  5.46
 Water Consumption,kg/s= 0.002   Pressure Drop,Pa= 20.80
 Air Velocity v,m/s=      1.65   Heat Transfer Area Ao,m**2= 21.6
 Power Demand, kW=        0.03
MEDIA : RIGID IMPREGNATED CELLULOSE  CASE=CELL12
 tdb1= 40.00 w1= 0.01300 twb1= 24.40 RH1= 0.281
 tdb2= 26.16 w2= 0.01861 twb2= 24.40 RH2= 0.868 B= 1013.25
 Air Flowrate L, m**3/s=  0.66   G,kg/s=  0.75
 Media Thickness lm, m= 0.3080   Face Area Ac,m**2=  0.20
 Effectiveness E =       0.887   Sensible Cooling Capacity Q,kW= 10.53
 Water Consumption,kg/s= 0.004   Pressure Drop,Pa= 83.18
 Air Velocity v,m/s=      3.30   Heat Transfer Area Ao,m**2= 21.6
 Power Demand, kW=        0.08
MEDIA : RIGID IMPREGNATED CELLULOSE  CASE=CELL12
 tdb1= 40.00 w1= 0.01300 twb1= 24.40 RH1= 0.281
 tdb2= 26.50 w2= 0.01848 twb2= 24.40 RH2= 0.845 B= 1013.25
 Air Flowrate L, m**3/s=  1.00   G,kg/s=  1.14
 Media Thickness lm, m= 0.3080   Face Area Ac,m**2=  0.20
 Effectiveness E =       0.866   Sensible Cooling Capacity Q,kW= 15.56
 Water Consumption,kg/s= 0.006   Pressure Drop,Pa=190.96
 Air Velocity v,m/s=      5.00   Heat Transfer Area Ao,m**2= 21.6
 Power Demand, kW=        0.24
MEDIA : ASPEN PAD                    CASE=ASPEN6
 tdb1= 40.00 w1= 0.01300 twb1= 24.40 RH1= 0.281
 tdb2= 27.47 w2= 0.01808 twb2= 24.40 RH2= 0.781 B= 1013.25
 Air Flowrate L, m**3/s=  0.33   G,kg/s=  0.38
 Media Thickness lm, m= 0.1530   Face Area Ac,m**2=  0.20
 Effectiveness E =       0.804   Sensible Cooling Capacity Q,kW=  4.76
 Water Consumption,kg/s= 0.002   Pressure Drop,Pa=190.96
 Air Velocity v,m/s=      1.65   Heat Transfer Area Ao,m**2= 10.7
 Power Demand, kW=        0.24
MEDIA : ASPEN PAD                    CASE=ASPEN6
 tdb1= 40.00 w1= 0.01300 twb1= 24.40 RH1= 0.281
 tdb2= 28.18 w2= 0.01779 twb2= 24.40 RH2= 0.738 B= 1013.25
 Air Flowrate L, m**3/s=  0.66   G,kg/s=  0.75
 Media Thickness lm, m= 0.1530   Face Area Ac,m**2=  0.20
 Effectiveness E =       0.758   Sensible Cooling Capacity Q,kW=  8.97
 Water Consumption,kg/s= 0.004   Pressure Drop,Pa=190.96
 Air Velocity v,m/s=      3.30   Heat Transfer Area Ao,m**2= 10.7
 Power Demand, kW=        0.24
MEDIA : ASPEN PAD                    CASE=ASPEN6
 tdb1= 40.00 w1= 0.01300 twb1= 24.40 RH1= 0.281
 tdb2= 28.63 w2= 0.01760 twb2= 24.40 RH2= 0.711 B= 1013.25
 Air Flowrate L, m**3/s=  1.00   G,kg/s=  1.14
 Media Thickness lm, m= 0.1530   Face Area Ac,m**2=  0.20
 Effectiveness E =       0.729   Sensible Cooling Capacity Q,kW= 13.06
 Water Consumption,kg/s= 0.005   Pressure Drop,Pa=190.96
 Air Velocity v,m/s=      5.00   Heat Transfer Area Ao,m**2= 10.7
 Power Demand, kW=        0.24
```

## Sample input file for dec2

```
 tdb1 twb1   B         H1     W1     D1     L1
  dbt  wbt  atmp    height width depth  flow
  [C]  [C]  [100Pa]  [m]     [m]    [m]   [m3/s]
------------------------------------------------
 40.0  25.0 101325. 0.30   0.30   0.20   0.5
 40.0  25.0 101325. 0.30   0.30   0.20   1.0
 40.0  25.0 101325. 0.30   0.30   0.20   1.5
 40.0  25.0 101325. 0.30   0.30   0.20   2.0
 40.0  25.0 101325. 0.30   0.30   0.20   2.5
 40.0  25.0 101325. 0.30   0.30   0.20   3.0
 40.0  25.0 101325. 0.30   0.30   0.20   3.5
 40.0  25.0 101325. 0.30   0.30   0.20   4.0
 35.0  22.5 101325. 0.60   0.60   0.4572 1.51
```

## Sample output file for dec2

```
HT[m]=0.300  WD[m]=0.300  DP[mm]=200.0  L1[m3/s]=0.500
dbe[C]= 40.0  wbe[C]= 25.0  w1= 0.0137  B[Pa]= 101325.
dbl[C]= 29.3  w2= 0.0180 h2o[g/s]= 2.602  E= 0.713
HT[m]=0.300  WD[m]=0.300  DP[mm]=200.0  L1[m3/s]=1.000
dbe[C]= 40.0  wbe[C]= 25.0  w1= 0.0137  B[Pa]= 101325.
dbl[C]= 30.1  w2= 0.0177 h2o[g/s]= 4.799  E= 0.658
HT[m]=0.300  WD[m]=0.300  DP[mm]=200.0  L1[m3/s]=1.500
dbe[C]= 40.0  wbe[C]= 25.0  w1= 0.0137  B[Pa]= 101325.
dbl[C]= 30.6  w2= 0.0175 h2o[g/s]= 6.866  E= 0.628
HT[m]=0.300  WD[m]=0.300  DP[mm]=200.0  L1[m3/s]=2.000
dbe[C]= 40.0  wbe[C]= 25.0  w1= 0.0137  B[Pa]= 101325.
dbl[C]= 30.9  w2= 0.0174 h2o[g/s]= 8.852  E= 0.607
HT[m]=0.300  WD[m]=0.300  DP[mm]=200.0  L1[m3/s]=2.500
dbe[C]= 40.0  wbe[C]= 25.0  w1= 0.0137  B[Pa]= 101325.
dbl[C]= 31.1  w2= 0.0173 h2o[g/s]=10.780  E= 0.592
HT[m]=0.300  WD[m]=0.300  DP[mm]=200.0  L1[m3/s]=3.000
dbe[C]= 40.0  wbe[C]= 25.0  w1= 0.0137  B[Pa]= 101325.
dbl[C]= 31.3  w2= 0.0172 h2o[g/s]=12.663  E= 0.579
HT[m]=0.300  WD[m]=0.300  DP[mm]=200.0  L1[m3/s]=3.500
dbe[C]= 40.0  wbe[C]= 25.0  w1= 0.0137  B[Pa]= 101325.
dbl[C]= 31.5  w2= 0.0172 h2o[g/s]=14.510  E= 0.569
HT[m]=0.300  WD[m]=0.300  DP[mm]=200.0  L1[m3/s]=4.000
dbe[C]= 40.0  wbe[C]= 25.0  w1= 0.0137  B[Pa]= 101325.
dbl[C]= 31.6  w2= 0.0171 h2o[g/s]=16.326  E= 0.560
HT[m]=0.600  WD[m]=0.600  DP[mm]=457.2  L1[m3/s]=1.510
dbe[C]= 35.0  wbe[C]= 22.5  w1= 0.0119  B[Pa]= 101325.
dbl[C]= 23.0  w2= 0.0168 h2o[g/s]= 8.849  E= 0.962
```

## Sample input file for idec

```
LINE1 PSYCHROMETRICS, EC SIZE AND TYPE
 tpdbl tpwbl tsdbl tswbl  B     Lp     Ls    eta   ITYPE
  pri   pri   sec   sec  atm   pri   sec    fan   1-TUBE
  dbt   wbt   dbt   wbt  pres flow  flow   eff   2-PLATE
  [C]   [C]   [C]   [C]  [Pa][m3/s][m3/s]        3-FIN
LINE 2 IF ITYPE - 1 (TUBE TYPE) :
  bs   L    di   do   n   nl   n2    s1   s2 rough coeff1coeff2coeffwcndcw
 inl  inl  tub  tub  tub  num        dis/ row ness  loss  loss  loss   k
 pht  len   id   od  no  /row rows   row  dis       to-t  fr-t  spray [kJ/
 [m]  [m] [mm] [mm]                 [mm] [mm] [mm]                     mC]
LINE 2 IF ITYPE - 2 (PLATE TYPE) :
 bl   h1   L1   dL1  np  cndcw coefp coefs coefw roughp roughs
 pri  pri  sec  plt  no.   k    pri   sec  spray   pri    sec
 wid  ht   wid  thk  plts [kJ/  loss  loss  loss
 [m]  [m]  [m]  [mm]      mC]                     [mm]    [mm]
LINE 2 IF ITYPE - 3 (PLATE FIN TYPE) :
  bl    hl    L1   dL1   np    Ap   As    dpeq  dseq cndcw coefcoef coef rough rough
  pri   pri   sec  plt   no.   pri  sec    pri  sec    k   pri  sec h2o   pri   sec
  wid   ht    wid  thk  plts  area  area   eqd  eqd  [kJ/ loss loss loss
  [m]   [m]   [m]  [mm]       [m2]  [m2]  [mm]  [mm]  mC]
---------------------------------------------------------------------------------
  30.0  26.0   0.0   0.0 101325. 0.472 0.378  0.85 TUBE     94TONGJI
 0.636 1.276 20.0 23.0 462.  12.  42. 56.0 20.0  0.025 0.50 0.46 26.0 0.3
 ----- ----- ---- ---- ----  ---  --- ---- ----  ----- ---- ---- ---- ---
  30.0  26.0   0.0   0.0 101325. 0.472 0.378  0.80 TUBE     VARICOOL
 0.316 1.365 25.4 27.4 160.   6.  15. 52.7 45.64 0.025 0.50 0.46 26.0 0.3
 ----- ----- ---- ---- ----   --  --- ---- ----- ----- ---- ---- ---- ---
  30.0  26.0   0.0   0.0 101325. 0.472 0.378  0.85 PLATE    94TONGJI
 0.310 0.355 0.57 0.2 68. 237.  0.60  0.60  2.0 0.40 2.0
 ----- ----- ---- --- --- ----  ----  -----  --- ---- ---
  30.0  25.0   0.0   0.0 101325. 0.472 0.378  0.85 PLATEFIN ADOBE
 1.000 1.070 0.267 0.40 200. 0.2105 0.1608 4.54 6.25 0.30 1.384 6.884 2.0 0.03 0.9
 ----- ----- ----- ---- ---- ------ ------ ---- ---- ---- ----- ----- --- ---- ---
```

### Sample output file for idec

```
IDEC TYPE : TUBE       (94TONGJI)
DBTin=30.00 WBTin=26.00 RHin=0.730 DBTout=27.60
B=101325.0   Lp= 0.472    Ls= 0.378     Ap= 0.1451    As= 0.4594
vp= 3.25     vs= 0.82     Ao=  42.60    Ai=  37.04    A=    0.00
gp= 0.545    gs= 0.437    tw=26.69      twp=26.86
Rep=  4078.6 Res=  1188.8 hcp= 21.53    hcs= 30.56
E=0.601      dW= 0.0011   Q=  1.372
dpp=  21.92  dps=   27.   powerp=  12.2 powers=  11.9 COP=  56.97
IDEC TYPE : TUBE       (VARICOOL)
DBTin=30.00 WBTin=26.00 RHin=0.730 DBTout=27.99
B=101325.0   Lp= 0.472    Ls= 0.378     Ap= 0.0811    As= 0.2069
vp= 5.82     vs= 1.83     Ao=  18.80    Ai=  17.43    A=    0.00
gp= 0.545    gs= 0.437    tw=26.69      twp=26.90
Rep=  9261.5 Res=  3139.5 hcp= 32.90    hcs= 38.51
E=0.502      dW= 0.0009   Q=  1.145
dpp=  53.73  dps=   63.   powerp=  31.7 powers=  29.6 COP=  18.66
IDEC TYPE : PLATE      (94TONGJI)
DBTin=30.00 WBTin=26.00 RHin=0.730 DBTout=27.48
B=101325.0   Lp= 0.472    Ls= 0.378     Ap= 0.0526    As= 0.0845
vp= 8.97     vs= 4.47     Ao=  18.80    Ai=  17.43    A=   13.76
gp= 0.546    gs= 0.436    tw=26.94      twp=26.94
Rep=  4777.5 Res=  2389.8 hcp= 69.45    hcs= 40.80
E=0.630      dW= 0.0009   Q=  1.439
dpp= 259.91  dps=  116.   powerp= 144.3 powers=  51.6 COP=   7.34
IDEC TYPE : PLATE FIN (ADOBE    )
DBTin=30.00 WBTin=25.00 RHin=0.669 DBTout=26.36
B=101325.0   Lp= 0.472    Ls= 0.378     Ap= 0.2105    As= 0.1608
vp= 2.24     vs= 2.35     Ao=  18.80    Ai=  17.43    A=   57.14
gp= 0.547    gs= 0.438    tw=26.05      twp=26.10
Rep=   641.5 Res=   925.4 hcp= 26.20    hcs= 24.05
E=0.728      dW= 0.0015   Q=  2.077
dpp=  17.34  dps=  105.   powerp=   9.6 powers=  46.7 COP=  36.90
```

# IEA Annex 28

## Low Energy Cooling

### Subtask 2: Detailed Design Tools

## Evaporative Cooling

### Excel oriented tools

Ecole des Mines de Paris, Centre dÉnergétique
D. Marchio, M. Orphelin, S. Bech

# Contents Chapter E

## 1. Technology area

Models of three air handling units using evaporative cooling as follows:

- an AHU with direct evaporative cooling and a cooling coil
- an AHU with indirect evaporative cooling and a cooling coil
- an AHU with both direct and indirect evaporative cooling and a cooling coil.

## 2. Developed by

**Name** : Dominique Marchio, Matthieu Orphelin, Sarah Bech

**Organisation** : Ecole des Mines de Paris, Centre d'Energétique

**Address** : 60 Boulevard St. Michel, 75272 Paris Cedex 06, France

**Phone** : 33 01 40 51 91 80

**Fax** : 33 01 46 34 24 91

**E-mail** : marchio@cenerg.ensmp.fr, orphelin@cenerg.ensmp.fr

## 3. General description

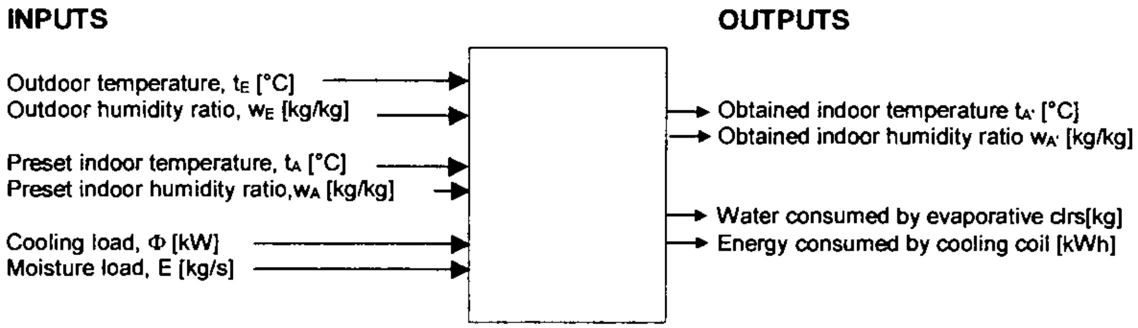
The algorithm determines the air treatment for any outdoor condition occurring throughout the year. Inputs are the outdoor and preset indoor dry bulb temperatures and humidity ratios, together with the cooling and moisture loads. Outputs are the annual water consumption of the evaporative coolers, the annual energy consumption of the cooling coil, and the indoor air conditions achieved.

The input values are the outputs of another programme, see references (Comfie, Comet). A data set consisting in hourly values of outdoor temperature, humidity ratio and loads can be used as input. Another possibility is to classify the data sets by outdoor temperature and humidity ratio to reorganise the hourly data into "boxes" with a temperature step of 3 °C and a step in humidity ratio of 2 g/kg. This classification reduces the number of data sets. For each box the mean values of the outputs are calculated, reducing the number of data sets from 8760 (the number of hours in a year) to approximately a hundred, see references (Clima 2000, 1992). In this way, the calculation can proceed using Excel work-sheets and macros.

Such a reduction of the existing data is not necessary for the application of the algorithm. It can be applied on the original 8760 data sets, but this would mean that the use of Excel work-sheets and macros would not be appropriate.

As the model only considers a fixed air renewal (fresh air) rate, the supply air mass flow rate is also constant, and the efficiency of the evaporation is assumed to be constant.

The information flow diagram of the algorithm is shown below :




# 4. Nomenclature

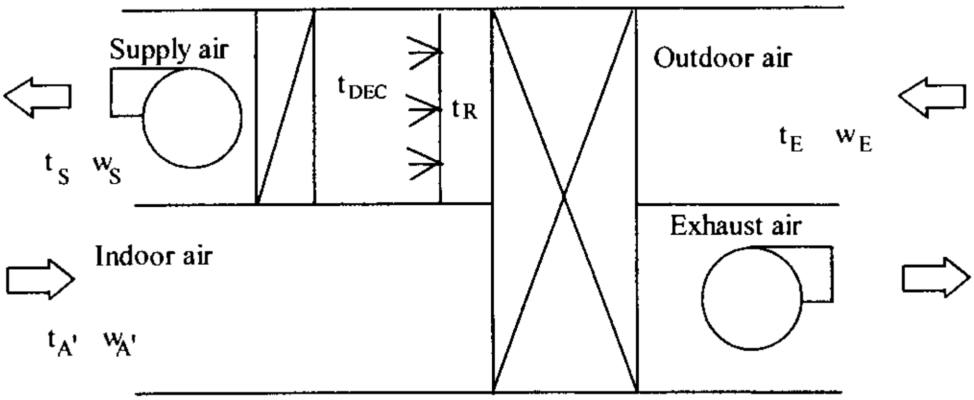
| | | |
|---|---|---|
| Φ | Cooling load | [kW] |
| E | Moisture load | [kg/s] |
| P | Power | [W] |
| n | Air mass flow rate | [kg/s] |
| t | Dry bulb temperature | [°C] |
| t' | Wet bulb temperature | [°C] |
| tr | Dew point temperature | [°C] |
| ε | Relative humidity | [%] |
| w | Humidity ratio | [kg water/kg air] |
| q' | Enthalpy | [kJ/kg] |
| m' | Water mass flow rate | [kg/h] |
| c | Specific heat | [kJ/kg/°C] |
| τ | Air renewal rate | [$h^{-1}$] |
| v' | Specific volume | [$m^3$/kg] |
| A | Latent heat of vaporization | (2501 kJ/kg) |
| η | Efficiency ratio | |
| V | Volume of the building | [$m^3$] |

**Indices:**

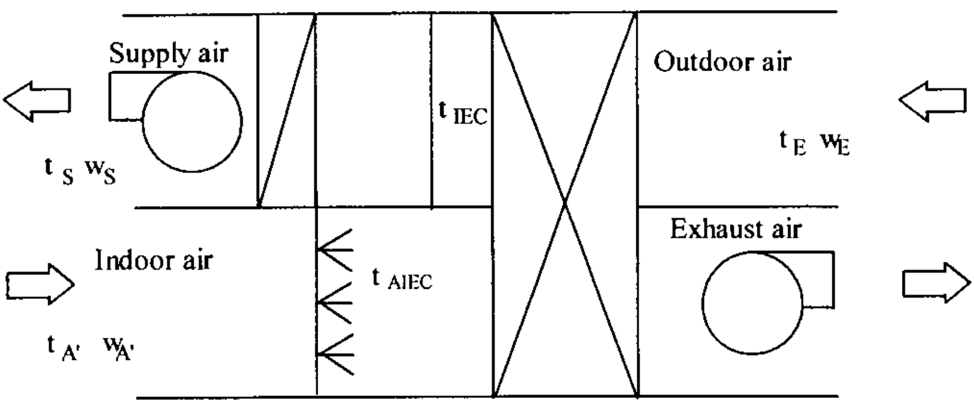
| | |
|---|---|
| E | Outdoor |
| A | Preset indoor air conditions |
| S | Preset supply air conditions |
| R | (After) heat exchanger |
| IEC | (After) indirect evaporative cooling |
| DEC | (After) direct evaporative cooling |
| AIEC | Exhaust air after indirect evaporative cooling |
| BF | (After) cooling coil |
| MS | Mean surface |
| S' | Actually obtained supply air conditions (after fan) |
| A' | Actually obtained indoor air conditions |
| a | Dry air |
| v | Water vapor |
| AN | Fresh air |

The following two diagrams of the air handling unit illustrate where in the AHU these air conditions are to be found:

**Direct evaporative cooling**




**Indirect evaporative cooling**



# 5. Mathematical description

## 5.1 Direct evaporative cooling

The standard AICVF equations for moist air calculations are applied. These equations are represented by the following numbers :

*1* $t'(t\,;w)$
*2* $w(t\,;\varepsilon)$
*3* $t(q'\,;w)$
*4* $q'(t\,;w)$

Otherwise the applied equations are given below:

**Preset supply air conditions:**

The supply air flow is all fresh air, and is characterised by the air renewal rate $\tau$.

*5* $$\Pi_S = \frac{\tau \cdot V}{3600 \cdot v'_E}$$

*6* $$q'_S = q'_A - \frac{\Phi}{\Pi_S}$$

*7* $$w_S = w_A - \frac{E}{\Pi_S}$$

*8* $$t_S = \frac{q'_S - w_S \cdot A}{c_a + w_S \cdot c_v}$$

**Heat exchanger:**

*9* $$P_R = \eta_R \cdot \Pi_S \cdot (q'_E - q'_A)$$

*10* $$q'_R = q'_E - \frac{P_R}{\Pi_S}$$

**Direct evaporative cooler:**

*11* $$t_{DEC,min} = t_R - \eta_{DEC}.(t_R - t'_R)$$

*12* $$w_{DEC,max} = w_R + \eta_{DEC} \cdot (w_R - w[t = t'_R\,;\varepsilon = 100\,\%])$$

*13* $$\eta_{DEC,real} = \frac{t_{DEC,real} - t_R}{t'_R - t_R}$$

*14* $$w_{DEC,real} = w_R + \eta_{DEC,real} \cdot (w_R - w[t = t'_R\,;\varepsilon = 100\,\%])$$

*15* $$E_{DEC} = \Pi_S \cdot (w_{DEC,real} - w_R)$$

*16* $$P_{DEC} = \Pi_S \cdot (q'_R - q'_{DEC,real})$$

**Cooling coil:**

17 $$\eta_{BF} = \frac{q'_{BF} - q'_{DEC,real}}{q'_{MS} - q'_{DEC,real}}$$

18 $$P_{BF} = \eta_{BF} \cdot (q'_{DEC,real} - q'_{MS}).\Pi_S$$

19 $$E_{BF} = \Pi_\Sigma \cdot \eta_{BF} \cdot (w_{DEC,real} - w_{MS})$$

20 $$w_{BF} = w_{DEC,real} - \frac{E_{BF}}{\Pi_S}$$

**Obtained supply air conditions:**

(We assume that the increment of the supply air temperature due to the fan is 1 °C.)

21 $$t_{S'} = t_{BF} + 1\ °C$$

22 $$w_{S'} = w_{BF}$$

**Obtained indoor air conditions:**

23 $$q'_{A'} = q'_A + (q'_{S'} - q'_S)$$

24 $$t_{A'} = t_A + (t_{S'} - t_S)$$

25 $$w_{A'} = w_A + (w_{S'} - w_S)$$

**Water consumed by direct evaporative cooler:**

26 $$\sum_{hours} E_{DEC}$$

**Energy consumed by cooling coil:**

27 $$\sum_{hours} P_{BF}$$

## 5.2 Indirect evaporative cooling

**Indirect evaporative cooler / heat exchanger:**

28 $t_{AIEC,min} = t_A - \eta_{IEC}.(t_A - t'_A)$

29 $w_{AIEC,max} = w_A + \eta_{IEC} \cdot (w_A - w[t = t'_A ; \varepsilon = 100\ \%])$

30 $t_{IEC,min} = t_E - \eta_R.(t_E - t'_{AIEC,min})$

31 $$\eta_{IEC,real} = \frac{t_{AIEC,real} - t_A}{t'_A - t_A}$$

32 $w_{AIEC,real} = w_A + \eta_{IEC,real} \cdot (w_A - w[t = t'_A ; \varepsilon = 100\ \%])$

33 $E_{IEC} = \Pi_S \cdot (w_{AIEC,real} - w_A)$

34 $P_{IEC} = \eta_R \cdot \Pi_S \cdot (q'_E - q'_{AIEC,real})$

**Cooling coil:**

35 $$\eta_{BF} = \frac{q'_{BF} - q'_{IEC,real}}{q'_{MS} - q'_{IEC,real}}$$

36 $P_{BF} = \eta_{BF} \cdot (q'_{IEC,real} - q'_{MS}).\Pi_S$

37 $E_{BF} = \Pi_S \cdot \eta_{BF} \cdot (w_{IEC,real} - w_{MS})$

38 $w_{BF} = w_{IEC,real} - E_{BF}$

**Obtained supply air and indoor air conditions:**

(See 5.1 Direct evaporative cooling.)

**Water consumed by indirect evaporative cooler:**

39 $$\sum_{hours} E_{IEC}$$

**Energy consumed by cooling coil:**

40 $$\sum_{hours} P_{BF}$$

## 5.3 Direct and indirect evaporative cooling

**Indirect evaporative cooler / heat exchanger:**

(See 5.2 Indirect evaporative cooling.)

**Direct evaporative cooler:**

*41* $t_{DEC,min} = t_{IEC,real} - \eta_{DEC}.(t_{IEC,real} - t'_{IEC,real})$

*42* $w_{DEC,max} = w_{IEC,real} + \eta_{DEC} \cdot (t_{IEC,real} - t'_{IEC,real})$

*43* $$\eta_{DEC,real} = \frac{t_{DEC,real} - t_{IEC,real}}{t'_{IEC,real} - t_{IEC,real}}$$

*44* $w_{DEC,real} = w_{IEC,real} + \eta_{DEC,real} \cdot (t_{IEC,real} - t'_{IEC,real})$

*45* $E_{DEC} = \Pi_S \cdot (w_{DEC,real} - w_{IEC,real})$

*46* $P_{DEC} = \Pi_S \cdot (q'_{IEC,real} - q'_{DEC,real})$

**Cooling coil:**

*17* $$\eta_{BF} = \frac{q'_{BF} - q'_{DEC,real}}{q'_{MS} - q'_{DEC,real}}$$

*18* $P_{BF} = \eta_{BF} \cdot (q'_{DEC,real} - q'_{MS}).\Pi_S$

*19* $E_{BF} = \Pi_\Sigma \cdot \eta_{BF} \cdot (w_{DEC,real} - w_{MS})$

*20* $w_{BF} = w_{DEC,real} - E_{BF}$

**Obtained supply air and indoor air conditions:**

(See 5.1 Direct evaporative cooling.)

**Water consumed by indirect evaporative cooler:**

*39* $$\sum_{hours} E_{IEC}$$

# 6. References


- COMFIE, a tool for bioclimatic and photovoltaic design, ISES Conference, Budapest, August 1993, Bernd Polster, Bruno Peuportier, Stéphane Biscaglia and Didier Mayer

- Clima 2000, "Air-conditioning energy consumption estimation", Roger Casari, Dominique Marchio, Sorin Stan, Ruxandra Dumitru, 1992

- Report D.E.S.S. Thermique et Régulation : Maîtrise des consommations de climatisation, Sarah Bech, Centre d'Energétique, Ecole des Mines de Paris, September 1996.

- COMET, "Cometres : a predesign tool for the improvement of summer comfort", European conference on energy and indoor air quality, Lyon, 24-26 Nov. 1995

# 7. Algorithms

## 7.1 Direct evaporative cooling

- Fixed air renewal rate → supply air mass flow rate
- Required supply air conditions
- Power of heat exchanger
- Supply air conditions after heat exchanger
- DEC power
- Supply air conditions after DEC
- DEC water load
- Power of cooling coil
- Supply air conditions after cooling coil
- Supply air conditions after fan
- Energy and water consumptions

## 7.2 Indirect evaporative cooling

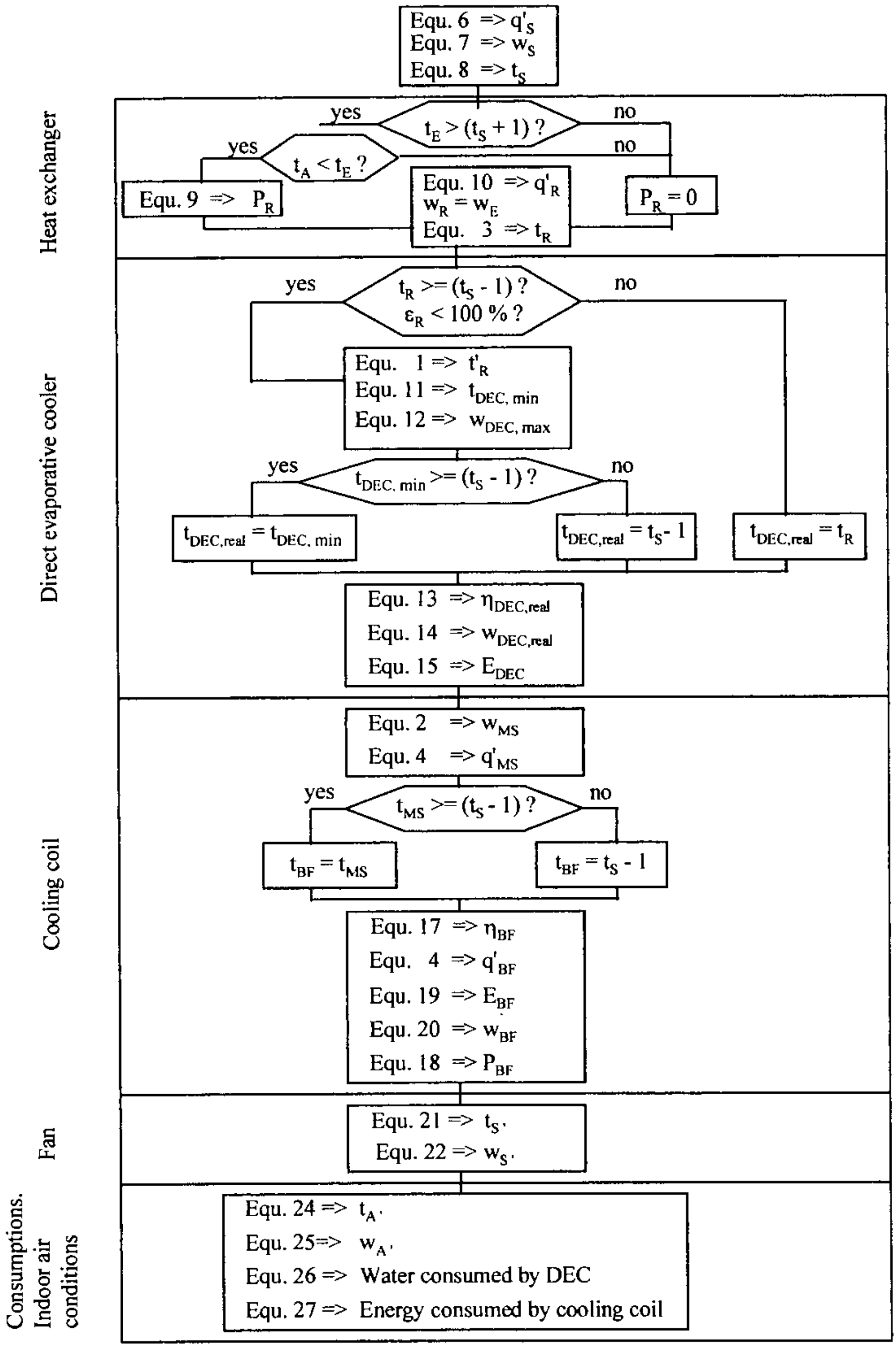
- Fixed air renewal rate → supply air mass flow rate
- Required supply air conditions
- Exhaust air conditions after IEC
- IEC water load
- Supply air conditions after IEC
- Power of IEC/heat exchanger
- Power of cooling coil
- Supply air conditions after cooling coil
- Supply air conditions after fan
- Energy and water consumptions

## 7.3 Direct and indirect evaporative cooling

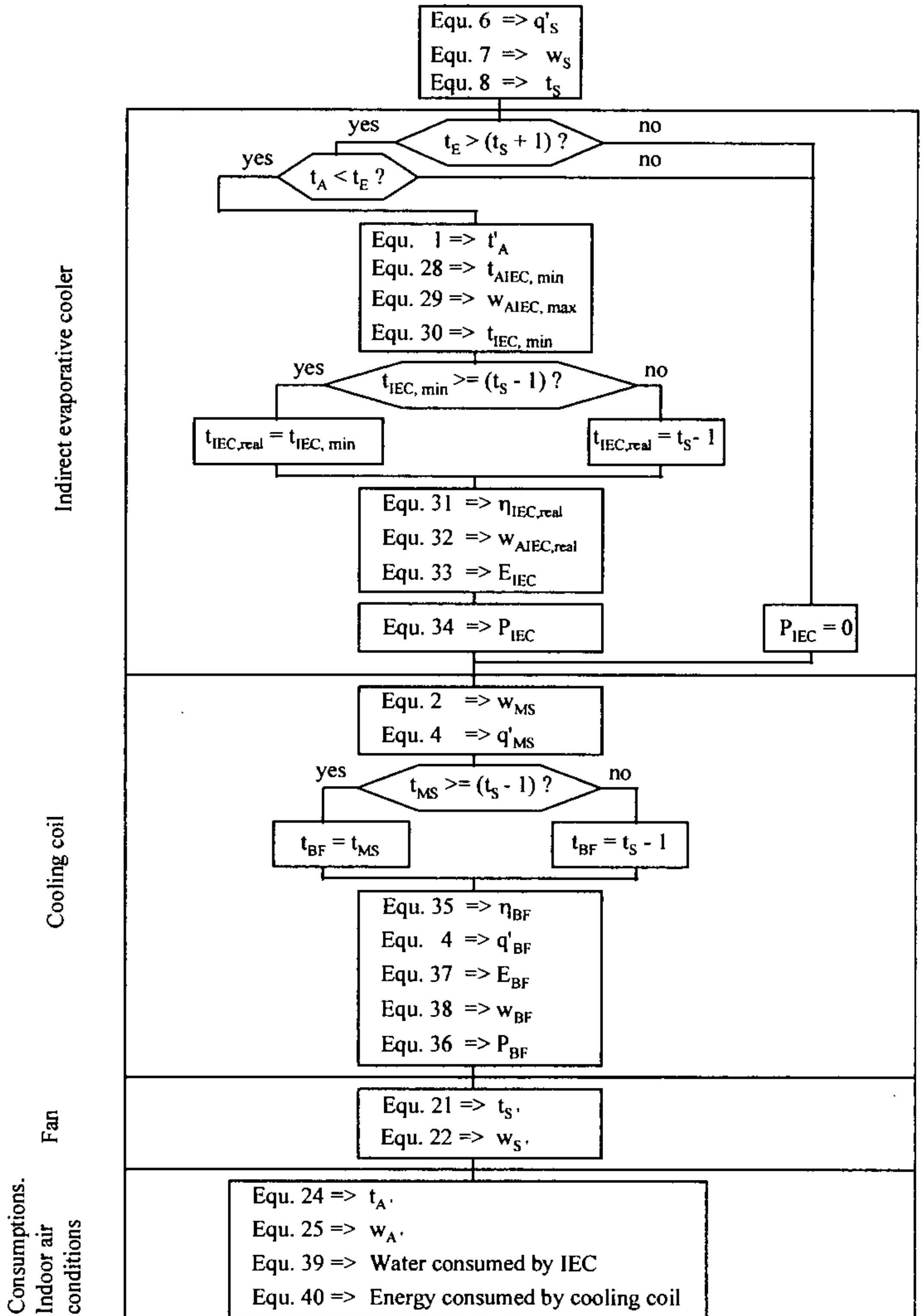
- Fixed air renewal rate → supply air mass flow rate
- Required supply air conditions
- Extract air conditions after IEC
- IEC water load
- Supply air conditions after IEC
- Power of IEC/heat exchanger
- Power of DEC
- Supply air conditions after DEC
- DEC water load
- Power of cooling coil
- Supply air conditions after cooling coil
- Supply air conditions after fan
- Energy and water consumptions

# 8. Flowcharts

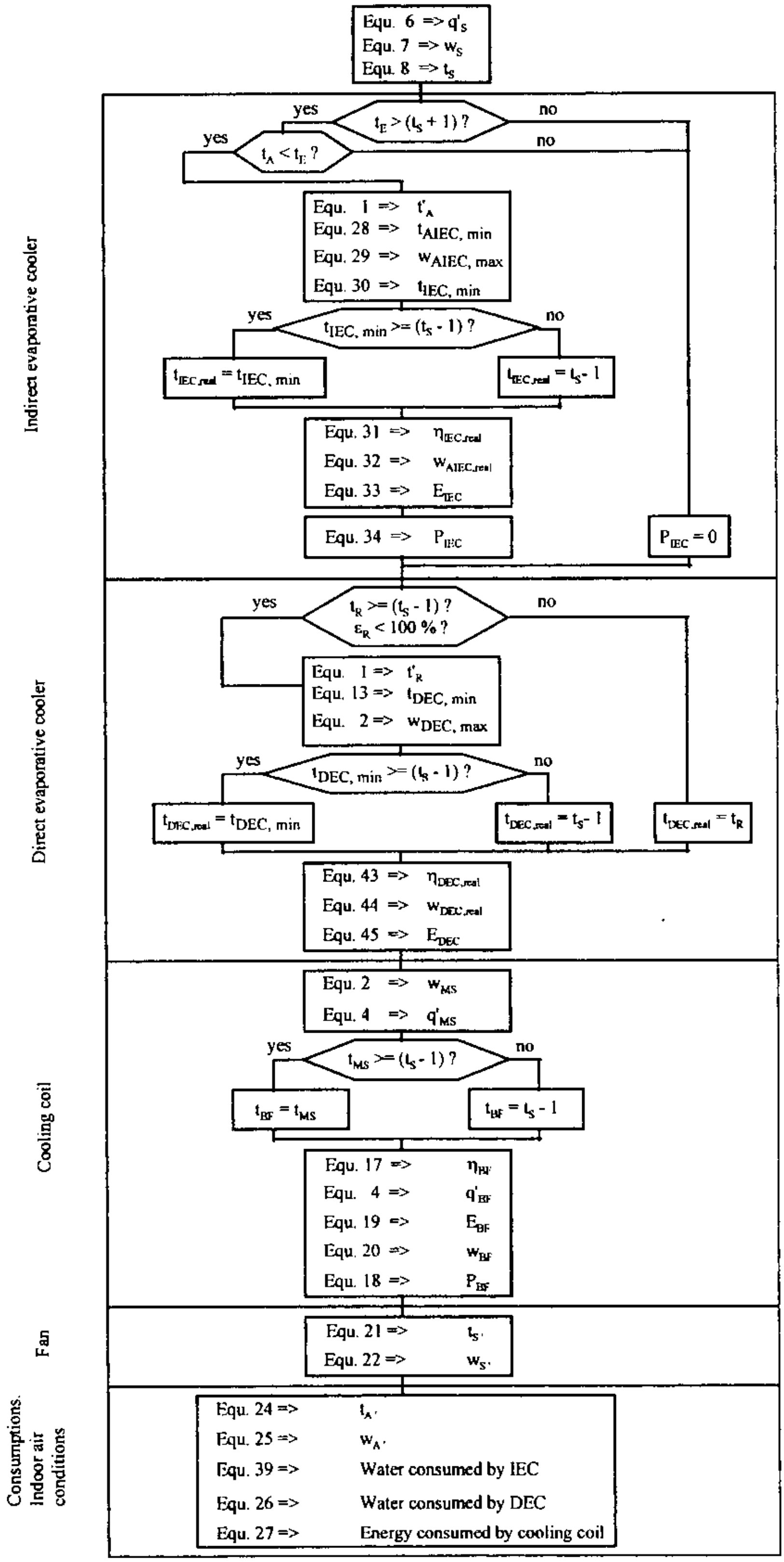
## 8.1 Direct evaporative cooling



## 8.2 Direct and indirect evaporative cooling



## 8.3 Direct and indirect evaporative cooling



# 9. Source code

These Excel Macros have been developed using a French version of Excel.

They can be used on an English (or other language) version with a few modifications, which can be automatically done using International Macros (see the Microsoft Excel User's Guide), or done by the user via about 10 instructions (select 'replace all' in the Edition menu).

If the translation is not automatically done, the translation of relevant terms from French to English is as follows.

| French | English |
|---|---|
| RESULTAT() | RESULT() |
| ARGUMENT() | ARGUMENT() |
| RETOUR() | RETURN() |
| POSER.VALEUR() | PASTE.VALUE() |
| SI() | IF() |
| SINON() | ELSE.IF() |
| FIN.SI() | END.IF() |

ASHRAE fundamentals :

**$\varepsilon$(q' ; w)**

```
eps_q_w

=RESULTAT(1)
=ARGUMENT("q";9)
=ARGUMENT("w";9)
=pv_w(w)
=t_q_w(q;w)
=pvs_temp(A8)
=A7*100/A9

=RETOUR(A10)
```

**$p_v$(w)**

```
pv_w

=RESULTAT(1)
=ARGUMENT("w";9)
=101325*w/(0,622+w)

=RETOUR(A20)
```

**$p_{vs}$(t)**

```
pvs_temp

=RESULTAT(1)
=ARGUMENT("temp";9)
=-5800,2206
=1,3914993
=-0,04860239
=0,000041764768
=-0,000000014452093
=6,5459673
=temp+273,15
=A30/A36+A31+A32*A36+A33*A36^2
=A37+A34*A36^3+A35*LN(A36)
=EXP(A38)

=RETOUR(A39)
```

**q'(t ; w)**

```
q_t_w

=RESULTAT(1)
=ARGUMENT("t";9)
=ARGUMENT("w";9)
=2500800*w+(1007+w*1846)*t

=RETOUR(A50)
```

**t(q' ; w)**

```
t_q_w

=RESULTAT(1)
=ARGUMENT("q";9)
=ARGUMENT("w";9)
=(q-2500800*w)/(1007+w*1846)

=RETOUR(C7)
```

**v'(t ; $\varepsilon$)**

```
v_t_eps

=RESULTAT(1)
=ARGUMENT("t";9)
=ARGUMENT("eps";9)
=pvs_temp(t)
=0,01*eps*C18
=t+273,15
=0,622*461,51*C20/(101325-C19)

=RETOUR(C21)
```

**v'(t ; w)**

```
v_t_w

=RESULTAT(1)
=ARGUMENT("t";9)
=ARGUMENT("w";9)
=t+273,15
=461,51*(0,622+w)*(t+273,15)/101325

=RETOUR(C33)
```

**w(t ; $\varepsilon$)**

```
w_t_eps

=RESULTAT(1)
=ARGUMENT("t";9)
=ARGUMENT("eps";9)
=pvs_temp(t)
=0,01*eps*C44
=0,622*C45/(101325-C45)

=RETOUR(C46)
```

**t(w ; v')**

```
t_w_v

=RESULTAT(1)
=ARGUMENT("w";9)
=ARGUMENT("v";9)
=101325*v/461,24/(0,622+w)-

=RETOUR(E7)
```

**w(q' ; t)**

```
w_q_t

=RESULTAT(1)
=ARGUMENT("q";9)
=ARGUMENT("t";9)
=(q-t*1007)/(2500800+1846*t)

=RETOUR(E18)
```

**t' : First estimation**

```
thum_cad

= ARGUMENT("thum0";9)
= ARGUMENT("x";9)
= ARGUMENT("y";9)
= fonc
=w_t_eps(thum0;100)
=q_t_w(thum0;E30)
=x
=y
=2500,8*(E30-y)/(1,006+1,83*y)
=x-E34
= POSER.VALEUR(E29;E35)
= ATTEINDRE(E39)

= RETOUR(E29)
```

**t' :Iteration**

```
thum_dicho

= ARGUMENT("a";9)
= ARGUMENT("b";9)
= ARGUMENT("e";9)
= ARGUMENT("x";9)
= ARGUMENT("y";9)

= thum

fa=thum_cad(a;x;y)
=POSER.VALEUR(G14;G12)
=fa
fb=thum_cad(b;x;y)
=POSER.VALEUR(G17;G15)
=fb
= SI(fa=a)
= POSER.VALEUR(G10;a)
= ATTEINDRE(G53)
= FIN.SI()
= SI(fb=b)
= POSER.VALEUR(G10;b)
= ATTEINDRE(G53)
= FIN.SI()
d=0,5*(a+b)
=POSER.VALEUR(G28;G26)
=d
= SI(ABS(a-b)<e)
= POSER.VALEUR(G10;d)
=ATTEINDRE(G53)
= FIN.SI()
fd=thum_cad(d;x;y)
=POSER.VALEUR(G35;G33)
=fd
= SI(fd=d)
= POSER.VALEUR(G10;d)
= ATTEINDRE(G53)
= FIN.SI()
= SI((fa-a)*(fd-d)<0)
fb=fd
=fb
b=d
=b
= SINON()
fa=fd
=fa
a=d
=a
= FIN.SI()
=ATTEINDRE(G53)

= RETOUR(G10)
```

## 9.1 Direct evaporative cooling

$P_R$

```
=ARGUMENT("eta";9)
= ARGUMENT("pis";9)
= ARGUMENT("te";9)
= ARGUMENT("ta";9)
= ARGUMENT("ts";9)
= ARGUMENT("we";9)
= ARGUMENT("qe";9)

=res

=SI(ts>te)
=POSER.VALEUR(A11;0)
=ATTEINDRE(A29)
=SINON()
=SI(te<ta)
=POSER.VALEUR(A11;0)
=ATTEINDRE(A29)
=SINON()
=eta*pis*1009*(te-ta)
=POSER.VALEUR(A11;A21)
=SINON()
=POSER.VALEUR(A11;0)
=ATTEINDRE(A29)
=FIN.SI()
=FIN.SI()

=RETOUR(A11)
```

$t_{DEC}$

```
=ARGUMENT("tsdem";9)
=ARGUMENT("tr";9)
=ARGUMENT("wr";9)
=ARGUMENT("epsr";9)
=ARGUMENT("efhd";9)
=ARGUMENT("pis";9)

=val

=SI(tr<=tsdem)
=POSER.VALEUR(C10;tr)
=ATTEINDRE(C33)
=FIN.SI()
=SI(epsr<100)
air après DEC
=thum_dicho(0;40;0,05;tr;wr)
=tr-(efhd*(tr-C18))
=SI(C19>=tsdem)
=POSER.VALEUR(C10;C19)
=ATTEINDRE(C33)
=SINON()
=tsdem
=POSER.VALEUR(C10;C24)
=ATTEINDRE(C33)
=FIN.SI()
=SINON()
=POSER.VALEUR(C10;tr)
=ATTEINDRE(C33)
=FIN.SI()

=RETOUR(C10)
```

$w_{DEC}$

```
=ARGUMENT("tdec";9)
=ARGUMENT("tr";9)
=ARGUMENT("wr";9)
=ARGUMENT("efhd";9)

=valeur

w DEC pour eta=1
=thum_dicho(0;40;0,05;tr;wr)
=w_t_eps(E11;100)
rendement reel DEC
=(tdec-tr)/(E11-tr)
w réel après DEC
=wr+E14*(E12-wr)
=POSER.VALEUR(E8;E16)

=RETOUR(E8)
```

**Equipment**

```
=ARGUMENT("te";9)
=ARGUMENT("tr";9)
=ARGUMENT("tdec";9)
=ARGUMENT("tbf";9)

=equipement

=SI(tr<te)
=SI(tdec<tr)
=SI(tbf<tdec)
=POSER.VALEUR(G8;"échangeur+DEC+batteriefroide")
=ATTEINDRE(G48)
=SINON()
=POSER.VALEUR(G8;"échangeur+DEC")
=ATTEINDRE(G48)
=FIN.SI()
=SINON()
=SI(tbf<tr)
=POSER.VALEUR(G8;"échangeur+batteriefroide")
=ATTEINDRE(G48)
=SINON()
=POSER.VALEUR(G8;"échangeur")
=ATTEINDRE(G48)
=FIN.SI()
=FIN.SI()
=SINON()
=SI(tdec<te)
=SI(tbf<tdec)
=POSER.VALEUR(G8;"DEC+batterie froide")
=ATTEINDRE(G48)
=SINON()
=POSER.VALEUR(G8;"DEC")
=ATTEINDRE(G48)
=FIN.SI()
=SINON()
=SI(tbf<te)
=POSER.VALEUR(G8;"batterie froide")
=ATTEINDRE(G48)
=SINON()
=POSER.VALEUR(G8;"pas de traitement")
=ATTEINDRE(G48)
=FIN.SI()
=FIN.SI()
=FIN.SI()

=RETOUR(G8)
```

$P_{BF}$

```
=ARGUMENT("tmin";9)
=ARGUMENT("qr";9)
=ARGUMENT("wr";9)
=ARGUMENT("tr";9)
=ARGUMENT("ts";9)
=ARGUMENT("pis";9)

=P batt fr.

=w_t_eps(tmin;100)
=q_t_w(tmin;I12)
gamma
=I13-qr
=I12-wr
=I15/I16
soufflage obtenu
=tr-ts
=tr-tmin
=I19/I20
delta q'
=I17*I16*I21
P batt.fr.
=-pis*I23
=SI(tr<=ts)
=POSER.VALEUR(I10;0)
=ATTEINDRE(I41)
=FIN.SI()
=SI(tr>ts)
=SI(ts>tmin)
=POSER.VALEUR(I10;I25)
=ATTEINDRE(I41)
=SINON()
=-pis*I15
=POSER.VALEUR(I10;I35)
=ATTEINDRE(I41)
=FIN.SI()
=FIN.SI()

=RETOUR(I10)
```

$W_{BF}$

```
=ARGUMENT("tmin";9)
=ARGUMENT("qr";9)
=ARGUMENT("wr";9)
=ARGUMENT("qs";9)

=w batt

=w_t_eps(tmin;100)
=q_t_w(tmin;K10)
gamma
=K11-qr
=K10-wr
=K13/K14
delta q' réel
=qr-qs
delta w réel
=K17/K15
w après bf
=wr-K19
=POSER.VALEUR(K8;K21)
=ATTEINDRE(K25)

=RETOUR(K8)
```

## 9.2 Indirect evaporative cooling

**$t_{IEC}$**

```
=ARGUMENT("tsdem";9)
=ARGUMENT("te";9)
=ARGUMENT("ta";9)
=ARGUMENT("wa";9)
=ARGUMENT("efhi";9)
=ARGUMENT("efec";9)

=val

=SI(te<ta)
=POSER.VALEUR(A10;ta)
=ATTEINDRE(A29)
=SINON()
air extrait après IEC
=thum_dicho(10;40;0,05;ta;wa)
=ta-efhi*(ta-A17)
=SI(te-efec*(te-A18)>=tsdem)
=POSER.VALEUR(A10;A18)
=ATTEINDRE(A29)
=SINON()
=te-(te-tsdem)/efec
=POSER.VALEUR(A10;A23)
=ATTEINDRE(A29)
=FIN.SI()
=FIN.SI()

=RETOUR(A10)
```

**$w_{IEC}$**

```
=ARGUMENT("tiec";9)
=ARGUMENT("ta";9)
=ARGUMENT("wa";9)
=ARGUMENT("efhi";9)

=valeur

w IEC pour eta=1
=thum_dicho(0;40;0,05;ta;wa)
=w_t_eps(A42;100)
rendement reel IEC
=(tiec-ta)/(A42-ta)
w réel après IEC
=wa+A45*(A43-wa)
=POSER.VALEUR(A39;A47)

=RETOUR(A39)
```

**Equipment**

```
=ARGUMENT("ta";9)
=ARGUMENT("tiec";9)
=ARGUMENT("Pr";9)
=ARGUMENT("Pbf";9)

=equipement

=SI(Pr>0)
=SI(tiec<ta)
=SI(Pbf>0)
=POSER.VALEUR(C8;"échangeur+IEC+batterie
=ATTEINDRE(C38)
=SINON()
=POSER.VALEUR(C8;"échangeur+IEC")
=ATTEINDRE(C38)
=FIN.SI()
=SINON()
=SI(Pbf>0)
=POSER.VALEUR(C8;"échangeur+batterie froide")
=ATTEINDRE(C38)
=SINON()
=POSER.VALEUR(C8;"èchangeur")
=ATTEINDRE(C38)
=FIN.SI()
=FIN.SI()
=SINON()
=SI(Pbf>0)
=POSER.VALEUR(C8;"batterie froide")
=ATTEINDRE(C38)
=SINON()
=POSER.VALEUR(C8;"pas de traitement")
=ATTEINDRE(C38)
=FIN.SI()
=FIN.SI()

=RETOUR(C8)
```

$P_{BF}$

```
=ARGUMENT("tmin";9)
=ARGUMENT("qr";9)
=ARGUMENT("wr";9)
=ARGUMENT("tr";9)
=ARGUMENT("ts";9)
=ARGUMENT("pis";9)

=P batt fr.

=w_t_eps(tmin;100)
=q_t_w(tmin;E12)
gamma
=E13-qr
=E12-wr
=E15/E16
soufflage obtenu
=tr-ts
=tr-tmin
=E19/E20
delta q'
=E17*E16*E21
P batt.fr.
=-pis*E23
=SI(tr<=ts)
=POSER.VALEUR(E10;0)
=ATTEINDRE(E41)
=FIN.SI()
=SI(tr>ts)
=SI(ts>tmin)
=POSER.VALEUR(E10;E25)
=ATTEINDRE(E41)
=SINON()
=-pis*E15
=POSER.VALEUR(E10;E35)
=ATTEINDRE(E41)
=FIN.SI()
=FIN.SI()

=RETOUR(E10)
```

$W_{BF}$

```
=ARGUMENT("tmin";9)
=ARGUMENT("qr";9)
=ARGUMENT("wr";9)
=ARGUMENT("qs";9)

=w batt

=w_t_eps(tmin;100)
=q_t_w(tmin;G10)
gamma
=G11-qr
=G10-wr
=G13/G14
delta q' réel
=qr-qs
delta w réel
=G17/G15
w aprés bf
=wr-G19
=POSER.VALEUR(G8;G21)
=ATTEINDRE(G25)

=RETOUR(G8)
```

## 9.3 Direct and indirect evaporative cooling

$t_{IEC}$

```
=ARGUMENT("tsdem";9)
=ARGUMENT("te";9)
=ARGUMENT("ta";9)
=ARGUMENT("wa";9)
=ARGUMENT("efhi";9)
=ARGUMENT("efec";9)

=val

=SI(te<ta)
=POSER.VALEUR(A10;ta)
=ATTEINDRE(A29)
=SINON()
air extrait après IEC
=thum_dicho(10;40;0,05;ta;wa)
=ta-efhi*(ta-A17)
=SI(te-efec*(te-A18)>=tsdem)
=POSER.VALEUR(A10;A18)
=ATTEINDRE(A29)
=SINON()
=te-(te-tsdem)/efec
=POSER.VALEUR(A10;A23)
=ATTEINDRE(A29)
=FIN.SI()
=FIN.SI()

=RETOUR(A10)
```

$w_{IEC}$

```
=ARGUMENT("tiec";9)
=ARGUMENT("ta";9)
=ARGUMENT("wa";9)
=ARGUMENT("efhi";9)

=valeur

w IEC pour eta=1
=thum_dicho(0;40;0,05;ta;wa)
=w_t_eps(A42;100)
rendement reel IEC
=(tiec-ta)/(A42-ta)
w réel après IEC
=wa+A45*(A43-wa)
=POSER.VALEUR(A39;A47)

=RETOUR(A39)
```

$t_{DEC}$

```
=ARGUMENT("tsdem";9)
=ARGUMENT("tr";9)
=ARGUMENT("wr";9)
=ARGUMENT("epsr";9)
=ARGUMENT("efhd";9)
=ARGUMENT("pis";9)

=val

=SI(tr<=tsdem)
=POSER.VALEUR(C10;tr)
=ATTEINDRE(C33)
=FIN.SI()
=SI(epsr<100)
air après DEC
=thum_dicho(0;40;0,05;tr;wr)
=tr-(efhd*(tr-C18))
=SI(C19>=tsdem)
=POSER.VALEUR(C10;C19)
=ATTEINDRE(C33)
=SINON()
=tsdem
=POSER.VALEUR(C10;C24)
=ATTEINDRE(C33)
=FIN.SI()
=SINON()
=POSER.VALEUR(C10;tr)
=ATTEINDRE(C33)
=FIN.SI()

=RETOUR(C10)
```

**$w_{DEC}$**

```
=ARGUMENT("tdec";9)
=ARGUMENT("tr";9)
=ARGUMENT("wr";9)
=ARGUMENT("efhd";9)

=valeur

w DEC pour eta=1
=thum_dicho(0;40;0,05;tr;wr)
=w_t_eps(E11;100)
rendement reel DEC
=(tdec-tr)/(E11-tr)
w réel après DEC
=wr+E14*(E12-wr)
=POSER.VALEUR(E8;E16)

=RETOUR(E8)
```

**Equipment**

```
=ARGUMENT("te";9)
=ARGUMENT("tr";9)
=ARGUMENT("tdec";9)
=ARGUMENT("tbf";9)

=equipement

=SI(tr<te)
=SI(tdec<tr)
=SI(tbf<tdec)
=POSER.VALEUR(G8;"échangeur+DEC+batterie
=ATTEINDRE(G48)
=SINON()
=POSER.VALEUR(G8;"échangeur+DEC")
=ATTEINDRE(G48)
=FIN.SI()
=SINON()
=SI(tbf<tr)
=POSER.VALEUR(G8;"échangeur+batterie
=ATTEINDRE(G48)
=SINON()
=POSER.VALEUR(G8;"échangeur")
=ATTEINDRE(G48)
=FIN.SI()
=FIN.SI()
=SINON()
=SI(tdec<te)
=SI(tbf<tdec)
=POSER.VALEUR(G8;"DEC+batterie froide")
=ATTEINDRE(G48)
=SINON()
=POSER.VALEUR(G8;"DEC")
=ATTEINDRE(G48)
=FIN.SI()
=SINON()
=SI(tbf<te)
=POSER.VALEUR(G8;"batterie froide")
=ATTEINDRE(G48)
=SINON()
=POSER.VALEUR(G8;"pas de traitement")
=ATTEINDRE(G48)
=FIN.SI()
=FIN.SI()
=FIN.SI()

=RETOUR(G8)
```

$P_{BF}$

```
=ARGUMENT("tmin";9)
=ARGUMENT("qr";9)
=ARGUMENT("wr";9)
=ARGUMENT("tr";9)
=ARGUMENT("ts";9)
=ARGUMENT("pis";9)

=P batt fr.

=w_t_eps(tmin;100)
=q_t_w(tmin;I12)
gamma
=I13-qr
=I12-wr
=I15/I16
soufflage obtenu
=tr-ts
=tr-tmin
=I19/I20
delta q'
=I17*I16*I21
P batt.fr.
=-pis*I23
=SI(tr<=ts)
=POSER.VALEUR(I10;0)
=ATTEINDRE(I41)
=FIN.SI()
=SI(tr>ts)
=SI(ts>tmin)
=POSER.VALEUR(I10;I25)
=ATTEINDRE(I41)
=SINON()
=-pis*I15
=POSER.VALEUR(I10;I35)
=ATTEINDRE(I41)
=FIN.SI()
=FIN.SI()

=RETOUR(I10)
```

$w_{BF}$

```
=ARGUMENT("tmin";9)
=ARGUMENT("qr";9)
=ARGUMENT("wr";9)
=ARGUMENT("qs";9)

=w batt

=w_t_eps(tmin;100)
=q_t_w(tmin;K10)
gamma
=K11-qr
=K10-wr
=K13/K14
delta q' réel
=qr-qs
delta w réel
=K17/K15
w après bf
=wr-K19
=POSER.VALEUR(K8;K21)
=ATTEINDRE(K25)

=RETOUR(K8)
```

# 10. Classification programme

As explained previously, the classification programme allows the number of data sets to be reduced from 8760 to less than 200.

As files containing meteorological data, loads, obtained indoor conditions or other information (electrical tariff structure, occupancy, ...) *have structures which vary from one building simulation software to* another, proposing a general source code program is not feasible. Therefore a source code is proposed which can be easily adapted by the user to meet requirements.

**INPUT FILE**

**c:\inputs.txt**
The inputs are the outdoor and preset indoor dry bulb temperatures and humidity ratios, plus the enthalpic and moisture loads calculated at each time step. This file containes 8760 (or less) rows, each is composed of the following information :
te ; we ; ta ; wa ; Load_enthalp ; Load_moisture

**OUTPUT FILES**

**c:\cooling.txt**
This file contains n<100 rows, each composed of the following information :
number of the box, te box, we box, nb of hours of occurrence ; te average ; we average ; ta average; wa average; Load_enthalp average; Load_moisture average

**c:\heating.txt**
This file contains n<100 rows, each composed of the following information :
number of the box, te box, we box, nb of hours of occurrence ; te average ; we average ; ta average; wa average; Load_enthalp average; Load_moisture average

**c:\free.txt**
nb of hours of occurrence of free evolution (no heating or cooling)

These output files could then be imported as inputs to the evaporative cooling modelling.

Initialisation of matrices :

cooling ("triclim"), heating ("trichauf") : 100 boxes defined and numerated
all output values = 0

free evolution ("trilibre") : 1 box defined; value = 0

⇓

Construction of boxes :

temperature/humidity step determined
mean value of $t_E$, $w_E$ determined for each box

⇓

Input files assigned

⇓

Hourly input values read
- situation classified in appropriate matrix ("sort_cool", "sort_heat", "sort_free")
- situation ($t_E$ , $w_E$) classified in appropriate box
- number of occurrences in a year of each box and
mean value of $t_E$, $w_E$, $t_A$, $w_A$, enthalpy load and moisture load calculated

⇓

Output files assigned

⇓

Boxes occurring minimum once a year selected from all three matrices

⇒ Output values written in output files

⇓

All files closed

Language : *Turbo Pascal for Windows*

SOURCE CODE

```
Uses Wincrt;

Type
results = array [1..100,1..10] of real;
results1 = array [1..1,1..1] of real;

Var
file_inputs : text;
file_cooling,file_heating,file_free : text;
te,we,ta,wa : real;
t_adim,w_adim :real,
i,j,k,l : integer;
sort_cool :  results;
sort_heat : results;
sort_free : results1;
tmax_h,tmin_c,step_tc,pas_cf,wemax_c,wemin_c,step_we : real;
tmax_h,tmin_h,step_th,pas_cc,wemax_h,wemin_h: real;
te_adim,we_adim : real;


{                                                                              }
{sorting for cooling hours}

Procedure cooling(te,we,ta,wa,tmax_c,tmin_c,wemax_c,wemin_c : real;
                                                var sort_cool : resultats);

Var

t_adim, we_adim : real;
n_case : integer;
i,j : integer;

Begin
        {Normalised values}
        t_adim:=(te-tmin_c)*100/(tmax_h-tmin_c);
        we_adim:=we*100/wemax_c;
        {Determination of corresponding box-numbers}
        i:=trunc(t_adim/10);
        j:=trunc(we_adim/10);
        n_case:=(10*j)+i+1;
        {implementation of new mean values}
        sort_cool[n_case,4]:=sort_cool[n_case,4]+1;
        sort_cool[n_case,5]:=((sort_cool[n_case,4]-1)*sort_cool[n_case,5]+te)/(sort_cool[n_case,4]);
        sort_cool[n_case,6]:=((sort_cool[n_case,4]-1)*sort_cool[n_case,6]+we)/(sort_cool[n_case,4]);
        sort_cool[n_case,7]:=((sort_cool[n_case,4]-1)*sort_cool[n_case,7]+ta)/(sort_cool[n_case,4]);
        sort_cool[n_case,8]:=((sort_cool[n_case,4]-1)*sort_cool[n_case,8]+wa)/(sort_cool[n_case,4]);
        sort_cool[n_case,9]:=((sort_cool[n_case,4]-
                1)*sort_cool[n_case,9]+(Load_enthalp))/(sort_cool[n_case,4]);
        sort_cool[n_case,10]:=((sort_cool[n_case,4]-
1)*sort_cool[n_case,10]+Load_moisture)/(sort_cool[n_case,4]);
End;


{                                                                              }
{sorting for heating hours}
```

```
Procedure triagechauf(te,we,ta,wa,tmax_h,tmin_h,wemax_h,wemin_h : real;
                                                                var sort_heat : resultats);

Var
t_adim, we_adim : real;
n_case : integer;
i,j : integer;

Begin
        {Normalised values}
        t_adim:=(te-tmin_h)*100/(tmax_h-tmin_h);
        we_adim:=we*100/wemax_h;
        {Determination of corresponding box-numbers}
        i:=trunc(t_adim/10);
        j:=trunc(we_adim/10);
        n_case:=(10*j)+i+1;
        sort_heat[n_case,4]:=sort_heat[n_case,4]+1;
        sort_heat[n_case,5]:=((sort_heat[n_case,4]-1)*sort_heat[n_case,5]+te)/(sort_heat[n_case,4]);
        sort_heat[n_case,6]:=((sort_heat[n_case,4]-1)*sort_heat[n_case,6]+we)/(sort_heat[n_case,4]);
        sort_heat[n_case,7]:=((sort_heat[n_case,4]-1)*sort_heat[n_case,7]+ta)/(sort_heat[n_case,4]);
        sort_heat[n_case,8]:=((sort_heat[n_case,4]-1)*sort_heat[n_case,8]+wa)/(sort_heat[n_case,4]);
        sort_heat[n_case,9]:=((sort_heat[n_case,4]-
                        1)*sort_heat[n_case,9]+(Load_enthalp))/(sort_heat[n_case,4]);
        sort_heat[n_case,10]:=((sort_heat[n_case,4]-
                1)*sort_heat[n_case,10]+Load_moisture)/(sort_heat[n_case,4]);
End;

{________________________________________________________________________________}
{sorting for free evolution hours}

Procedure triagelibre(var trilibre : resultats1);

Begin
        sort_free[1,1]:= sort_free[1,1]+1;
End;

{________________________________________________________________________________}

Begin

        {Initialisation of the matrices "sort_cool", "sort_heat" and "sort_free"}
        for i:=1 to 100 do
        begin
                sort_cool[i,1]:=i;
                sort_heat[i,1]:=i;
                for j:=2 to 10 do
                begin
                        sort_cool[i,j]:=0;
                        sort_heat[i,j]:=0;
                end;
        end;
        begin
                sort_free[1,1]:=0;
        end;

        {Construction of boxes in "sort_cool"}
        tmax_c:=40;
        tmin_c:=10;
        step_tc:=3;
        wemax_c:=20/1000;
        wemin_c:=0;
        step_we:=2/1000;
```

```
for i:=0 to 9 do
begin
        for j:=0 to 9 do
        begin
                sort_cool[10*j+i+1,2]:=tmin_c+(2*i+1)*step_tc/2;
                sort_cool[10*j+i+1,3]:=wemin_c+(2*j+1)*step_we/2
        end;
end;

{Construction of boxes in "sort_heat"}
tmax_h:=21;
tmin_h:=-9;
step_th:=3;
wemaxc:=20/1000;
wemin_h:=0;
step_we:=2/1000;
for i:=0 to 9 do
begin
        for j:=0 to 9 do
        begin
                sort_heat[10*j+i+1,2]:=tmin_h+(2*i+1)*step_th/2;
                sort_heat[10*j+i+1,3]:=wemin_h+(2*j+1)*step_we/2
        end;
end;

Assign(file_input,'c:\inputs.txt');
reset(file_input);

for i:=1 to 8736 do
begin
        readln(file_input,te);
        readln(file_input,we);
        readln(file_input,ta);
        readln(file_input,wa);
        readln(file_input,Load_enthalp);
        readln(file_input,Load_moisture);

{Test on Load_enthalp to determine the type of considered situation between cooling, heating, and free
evolution}
        if (Load_enthalp>1) then
        begin
                cooling(te,we,ta,wa,tmax_h,tmin_c,wemax_c,wemin_c, sort_cool);
        end;
        if (Load_enthalp<-1) then
        begin
                heating(te,we,ta,wa,tmax_h,tmin_h,wemax_h,wemin_h, sort_heat);
        end;
        if (Load_enthalp<=1) and (Load_enthalp>=-1) then
        begin
                free(trilibre);
        end;
end;

{The results are written in three text files};
Assign(file_cooling,'c:\cooling.txt');
rewrite(file_cooling);
Assign(file_heating,'c:\heating.txt');
rewrite(file_heating);
Assign(file_free'c:\free.txt');
rewrite(file_free);

{Selection of the boxes with number of hours non zero}
```

```
        {cooling results}
        for i:=1 to 100 do
        begin
                if (sort_cool[i,4]>0) then
                begin
                        for j:=1 to 11 do
                        begin
                                write(file_cooling,sort_cool[i,j]);
                        end;
                        writeln(file_cooling);
                end;

        {cooling results}
                if (sort_heat[i,4]>0) then
                begin
                        for j:=1 to 11 do
                        begin
                                write(file_heating, sort_heat[i,j]);
                        end;
                        writeln(file_heating);
                end;
        end;

        {free evolution results}
        begin
                write(file_free,sort_free[1,1]);
        end;
        writeln(file_free);

{End of the program}

        close(file_inputs);
        close(file_cooling);
        close(file_heating);
        close(file_free);

End.
```

# IEA Annex 28

## Low Energy Cooling

### Subtask 2: Detailed Design Tools

# Displacement Ventilation and Chilled Ceiling Multi-node Model

Loughborough University, Loughborough (UK)

Simon J. Rees, Philip Haves

# Contents Chapter F

## 1. Technology area

A multi-node model of a displacement ventilation and chilled ceiling system.

## 2. Developed by

**Name** : **Simon J. Rees & Philip Haves**

**Organisation** : **Loughborough University**

**Address** : **Loughborough University**
**Department of Civil & Building Engineering**
**Loughborough**
**Leicestershire**
**LE11 3TU**
**UK**

**Phone** : **+44 (0)1509 263171**

**Fax** : **+44(0)1509 263981**

**E-mail** : **S.J.Rees@lboro.ac.uk, P.Haves@lboro.ac.uk**

## 3. General description

The purpose of the model is to allow the simulation of the bulk air movement, convective and radiant heat transfer that occurs in rooms with displacement ventilation and chilled ceilings. The model has been developed using the nodal modelling program LIGHTS (Sowell 1989) as a prototyping environment and is defined by a network of room air and surface nodes and associated conductances along with the room geometry and loads. The model has also been implemented in the HVACSIM+ simulation environment using the HSLIGHTS program (Sowell 1991).

The model consists of a series of air nodes and associated capacity rates which describe the bulk air movement from the supply air terminal and the development of a plume over the load extending to ceiling level. Some recirculation of the air as it flows from the plume, across the ceiling, and down part of the walls is also represented. The model divides the room in to four horizontal layers to capture the effects of vertical temperature gradients. The model in its current state of development treats a limited number of load types.

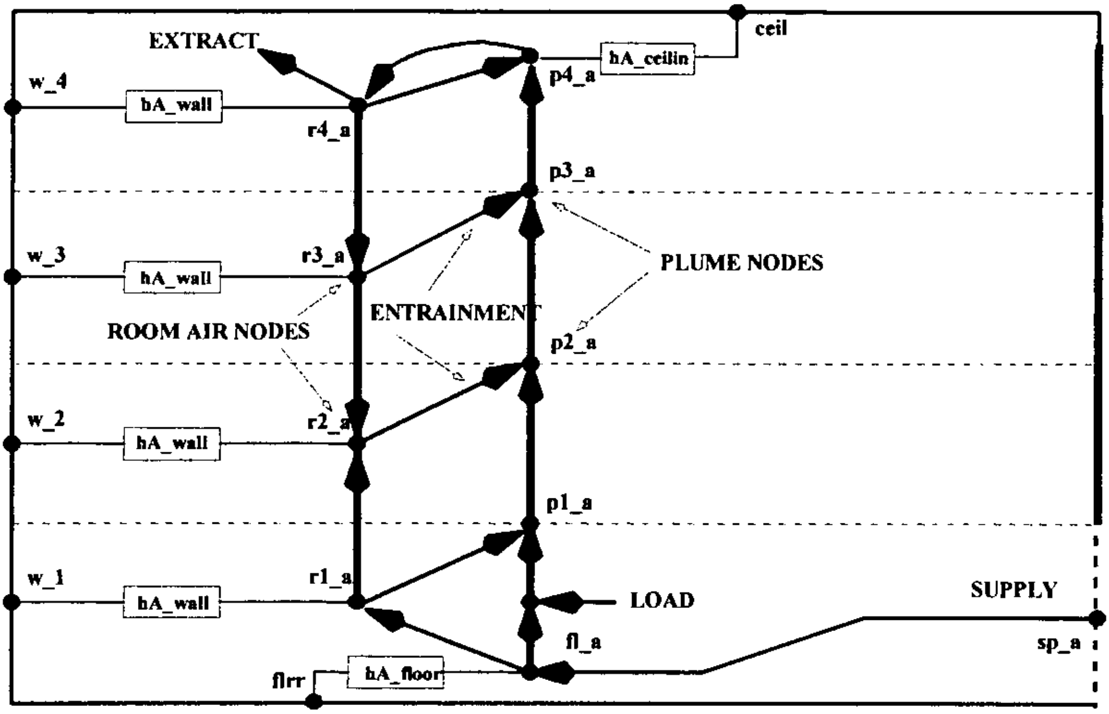



**Figure 1** : Schematic diagram of the model and the nodal network

The air flow over the load is divided between two sets of air nodes, one representing air in the plume, and the second representing the surrounding air in the room (see figure 1). Flows between these two sets of air nodes represent the entrainment of air from the room into the plume. Convective heat transfer is represented by conductances between these surrounding room air nodes and wall surface nodes at each level, along with conductances between the floor and ceiling surfaces and adjacent air nodes.

Only the internal surfaces of the wall, floor and ceiling are included in the model, allowing the conduction and thermal mass of the room fabric to by treated by other models. This allows a flexible approach, so that a number of instances of the zone model could be linked to different arrangements of boundary elements within a simulation. Radiation between room surface nodes is dealt with in an exact manner on the basis of view factors provided by the user (see Sowell and O"Brien 1972).

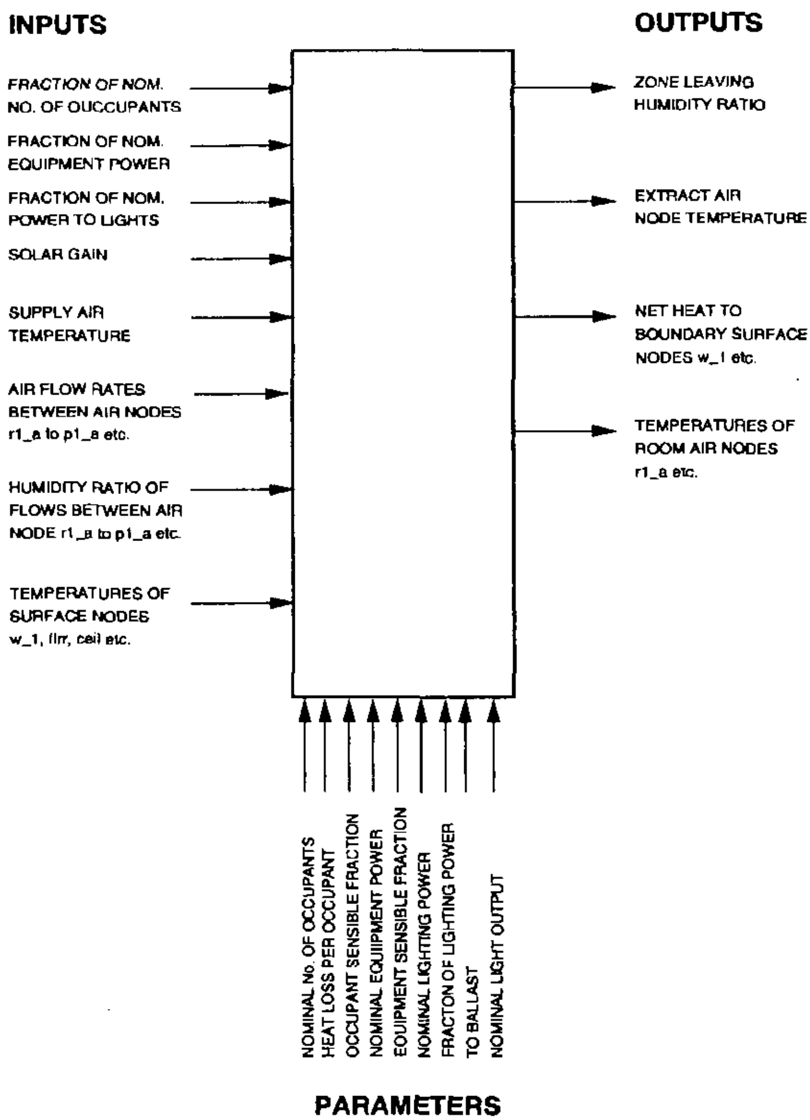



**Figure 2** Model information flow diagram

The model in the form reported here has been implemented in a nodal modelling program HSLIGHTS ( The model may be transfered to other simulation environments after further prototype development). HSLIGHTS allows the representation of convective conductances, capacity rates and radiant heat fluxes between nodes that are either air or surface nodes and is capable of solving the non-linear algebraic equations defined by the nodal network. The method of solving these equations for the node temperatures and heat fluxes is not described here. For further details of the HSLIGHTS program see Sowell 1991.

# 4. Nomenclature

Mathematical nomenclature

| Name | | Description | Units |
|---|---|---|---|
| $C_S$ | - | supply capacity rate | (W/°C) |
| $C_{e1-4}$ | - | plume entrainment capacity rates | (W/°C) |
| $C_{R1-4}$ | - | room air node capacity rates | (W/°C) |
| $C_{P1-4}$ | - | plume air node capacity rates | (W/°C) |
| $C_L$ | - | capacity rate to load | (W/°C) |
| $h_f$ | - | floor convective heat transfer coefficient | (W/m$^2$.°C) |
| $h_c$ | - | ceiling convective heat transfer coefficient | (W/m$^2$.°C) |
| $h_w$ | - | wall convective heat transfer coefficient | (W/m$^2$.°C) |
| $A_w$ | - | 0.25 x total wall area | (m$^2$) |
| $A_f$ | - | floor area | (m$^2$) |
| $A_c$ | - | ceiling area | (m$^2$) |

Node names

| | | |
|---|---|---|
| R1-4 | - | room air nodes |
| P1-4 | - | plume air nodes |
| W1-4 | - | wall surface nodes |
| S | - | supply air node |
| fla | - | floor air node |
| flr | - | floor surface node |
| ceil | - | ceiling surface node |

# 5. Mathematical description

The model is defined mathematically by the set of equations representing the heat balance at each node and in addition the mass balance at each air node. The heat and mass balances at the air nodes are defined by the following equations:

$$
\begin{aligned}
&C_S T_S + h_f A_f (T_{flr} - T_{fla}) - C_{R1} T_{fla} - C_L T_{fla} = 0 \\
&C_S - C_L - C_{R1} = 0 \\
&C_{R1} T_{fla} + h_w A_w (T_{W1} - T_{R1}) - C_{e1} T_{R1} - C_{R1} T_{R1} = 0 \\
&C_{R1} - C_{R2} - C_{e1} = 0 \\
&C_{R2} T_{R1} + h_w A_w (T_{W2} - T_{R2}) - C_{e2} T_{R2} + C_{R3} T_{R3} = 0 \\
&C_{e2} - C_{R3} - C_{e3} = 0 \\
&C_{R3} T_{R3} + h_w A_w (T_{W3} - T_{R3}) - C_{e3} T_{R3} - C_{R3} T_{R3} = 0 \\
&C_{R4} - C_{R3} - C_{e3} = 0 \\
&C_{P4} T_{P4} + h_w A_w (T_{W4} - T_{R4}) - C_{e4} T_{R4} - C_{R4} T_{R4} - C_S T_{R4} = 0 \\
&C_{P4} - C_{R4} - C_{e4} - C_S = 0 \\
&C_{e1} T_{R1} + C_L T_{fla} + Q - C_{P1} T_{P1} = 0 \\
&C_{P1} - C_{e1} - C_L = 0 \\
&C_{P1} T_{P1} - C_{e2} T_{R2} - C_{P2} T_{P2} = 0 \\
&C_{P2} - C_{P1} - C_{e2} = 0 \\
&C_{P2} T_{P2} + C_{e3} T_{R3} - C_{P3} T_{P3} = 0 \\
&C_{P2} - C_{P1} - C_{e2} = 0 \\
&C_{P2} T_{P2} + C_{e3} T_{R3} - C_{P3} T_{P3} = 0 \\
&C_{P3} - C_{e3} - C_{P2} = 0 \\
&C_{P3} T_{P3} + C_{e4} T_{R4} + h_C A_C (T_{ceil} - T_{P4}) - C_{P4} T_{P4} = 0 \\
&C_{P4} - C_{P3} - C_{e4} = 0
\end{aligned}
$$

The capacity rates are the most significant parameters of the model. The supply capacity rate is an input to the model. The other capacity rates are a function of the supply capacity rate and the size of the load. In total four other capacity rate parameters need to be given - the rest being calculated from the mass balance equations. To date the capacity rates have been derived to fit experimental data. A generalised set of parameters is under development.

## 6. References

Li, Y., Fuchs, L. and Sandberg, M., 1993. *Vertical Temperature Profiles in rooms Ventilated by Displacement: Full-scale Measurement and Nodal Modelling.* Indoor Air, (2), pages 225-243.

Sowell, E.F. and O'Brien, P.F., 1972 *Efficient Computation of Radiant-Interchange Configuration Factors within the Enclosure.*, A.S.M.E. transactions, pages 326–328.

Sowell, E.F., 1989. *LIGHTS User Guide.* Department of Computer Science, California State Universtiy, CA, USA.

Sowell, E.F., 1991. *A General Zone Model for HVACSIM+: Users Manual.*, Oxford University Dept. of Engineering Science, report No. 1889/91.

## 7. Algorithm

INITIALIZATION

- Set up data structures for each instance of the zone model
- Model parameters passed from HVACSIM+ to HSLIGHTS

START TIME STEP

- HVACSIM+ pass air flows, loads and boundary temperatures to HSLIGHTS
- HSLIGHTS performs a heat balance on the nodal network for current time step
- HSLIGHTS performs a moisture balance on the air stream
- HSLIGHTS passes boundary node heat fluxes to HVACSIM+
- HSLIGHTS passes room air temperature information to HVACSIM+ along with the leaving air humidity ratio
- HVACSIM+ uses heat fluxes to calculate new wall surface temperatures etc.

START NEW TIME STEP

# 8. Flowchart

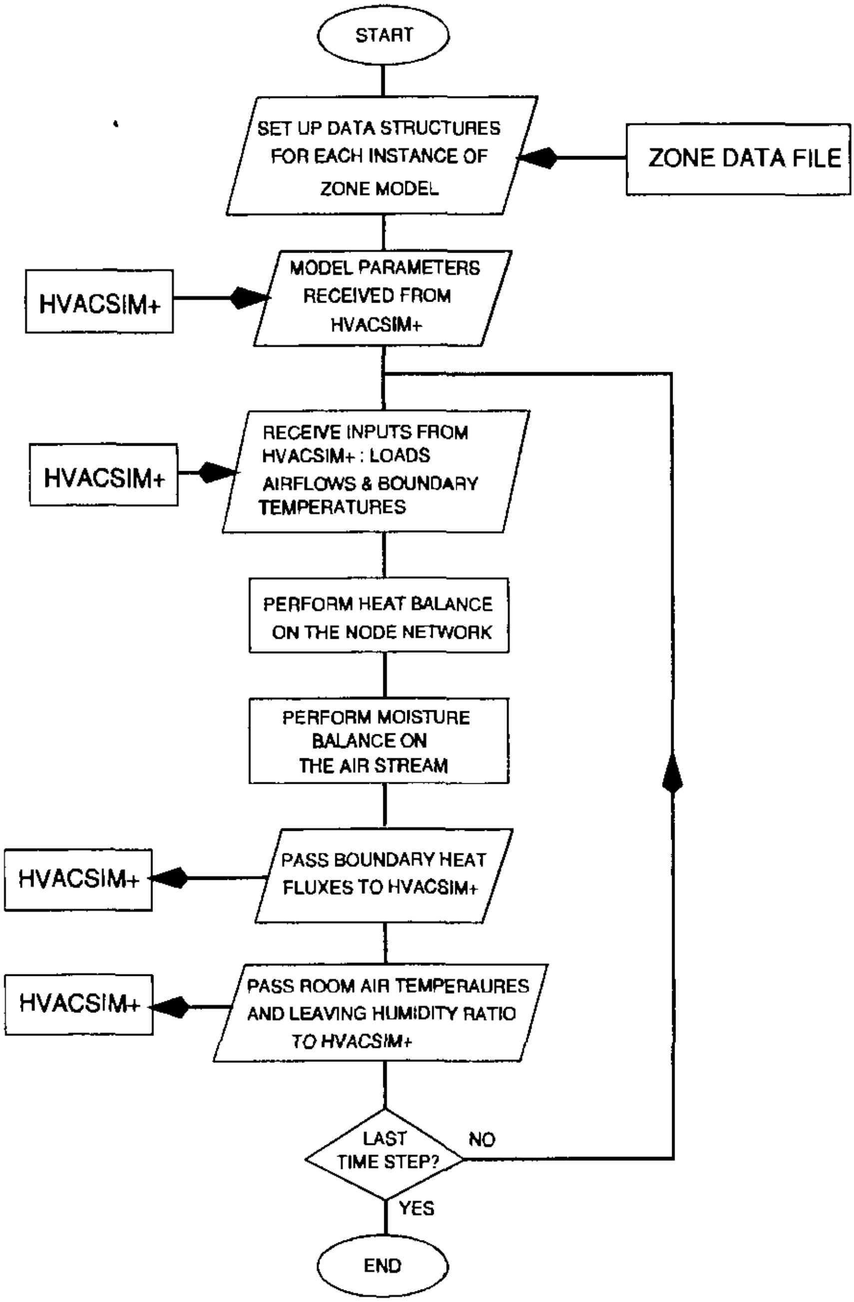



**Figure 3** Algorithm Flowchart

# 9. Source code

The source code used to implement this model as a HVACSIM+ model consists of a number of elements:

1. A FORTRAN subroutine forming the front end to HSLIGHTS which is called by HVACSIM+

2. An entry in the HVACSIM+ TYPAR.DAT file which describes the inputs, outputs and parameters of the model.

3. A HSLIGHTS zone data file which defines all the nodes, properties, heat transfer coefficients and view factors of the room model

4. The source code for the HSLIGHTS program. This is written in ANSI C.

The first three of the above are given below. The HSLIGHTS source code is omitted because of its length.

1.) The FORTRAN Subroutine front end to the HSLIGHTS program

```
SUBROUTINE TYPE288(XIN,OUT,PAR,SAVED,IOSTAT)
C
C *******************************************************************
C                          OXFORD ZONE MODEL
C                            E. F. SOWELL
C                              JUNE 1991
C * --------------------------------------------------------------- *
C * INPUT:
C *  MISCELLANEOUS (NOT AFFECTED BY ZONE DEFINITION FILE):
C *      1. FRACTION OF NOMINAL NUMBER OF OCCUPANTS (-)
C *      2. FRACTION OF NOMINAL EQUIPMENT POWER (-)
C *      3. FRACTION OF NOMINAL POWER TO LIGHTS (-)
C *      4. SOLAR GAIN THROUGH GLASS (KW)
C *      5. ZONE HUMIDITY (KG/KG-DRY AIR)
C
C *  INFLOWS (MASS, HUMIDITY PAIRS):
C *     NMISC+1. SUPPLY AIR FLOW (KG/S)
C *     NMISC+2. SUPPLY AIR HUMIDITY RATIO  (KG/KG-DRY AIR)
C *                ...
C *     NMISC+2*NIF-1. SUPPLY AIR FLOW (KG/S)
C *     NMISC+2*NIF.   SUPPLY AIR HUMIDITY RATIO  (KG/KG-DRY AIR)
C
C *  TEMPERATURES OF INTERNAL MASS NODES:
C *     NMISC+2*NIF+1.       MASSIVE NODE TEMPERATURE (C)
C *                ...
C *     NMISC+2*NIF+NMASS.  MASSIVE NODE TEMPERATURE (C)
C
C *  TEMPERATURES OF BOUNDARY NODES (INCLUDING SUPPLY AIR STREAMS):
C *     NMISC+2*NIF+NMASS+1. BOUNDARY NODE TEMPERATURE (C)
C *                ...
C *     NMISC+2*NIF+NMASS+NBOUND. BOUNDARY NODE TEMPERATURE (C)
C
C * OUTPUTS:
C *   DERIVATIVES OF INTERNAL MASS NODE TEMPERATURES
C *     1.      MASSIVE NODE TEMPERATURE DERIVATIVE (C/S)
C *                ...
C *     NMASS.  MASSIVE NODE TEMPERATURE DERIVATIVE (C/S)
C
C *   DERIVATIVE OF  ZONE HUMIDITY
C *     NMASS+1.  DERIVATIVE OF  ZONE HUMIDITY  (KG-MOISTURE/KG DRY AIR/S)
C
C *  TEMPERATURES OF LEAVING AIR STREAMS
C *     NMASS+1+1.        LEAVING AIR STREAM TEMPERATURE (C)
C *                        ...
C *     NMASS+1+NOF.  LEAVING AIR STREAM TEMPERATURE (C)
C
C *  NET HEAT GAIN AT BOUNDARY NODES:
C *     NMASS+1+NOF+1.       NET HEAT GAIN AT BOUNDARY NODE (KW)
C *                            ...
C *     NMASS+1+NOF+NBOUND. NET HEAT GAIN AT BOUNDARY NODE (KW)
C
C *  ADDITIONAL ZONE OUTPUTS (CAN BE ANY T OR Q AS SPECIFIED IN ZONE.DAT)
```

```
C *      NMASS+1+NOF+NBOUND+1.  ADDITIONAL OUTPUT
C *                              ...
C *      NMASS+1+NOF+NBOUND+NZVARS.  ADDITIONAL OUTPUT
C
C * PARAMETERS:
C *      1. NOMINAL NUMBER OF OCCUPANTS (-)
C *      2. TOTAL HEAT LOSS PER OCCUPANT (KW/PERSON)
C *      3. OCCUPANT  SENSIBLE FRACTION (-)
C *      4. NOMINAL EQUIPMENT POWER (KW)
C *      5. EQUIPMENT SENSIBLE FRACTION (-)
C *      6. NOMINAL LIGHTING POWER (KW)
C *      7. FRACTION LIGHTING POWER TO BALLAST (-)
C *      8. NOMINAL LIGHT OUTPUT (LUMENS)

C ********************************************************************
C        LIGHTS PROGRAM INTERFACE:
C        (THESE ARE SET FROM THE LIGHTS PROGRAM)

C:  GENERAL:
C        N: TOTAL NUMBER OF NODES IN THE ZONE MODEL
C        F(N): DERIVATIVE OF MASSIVE NODE OR HEAT RATE OF BOUNDARY NODE
C        TIMEL: SIMULATION TIME

C   INTERNAL MASS INFORMATION:
C        NMASS:   NUMBER OF MASSIVE NODES INTERNAL TO THE ZONE MODEL
C        IMASS(NMASS):  ZONE MODEL MASSIVE NODE INDICES
C        TM(NMASS):  MASSIVE NODE TEMPERATURES
C        AIRMASS: TOTAL MASS OF ALL AIR NODES
C        THRMASS: TOTAL THERMAL MASS OF ALL AIR NODES
C
C   BOUNDARY NODE INFORMATION:
C        NBOUND:  NUMBER OF BOUNDARY NODES SET BY HVACSIM
C        IBOUND(NBOUND): ZONE MODEL BOUNDARY NODE INDICES
C        TBOUND(NBOUND): BOUNDARY NODE TEMPERATURES

C    MASS FLOW CONDUCTOR INFORMATION:
C               NOTE-- FLOWS AFFECTING CONDUCTORS  ARE NOT NECESSARILY
C               SAME AS ZONE SUPPLY OR LEAVING FLOW STREAMS. SOME SCHEMES
C               FOR ROUTING AIR THROUGH THE ROOM AND PLENUM
C               HAVE A SINGLE PHYSICAL "SUPPLY" STREAM, BUT
C               DUE TO INTERNAL DIVERSION MAY HAVE TWO OR MORE
C               MASS FLOW RATE TYPE CONDUCTORS. ON THE OTHER HAND,
C               A SINGLE MASS FLOW RATE MAY SET SEVERAL CONDUCTORS.

C        NALF:      NUMBER OF ZONE MODEL CONDUCTORS SET BY HVACSIM+
C        IALF(NALF):    ZONE MODEL CONDUCTOR INDICES
C        ALF(NALF):     CONDUCTANCE OF ZONE MODEL CONDUCTORS
C        NIF: NUMBER OF TYPE288 AIR FLOW INPUTS
C        INFLOW(NALF): CONDUCTANCE INPUT AIR FLOW INDEX (1..NIF)

C   NEEDED FOR SETTING UP TYPE288 OUTPUTS:
C        NOF: NUMBER OF LEAVING AIRSTREAMS
C        IOUT(NOF): INDICES OF NODES FOR SETTING OUT FLOW TEMPERATURES
C       NZVARS: NUMBER OF ADDITIONAL ZONE VARIAVLES OUTPUT
C
C     THINGS IN THE HVACSIM+ SAVED ARRAY:
C.      SAVED(1) = INSTANCE COUNTER
C.      SAVED(2) = POINTER TO PRIVATE DATA (SET IN THE C CODE)
C.      SAVED(3) = TIME AT LAST CALL
C.      SAVED(4) = TIME AT PREVIOUS TIME STEP

C   UNITS OF MEASURE:
C     NORMALLY THE LIGHTS PROGRAM WILL BE COMPILED TO WORK INTERNALLY IN
C     SI UNITS. THE FOLLOWING PARAMETER SETTINGS SUPPORT THIS UNIT SYSTEM,
C     PROVIDING AN SI INTERFACE TO HVACSIM+:
 PARAMETER (TBASE=273.11111,TCONV=1.0,QCONV=1.0,TSCALE=1.0)
 PARAMETER (ACONV= 1.0)

C    IF THE LIGHTS PROGRAM IS COMPILED TO WORK INTERNALLY IN IP UNITS,
C    USE THE FOLLOWING PARAMETER SETTINGS TO GET AN SI HVACSIM+ INTERFACE:
C        PARAMETER (TBASE=491.6,TCONV=1.8,QCONV=3414.0,TSCALE=3600.0)
C        PARAMETER (ACONV= 0.092903)

 PARAMETER (MAXIN=30, MAXOUT=30, MAXIO=30)
 PARAMETER (MAXNODES=50, MAXALF=50, MAXPAR=8, MAXSAV=4)
 PARAMETER (MAXREPT=30)

       DIMENSION XIN(MAXIN), OUT(MAXOUT), IOSTAT(MAXIO), PAR(MAXPAR)
 DIMENSION SAVED(MAXSAV)
       COMMON/CHRONO/ TIME,TSTEP,TTIME,TMIN,ITIME
 DOUBLE PRECISION  F(MAXNODES), TIMEL, ALF(MAXALF)
 DOUBLE PRECISION TM(MAXNODES), TBOUND(MAXNODES), TOF(MAXNODES)
 DOUBLE PRECISION QOCCS,QOCCL,QEQPS,QEQPL, QLIGHTS,QSOL,QBALST
       DOUBLE PRECISION LUMENS
```

```
 CHARACTER*15 FSTRING
 INTEGER IFIRST
 INTEGER IMASS(MAXNODES),NMASS, N, NALF, IALF(MAXALF)
 INTEGER IBOUND(MAXNODES),NBOUND, INFLOW(MAXALF), IOUT(MAXNODES)
 INTEGER NIF, NOF, NPAR, NSAVED, NMISC
 INTEGER CODES(MAXREPT)
        DOUBLE PRECISION ZVARS(MAXREPT), AIRMASS, THRMASS
 REAL SUMIN,SUMMASS,HV,CPA
 SAVE IFIRST,IMASS, N, NMASS, NALF, IALF, IBOUND, NBOUND
 SAVE INFLOW,  NIF, NOF, INSTANCE, NZVARS, AIRMASS, THRMASS
        DATA IFIRST /1/
 DATA CPA/1.006/, HV/2554.91/
 DATA INSTANCE /1/
C.
C.   ZONE288.DAT DEFINES THE ZONE GEOMETRY, ETC.
C.
 FSTRING ='zone288.dat'

C.     SETUP CREATES A NEW INSTANCE OF THE ZONE.  IF IT IS THE
C.     FIRST INSTANCE OF THIS ZONE,  IT READS DATA FILE "zone288.dat"
C.     AND SETS UP CLASS DATA STRUCTURES. ON SUBSEQUENT CALLS
C.     IT WILL ONLY ALLOCATE STORAGE FOR THE PRIVATE DATA FOR THE
C.     NEW INSTANCE.


C.     THE IF-CHECK DEPENDS UPON INITIALIZATION OF THE SAVED ARRAY
C.     TO 0.0 AT A HIGHER LEVEL.  SUN OS SEEMS TO DO SO.

        IF(INT(SAVED(1)).EQ. 0)  THEN
          IF(INSTANCE .EQ. 1) THEN
              CALL SETUP(FSTRING, N, NMASS, IMASS, NBOUND, IBOUND,
     +             NALF, IALF, INFLOW, NIF, NOF, IOUT, NMISC, NPAR,
     +              NSAVED, NZVARS, THRMASS, INSTANCE, SAVED(2))
        AIRMASS = THRMASS/CPA
       NINPUT = NMISC+NMASS+2*NIF+NBOUND
       NOUT = NMASS+1+NOF+NBOUND+NZVARS
       NMAXIO = NINPUT
       IF(NOUT .GT. NINPUT) THEN
        NMAXIO = NOUT
       ENDIF
          ELSE
              CALL INSTAN(INSTANCE, SAVED(2))
          ENDIF
          SAVED(1) = INSTANCE
   INSTANCE  = INSTANCE +1
        ENDIF
C.
C.  CALL TO THE ZONE FILE WRITER
C.
 IF(ITIME .GT. 1) THEN
        IF(TIME .GT. SAVED(3)) THEN
               TIMEL = SAVED(4)/TSCALE
               SAVED(4) = SAVED(3)
               CALL FWZONE(TIMEL, SAVED(2))
        ENDIF
 ELSE
               SAVED(4) = SAVED(3)
 ENDIF
 SAVED(3) = TIME
        TIMEL = TIME/TSCALE

 DO 10 I = 1, NMASS
        TM(I) = TCONV*XIN(NMISC+2*NIF+I) + TBASE
10      CONTINUE

 DO 20 I = 1, NBOUND
        TBOUND(I) = TCONV*XIN(NMISC+2*NIF+NMASS+I) + TBASE
20      CONTINUE

 DO 30 I = 1, NALF
C       INFLOW(I) IS THE SUPPLY AIR FLOW INDEX: 1,2..NIF
C       NOTE THAT THERE IS A (MDOT,W) DOUBLET FOR EACH INFLOW IN XIN
        ALF(I) = (XIN(2*INFLOW(I)+NMISC-1)*CPA)* QCONV/TCONV
30      CONTINUE

 QOCCS   = PAR(1) * PAR(2) * PAR(3) * XIN(1) * QCONV
 QOCCL   = PAR(1) * PAR(2) * (1.0-PAR(3)) * XIN(1) * QCONV
 QEQPS   = PAR(4) * PAR(5) * XIN(2) * QCONV
 QEQPL   = PAR(4) *(1.0 - PAR(5)) * XIN(2) * QCONV
 QSOL    = XIN(4) * QCONV
 QLIGHTS = PAR(6) * (1.0 - PAR(7)) * XIN(3) * QCONV
 QBALST  = PAR(6) * PAR(7) * XIN(3) * QCONV
 LUMENS  = PAR(8) * XIN(3)
```

```
C
C        EVALUATE DERIVATIVES OF DYNAMIC TEMPERATURES
C        AND THE HEAT FLOW RATES AT THE SHARED MASS NODES

C                THIS IS A CALL TO THE C PROGRAM
C    TO CALCULATE TEMPERATURE DERIVATIVES AND BOUNDARY TEMPERATURES
C
 CALL EVALFT(TIMEL, NMASS, IMASS, TM, F, NBOUND, IBOUND, TBOUND,
     +  NALF, IALF, ALF, NOF, TOF, IOUT,
     +  QOCCS,  QEQPS, QSOL, QLIGHTS, LUMENS,
     +  QBALST,SAVED(2))

C        SET UP OUTPUTS

C   TEMPERATURE DERIVATIVES FOR MASSIVE INTERNAL NODES:
 DO 40 I = 1, NMASS
        OUT(I)= (F(IMASS(I)+ 1)/TCONV)/TSCALE
40      CONTINUE

C   ZONE HUMIDITY DERIVATIVE:
 SUMIN   = 0.0
 SUMMASS = 0.0
 J = -2
 DO 45 I=1, NIF
        J = J + 2
        SUMMASS = SUMMASS + XIN(NMISC+1+J)
        SUMIN   = SUMIN + XIN(NMISC+1+J)*XIN(NMISC+1+J+1)
45      CONTINUE
 OUT(NMASS+1)  =  (SUMIN-SUMMASS*XIN(5)+(QOCCL+QEQPL)/HV)/AIRMASS

C   OUTFLOW TEMPERATURES:
 DO 50 I = 1, NOF
        OUT(NMASS+1+I)=  (TOF(I)  - TBASE )/TCONV
50      CONTINUE

C ZONE BOUNDARY HEAT FLOWS:
 DO 60 I = 1, NBOUND
        OUT(NMASS+1+NOF+I)= F(IBOUND(I)+ 1)/QCONV
60      CONTINUE
C
C        GET ADDITIONAL VARIABLES FROM ZONE MODEL FOR OUTPUT
C
 IF(NZVARS .GT. 0) THEN
   CALL GZVARS(SAVED(2), ZVARS, CODES)

   DO 70 I = 1, NZVARS
      IF(CODES(I) .EQ. I) THEN
        OUT(NMASS+1+NOF+NBOUND+I)=  ZVARS(I)/TCONV
      ELSEIF(CODES(I) .EQ. 2 .OR. CODES(I) .EQ. 6 .OR.  CODES(I) .EQ. 7
     +                  .OR.   CODES(I) .EQ. 8) THEN
        OUT(NMASS+1+NOF+NBOUND+I)= ZVARS(I)/QCONV
      ELSE
        OUT(NMASS+I+NOF+NBOUND+I)= ZVARS(I)/(QCONV*ACONV)
     ENDIF
70         CONTINUE
 ENDIF

C        DERIVATIVES CAN BE FROZEN (I SUPPOSE!)

        DO 100 I=1,NMASS+1
            IOSTAT(I) = 1
100     CONTINUE
C
        DO 200 I=NMASS+2, NMAXIO
            IOSTAT(I) = 0
200     CONTINUE
C
        RETURN
      END
```

## 2.) The entry in the HVACSIM+ TYPAR.DAT file for the model

```
************************************************************************
288 'LIGHTS-based Displacement Ventilation Model: one outside wall'
4 1 34 13 8
8  'Zone leaving humidity ratio' 'kg/kg'
3  'ex_a Temperature' 'C'
7  'Net heat to boundary sp_a' 'kW'
```

```
7  'Net heat to boundary flrr' 'kW'
7  'Net heat to boundary ceil' 'kW'
7  'Net heat to boundary w_e1' 'kW'
7  'Net heat to boundary w_e2' 'kW'
7  'Net heat to boundary w_e3' 'kW'
7  'Net heat to boundary w_e4' 'kW'
3  'Temperature of fl_a' 'C'
3  'Temperature of r1_a' 'C'
3  'Temperature of r2_a' 'C'
3  'Temperature of r3_a' 'C'
4  'Fraction of nominal number of occupants' '-'
4  'Fraction of nominal equipment power' '-'
4  'Fraction of nominal power to lights' '-'
7  'Solar gain through glass' 'kW'
8  'Zone leaving humidity ratio' 'kg/kg'
2  'Supply air flow 1' 'kg/s'
8  'Humidity ratio of supply air flow 1' 'kg/kg'
2  'Supply air flow 2' 'kg/s'
8  'Humidity ratio of supply air flow 2' 'kg/kg'
2  'Supply air flow 3' 'kg/s'
8  'Humidity ratio of supply air flow 3' 'kg/kg'
2  'Supply air flow 4' 'kg/s'
8  'Humidity ratio of supply air flow 4' 'kg/kg'
2  'Supply air flow 5' 'kg/s'
8  'Humidity ratio of supply air flow 5' 'kg/kg'
2  'Supply air flow 6' 'kg/s'
8  'Humidity ratio of supply air flow 6' 'kg/kg'
2  'Supply air flow 7' 'kg/s'
8  'Humidity ratio of supply air flow 7' 'kg/kg'
2  'Supply air flow 8' 'kg/s'
8  'Humidity ratio of supply air flow 8' 'kg/kg'
2  'Supply air flow 9' 'kg/s'
8  'Humidity ratio of supply air flow 9' 'kg/kg'
2  'Supply air flow 10' 'kg/s'
8  'Humidity ratio of supply air flow 10' 'kg/kg'
2  'Supply air flow 11' 'kg/s'
8  'Humidity ratio of supply air flow 11' 'kg/kg'
3  'Temperature of boundary sp_a' 'C'
3  'Temperature of boundary flrr' 'C'
3  'Temperature of boundary ceil' 'C'
3  'Temperature of boundary w_e1' 'C'
3  'Temperature of boundary w_e2' 'C'
3  'Temperature of boundary w_e3' 'C'
3  'Temperature of boundary w_e4' 'C'
#
1  'Nominal Number Of Occupants (-)'
2  'Total Heat Loss Per Occupant (kW/person)'
3  'Occupant  sensible fraction (-)'
4  'Nominal Equipment Power (kW)'
5  'Equipment Sensible Fraction (-)'
6  'Nominal Lighting Power (kW)'
7  'Fraction lighting power to ballast (-)'
8  'Nominal Light output (Lumens)'

*************************************************************
```

### 3.) Zone data file for the HSLIGHTS program.

```
 /*
  *             Test data for Oxford HVACSIM+ zone model
  *          Rectangular room with walls, size as Li et al
  *     Walls split into four equal layers - air flow with entrainment
  *             room and plume have three air nodes
  *         Outside temps. and U values set as Li's case B3.
  *                     S.J.Rees - 21/2/95
  */
/* Program Control Parameters */
/* Newton-Raphson controls */
/*eps   max tries */
 0.000000001      100

/*  shortwave limits (microns)*/
/* ms     1st     last */
   1     0.3     0.8
/* longwave limits (microns)*/
/* ml     1st     last */
   1     1.0    200.0
 /* n (number of nodes)*/
    30
```

```
/* node data */
/*node      mass    area    r-sw    r-lw    t-sw   t-lw    t  */
/*====      ====    ====    ====    ====    ====   ====  =====*/

  ceil      0.0    15.12   0.50     0.05    0.0    0.0    22.0
  flrr      0.0    15.12   0.50     0.05    0.0    0.0    22.0
/*  wall nodes */

  w_n1      0.0    2.475   0.50     0.05    0.0    0.0    22.0
  w_n2      0.0    2.475   0.50     0.05    0.0    0.0    22.0
  w_n3      0.0    2.475   0.50     0.05    0.0    0.0    22.0
  w_n4      0.0    2.475   0.50     0.05    0.0    0.0    22.0
  w_s1      0.0    2.475   0.50     0.05    0.0    0.0    22.0
  w_s2      0.0    2.475   0.50     0.05    0.0    0.0    22.0
  w_s3      0.0    2.475   0.50     0.05    0.0    0.0    22.0
  w_s4      0.0    2.475   0.50     0.05    0.0    0.0    22.0
  w_w1      0.0    2.8875  0.50     0.05    0.0    0.0    22.0
  w_w2      0.0    2.8875  0.50     0.05    0.0    0.0    22.0
  w_w3      0.0    2.8875  0.50     0.05    0.0    0.0    22.0
  w_w4      0.0    2.8875  0.50     0.05    0.0    0.0    22.0
  w_e1      0.0    2.8875  0.50     0.05    0.0    0.0    22.0
  w_e2      0.0    2.8875  0.50     0.05    0.0    0.0    22.0
  w_e3      0.0    2.8875  0.50     0.05    0.0    0.0    22.0
  w_e4      0.0    2.8875  0.50     0.05    0.0    0.0    22.0

/* air nodes */
  ex_a      0.0     0.0     1.0     1.0     0.0    0.0    22.0
  fl_a      0.0     0.0     1.0     1.0     0.0    0.0    22.0
  sp_a      0.0     0.0     1.0     1.0     0.0    0.0    22.0
  p1_a      0.0     0.0     1.0     1.0     0.0    0.0    22.0
  p2_a      0.0     0.0     1.0     1.0     0.0    0.0    22.0
  p3_a      0.0     0.0     1.0     1.0     0.0    0.0    22.0
  r1_a      0.0     0.0     1.0     1.0     0.0    0.0    22.0
  r2_a      0.0     0.0     1.0     1.0     0.0    0.0    22.0
  r3_a      0.0     0.0     1.0     1.0     0.0    0.0    22.0
  lo_a      0.0     0.0     1.0     1.0     0.0    0.0    22.0

/* lamp nodes */
  lamp      0.0     1.0     0.5     0.05    0.0    0.0    22.0
  blst      0.0     0.1     0.5     0.05    0.0    0.0    22.0


/*                                                              */
/*                View Factor Data                              */
/*                                                              */
/* number of I,J,F(I,J) triplets*/
   152
/* I   J       F(I,J) */

ceil ceil 0.0
flrr flrr 0.0
w_n1 w_n1 0.0
w_n2 w_n2 0.0
w_n3 w_n3 0.0
w_n4 w_n4 0.0
w_s1 w_s1 0.0
w_s2 w_s2 0.0
w_s3 w_s3 0.0
w_s4 w_s4 0.0
w_w1 w_w1 0.0
w_w2 w_w2 0.0
w_w3 w_w3 0.0
w_w4 w_w4 0.0
w_e1 w_e1 0.0
w_e2 w_e2 0.0
w_e3 w_e3 0.0
w_e4 w_e4 0.0
w_n1 w_n2 0.0
w_n1 w_n3 0.0
w_n1 w_n4 0.0
w_n2 w_n3 0.0
w_n2 w_n4 0.0
w_n3 w_n4 0.0
w_s1 w_s2 0.0
w_s1 w_s3 0.0
w_s1 w_s4 0.0
w_s2 w_s3 0.0
w_s2 w_s4 0.0
w_s3 w_s4 0.0
w_w1 w_w2 0.0
w_w1 w_w3 0.0
w_w1 w_w4 0.0
w_w2 w_w3 0.0
w_w2 w_w4 0.0
```

```
w_w3 w_w4 0.0
w_e1 w_e2 0.0
w_e1 w_e3 0.0
w_e1 w_e4 0.0
w_e2 w_e3 0.0
w_e3 w_e4 0.0
ceil flrr 0.300365
w_n1 w_s1 0.036622
w_n2 w_s2 0.036622
w_n3 w_s3 0.036622
w_n4 w_s4 0.036622
w_w1 w_e1 0.051852
w_w2 w_e2 0.051852
w_w3 w_e3 0.051852
w_w4 w_e4 0.051852
w_n1 w_w1 0.087662
w_n2 w_w2 0.087662
w_n4 w_w4 0.087662
w_s1 w_e1 0.087662
w_s2 w_e2 0.087662
w_s3 w_e3 0.087662
w_s4 w_e4 0.087662
w_n1 w_e1 0.087662
w_n2 w_e2 0.087662
w_n3 w_e3 0.087662
w_n4 w_e4 0.087662
w_s1 w_w1 0.087662
w_s2 w_w2 0.087662
w_s3 w_w3 0.087662
w_s4 w_w4 0.087662
flrr w_w1 0.076186
flrr w_e1 0.076186
ceil w_w4 0.076186
flrr w_n1 0.064498
flrr w_s1 0.064498
ceil w_n4 0.064498
ceil w_s4 0.064498
w_n1 w_w2 0.046227
w_n2 w_w3 0.046227
w_n3 w_w4 0.046227
w_n3 w_w2 0.046227
w_n4 w_w3 0.046227
w_s1 w_e2 0.046227
w_s2 w_e3 0.046227
w_s2 w_e1 0.046227
w_s3 w_e4 0.046227
w_s3 w_e2 0.046227
w_s4 w_e3 0.046227
w_n1 w_e2 0.046227
w_n2 w_e1 0.046227
w_n2 w_e3 0.046227
w_n3 w_e2 0.046227
w_n3 w_e4 0.046227
w_n4 w_e3 0.046227
w_s1 w_w2 0.046227
w_s2 w_w1 0.046227
w_s2 w_w3 0.046227
w_s3 w_w2 0.046227
w_s3 w_w4 0.046227
w_s4 w_w3 0.046227
w_n1 w_s2 0.034921
w_n2 w_s1 0.034921
w_n2 w_s3 0.034921
w_n3 w_s2 0.034921
w_n3 w_s4 0.034921
w_n4 w_s3 0.034921
w_w1 w_e2 0.048746
w_w2 w_e1 0.048746
w_w3 w_e4 0.048746
w_w4 w_e3 0.048746
flrr w_n2 0.043649
flrr w_s2 0.043649
ceil w_s3 0.043649
flrr w_w2 0.051742
flrr w_e2 0.051742
ceil w_w3 0.051742
flrr w_n3 0.030695
ceil w_n2 0.030695
ceil w_s2 0.030695
flrr w_s3 0.030695
flrr w_w3 0.035960
flrr w_e3 0.035960
ceil w_w2 0.035960
ceil w_e2 0.035960
```

```
w_n1 w_s3 0.030468
w_n3 w_s1 0.030468
w_n4 w_s2 0.030468
w_n2 w_s4 0.030468
w_w1 w_e3 0.04093
w_w3 w_e1 0.04093
w_w4 w_e2 0.04093
w_w2 w_e4 0.04093
w_n1 w_s4 0.024725
w_n4 w_s1 0.024725
w_w1 w_e4 0.031522
w_w4 w_e1 0.031522
w_n1 w_e3 0.024334
w_n1 w_w3 0.024334
w_n2 w_e4 0.024334
w_n3 w_e1 0.024334
w_n4 w_e2 0.024334
w_n2 w_w4 0.024334
w_n3 w_w1 0.024334
w_n4 w_w2 0.024334
w_s1 w_w3 0.024334
w_s1 w_e3 0.024334
w_s2 w_w4 0.024334
w_s3 w_w1 0.024334
w_s4 w_w2 0.024334
w_s4 w_e2 0.024334
w_n1 w_e4 0.014508
w_n1 w_w4 0.014508
w_s1 w_w4 0.014508
w_n4 w_w1 0.014508
w_n4 w_e1 0.014508
w_s4 w_w1 0.014508
w_s4 w_e1 0.014508

/* The JJF vector */
/* row  col det. by conservation */

  ceil  w_n3
  flrr  w_e4
  w_n1  ceil
  w_n2  w_w1
  w_n3  w_w3
  w_n4  flrr
  w_s1  ceil
  w_s2  w_e4
  w_s3  w_e1
  w_s4  flrr
  w_w1  ceil
  w_w2  w_e3
  w_w3  w_e2
  w_w4  flrr
  w_e1  ceil
  w_e2  w_e4
  w_e3  ceil
  w_e4  ceil
  ex_a  ex_a
  fl_a  fl_a
  sp_a  sp_a
  p1_a  p1_a
  p2_a  p2_a
  p3_a  p3_a
  r1_a  r1_a
  r2_a  r2_a
  r3_a  r3_a
  lo_a  lo_a
  lamp  lamp
  blst  blst


/* Radiant transmission coupling vector */


  ceil  0
  flrr  0
  w_n1  0
  w_n2  0
  w_n3  0
  w_n4  0
  w_s1  0
  w_s2  0
  w_s3  0
  w_s4  0
  w_w1  0
  w_w2  0
```

```
   w_w3  0
   w_w4  0
   w_e1  0
   w_e2  0
   w_e3  0
   w_e4  0
   ex_a  0
   fl_a  0
   sp_a  0
   p1_a  0
   p2_a  0
   p3_a  0
   r1_a  0
   r2_a  0
   r3_a  0
   lo_a  0
   lamp  0
   blst  0


/*                Conduction/convection data              */
/* Num. of Conductors  */
   35

/* from  to    alpha    beta     gama    delta        */
/* convection from floor (hf = 6) */
  flrr   fl_a  0.09072     0.0        0.0      1.0
/* convection to ceiling (hc = 6) */
  ex_a   ceil  0.09072     0.0        0.0      1.0
/* air flow conductances (a/c = 3) */
  sp_a  -ex_a  0.04158   0.0        0.0      1.0
  fl_a  -sp_a  0.04158   0.0        0.0      1.0
  lo_a  -fl_a  0.02079   0.0        0.0      1.0
  p1_a  -lo_a  0.02079   0.0        0.0      1.0
  r1_a  -fl_a  0.02079   0.0        0.0      1.0
  p1_a  -r1_a  0.00693   0.0        0.0      1.0
  r2_a  -r1_a  0.01386   0.0        0.0      1.0
  p2_a  -p1_a  0.02772   0.0        0.0      1.0
  p2_a  -r2_a  0.00693   0.0        0.0      1.0
  r3_a  -r2_a  0.02216   0.0        0.0      1.0
  p3_a  -r3_a  0.03876   0.0        0.0      1.0
  p3_a  -p2_a  0.03465   0.0        0.0      1.0
  r2_a  -p3_a  0.0083    0.0        0.0      1.0
  r3_a  -p3_a  0.0166    0.0        0.0      1.0
  ex_a  -p3_a  0.04158   0.0        0.0      1.0


/* convection from surfaces to air nodes (hw = 3)*/

/*  w_e3    r3_a  0.008664  0.0            0.0     1.0
  w_e4    ex_a  0.008664  0.0      0.0     1.0
  w_e1    r1_a  0.008664  0.0      0.0     1.0
  w_e2    r2_a  0.008664  0.0      0.0     1.0
  w_w3    r3_a  0.008664  0.0      0.0     1.0
  w_w4    ex_a  0.008664  0.0      0.0     1.0
  w_w1    r1_a  0.008664  0.0      0.0     1.0
  w_w2    r2_a  0.008664  0.0      0.0     1.0
  w_n3    r3_a  0.006189        0.0     0.0     1.0
  w_n4    ex_a  0.006189        0.0     0.0     1.0
  w_n1    r1_a  0.006189        0.0     0.0     1.0
  w_n2    r2_a  0.006189        0.0     0.0     1.0
  w_s3    r3_a  0.006189        0.0     0.0     1.0
  w_s4    ex_a  0.006189        0.0     0.0     1.0
  w_s1    r1_a  0.006189        0.0     0.0     1.0
  w_s2    r2_a  0.006189        0.0     0.0     1.0
  lamp    ex_a  0.001    0.0       0.0     1.0
  blst    ex_a  0.001    0.0       0.0     1.0*/

/* hw = 6.0 */
  w_e3    r3_a  0.01733  0.0       0.0     1.0
  w_e4    ex_a  0.01733  0.0       0.0     1.0
  w_e1    r1_a  0.01733  0.0       0.0     1.0
  w_e2    r2_a  0.01733  0.0       0.0     1.0
  w_w3    r3_a  0.01733  0.0       0.0     1.0
  w_w4    ex_a  0.01733  0.0       0.0     1.0
  w_w1    r1_a  0.01733  0.0       0.0     1.0
  w_w2    r2_a  0.01733  0.0       0.0     1.0
  w_n3    r3_a  0.01238 0.0        0.0     1.0
  w_n4    ex_a  0.01238 0.0        0.0     1.0
  w_n1    r1_a  0.01238 0.0        0.0     1.0
  w_n2    r2_a  0.01238 0.0        0.0     1.0
  w_s3    r3_a  0.01238 0.0        0.0     1.0
  w_s4    ex_a  0.01238 0.0        0.0     1.0
  w_s1    r1_a  0.01238 0.0        0.0     1.0
```

```
  w_s2   r2_a  0.01238 0.0       0.0     1.0
  lamp   ex_a  0.001   0.0       0.0     1.0
  blst   ex_a  0.001   0.0       0.0     1.0

/* Vector indicating which is lamp surface */
/*  NOTE: lamp nodes must have power   */


  ceil  0
  flrr  0
  w_n1  0
  w_n2  0
  w_n3  0
  w_n4  0
  w_s1  0
  w_s2  0
  w_s3  0
  w_s4  0
  w_w1  0
  w_w2  0
  w_w3  0
  w_w4  0
  w_e1  0
  w_e2  0
  w_e3  0
  w_e4  0
  ex_a  0
  fl_a  0
  sp_a  0
  p1_a  0
  p2_a  0
  p3_a  0
  r1_a  0
  r2_a  0
  r3_a  0
  lo_a  0
  lamp  1
  blst  0

/* Vector indicating which is ballast surface */


  ceil  0
  flrr  0
  w_n1  0
  w_n2  0
  w_n3  0
  w_n4  0
  w_s1  0
  w_s2  0
  w_s3  0
  w_s4  0
  w_w1  0
  w_w2  0
  w_w3  0
  w_w4  0
  w_e1  0
  w_e2  0
  w_e3  0
  w_e4  0
  ex_a  0
  fl_a  0
  sp_a  0
  p1_a  0
  p2_a  0
  p3_a  0
  r1_a  0
  r2_a  0
  r3_a  0
  lo_a  0
  lamp  0
  blst  1


/* Lamp luminous distribution vs. wave length */
/* No. of points on the curve */
  39

/* wavelength,output pairs */

          0       0       .3080       0        .3081        15e-4
          .3179   15e-4   .3180       0        .3290        0
          .3291   5e-4    .3389       5e-4     .3390        0
          .3610   10e-4   .3611       40e-4    .3709        40e-4
```

```
          .3710   15e-4   .4000        30e-4      .4001            90e-4
          .4099   90e-4   .4100        40e-4      .4310            50e-4
          .4311   195e-4  .4409        195e-4     .4410            65e-4
          .4750   85e-4   .5200        80e-4      .5410            108e-4
          .5411   190e-4  .5509        190e-4     .5510            135e-4
          .5730   175e-4  .5731        225e-4     .5829            225e-4
          .5830   190e-4  .6000        185e-4     .6150            150e-4
          .6250   110e-4  .6500        55e-4      .6750            30e-4
          .7000   20e-4   .7500        0          0.9999           0

/* Lamp relative power and luminous output curves */
/* No. points on curve*/

 2
   -50000    1.0     1.0
     60000   1.0     1.0

 /* Node reporters */
/*  No. of reporters */
       1
/*     File name        Writing interval   Nodes reported */
/*        out                 0.01                 5*/
/* This one writes to a file.  Doesn't work too well due to lack of
 * synchronization with  calculation interval. Best we can do is make
 * requested interval small, causing printing at the calculation points.
 */
     /*Node     No. items reported at node */


/* This one sets up additional zone variables to pass back for TYPE288 output*/
/*     File name        Writing interval   Nodes reported */
        none                 0.0                4
     /*Node     No. items reported at node */
       fl_a     1       -1
       rl_a     1       -1
       r2_a     1       -1
       r3_a     1       -1


/* HVACSIM+ TYPE188.f interface definition */


/* Description string  (60 char max on a separate line)  */
LIGHTS-based Displacement Ventilation Model: one outside wall

/* n_misc: Number of miscellaneous inputs to TYPE188 (always 5):*/
5
/* n_par: Number of TYPE188 parameters  (always 8)*/
8
/* n_saved: Number of TYPE188 saved values(always 4) */
4

/*  Massive nodes  in order of TYPE188 input list:. Include every node with
 *  mass greater than  min. recognized mass.
 */
/* n_mass: Number of massive nodes:*/
0
/* Node labels */


/*  Boundary nodes. Include enclosing surfaces and supply air nodes.
 *  List in order of TYPE188 input list.
 */

/* n_bound: Number of boundary nodes:*/
7
/* Node labels */
sp_a flrr ceil w_e1 w_e2 w_e3 w_e4

/* nif: Number of TYPE 188 input flows (used to set mass flow type conductances */
11

/* nalf = Number of mass flow type conductances -- m_dot*Cp */
14

/* nalf triplets: (downstream-node upstream-node input-flow-index) */

/*  sp_a  -ex_a   1*/
  fl_a  -sp_a  1
  lo_a  -fl_a  2
  pl_a  -lo_a  2
  rl_a  -fl_a  3
  pl_a  -rl_a  6
```

```
  r2_a  -r1_a  5
  p2_a  -p1_a  4
  p2_a  -r2_a  6
  r3_a  -r2_a  8
  p3_a  -r3_a  9
  p3_a  -p2_a  7
  r2_a  -p3_a  11
  r3_a  -p3_a  10
  ex_a  -p3_a  1


/*nof: Number of TYPE188 output flows */
1
/* Node labels for temperatures of output flows */
ex_a


/* Distribution of occupant, equipment,  and solar, SENSIBLE heat gains*/
/* node   solar    occupant      equipment        */

  ceil     0.0   0.0   0.0
  flrr     1.0   0.0   0.0
  w_n1     0.0   0.0   0.0
  w_n2     0.0   0.0   0.0
  w_n3     0.0   0.0   0.0
  w_n4     0.0   0.0   0.0
  w_s1     0.0   0.0   0.0
  w_s2     0.0   0.0   0.0
  w_s3     0.0   0.0   0.0
  w_s4     0.0   0.0   0.0
  w_w1     0.0   0.0   0.0
  w_w2     0.0   0.0   0.0
  w_w3     0.0   0.0   0.0
  w_w4     0.0   0.0   0.0
  w_e1     0.0   0.0   0.0
  w_e2     0.0   0.0   0.0
  w_e3     0.0   0.0   0.0
  w_e4     0.0   0.0   0.0
  ex_a    0.0   0.0   0.0
  fl_a     0.0   0.0   0.0
  sp_a     0.0   0.0   0.0
  p1_a     0.0   0.0   0.0
  p2_a     0.0   0.0   0.0
  p3_a     0.0   0.0   0.0
  r1_a     0.0   0.0   0.0
  r2_a     0.0   0.0   0.0
  r3_a     0.0   0.0   0.0
  lo_a     0.0   1.0   1.0
  lamp     0.0   0.0   0.0
  blst     0.0   0.0   0.0

/* Number of air nodes */
 10
/* Air node labels  (used to determine air mass for humidity diff eq)*/
  ex_a fl_a sp_a p1_a p2_a p3_a r1_a r2_a r3_a lo_a

 /* That's all folks! */
```

# IEA Annex 28

## Low Energy Cooling

### Subtask 2: Detailed Design Tools

# Night Cooling Control Strategies for Commercial Buildings

## A description of the model in ASHRAE Toolkit format

Building Research Establishment
M. Kolokotroni

BSRIA
A.Martin

# Contents Chapter G

## 1. Technology area

Night cooling control strategies for commercial buildings.

## 2. Developed by


| | | | |
|---|---|---|---|
| **Name** | : | Maria Kolokotroni | Andrew Martin |
| **Organisation** | : | Building Research Establishment | BSRIA |
| **Address** | : | Environmant Group, Garston, Watford, WD2 7JR, UK | Old Bracknell Lane West, Bracknell Berkshire, RG12 7AH, UK |
| **Phone** | : | +44 1923 664000 | +44 1344 426511 |
| **Fax** | : | ++44 1923 664796 | +44 1344 487575 |
| **E-mail** | : | kolokotronim@bre.co.uk | |


## 3. General description

Three control strategies suitable for night cooling for commercial buildings have been identified. The effectiveness of the strategies has been monitored in buildings with mixed mode (combination of natural and mechanical) ventilation strategies.

The strategies are as follows:

**Setpoint Control**

This control strategy for night cooling typically calculates the mean external temperature for a time period during the afternoon. In the event that the mean outside temperature is above the "precool initiation setpoint" (e.g. 20°C) for this period and the internal temperature is greater than the external temperature then precooling will take place at the end of the occupation period. Precooling is carried out until the zone temperature drops to the minimum allowable space temperature (e.g.16°C) at which point all the inlet and outlet vents will be closed. Once this has occurred then the building will slowly increase in temperature due to heat gains re-emitted from the building fabric, furniture and fittings.

The vents remain shut until this passive heating process has allowed the internal temperature to rise (e.g. 19°C) at which point the inlet and outlet vents again open. This cooling and heating process continues until such time as the "preheat" period is reached. The preheat period is the time at which the inlet and outlet vents must shut in order that the heating effects of the building fabric and fittings will heat the space to the heating setpoint (e.g. 19°C) by the start of occupation. A further calculation is continuously carried out after preheat has started to assess if the building will reach the heating setpoint by the start of the occupation period. If this setpoint is not reached then the heating plant will be enabled. Although this is not anticipated to occur it does ensure that any over-cooling will not affect the comfort conditions. Wind and rain interlocks are utilised to prevent water ingress and damage due to high air velocities. A low external temperature interlock (eg 12°C) is also provided to prevent any risk of condensation.

**Slab Temperature Control**

This precooling strategy aims to cool the slab to a pre-defined slab temperature setpoint during the night in order to offset the heat gains of the next day. If the control strategy is applied to a mixed-mode building utilising both automatic control of casement windows and mechanical ventilation plant, the mechanical plant shuts down at the end of the occupation period. In the event that the space temperature is more than say, 0.5°C above the cooling setpoint (eg 23°C), passive cooling utilising the casement windows will be maintained in order to reduce the internal space temperature. The amount of cooling is controlled by the internal air temperature. At a pre-determined time, and providing that the building is to be occupied the following day, the slab temperature is compared with the required slab temperature setpoint and if this is higher then slab cooling will commence. Passive cooling is initially used to facilitate cooling of the slab and this is controlled by the slab temperature setpoint. The calculation of the slab temperature setpoint is self-learning and is based on equalising the slab temperature, the room temperature and the slab temperature setpoint. An adjustment factor is provided in order that towards the end of the occupancy period a cooling effect is still available from the slab (if the slab and the room are at the same temperature then there is no cooling available). A further factor allows the amount of self-learning to be varied between full self-learning and no self-learning. Wind and rain interlocks are utilised to prevent water ingress and damage due to high air velocities as well as a low external temperature interlock (e.g. 12°C) to prevent any risk of condensation.

If at the start of the low electricity tariff period the slab temperature has not achieved the slab setpoint, the time is calculated so that the fan assisted cooling is enabled to achieve the slab temperature setpoint by the end of the low tariff period. This calculation is based upon the difference between the internal and the external temperature and the rate of change of the slab temperature using the fans under these conditions.

If the building is to be unoccupied for more than 24 hours then at the end of the occupancy period all the plant will shut down. Natural ventilation will be employed to maintain the space temperature conditions. At say, 18:00 hours the day before the next occupancy period, the precooling strategy detailed above will be initiated for slab cooling.

**Degree Hours Control**

This precooling strategy aims to measure the daytime heat gains in the space and the cooling gains at night, using these to maintain the equilibrium between the building fabric temperature and the space temperature. The method of estimating the daytime heat gains is based upon measuring the degree hours of heating. The heat gain degree hours is defined as the amount by which the temperature is above the chosen setpoint, totalled for all the hours in the period. The decision as to precool or not is based upon the number of hours that the internal temperature is above the room temperature setpoint. If at the end of the occupied period the degree hours are greater than say, three degree hours, and the internal temperature is greater than the external temperature then the decision is made to precool the building during that night. Normal wind and rain interlocks still apply as well as the proviso that the external temperature is above the low limit setpoint (12°C) to prevent any risk of condensation.

Once precooling has been initiated the inlet and outlet vents modulate to maintain the space temperature at the precool setpoint (e.g. 18°C). The amount of time that the internal temperature is below the cooling setpoint is calculated and a figure for the night cooling gains obtained. When the degree hours of night cooling gains are equal to those of the daytime heating gains the precooling is complete. However further cooling will take place since the internal temperature will still be below the room temperature setpoint.

If night cooling is not completed then the control system calculates the time that the ventilation should shut down in order that the heat gains (from the building fabric, furniture and fittings) will provide sufficient heating to raise the space temperature to the space temperature setpoint by the start of occupation period.

**Summary**

To help choose between the different strategies, a recommendation has been derived based on monitored results and thermal modelling defined by the following rules:

- Select one or a combination of the following criteria to initiate night cooling
  - ⇒ *peak zone temperature* (any zone) > 23°C
  - ⇒ average daytime zone temperature (any zone) >22 °C
  - ⇒ average afternoon outside air temperature > 20 °C
  - ⇒ slab temperature >23 °C

- Night cooling should continue through the night providing that all the following conditions are satisfied
  - ⇒ zone temperature (any zone) > outside air temperature (+2K, or more, for mechanical ventilation , ie to allow for fan pick up, +0K for passive ventilation)
  - ⇒ zone temperature (any zone) > heating setpoint
  - ⇒ minimum outside air temperature >12 °C

Night cooling should be potentially avaialable seven days per week and thoughout the entire non-occupied period in the building.

## 4. Mathematical description

**Setpoint Control** will operate night cooling if the following conditions are satisfied:

the time is between midnight and 7am AND
inside air temperature > cooling setpoint (e.g. 18°C) AND
the outside temperature > 12°C AND
the outside air temperature < inside air temperature

**Slab Temperature Control** will operate night cooling if the following conditions are satisfied:

the time is between midnight and 7am AND
slab temperature > cooling setpoint (e.g. 23°C) AND
*the outside temperature* > 12°C

**Degree Hours Control** will operate night cooling if the following conditions are satisfied:

the time is between midnight and 7am AND
heat gain degree hours > preset degree hours (e.g. 3) AND
the outside temperature > 12°C

heat gain degree hours = (number of hours) x pos(internal temperature - cooling setpoint (e.g. 18°C))

## 5. References


1. Martin A., 'Night Cooling Control Strategies', in Proc. CIBSE National Conference 1995, Volume II, CIBSE, London, pp215-222.

2. Martin A and Fletcher J, 'Night Time is the Right Time', Building Services Journal, August 1996, pp 25-26

3. Fletcher J.S., Martin A.J., 'Night Cooling Control Strategies', Technical Appraisal 14/96, BSRIA 1996.

# 6. Flowcharts

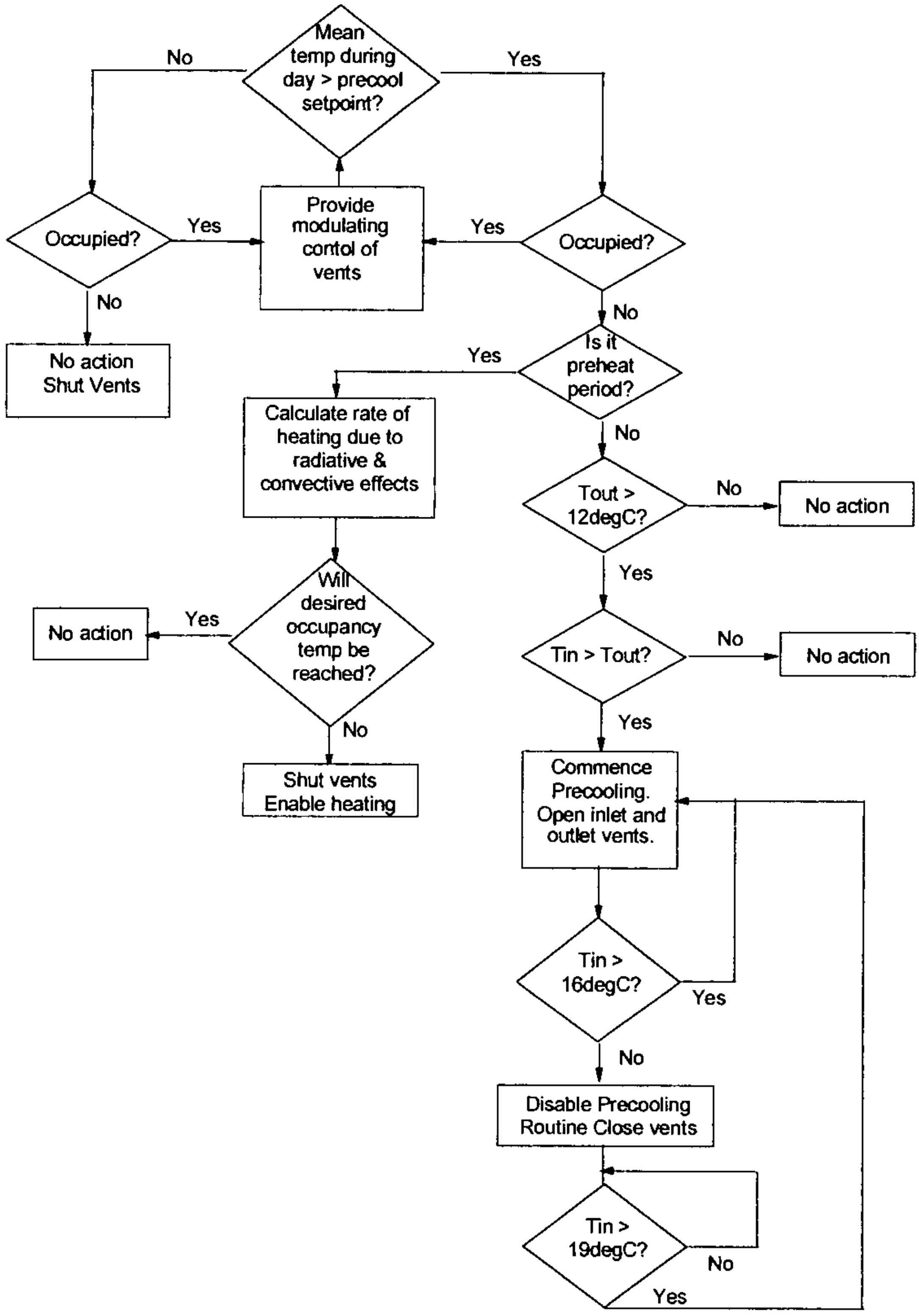



**Figure 1:** Control Strategy 1- Setpoint Control

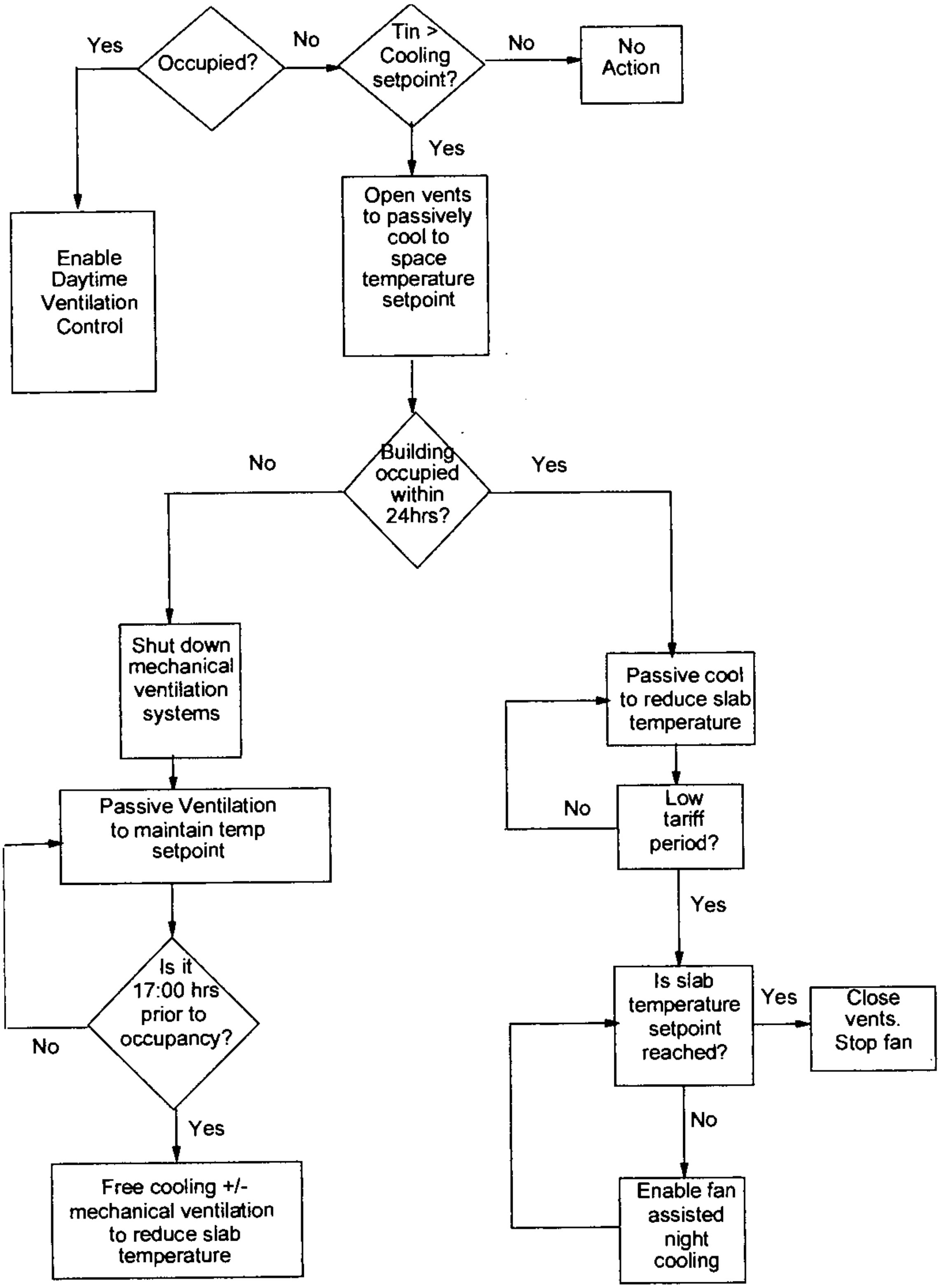



**Figure 2:** Control Strategy 2 - Slab Temperature Control

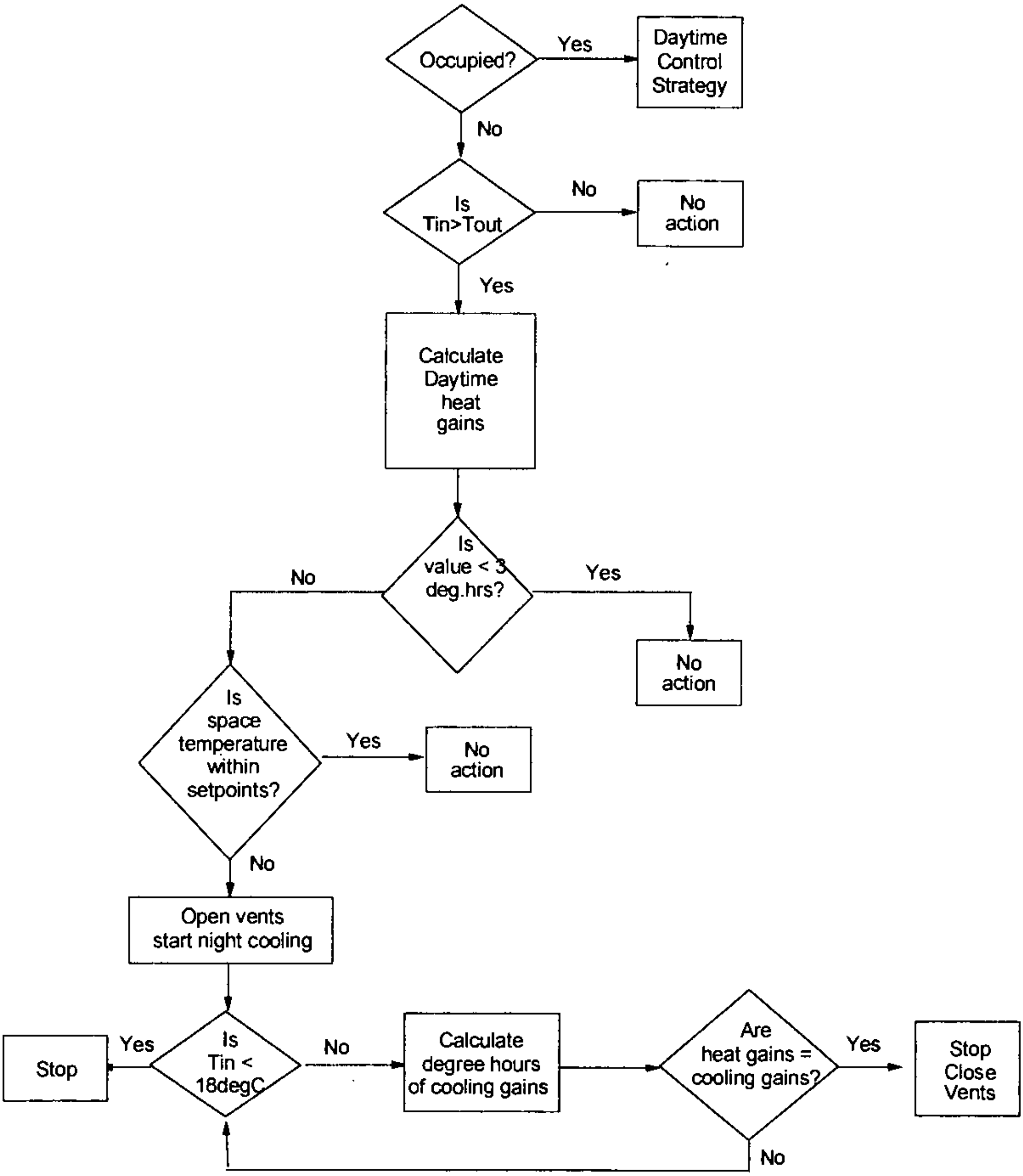



**Figure 3:** Control Strategy 3: Degree Hours Control

## 7.1 Sample results using Kew weather data

The following three graphs are an illustration of the internal zone air temperatures **predicted** by each control strategy for (a) a typical summer day, (b) a peak summer day and (c) a typical spring day, using Heathrow UK weather data.

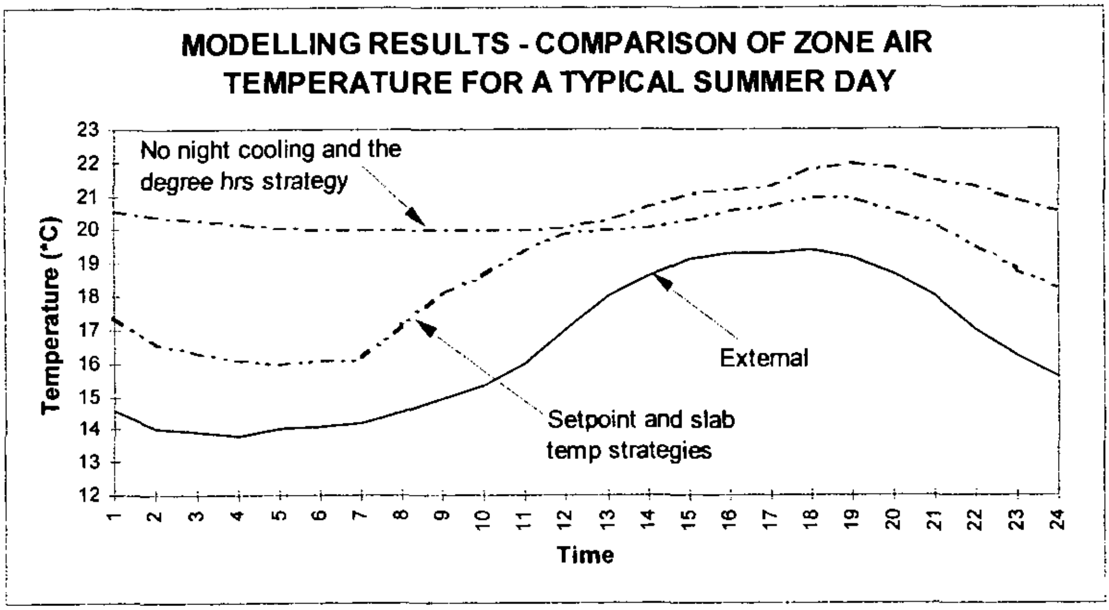



**Figure 4:** Typical summer's day zone air temperature. This graph demonstrates that the degree hours night cooling strategy did not operate since that the internal temperature did not rise above 24°C. The setpoint and slab temperature control strategies did operate, both for the same amount of time.

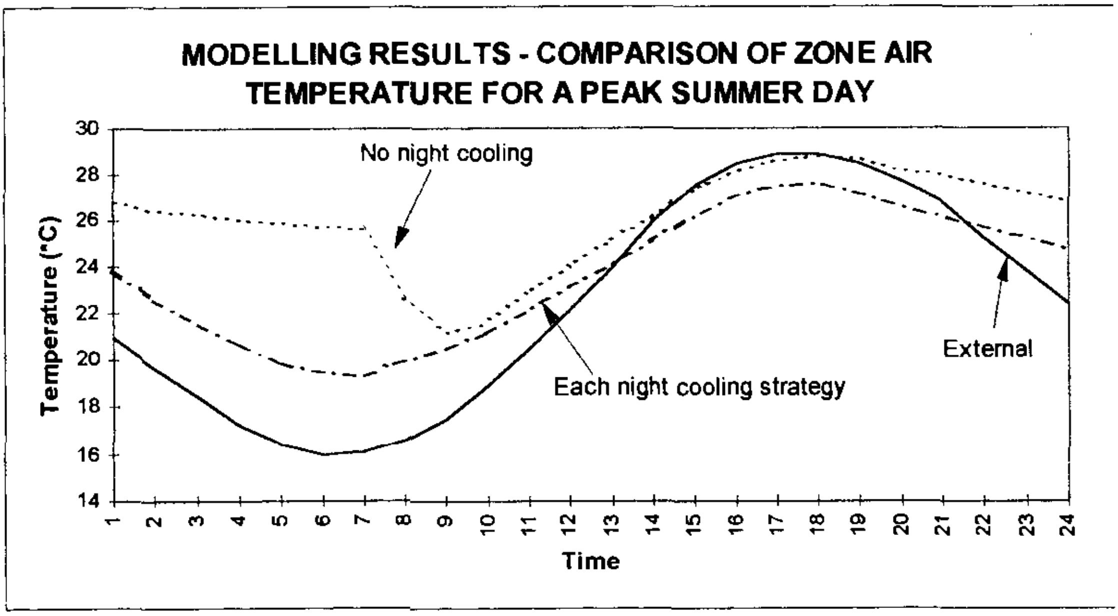



***Figure 5:*** *Peak summer's day zone air temperature.* This graph shows that all the night cooling control strategies operated for the same amount of time ie maximum night cooling

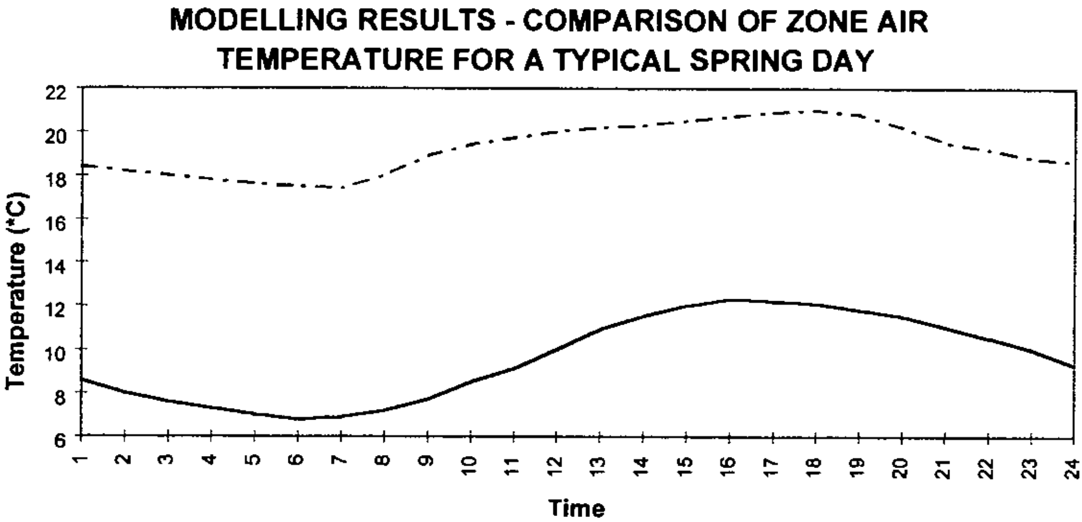



**Figure 6:** Spring day zone air temperature. This graph shows that the conditions were not satisfied for any of the night cooling strategies to initiate night cooling.

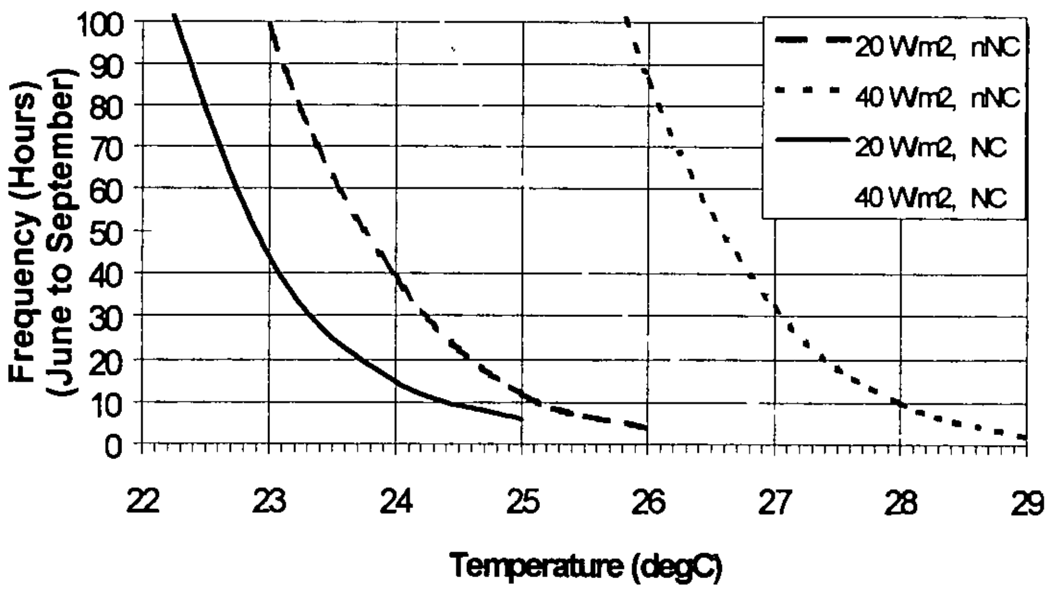
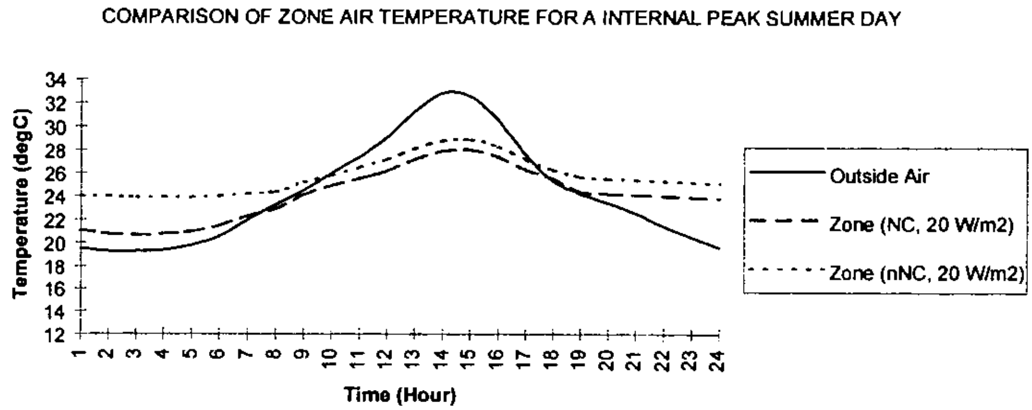
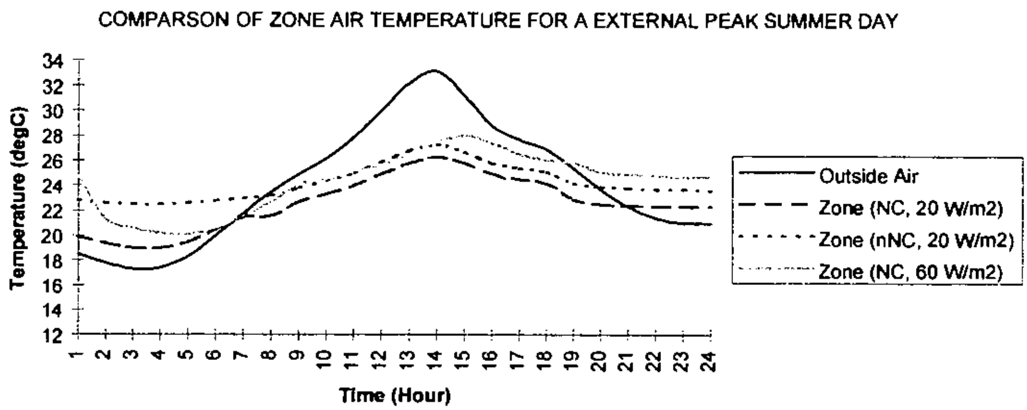



**Figure 7:** Internal temperature frequency distribution of a typical office module with two levels of internal + solar gains. NC denotes day ventilation and night ventilation at 4 AC/h. nNC denotes day ventilation at 4 AC/h but no night ventilation.

## 7.2 Sample results using Zurich weather data

The following graphs are the predicted temperatures using the Annex 28 reference building model and weather data (Zurich) provided by Mark Zimmermann (Switzerland). The Setpoint Control strategy was used for the cases where night cooling is used (NC). For comparison simulations were run without night cooling (nNC).



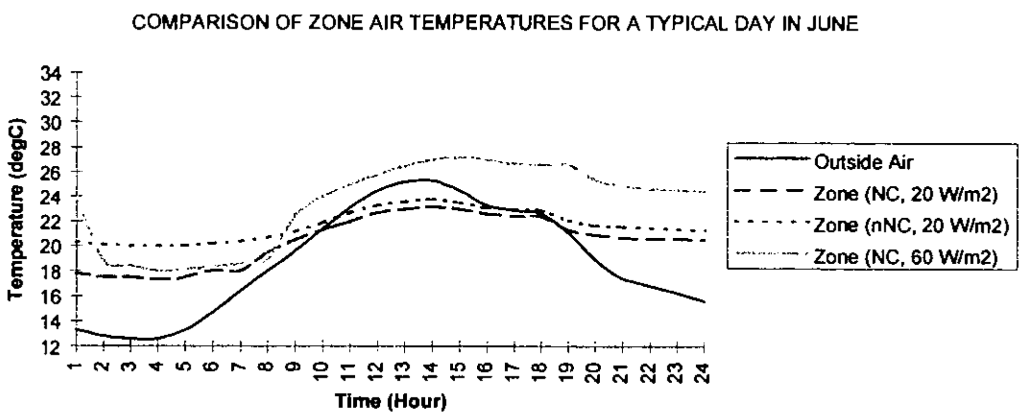




**Figure 8:** Modelling results using Zurich weather data, Annex 28 reference building and the setpoint control strategy.

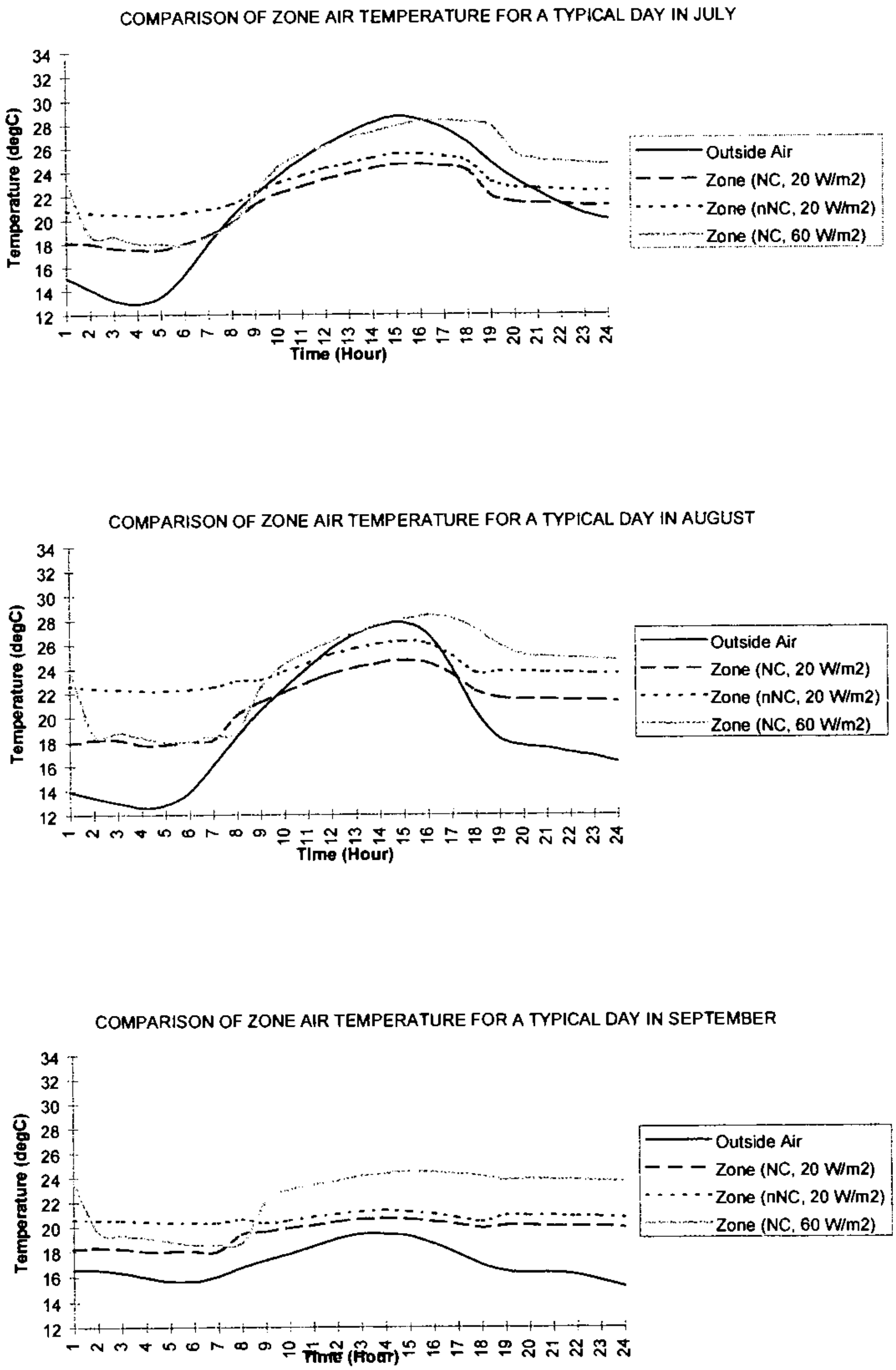



**Figure 9:** Modelling results using Zurich weather data, Annex 28 reference building and the setpoint control strategy.

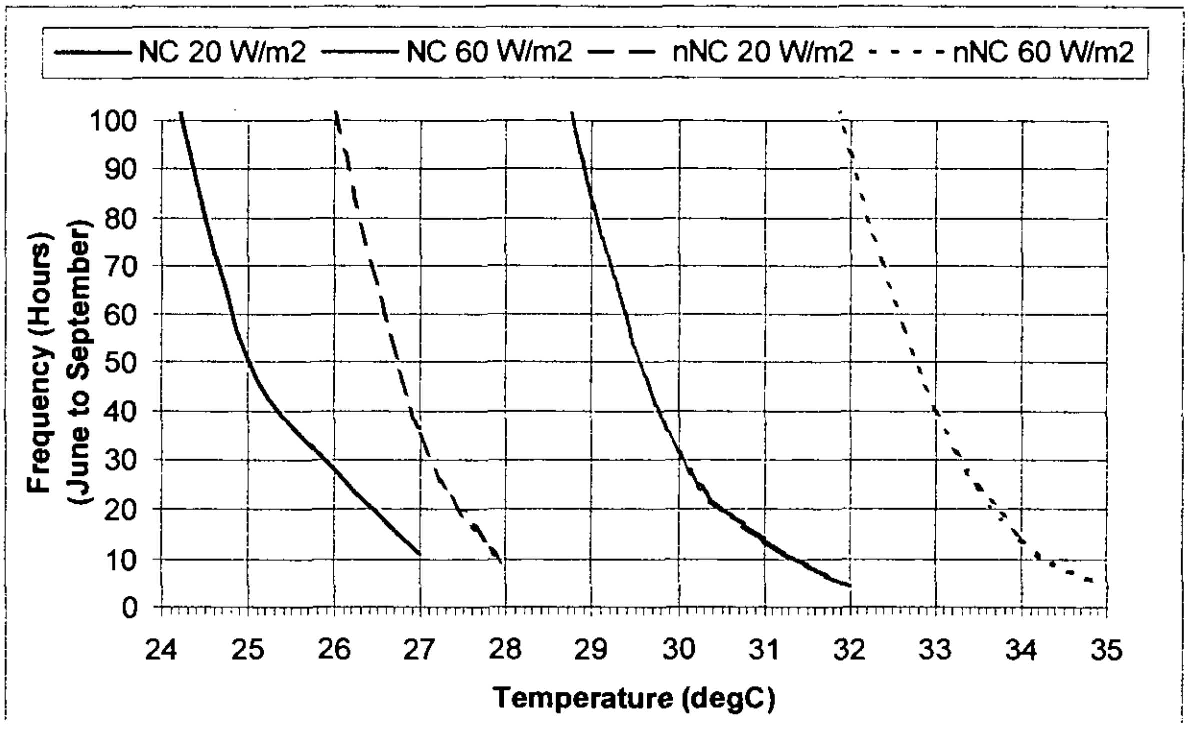



**Figure 10:** Internal temperature frequency distribution of Annex 28 reference building and Zuricjh weather data with two levels of internal + solar gains. NC denotes day ventilation and night ventilation at 4 AC/h. nNC denotes day ventilation at 4 AC/h but no night ventilation.

# IEA Annex 28

## Low Energy Cooling

### Subtask 2: Detailed Design Tools

**NIGHT VENTILATION**
**IN RESIDENTIAL BUILDINGS**

**VENTILATION DUE TO WINDOWS OPENING**


CSTB, Marne la Vallée, France
J.R. Millet

C.E.T.E. Méditerrannée, Aix en Provence, France
J.A.Bouchet

# Contents chapter H

## 1. Technology area

Night ventilation in residential buildings – ventilation due to window opening

## 2. Developed by


**Name** : J.R. MILLET and *J.A. BOUCHET

**Organisation** : CSTB - *C.E.T.E. Méditerranée

**Address** : BP 02 - F-77421 Marne-la-Vallée Cedex 02
*BP 37000 - F-13791 Aix-en-Provence Cedex 03

**Phone** : 33 01 64688300
*33 04 42247966

**Fax** : 33 01 64688350
*33 04 42247777

**E-mail** : millet@cstb.fr


## 3. General description

This algorithm can be used to calculate air flow rates in residential buildings in summer due to window opening. It is based on typical occupant behaviour regarding noise exposure of windows.

Firstly, the state of opening of windows is calculated for three periods (night, early evening, day) taking into account window characteristics and the type of room (bedroom or not). This provides equivalent opening areas which are then used in a second step to calculate the air change rate.

## 4. Nomenclature

| name | description | unit |
|---|---|---|
| A | window area | $m^2$ |
| Aeq | equivalent window area | $m^2$ |
| Aeqt | equivalent total area for each of 4 orientations | $m^2$ |
| Cpr | equivalent area coefficient | ad. 0 to 1 |
| Deb | air flow due to window openings | $m^3/h$ |
| IBR | exposure to outdoor noise coefficient | ad. 1,2,3,4 |
| IJN | room status | ad. 1,2 |
| IPS | type of solar shading device | ad. 1,2,3,4 |
| IPV | exposure to breaking or entering coefficient | ad. 1,2 |
| or | window orientation coefficient | ad. 1,2,3,4 |
| ph | period of day | ad. 1,2,3 |
| ROL | free window area | ad. 0 to 1 |

# 5. Mathematical description

## 5.1. Calculation of the air change rate due to window opening

### 5.1.1 Algorithm

a) each window is characterised by the following parameters :

**ROL** ratio between the maximum free opening area and the window area

**IBR** exposure to outdoor noise (expressed in terms of required sound insulation)
1 : no requirement,
2 : 30 dB,
3 : 35 dB,
4 : 40 dB.

**IPV** exposure to breaking or entering
1 : no risks or non accessible or protected window,
2 : others.

Each window with lower part higher than 2 m from the ground is considered as non accessible.

**IJN** occupation of the room behind the window
1 : sleeping room
2 : others

**IPS** type of solar device
1 : permeable devices giving a protection against entering when closed,
2 : other permeable devices,
3 : other devices,
4 : no solar device.

A device is considered as permeable to air is the air openings area is greater than 0.3 times the window area.

b) **ph** is defined as a period of day (night, early morning, day)
For each window b, the equivalent air area Aeq (b,ph) is calculated as:

$$Aeq\ (b,\ ph) = Ab\ .\ ROL\ .\ Cpr\ (IJN,\ IBR,\ IPV,\ IPS)$$

where Ab ($m^2$) is the window area.
The equivalent areas for the dwelling windows are summed for each orientation or (East :1, South :2, West : 3, North :4). Equivalent total areas are then obtained for each orientation :

$$Aeqt\ (ph,or) = \Sigma_b\ Aeq\ (b,ph)$$

c) The air change rate **Deb(ph)** for the dwelling ($m^3/h$) is then calculated for each period of time ph by:

$$Deb(ph) = 100\ (\ \Sigma_{or}\ Aeqt\ (ph,or) + 3\ \surd\ (\ \Sigma_{or2>or1r}\ Aeqt\ (ph,or1)\ .\ Aeqt\ (ph,or2))$$

where 100 (m/h) is an equivalent wind speed for thermal stack effect and cross ventilation due to the wind. The first term of this equation is related to the stack effect for one room. The second one takes into account cross ventilation [1].

### 5.1.2 Parameters values

| room type | exposure to noise | safety exposure | solar device | night | early morning | day |
|---|---|---|---|---|---|---|
| IJN | IBR | IPV | IPS | ph1 | ph2 | ph3 |
| 1 sleeping rooms | 1 | 1 | 1 | 0.7 | 0.7 | 0 |
| | | | 2 | 0.7 | 0.7 | 0 |
| | | | 3 | 0.3 | 0.3 | 0 |
| | | | 4 | 0.3 | 1 | 0 |
| | | 2 | 1 | 0.7 | 0.7 | 0 |
| | | | 2 | 0 | 0.7 | 0 |
| | | | 3 | 0 | 0.3 | 0 |
| | | | 4 | 0 | 1 | 0 |
| | 2 or 3 | all | 1 | 0 | 0.7 | 0 |
| | | | 2 | 0 | 0.7 | 0 |
| | | | 3 | 0 | 0.3 | 0 |
| | | | 4 | 0 | 1 | 0 |
| 2 other rooms | 1 or 2 | 1 | 1 | 1 | 0.7 | 0 |
| | | | 2 | 1 | 0.7 | 0 |
| | | | 3 | 1 | 0.3 | 0 |
| | | | 4 | 1 | 1 | 0 |
| | | 2 | 1 | 0.7 | 0.7 | 0 |
| | | | 2 | 0 | 0.7 | 0 |
| | | | 3 | 0 | 0.3 | 0 |
| | | | 4 | 0 | 1 | 0 |
| | 3 | all | 1 | 0 | 0.3 | 0 |
| | | | 2 | 0 | 0.3 | 0 |
| | | | 3 | 0 | 0.3 | 0 |
| | | | 4 | 0 | 0.3 | 0 |
| all | 4 | all | all | 0 | 0 | 0 |

**Cpr coefficients Cpr (IJN, IBR, IPV, IPS) for each period of day ph1,2,3**

the period of times ph are constant and defined as (in solar time)
night : from 18h to 5h (e.g. 20h to 7h legal time)
early morning : from 5h to 7h (e.g. 7h to 9h legal time)
day : from 7 to 18h (e.g. 9h to 20 legal time)
The Cpr coefficients Cpr (IJN, IBR, IPV, IPS) are given in the Table above. These coefficients are based on *in situ* observations of the inhabitant behaviour (CETE Méditerranée).

### 5.2. Calculation of the overall air change rate

The overall air change rate is calculated by adding the air change rate due to window opening to the mechanical air change rate due to the conventional mechanical system if there is one (typically providing about 0.5 AC/h). If the mechanical ventilation system is designed to provide higher air flow rates to improve summer comfort, then the maximum value between it and the one obtained by window opening should be adopted.

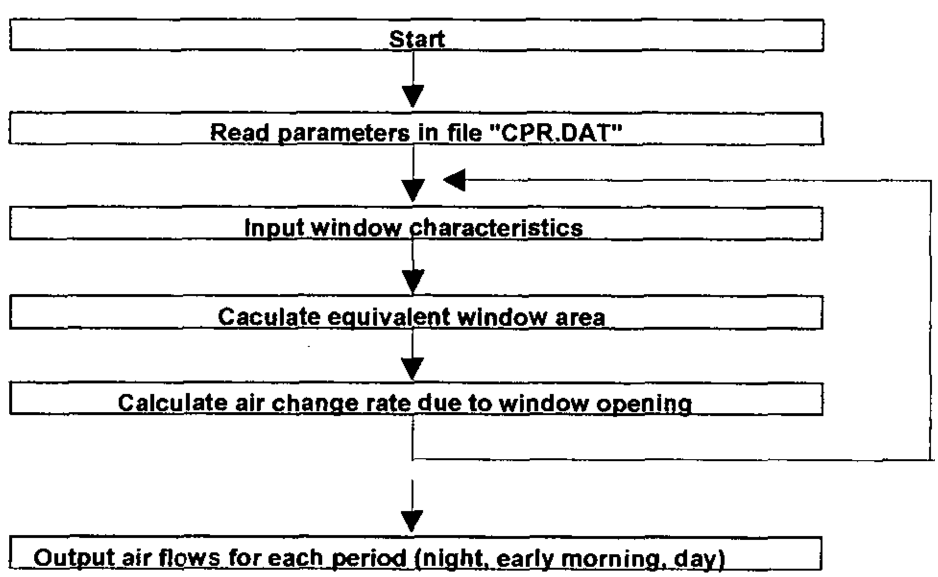
## 6. References

[1] The equation is based on in situ temperature measurements made and used by the CETE Méditerranée.

## 7. Algorithm

1. read the Cpr values
2. read the window characteristics, orientations and noise exposure
3. calculate the effective areas for the fourth orientations (N,E,S,O)
4. calculate the air flows for each period (night, early morning, day)

## 8. Flowchart



## 9. Source code (Fortran)

```
PROGRAM NATVTres

     REAL DEB(3),DEBVE,DEBTT,VOL
     REAL SEN(3),SEE(3),SES(3),SEO(3)
     REAL CPR_V(4,2,4,4)

c***** INPUTS
c        VOL Volume of buiding (m3)
c        For each Window
c        AP  Surface of Window
c        IOR Orientation (South: 0°; West: 90°; North: 180°; East: 270°)
c        ROL Ratio between the maximum free opening area and the window area
c        IBR Exposure to outdoor noise
c           (1: no requirement; 2: 30dB; 3: 35dB; 4: 40dB)
c        IPV Exposure to breaking or entering
c           (1: no risks or non accessible or protected window
c            2: others)
c        IJN Occupation of the room behind the window
c           (1: sleeping room ; 2: others)
c        IPS Type of solar device
c           (1: permeable device giving a protection against entering when close
c            2: other permeable devices
c            3: other device
c            4: no solar device)
c
c****  PARAMETERS
c        CPR(IJN,IBR,IPV,IPS) read file "CPR.DAT"
c
c****  OUTPUTS
c        DEB(1) Air Flow due to window openings from 18h to 5h
c        DEB(2) Air Flow due to window openings from 5h to 7h
c        DEB(3) Air Flow due to window openings from 7h to 18h

c****  Initialisation
     do ih=1,3
     Deb(ih)=0
     sen(ih)=0
     see(ih)=0
     ses(ih)=0
     seo(ih)=0
     end do

c****  Read CPR values
     open (unit=9,file='CPR.DAT',status='unknown')
     DO IBR=1,4
         DO IJN=1,2
         READ(9,*) ((CPR_V(IBR,IJN,IPS,IPH),IPH=1,4),IPS=1,4)
         END DO
     END DO
     close(9)
```

```
c**** Read Windows characteristics, orientation and noise exposure
      Write(*,*) 'VOL'
      READ (*,*) VOL
      Write(*,*) 'AP,IOR,ROL,IBR,IPV,IJN,IPS'

      DO 30 Ip=1,100
      READ(*,*) AP,IOR,ROL,IBR,IPV,IJN,IPS
      IF (AP.EQ.-99) goto 40

c**** calculation of equivalent surfaces for ventilation
         DO IPH=1,3
         IPHV=IPH+1
         IF(IPV.EQ.1.AND.IPH.EQ.1) IPHV=1
         SE=AP*ROL*CPR_V(IBR,IJN,IPS,IPHV)
         IF ((IOR.GE.135).AND.(IOR.LT.225)) SEN(IPH)=SEN(IPH)+SE
         IF ((IOR.GE.225).AND.(IOR.LT.315)) SEE(IPH)=SEE(IPH)+SE
         IF ((IOR.GE.315).OR.(IOR.LT.45)) SES(IPH)=SES(IPH)+SE
         IF ((IOR.GE.45).AND.(IOR.LT.135)) SEO(IPH)=SEO(IPH)+SE
         END DO

c**** calculation of air change rate due to windows openings
c     IPH=1 from 18h to 5h
c     IPH=2 from 5h to 7h
c     IPH=3 from 7h to 18h

      DO IPH=1,3
      DEBTT=SEN(IPH)+SEE(IPH)+SES(IPH)+SEO(IPH)
      DEBVE=3*(SEN(IPH)*(SEE(IPH)+SES(IPH)+SEO(IPH))
     1 +SEE(IPH)*(SES(IPH)+SEO(IPH))+SES(IPH)*SEO(IPH))**0.5
      DEB(IPH)=100*(DEBTT+DEBVE)/VOL
      END DO

30    CONTINUE

40    CONTINUE

      WRITE(*,*) 'AIR CHANGE RATE FOR THE WINDOWS OPENINGS (vol/h) :'
      WRITE(*,*) 'Solar time ->'
      WRITE(*,2000) 'from 18h to  5h :',DEB(1)
      WRITE(*,2000) 'from  5h to  7h :',DEB(2)
      WRITE(*,2000) 'from  7h to 18h :',DEB(3)

2000  FORMAT(A18,F10.2)

      END
```

Note: the executable version given on the disket must be run under DOS directly to obtain the screen displayed results

## 10. Sample results

Sample results are given below for a dwelling (200 $m^3$) with window areas of 6 $m^2$ for the bedroom and 8 $m^2$ for the other rooms. Results are given with and without cross ventilation for IBR 1 to 3 and for cases without safety risks (IPS1 and 2 are not separated).

**air change rate per hour due to window opening**

| window areas | | | | |
|---|---|---|---|---|
| | Aje | 8 | Ane | 0 |
| | Ajo | 0 | Ano | 6 |

| dwelling volume | 200 |
|---|---|

**night**
18 h - 5h solar time

| | | | sleeping room windows | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | IBR1 | | | IBR2 | | | IBR3 | | | |
| | | | IPS12 | IP3 | IPS4 | IPS12 | IPS3 | IPS4 | IPS12 | IPS3 | IPS4 | CPR |
| other wind. | IBR1 | IPS12 | 14.8 | 10.6 | 10.6 | 4.0 | 4.0 | 4.0 | 4.0 | 4.0 | 4.0 | 1 |
| | | IP3 | 14.8 | 10.6 | 10.6 | 4.0 | 4.0 | 4.0 | 4.0 | 4.0 | 4.0 | 1 |
| | | IPS4 | 14.8 | 10.6 | 10.6 | 4.0 | 4.0 | 4.0 | 4.0 | 4.0 | 4.0 | 1 |
| | IBR2 | IPS12 | 14.8 | 10.6 | 10.6 | 4.0 | 4.0 | 4.0 | 4.0 | 4.0 | 4.0 | 1 |
| | | IPS3 | 14.8 | 10.6 | 10.6 | 4.0 | 4.0 | 4.0 | 4.0 | 4.0 | 4.0 | 1 |
| | | IPS4 | 14.8 | 10.6 | 10.6 | 4.0 | 4.0 | 4.0 | 4.0 | 4.0 | 4.0 | 1 |
| | IBR3 | IPS12 | 2.1 | 0.9 | 0.9 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0 |
| | | IPS3 | 2.1 | 0.9 | 0.9 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0 |
| | | IPS4 | 2.1 | 0.9 | 0.9 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0 |
| | | CPR | 0.7 | 0.3 | 0.3 | 0 | 0 | 0 | 0 | 0 | 0 | |

**early morning**
5h - 7h solar time

| | | | sleeping room windows | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | IBR1 | | | IBR2 | | | IBR3 | | | |
| | | | IPS12 | IPS3 | IPS4 | IPS12 | IPS3 | IPS4 | IPS12 | IPS3 | IPS4 | CPR |
| other wind. | IBR1 | IPS12 | 12.2 | 8.5 | 14.5 | 12.2 | 8.5 | 14.5 | 12.2 | 8.5 | 14.5 | 0.7 |
| | | IPS3 | 8.1 | 5.2 | 9.9 | 8.1 | 5.2 | 9.9 | 8.1 | 5.2 | 9.9 | 0.3 |
| | | IPS4 | 14.8 | 10.6 | 17.4 | 14.8 | 10.6 | 17.4 | 14.8 | 10.6 | 17.4 | 1 |
| | IBR2 | IPS12 | 12.2 | 8.5 | 14.5 | 12.2 | 8.5 | 14.5 | 12.2 | 8.5 | 14.5 | 0.7 |
| | | IPS3 | 8.1 | 5.2 | 9.9 | 8.1 | 5.2 | 9.9 | 8.1 | 5.2 | 9.9 | 0.3 |
| | | IPS4 | 14.8 | 10.6 | 17.4 | 14.8 | 10.6 | 17.4 | 14.8 | 10.6 | 17.4 | 1 |
| | IBR3 | IPS12 | 8.1 | 5.2 | 9.9 | 8.1 | 5.2 | 9.9 | 8.1 | 5.2 | 9.9 | 0.3 |
| | | IPS3 | 8.1 | 5.2 | 9.9 | 8.1 | 5.2 | 9.9 | 8.1 | 5.2 | 9.9 | 0.3 |
| | | IPS4 | 8.1 | 5.2 | 9.9 | 8.1 | 5.2 | 9.9 | 8.1 | 5.2 | 9.9 | 0.3 |
| | | CPR | 0.7 | 0.3 | 1 | 0.7 | 0.3 | 1 | 0.7 | 0.3 | 1 | |

**window effective area**
**Aje** : no sleeping, East
**Ajo** : no sleeping, West
**Ane** : sleeping, East
**Ano** : sleeping, West

**noise exposure(IBR)**
**1**: not exposed
**2:** isol. 30 dB
**3:** isol 35 dB

**solar prot. (IPS)**
**1 and 2** : permeable
**3**: not permable
**4:** without

**with cross ventilation (bedroom west, other rooms east)**

**air change rate per hour due to window opening**

| | | | | |
|---|---|---|---|---|
| window areas | Aje | 8 | Ane | 6 |
| | Ajo | 0 | Ano | 0 |

| | |
|---|---|
| dwelling volume | 200 |

**night**
18 h - 5h solar time

| | | | sleeping room windows | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | IBR1 | | | IBR2 | | | IBR3 | | | |
| | | | IPS12 | IP3 | IPS4 | IPS12 | IPS3 | IPS4 | IPS12 | IPS3 | IPS4 | CPR |
| other wind. | IBR1 | IPS12 | 6.1 | 4.9 | 4.9 | 4.0 | 4.0 | 4.0 | 4.0 | 4.0 | 4.0 | 1 |
| | | IP3 | 6.1 | 4.9 | 4.9 | 4.0 | 4.0 | 4.0 | 4.0 | 4.0 | 4.0 | 1 |
| | | IPS4 | 6.1 | 4.9 | 4.9 | 4.0 | 4.0 | 4.0 | 4.0 | 4.0 | 4.0 | 1 |
| | IBR2 | IPS12 | 6.1 | 4.9 | 4.9 | 4.0 | 4.0 | 4.0 | 4.0 | 4.0 | 4.0 | 1 |
| | | IPS3 | 6.1 | 4.9 | 4.9 | 4.0 | 4.0 | 4.0 | 4.0 | 4.0 | 4.0 | 1 |
| | | IPS4 | 6.1 | 4.9 | 4.9 | 4.0 | 4.0 | 4.0 | 4.0 | 4.0 | 4.0 | 1 |
| | IBR3 | IPS12 | 2.1 | 0.9 | 0.9 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0 |
| | | IPS3 | 2.1 | 0.9 | 0.9 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0 |
| | | IPS4 | 2.1 | 0.9 | 0.9 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0 |
| | | CPR | 0.7 | 0.3 | 0.3 | 0 | 0 | 0 | 0 | 0 | 0 | |

**early morning**
5h - 7h solar time

| | | | sleeping room windows | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | IBR1 | | | IBR2 | | | IBR3 | | | |
| | | | IPS12 | IPS3 | IPS4 | IPS12 | IPS3 | IPS4 | IPS12 | IPS3 | IPS4 | CPR |
| other wind. | IBR1 | IPS12 | 4.9 | 3.7 | 5.8 | 4.9 | 3.7 | 5.8 | 4.9 | 3.7 | 5.8 | 0.7 |
| | | IPS3 | 3.3 | 2.1 | 4.2 | 3.3 | 2.1 | 4.2 | 3.3 | 2.1 | 4.2 | 0.3 |
| | | IPS4 | 6.1 | 4.9 | 7.0 | 6.1 | 4.9 | 7.0 | 6.1 | 4.9 | 7.0 | 1 |
| | IBR2 | IPS12 | 4.9 | 3.7 | 5.8 | 4.9 | 3.7 | 5.8 | 4.9 | 3.7 | 5.8 | 0.7 |
| | | IPS3 | 3.3 | 2.1 | 4.2 | 3.3 | 2.1 | 4.2 | 3.3 | 2.1 | 4.2 | 0.3 |
| | | IPS4 | 6.1 | 4.9 | 7.0 | 6.1 | 4.9 | 7.0 | 6.1 | 4.9 | 7.0 | 1 |
| | IBR3 | IPS12 | 3.3 | 2.1 | 4.2 | 3.3 | 2.1 | 4.2 | 3.3 | 2.1 | 4.2 | 0.3 |
| | | IPS3 | 3.3 | 2.1 | 4.2 | 3.3 | 2.1 | 4.2 | 3.3 | 2.1 | 4.2 | 0.3 |
| | | IPS4 | 3.3 | 2.1 | 4.2 | 3.3 | 2.1 | 4.2 | 3.3 | 2.1 | 4.2 | 0.3 |
| | | CPR | 0.7 | 0.3 | 1 | 0.7 | 0.3 | 1 | 0.7 | 0.3 | 1 | |

**window effective area**
**Aje** : no sleeping, East
**Ajo** : no sleeping, West
**Ane** : sleeping, East
**Ano** : sleeping, West

**noise exposure(IBR)**
**1**: not exposed
**2:** isol. 30 dB
**3:** isol 35 dB

**solar prot. (IPS)**
**1 and 2** :
permeable
**3**: not permable
**4:** without

**without cross ventilation (all rooms east)**

# IEA Annex 28

## Low Energy Cooling

### Subtask 2: Detailed Design Tools

## Seasonal Groundwater Cold
## Water Storage

### Calculation and Simulation model in spreadsheet format

DWA Consultants
J.J.Buitenhuis

Roel Consultants
H.C.Roel

# Contents Chapter I

# 1. Technology Area

Model for calculation and simulation of a ground coupled seasonal storage cooling system

# 2. Developed by


| | | |
|---|---|---|
| **Name:** | DWA Consultants | J.J. Buitenhuis |
| **Contact and editor:** | Roel Consultants | H.C.Roel |
| **Phone:** | 0031 30 687 59 69 | |
| **Fax:** | 0031 30 687 59 70 | |
| **E-mail:** | hcroel@wxs.nl | |


# 3. General description

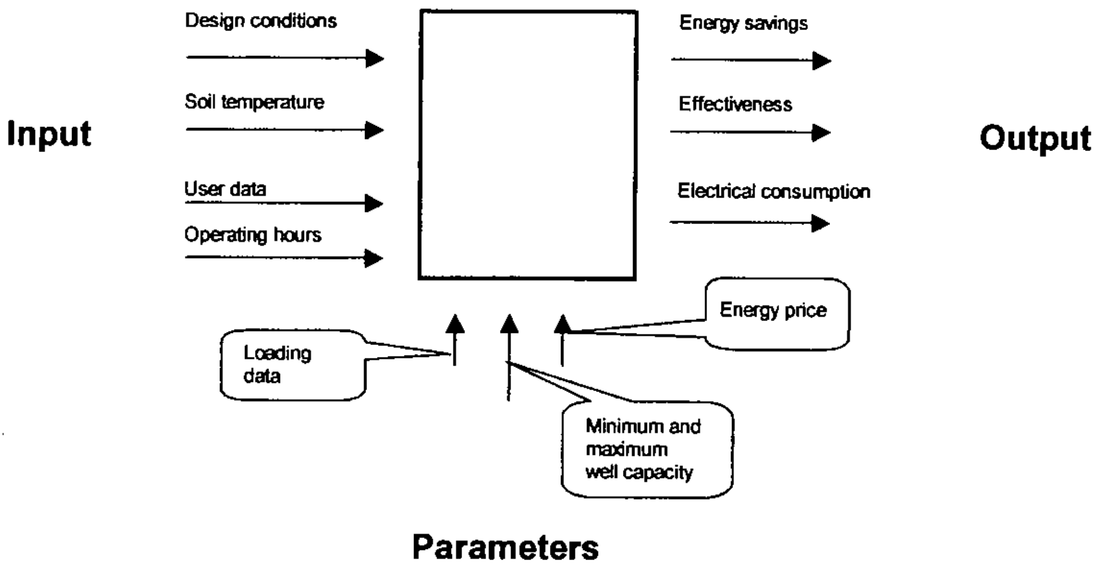
The technology is based on the storage of cold and/or heat over a long period (ie seasons) in soil aquifers. Cold water produced in wintertime by exposing to cold air can be stored until far into the summer.

The basis of the model described is an air conditioning system with constant ventilation rate. It is also possible to use the stored water for other processes and purposes. The stored cold can be used as a standalone source of cold energy, or in combination with cooling machines and cooling towers.

The model analyses the storage process. Transport energy has been included in the calculation. The store loading and unloading volume flow rates depend on the soil conditions. The calculation starts with experiential values depending on soil system and pump installation. The soil composition into aquifer and blocking layers (on top and bottom) are put in as default values (refer to table). The temperatures in the air handling unit (or other process) determine the return temperature to the warm well and are ignored further in this model. (NB This would obviously be important in case of a two well-installation where the warm water is to be used in the winter, eg for preheating or as a source for a heat pump).

One of the following may be used for loading the cold-well during the winter:

- The coil of the air handling unit during periods when it is not in use for ventilation purposes.
- A cooling tower connected into the cold water circuit.
- By open water when the temperature is low enough.


Input
Design conditions
Soil temperature
User data
Operating hours
Energy savings
Effectiveness
Electrical consumption
Output
Energy price
Loading data
Minimum and maximum well capacity
Parameters

# 4. Mathematical description

The following equations are used to calculate the well parameters, the pump head, efficiency of cold water storage and open cooling tower performance.

**Experimental formulas for calculating well diameters and separation**

$$D_{well} = \frac{Q_{max}}{1.5 * \pi * D_{aq}}$$

Where:

$D_{well}$= diameter of the well (m)
$Q_{max}$= maximum well capacity. ($m^3/h$)
$D_{aq}$ = thickness of aquifer- layer (m)

**1.5 is a default value for water velocity on filter surface in m/h.**

The distance between two wells must be at least:

$$L = 2.25 * \sqrt{\frac{V_w}{\varepsilon . \pi . D_{aq}}}$$

Where:

$L$ = distance between wells
$\varepsilon$ = porosity of the aquifer
$V_w$= water volume per season

**Calculation of pump head**

The pump head needs to take account of the changing goundwater level due to defiltration or infiltration. The level change can be determined by the following experimental equations:

$$dHgw = \frac{24 * Q\max}{2 * \pi * K * Daq} * \ln\left(\frac{L}{(0.5) * Dwell}\right)$$

Where:

$K$ = permeability of the aquifer (m/day)

The total well pump pressure will be given by:

$$dH_{tot} = \left(dH_{gw} + dH_{a}\right)_{defil} + \left(dH_{gw} - dH_{a}\right)_{infil} + dH_{circ} + 2$$

Where:

| | | | |
|---|---|---|---|
| $dH_p$ | = | *total pressure of the pump in meters water gauge* | *($mw_p$)* |
| $dH_a$ | = | *difference between ground water level of the aquifer and zero-level* | |
| $dH_{circ}$ | = | *pressure loss into the aquifer between the wells* | |

The factor **2** is a safety margin for filthiness of wells etc.

**Efficiency factor of the coldwater storage**

This efficiency factor can be determined by:

$$\eta_{sto} = \frac{\int Q_{out}(t) * (T_{ref} - T_{out}(t)) * \rho * C dt}{\int Q_{in}(t) * (T_{ref} - T_{inj}(t)) * \rho * C dt}$$

Where:

| | | | |
|---|---|---|---|
| $Q_{in}, Q_{out}$ | = | *capacity of infiltration and defiltration* | *($m^3/s$)* |
| $T_{inj}, T_{out}$ | = | *mean temp.of stored and rejected water.* | *(°C)* |
| $T_{ref}$ | = | *reference temp.mostly mean temp into the warm well* | *(°C)* |
| $\rho$ | = | *specific mass of water* | *($kg/m^3$)* |
| $C$ | = | *specific heat factor* | *(J/kgK)* |

**Open cooling tower**

In the case of loading the cold well by use of an open cooling tower we can model the ideal cooling tower as:

$$Q_w * \rho_w * C_w (T_{wi} - T_{wo}) = Q_a * \rho_a * (H_{ao} - H_g)$$

Where:

| | | | |
|---|---|---|---|
| $Q_w, W_a$ | = | *water and air volumes* | *($m^3/s$)* |
| $\rho_w, \rho_a$ | = | *specific mass of water and air* | *($kg/m^3$)* |
| $C_w$ | = | *specific heat of water* | *(J/kgK)* |
| $T_{wi}, T_{wo}$ | = | *supply and return temp.of water* | *(°C)* |
| $H_{ao}, H_g$ | = | *supply and return enthalpy of air* | *(J/kg)* |

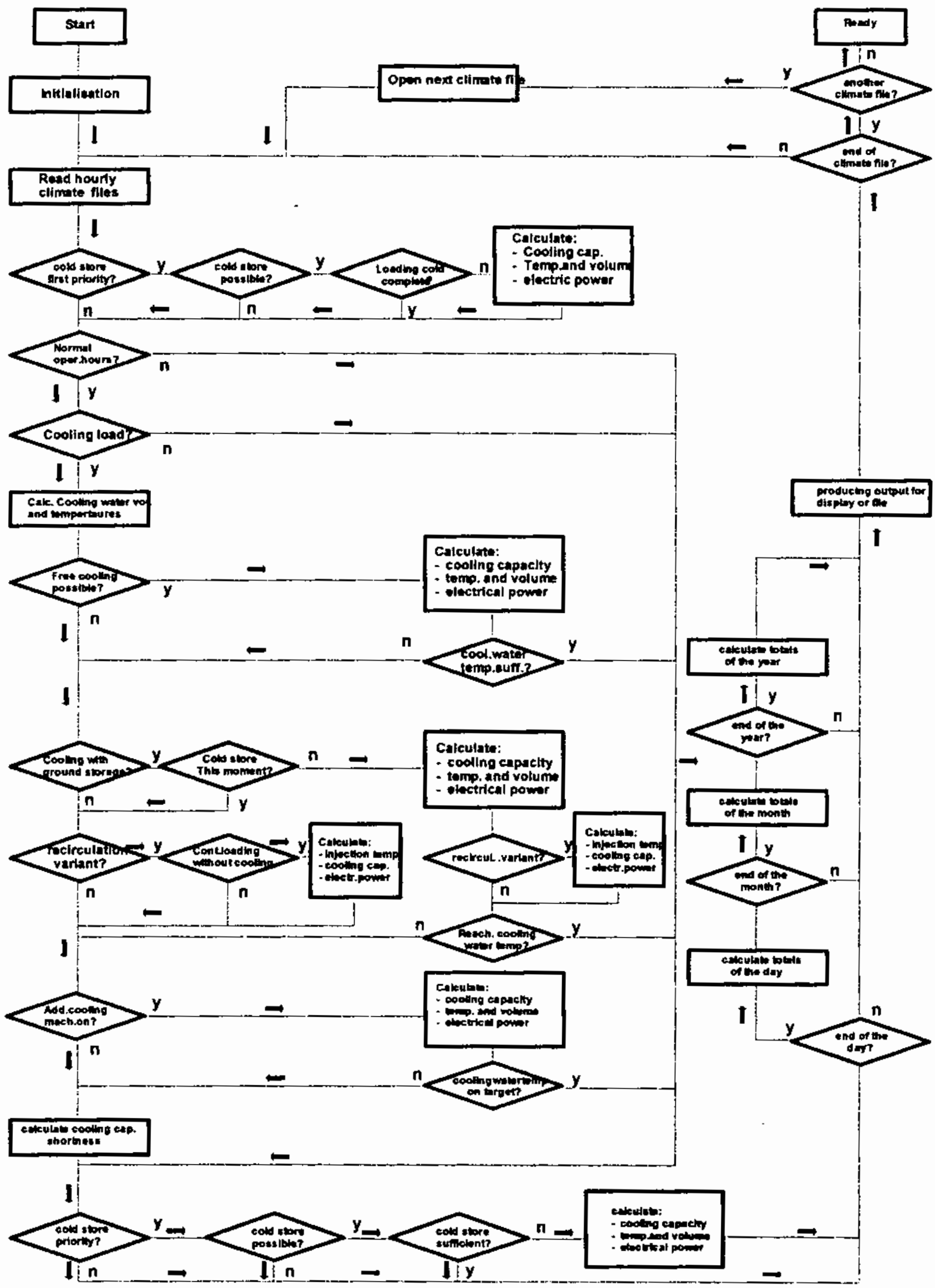
In practice, the cooling tower will require an over-measure of air through the tower. The cooling tower efficiency is given by:

$$\eta = \frac{T_{wi} - T_{wo}}{T_{wi} - T_{wb}}$$

Where:

| | | |
|---|---|---|
| $\eta$ | = | *Cooling tower efficiency* |
| $T_{wb}$ | = | *Wet bulb temp.* |

# 5. Flow chart


Start
Initialisation
Open next climate file
Ready
another climate file?
end of climate file?
Read hourly climate files
cold store first priority?
cold store possible?
Loading cold complete?
Calculate:
- Cooling cap.
- Temp.and volume
- electric power
Normal oper.hours?
Cooling load?
Calc. Cooling water vol. and tempertaures
producing output for display or file
Free cooling possible?
Calculate:
- cooling capacity
- temp. and volume
- electrical power
cool.water temp.suff.?
calculate totals of the year
end of the year?
Cooling with ground storage?
Cold store This moment?
Calculate:
- cooling capacity
- temp. and volume
- electrical power
calculate totals of the month
recirculation variant?
Cont.loading without cooling
Calculate:
- injection temp
- cooling cap.
- electr.power
recircul. variant?
Calculate:
- injection temp
- cooling cap.
- electr.power
end of the month?
Reach. cooling water temp?
calculate totals of the day
Add.cooling mech.on?
Calculate:
- cooling capacity
- temp. and volume
- electrical power
end of the day?
cooling water temp on target?
calculate cooling cap. shortness
cold store priority?
cold store possible?
cold store sufficient?
calculate:
- cooling capacity
- temp.and volume
- electrical power

# 6. References

1. Underground Thermal Storage.
State of the art 1994
ISBN 90-802769-1-x
G.Bakama,A.L. Snijders IF Technology Arnhem The Netherlands
B.Nordell Lulea University of Technology

2. PIA 12 Model for Design and simulation of cooling systems with seasonal cold storage.
(Dutch) DWA Consultants, Bodegraven, The Netherlands

3. Seasonal Thermal Storage in North West Europe
Report for EC DG XVII Thermie B-program DIS-0463-95-NL
Coordinator Grontmij De Bilt The Netherlands

4. A lot of brochures in Dutch, edited by The Netherlands Agency for Energy and the Environment.
(NOVEM)

# 7 Sample results

## Project information

| | | |
|---|---|---|
| Name of the project | : | KPN office |
| Adress | : | |
| City | : | Amersfoort |
| Phone | : | |
| Subject | : | Seasonal Cold Storage |
| Project ID | : | |
| Date | : | 15 January 1997 |

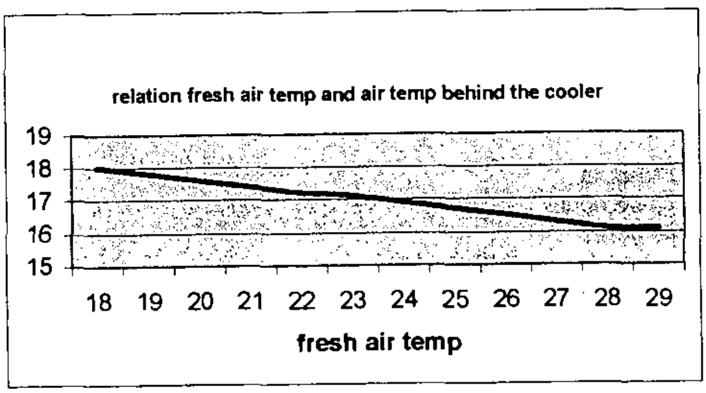
Calculation PIA 12 model

### Ventilation

| | |
|---|---|
| Number of air handling units | 2 |
| Capacity per unit | 89000 $m^3/h$ |
| Percentage recirculation (max.) | 0% |
| Fan power | 45 kW |

### Cooling

| | Temp. | R.H. | Cooling load by design cond. | |
|---|---|---|---|---|
| Outside Cond. | 28.0 °C | 60% | Room Cond. | 24.0 °C 50 % |
| | | | | -------- |
| | | | | 0 kW |




| | | |
|---|---|---|
| Space sensible cooling load | : | 471 kW |
| Requirement for secondary (mechanical) cooling | : | 0 kW |

Operation hours

| | |
|---|---|
| Sunday | none |
| Monday-Friday | 08.00-18.00 h |
| Saturday | none |

**Estimated cooling load**

| | Maximum Cooling Capacity KW | Cooling load Year MWh | Full load hours | Operating hours |
|---|---|---|---|---|
| Cooling with AHU | 1160 | 145 | 125 | 540 |
| Secondary cooling | 0 | 0 | 0 | 0 |
| Apps/processes | 0 | 0 | 0 | 0 |
| **Total** | **1160** | **145** | **125** | **540** |

**Soil data**

| | Aquifer | Blocking Layers |
|---|---|---|
| Thickness in meters | 60 | 20 |
| Heat conductivity factor W/m.K | 2.4 | 1.2 |
| Heat capacity $MJ/m^3.K$ | 2.6 | 2.0 |
| Horizontal Permeability m/day | 30.0 | |
| Porosity $m^3/m^3$ | 0.35 | |

Natural soil temperature 12 °C
Water level of the aquifer against zero level 0 meters wg

**Energy prices**

Electricity day tariff: 0.15 Dfl.kWh
Electricity night tariff 0.12 Dfl/kWh

Night tariff hours:

Sunday 24 hours
Monday/Friday 23-06 hours
Saturday 24 hours

**System overview**

| | | |
|---|---|---|
| AHU cooler | Maximum capacity | 1216 kW |
| | Water volume | 149 $m^3/h$ |
| | Temperature range | 13-20 °C |
| Storage capacity | Maximum capacity | 1814 kW |
| | Water volume | 149 $m^3/h$ |
| | Temperature range | 6-16,5 °C |

**Storage installation**

| | | |
|---|---|---|
| Storage cooling capacity | 145 MWh | 100 % |
| | **Cold well** | **Warm well** |
| Average injection temperature | 6.0 °C | 16.5 °C |
| Cut off temperature | 11.0 °C | |
| Maximum storage cooling capacity at: | | |
| Cold storage temperature 6.0 °C | 1814 kW | |
| Cold storage temperature 11.0 °C | 950 kW | |
| Estimated volume of ground water withdrawn | 15641 $m^3$ | |
| Distance between wells | 35 m | |
| Estimated storage efficiency in first cycle (allowance for capacity loss in first loading, mainly due to the heat capacity of the aquifer material) | 88 % | |
| Estimated storage efficiency in balance | 100 % | |
| | **Unloading** | **Loading** |
| Maximum well capacity | 149 $m^3/h$ | 149 $m^3/h$ |
| Minimum well capacity | 28 $m^3/h$ | 28 $m^3/h$ |
| Maximum water velocity on well surface | 1.5 m/h | |
| **Pressure loss between the wells:** | | |
| - during unloading at maximum capacity | 150 kPa | |
| - *during loading at maximum capacity* | 150 kPa | |
| Efficiency of ground water pumps | 50 % | |
| | **Cold well** | **Warm well** |
| Number of wells | 1 | 1 |
| Hole diameter | 0.53 m | 0.53 m |
| **Well pump at maximum capacity:** | | |
| - Total pressure loss | 201 kPa | 201 kPa |
| - Electrical power | 16.6 kW | 16.6 kW |

**Cooling coil**

| | Front side | Back side |
|---|---|---|
| Air temperature | 28.0 °C | 16.0 °C |
| Relative humidity | 60 % | 98 % |
| Enthalpy | 65.0 kJ/kg | 44.6 kJ/kg |
| Water temperature | 13.0 °C | 20.0 °C |
| | **Air side** | **Water side** |
| Volume | 89000 $m^3/h$ | 74.7 $m^3/h$ |
| Velocity | 2.2 m/s | 1.0 m/s |
| Pressure loss | 120 Pa | 39 kPa |
| Transported capacity | 608 kW | |
| Medium | Water | |
| Condensation ? | Yes | |
| Manufacturer | A | |
| Type | type 3 | |
| Fin thickness | 0.15 mm | |
| Fin pitch | 2.0 mm | |
| Horizontal tube pitch | 30.0 mm | |
| Vertical tube pitch | 30.0 mm | |
| Tube inside diameter | 15.7 mm | |
| Tube outside diameter | 16.5 mm | |
| Cooler sizes(h*b*l) | 336*335*24 cm | |
| Number of tube rows | 8 | |
| Air side surface | 2134 $m^2$ | |

**Chilled water control**

During the whole fresh air temp range 16 °C – 29 °C the chilled water temp will be constant 13 °C

| | |
|---|---|
| Control water volume | 2-way valve |

**Heat exchanger loading/unloading cold storage**

| | | |
|---|---|---|
| Film coefficient heat transfer | 4500 $W/m^2K$ | |
| Heat transfer surface | 138 $m^2$ | |
| | Primary (wells) | Secondary (building |
| water volume | 149 $m^3/h$ | 149 $m^3/h$ |
| entry temperature | 11.0 °C | 20.0 °C |
| discharge temperature | 18.0 °C | 13.0 °C |
| *pressure drop* | 50 kPa | 50 kPa |
| medium | Water | Water |
| transported capacity | 1217 kW | |
| ka-value | 621 kW/K | |
| *average temperature diff* | 2.0 K | |
| temperature efficiency | 78 | |

**Operation**

Operating hours for loading cold:

| | |
|---|---|
| Sunday | none |
| Monday/Friday | 8-18 h |
| Saturday | none |

Freeze control:

| | |
|---|---|
| Minimum *outside temperature* | -30.0 °C |
| Minimum injection temperature | 0.5 °C |
| Minimum cooler entrance temp | -30.0 °C |

When loading and unloading can occur at the same time loading has priority.

There is no requirement for a switch buffer for unloading of the storage installation.

# 8 Results of the simulation

Simulation year **1 of 3**

Start date 01-10
Last date 30-09

Seasons

Winter 1964-1965 (100%)
Summer1964 (100%)

**Cooling load [MWhl :**

| | Air handling unit AHU | Secondary | Apps/Processes | Total |
|---|---|---|---|---|
| Oct | | 0 | 0 | 0 |
| Nov | 0 | 0 | 0 | 0 |
| Dec | 0 | 0 | 0 | 0 |
| Jan | 0 | 0 | 0 | 0 |
| Feb | 0 | 0 | 0 | 0 |
| Mar | 0 | 0 | 0 | 0 |
| Apr | 2 | 0 | 0 | 2 |
| May | 17 | 0 | 0 | 17 |
| Jun | 44 | 0 | 0 | 44 |
| Jul | 51 | 0 | 0 | 51 |
| Aug | 38 | 0 | 0 | 38 |
| Sep | 17 | 0 | 0 | 17 |
| **Total** | **169** | **0** | **0** | **169** |

**Cooling production [MWh]**

| | Free Cooling | Storage | Mechanical | Total prod | Shortfall |
|---|---|---|---|---|---|
| Oct | 0 | 0 | 0 | 0 | 0 |
| Nov | 0 | 0 | 0 | 0 | 0 |
| Dec | 0 | 0 | 0 | 0 | 0 |
| Jan | 0 | 0 | 0 | 0 | 0 |
| Feb | 0 | 0 | 0 | 0 | 0 |
| Mar | 0 | 0 | 0 | 0 | 0 |
| Apr | 0 | 2 | 0 | 2 | 0 |
| May | 0 | 17 | 0 | 17 | 0 |
| Jun | 0 | 44 | 0 | 44 | 0 |
| Jul | 0 | 51 | 0 | 51 | 0 |
| Aug | 0 | 38 | 0 | 38 | 0 |
| Sep | 0 | 17 | 0 | 17 | 0 |
| **Total** | **0** | **169** | **0** | **169** | **0** |

**Storage Installation**

**Stored cooling (when possible) : Maximum**

| | Loading | Unloading |
|---|---|---|
| Water quantity | 29698 $m^3$ | 20292 $m^3$ |
| *Energy quantity* | 206 MWh | 169 MWh |
| Number of hours | 515 h | 578 h |
| Average injection temperature | 6.0 °C | 16.5 °C |
| *Average rejection temperature* | 12.0 °C | 9.3 °C |

| | | |
|---|---|---|
| Well temperature at 30-09 | Cold well | 11.5 °C |
| | Warm well | 14.1 °C |

Cut off temperature reached at 28-08

**Electrical energy**

| **Loading storage** | **Power kWh** | **Day In Dfl** | **Night In Dfl** | **Total In Dfl** |
|---|---|---|---|---|
| Cooler | 0 | 0 | *0* | 0 |
| Well pumps | 1127 | 169 | 0 | 169 |
| Circulation pumps | 269 | 40 | 0 | 40 |
| **Subtotal** | **1397** | **209** | ***0*** | **209** |
| | | | | |
| **Cooling Production** | | | | |
| Free cooling | 0 | 0 | 0 | 0 |
| Storage installation | 597 | 90 | 0 | 90 |
| Mechanical cooling | 0 | 0 | 0 | 0 |
| Circulation pumps | 169 | 25 | 0 | 25 |
| **Subtotal** | **766** | **115** | **0** | **115** |
| | | | | |
| **Total** | **2163** | **324** | **0** | **324** |

| | | | |
|---|---|---|---|
| Simulation year | **2 of 3** | Start date | 01-10 |
| | | Last date | 30-09 |
| Seasons | | Winter 1964-1965 | (100%) |
| | | Summer1964 | (100%) |

**Cooling load [MWhl :**

| | Air handling unit AHU | Secondary | Apps/Processes | Total |
|---|---|---|---|---|
| Oct | | 0 | 0 | 0 |
| Nov | 0 | 0 | 0 | 0 |
| Dec | 0 | 0 | 0 | 0 |
| Jan | 0 | 0 | 0 | 0 |
| Feb | 0 | 0 | 0 | 0 |
| Mar | 0 | 0 | 0 | 0 |
| Apr | 2 | 0 | 0 | 2 |
| May | 17 | 0 | 0 | 17 |
| Jun | 44 | 0 | 0 | 44 |
| Jul | 51 | 0 | 0 | 51 |
| Aug | 38 | 0 | 0 | 38 |
| Sep | 17 | 0 | 0 | 17 |
| **Total** | **169** | **0** | **0** | **169** |

**Cooling production [MWh]**

| | Free Cooling | Storage | Mechanical | Total prod | Shortfall |
|---|---|---|---|---|---|
| Oct | 0 | 0 | 0 | 0 | 0 |
| Nov | 0 | 0 | 0 | 0 | 0 |
| Dec | 0 | 0 | 0 | 0 | 0 |
| Jan | 0 | 0 | 0 | 0 | 0 |
| Feb | 0 | 0 | 0 | 0 | 0 |
| Mar | 0 | 0 | 0 | 0 | 0 |
| Apr | 0 | 2 | 0 | 2 | 0 |
| May | 0 | 17 | 0 | 17 | 0 |
| Jun | 0 | 44 | 0 | 44 | 0 |
| Jul | 0 | 51 | 0 | 51 | 0 |
| Aug | 0 | 38 | 0 | 38 | 0 |
| Sep | 0 | 17 | 0 | 17 | 0 |
| **Total** | **0** | **169** | **0** | **169** | **0** |

**Storage Installation**

**Stored cooling (when possible) :**

**Stored cold in the year before with correction for efficiency** **9%**

**Extra security** **5%**

(Correction factors for pre-conditioning phase in a decreasing series)

This quantity has been reached at 06 April

| | Loading | Unloading |
|---|---|---|
| Water quantity | 23233m$^3$ | 20385m$^3$ |
| Energy quantity | 206 MWh | 169 MWh |
| Number of hours | 456 h | 578 h |
| Average injection temperature | 6.0 °C | 16.5 °C |
| Average rejection temperature | 13.7 °C | 9.4 °C |

| | | |
|---|---|---|
| Well temperature at 30-09 | Cold well | 11.6 °C |
| | Warm well | 14.1 °C |

Cut off temperature reached at 27-08

**Electrical energy**

| **Loading storage** | **Power kWh** | **Day In Dfl** | **Night In Dfl** | **Total In Dfl** |
|---|---|---|---|---|
| Cooler | 0 | 0 | 0 | 0 |
| Well pumps | 751 | 113 | 0 | 113 |
| Circulation pumps | 155 | 23 | 0 | 23 |
| **Subtotal** | **906** | **136** | **0** | **136** |
| | | | | |
| **Cooling Production** | | | | |
| Free cooling | 0 | 0 | 0 | 0 |
| *Storage installation* | 606 | 91 | 0 | 91 |
| Mechanical cooling | 0 | 0 | 0 | 0 |
| Circulation pumps | 169 | 25 | 0 | 25 |
| **Subtotal** | **775** | **116** | **0** | **116** |
| | | | | |
| **Total** | **1681** | **252** | **0** | **252** |

| | | | |
|---|---|---|---|
| Simulation year | 3 of 3 | Start date | 01-10 |
| | | Last date | 30-09 |
| Seasons | | Winter 1964-1965 | (100%) |
| | | Summer1964 | (100%) |

**Cooling load [MWhl :**

| | Air handling unit AHU | Secondary | Apps/Processes | Total |
|---|---|---|---|---|
| Oct | | 0 | 0 | 0 |
| Nov | 0 | 0 | 0 | 0 |
| Dec | 0 | 0 | 0 | 0 |
| Jan | 0 | 0 | 0 | 0 |
| Feb | 0 | 0 | 0 | 0 |
| Mar | 0 | 0 | 0 | 0 |
| Apr | 2 | 0 | 0 | 2 |
| May | 17 | 0 | 0 | 17 |
| Jun | 44 | 0 | 0 | 44 |
| Jul | 51 | 0 | 0 | 51 |
| Aug | 38 | 0 | 0 | 38 |
| Sep | 17 | 0 | 0 | 17 |
| **Total** | **169** | **0** | **0** | **169** |

**Cooling production [MWh]**

| | Free Cooling | Storage | Mechanical | Total prod | Shortfall |
|---|---|---|---|---|---|
| Oct | 0 | 0 | 0 | 0 | 0 |
| Nov | 0 | 0 | 0 | 0 | 0 |
| Dec | 0 | 0 | 0 | 0 | 0 |
| Jan | 0 | 0 | 0 | 0 | 0 |
| Feb | 0 | 0 | 0 | 0 | 0 |
| Mar | 0 | 0 | 0 | 0 | 0 |
| Apr | 0 | 2 | 0 | 2 | 0 |
| May | 0 | 17 | 0 | 17 | 0 |
| Jun | 0 | 44 | 0 | 44 | 0 |
| Jul | 0 | 51 | 0 | 51 | 0 |
| Aug | 0 | 38 | 0 | 38 | 0 |
| Sep | 0 | 17 | 0 | 17 | 0 |
| **Total** | **0** | **169** | **0** | **169** | **0** |

**Storage Installation**

**Stored cooling (if possible) :**

| | |
|---|---|
| **Stored cold in the year before with correction for efficiency** | **5%** |
| **Extra security** | **5%** |

(Correction factors for pre-conditioning phase in a decreasing series)

This situation has been reached at 04 April

| | Loading | unloading |
|---|---|---|
| Water quantity | 22394 $m^3$ | 20477 $m^3$ |
| Energy quantity | 200 MWh | 169 MWh |
| Number of hours | 437 h | 578 h |
| Average injection temperature | 6.0 °C | 16.5 °C |
| Average rejection temperature | 13.7 °C | 9.3 °C |

| | | |
|---|---|---|
| Well temperature at 30-09 | Cold well | 11.5 °C |
| | Warm well | 14.1 °C |

Cut off temperature reached at 24-08

**Electrical energy**

| **Loading storage** | **Power kWh** | **Day In Dfl** | **Night In Dfl** | **Total In Dfl** |
|---|---|---|---|---|
| Cooler | 0 | 0 | 0 | 0 |
| Well pumps | 728 | 109 | 0 | 109 |
| Circulation pumps | 151 | 23 | 0 | 23 |
| **Subtotal** | **879** | **132** | **0** | **132** |
| | | | | |
| **Cooling Production** | | | | |
| Free cooling | 0 | 0 | 0 | 0 |
| Storage installation | 614 | 92 | 0 | 92 |
| Mechanical cooling | 0 | 0 | 0 | 0 |
| *Circulation pumps* | 169 | 25 | 0 | 25 |
| **Subtotal** | **783** | **117** | **0** | **117** |
| | | | | |
| **Total** | **1662** | **249** | **0** | **249** |

# IEA Annex 28

## Low Energy Cooling

### Subtask 2: Detailed Design Tools

# Program for the Simulation of Air-Earth Heat Exchangers

### Resistance Capacity Model

Arthur Huber
Huber EnergieTechnik, Zurich, Switzerland

Mark Zimmermann
EMPA, Swiss Federal Laboratories for Materials Testing and Research, Duebendorf, Switzerland

# Contents Chapter J

## 1. Technology area

Widerstands-Kapazitäten-Modell (WKM_LTe, resistance capacity model) is a calculation model for the simulation of air-earth heat exchangers. The model is based on the analogy of heat and electricity.

## 2. Developed by


| | | |
|---|---|---|
| Developed by | : | **Huber Energietechnik , CH-8032 Zurich (huber@igjzh.com)** |
| Contact name | : | **Mark Zimmermann** |
| Organisation | : | **EMPA, Swiss Federal Laboratories for Materials Testing and Research** |
| Address | : | **Ueberlandstrasse 129, CH-8600 Duebendorf** |
| Fax | : | **0041 1 821 62 44** |
| E-mail | : | **mark.zimmermann@empa.ch** |

# 3. General description

The WKM_LTe simulates the performance of an underground piping system. The outlet temperature is determined as a function of the inlet temperature. The mathematical analogy of heat and electricity allows to use an electrical circuit as model for the thermal behaviour of the ground. Heat capacity corresponds to electrical capacity and heat transfer resistance corresponds to electrical resistance.

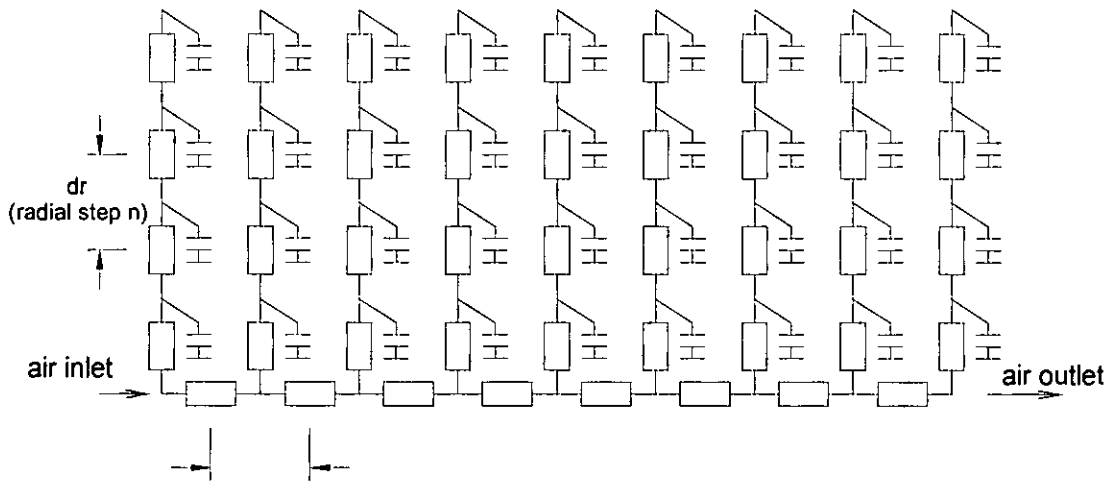



*Figure 1: An electrical circuit consisting of capacities and resistors is used as model for an air-earth heat exchanger*

To simplify the model the following assumptions are made:

- Heat conductivity of the ground in axial direction is neglected
- Heat capacity of air flowing in the piping system is also neglected

The circuit shown in figure 1 represents a single pipe. This is used as the basis for an underground piping system. The pipe is surrounded symmetrically in the radial direction by soil. It is useful to split the soil into several layers with different properties. In axial direction the geometry is divided into short sections (dl). Each section is represented by a branch of the electrical circuit. The thermal properties of each layer are described by a resistance and a capacity.

In addition the electrical branches are considered as discrete sticks. This geometry allows the use of the one-dimensional heat equation in unsteady state. The WKM_LTe solves this equation by using the numerical implicit Crank-Nicolson method.

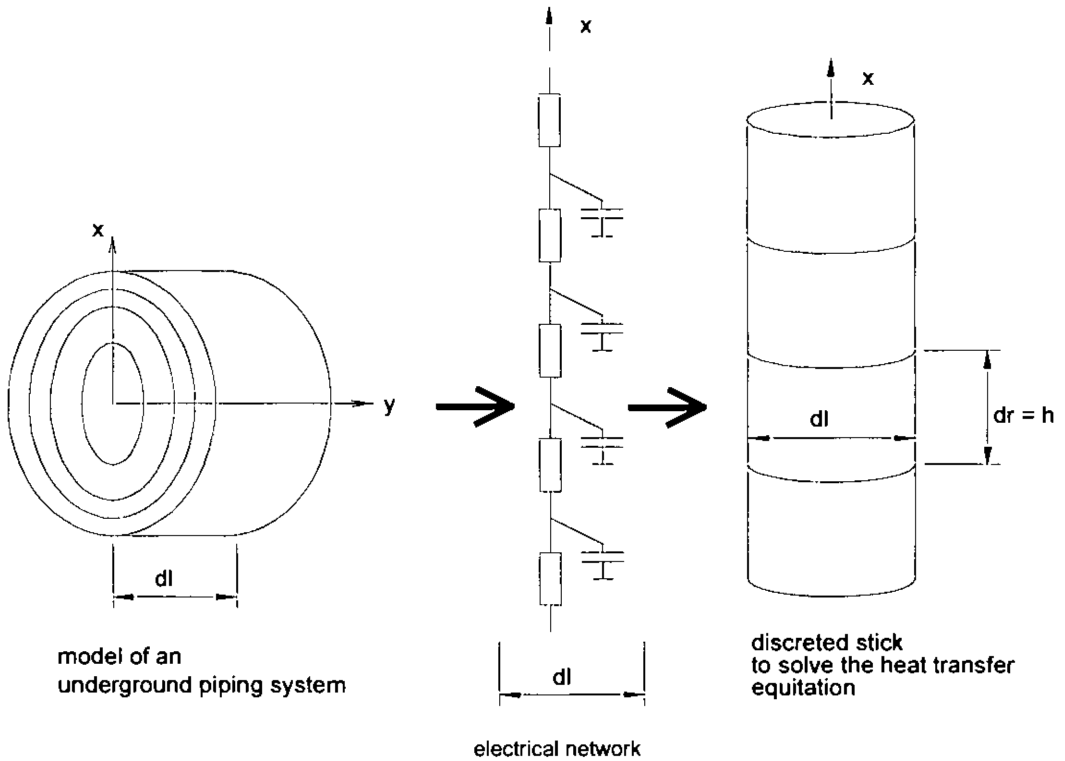



**Figure 2: Two steps to get the calculation model:**

**1.** A short section of a pipe surrounded by soil is represented by branch of an electrical circuit.
**2.** The branch is considered as a (one-dimensional) discreted stick. In the basic model the properties of each layer are used all around the pipe. Only one outer boundary condition can be considered. This doesn't agree exactly with the real conditions. Therefore the circumference is divided into three segments. Each of the three segments may have its own boundary conditions. Through this improvement different influences can be considered.

Usually different boundary conditions are set to model the influence from the ground, the top (influence of the basement of a building), and the side (adiabatic, if there are parallel pipes).

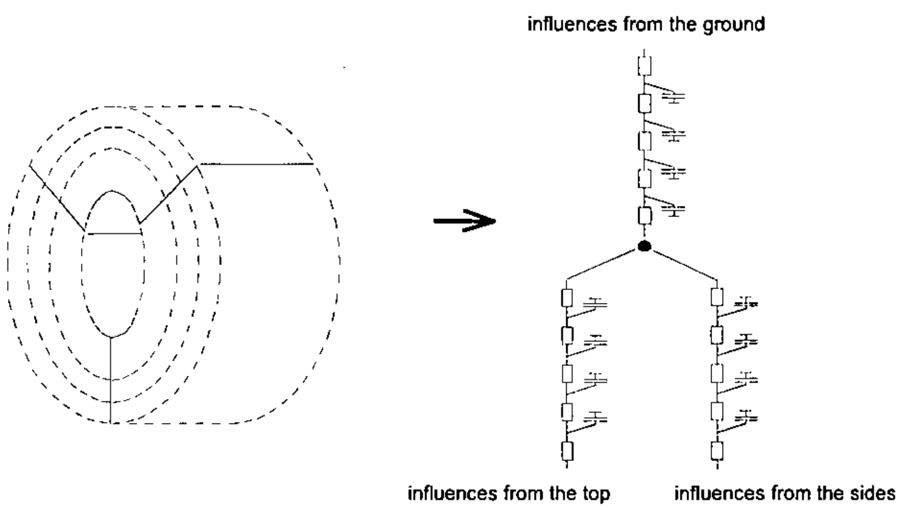



*Figure 3: Dividing the circumference into segments to consider more than one outer boundary condition. The dividing does not correspond to the real geometrical conditions. It serves only to consider different influences.*

To determine the resulting influence on the air in the pipe three independent branches have to be calculated and superimposed.
Therefore the arithmetical demands grow by approximately a factor of 3.

The size of the segments and with it the influence of the corresponding boundary condition has to be determined separately.

The inlet air temperature corresponds to the outside temperature. This temperature is the starting value for the first section of the pipe. The outlet air temperature of a section is passed to the next section as inlet air temperature. The outlet air temperature of the last pipe section is the outlet air temperature of the piping system.

The WKM_LTe is designed to work with hourly temperature values. Once an hour the calculation is made simultaneously for the whole piping system.

# 4. Nomenclature

| | | | |
|---|---|---|---|
| $A$ | = | matrix | |
| $C$ | = | heat capacity | J/K |
| $c_{earth}$ | = | specific heat capacity of earth | J/kg K |
| $c_{air}$ | = | specific heat capacity of air | J/kg K |
| $cp$ | = | specific heat capacity | J/kg K |
| $c_{pipe}$ | = | specific heat capacity of the pipe | J/kg K |
| $D$ | = | diameter | m |
| $dl$ | = | length of a pipe-section | m |
| $dt$ | = | time step | s |
| $f$ | = | grid factor | |
| $F$ | = | matrix | |
| $L$ | = | heat conductance ( = 1/R) | W/K |
| $Nu$ | = | Nusselt number | |
| $Pr$ | = | Prandtl number | |
| $\dot{Q}$ | = | heat flux | W |
| $R$ | = | heat resistance | K/W |
| $r$ | = | radius | m |
| $Re$ | | Reynolds number | |
| $T$ | = | temperature | °C |
| $t$ | = | time | s |
| $Ta$ | = | time period | s |
| $Te$ | = | ground temperature | °C |
| $Tm$ | = | average temperature | °C |
| $To$ | = | temperature amplitude | °C |
| $x$ | = | x-coordinate | m |
| | | | |
| $\alpha$ | = | heat transfer coefficient | $W/m^2K$ |
| $\alpha_{out}$ | = | heat transfer coefficient (earth surface) | $W/m^2K$ |
| $\lambda$ | = | heat conductivity | W/m K |
| $\mu_{air}$ | = | dynamic viscosity of air | kg/m s |
| $\nu_{air}$ | = | kinematic viscosity of air | $m^2/s$ |
| $\rho$ | = | density | $kg/m^3$ |
| $\rho_{earth}$ | = | density of earth | $kg/m^3$ |
| $\rho_{air}$ | = | density of air | $kg/m^3$ |
| $\rho_{pipe}$ | = | density of pipe material | $kg/m^3$ |

indices

| | | |
|---|---|---|
| $i$ | = | axial direction |
| $j$ | = | radial direction |
| $k$ | = | time step |
| $W$ | = | wall |
| $\alpha$ | = | inlet |
| $\omega$ | = | outlet |

# 5. Mathematical description

The calculation consists of a radial and an axial part. The radial part determines the radial temperature field by solving the one-dimensional heat equation for each layer. The axial step determines the heat transport by the air in the pipe.

## 5.1 Heat transport

There are two different kind of heat transport:

- Enforced convective heat transport of the air
- Molecular heat conduction in the ground

Also important is also the heat transfer from the air to the wall of the pipe. Turbulent flow conditions are assumed, which are described by the following expression for the Nusselt number:

$$Nu = 0.021\, Pr^{0.5}\, Re^{0.8} = \frac{\alpha\, D}{\lambda}$$

$$Re = \frac{v\, D\, \rho_{air}}{\mu_{air}} \quad , \quad \nu_{air} = \frac{\mu_{air}}{\rho_{air}}$$

$$Pr = \frac{\nu_{air}\, \rho_{air}\, c_p}{\lambda} \tag{1}$$

The heat transfer coefficient can be calculated from this:

$$\alpha = 0.021\, Pr^{0.5}\, v^{0.8}\, D^{-0.2}\, \mu_{air}^{-0.8}\, \rho_{air}^{0.8}\, \lambda_{air} \tag{2}$$

## 5.2 Radial direction

In radial direction the one-dimensional heat equation or Fourier equation has to be solved.

$$\frac{\partial T}{\partial t} = a\frac{\partial^2 T}{\partial x^2} \quad \text{with } T = T(t,x) \text{ and } \mathrm{a} = \frac{\lambda}{\rho \mathrm{c_p}} \tag{3}$$

As an implicit equation of differences it is written as:

$$\begin{aligned} &T_{k+1,j} - \frac{dt}{2}\frac{L_j}{C_j}\left(T_{k+1,j-1} - T_{k+1,j}\right) - \frac{dt}{2}\frac{L_{j+1}}{C_j}\left(T_{k+1,j+1} - T_{k+1,j}\right) = \\ &T_{k,j} + \frac{dt}{2}\frac{L_j}{C_j}\left(T_{k,j-1} - T_{k,j}\right) + \frac{dt}{2}\frac{L_{j+1}}{C_j}\left(T_{k,j+1} - T_{k,j}\right) \end{aligned} \tag{4}$$

Index k belongs to the time coordinate and index j to the radial coordinate. C is the capacity which is described below. L is the conductance, the reciprocal of resistance:

$$L = \frac{1}{R} = \frac{\dot{Q}}{\Delta T} \tag{5}$$

### 5.3 Definition of capacities and resistors

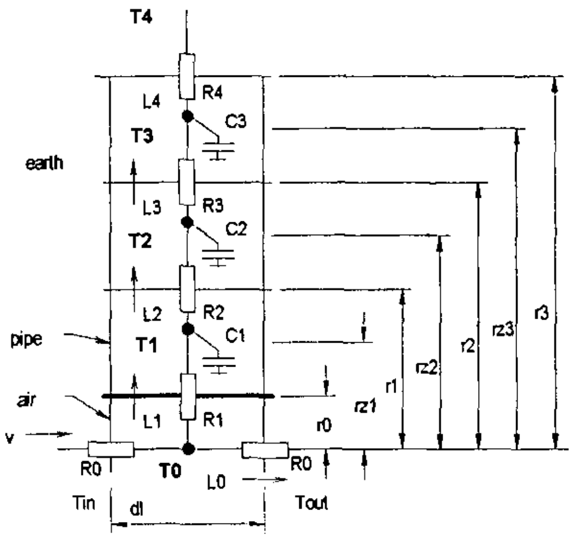



*Figure 3: Overview of the naming of a pipe section*

Heat capacities are defined for the pipe wall and for all layers of the surrounding ground. The heat capacity of the air is ignored.

**Capacity of the pipe wall:**

$$C_1 = c_{\text{pipe}}\, \rho_{\text{pipe}}\, \pi\, (r_1^2 - r_0^2)\, dl \tag{6}$$

**Capacity of the ground (depending on the distance):**

$$C_2 = c_{\text{earth}}\, \rho_{\text{earth}}\, \pi\, (r_2^2 - r_1^2)\, dl \tag{7}$$
$$C_3 = c_{\text{earth}}\, \rho_{\text{earth}}\, \pi\, (r_3^2 - r_2^2)\, dl$$

**A heat resistance for the flowing air in axial direction is defined:**

$$R_0 = \frac{1}{\pi\, r_0^2\, v\, c_{air}\, \rho_{air}} \tag{8}$$

$$
\begin{aligned}
R_1 &= \frac{1}{2\,\pi\,\alpha\,r_0\,dl} + \frac{1}{2\,\pi\,\lambda_{\text{pipe}}\,dl} ln\frac{r_{z1}}{r_0} \\
R_2 &= \frac{1}{2\,\pi\,dl}\left(\frac{1}{\lambda_{\text{pipe}}} ln\frac{r_1}{r_{z1}} + \frac{1}{\lambda_{\text{earth}}} ln\frac{r_{z2}}{r_1}\right) \\
R_3 &= \frac{1}{2\,\pi\,dl}\frac{1}{\lambda_{\text{earth}}} ln\frac{r_{z3}}{r_{z2}} \\
R_4 &= R_3 \quad \text{or at a surface:} \\
R_4 &= \frac{1}{2\,\pi\,dl}\frac{1}{\lambda_{\text{earth}}} ln\frac{r_3}{r_{z3}} + \frac{1}{2\,\pi\,\alpha_{out}\,r_3\,dl}
\end{aligned}
\tag{9}
$$

### 5.4 Arithmetical grid

The grid in radial direction is variable. It is defined by the grid factor f:

$$
\text{grid factor } f = \frac{r_{j+1} - r_j}{r_j - r_{j-1}} \tag{10}
$$

A grid factor of 2 doubles the difference of the radiuses of two neighbouring calculation volumes. The simulation area is defined by pre-setting a maximum radius. The grid is then given by the following expression:

$$
\begin{aligned}
r_0 &= \text{ inner radius of the pipe} \\
r_1 &= \text{ outer radius of the pipe} \\
r_m &= \text{ maximum radius} \\
j \ge 2:\;\; r_j &= r_{j-1} + (r_m - r_1)\frac{1-f}{1-f^{m-1}} f^{j-2}
\end{aligned}
\tag{11}
$$

Each layer around the pipe corresponds to an arithmetical node, which is described by a heat equation (4). All equations for a pipe section segment are put together in matrices (example with 3 arithmetical nodes, see also figure 3):

$$
\begin{bmatrix}
1 & 0 & 0 & 0 & 0 \\
-dt\,L_1 & 2C_1 + dt\,L_1 + dt\,L_2 & -dt\,L_2 & 0 & 0 \\
0 & -dt\,L_2 & 2C_2 + dt\,L_2 + dt\,L_3 & -dt\,L_3 & 0 \\
0 & 0 & -dt\,L_3 & 2C_3 + dt\,L_3 + dt\,L_4 & -dt\,L_4 \\
0 & 0 & 0 & 0 & 1
\end{bmatrix}
\begin{Bmatrix} T_0 \\ T_1 \\ T_2 \\ T_3 \\ T_4 \end{Bmatrix}_i^{k+1} =
$$

$$
\begin{bmatrix}
1 & 0 & 0 & 0 & 0 \\
dt\,L_1 & 2C_1 - dt\,L_1 - dt\,L_2 & dt\,L_2 & 0 & 0 \\
0 & dt\,L_2 & 2C_2 - dt\,L_2 - dt\,L_3 & dt\,L_3 & 0 \\
0 & 0 & dt\,L_3 & 2C_3 - dt\,L_3 - dt\,L_4 & dt\,L_4 \\
0 & 0 & 0 & 0 & 1
\end{bmatrix}
\begin{Bmatrix} T_0 \\ T_1 \\ T_2 \\ T_3 \\ T_4 \end{Bmatrix}_i^{k}
\tag{12}
$$

or symbolically:

$$[A]\cdot\{T\}_i^{k+1} = [F]\cdot\{T\}_i^k \tag{13}$$

To find the new temperature field the Matrix A has to be inverted:

$$\{T\}_i^{k+1} = [A]^{-1}\cdot[F]\cdot\{T\}_i^k \tag{14}$$

This is the description of the temperature field for the pipe section i at the time k+1. Now the air temperature (T0) then has to be passed from section i to section i+1.

In the program the two steps are not performed together for each segment. The air temperature is calculated first for all the sections, then the radial calculations for all the sections, and so on.

## 5.5 Axial heat transport

In axial direction heat is transferred only by the flowing air. The axial heat transport in the ground is ignored. Because the heat capacity of the air is ignored a steady statement for the axial heat transport can be used. It is based on an energy balance for a section of the pipe.

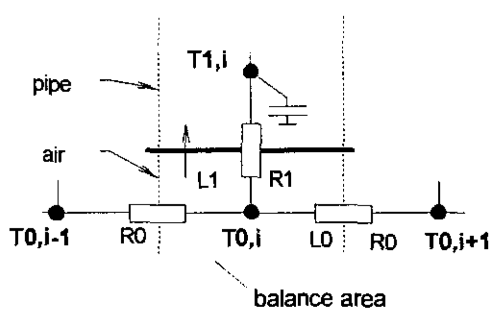



*Figure 4: Balance area to determine the axial heat transport*

$$(T_{k,0})_i = (T_{k,0})_{i-1}\frac{L_0}{L_0+L_1} + (T_{k,1})_i\frac{L_1}{L_0+L_1} \tag{15}$$

## 5.6 Starting condition

At the beginning of the simulation the temperature in the whole area is set to earth temperature. To warm up the system three months preconditioning is used prior to the actual calculation.

For each electrical branch (or discreted stick) an inner and an outer boundary condition have to be defined. There are several possibilities:

## 5.7 Boundary conditions

***Inner boundary condition:***

1. The ventilation is running. Heat transfer coefficient $\alpha > 0$
2. The ventilation is not operating. Heat transfer coefficient is set to $\alpha = 0$

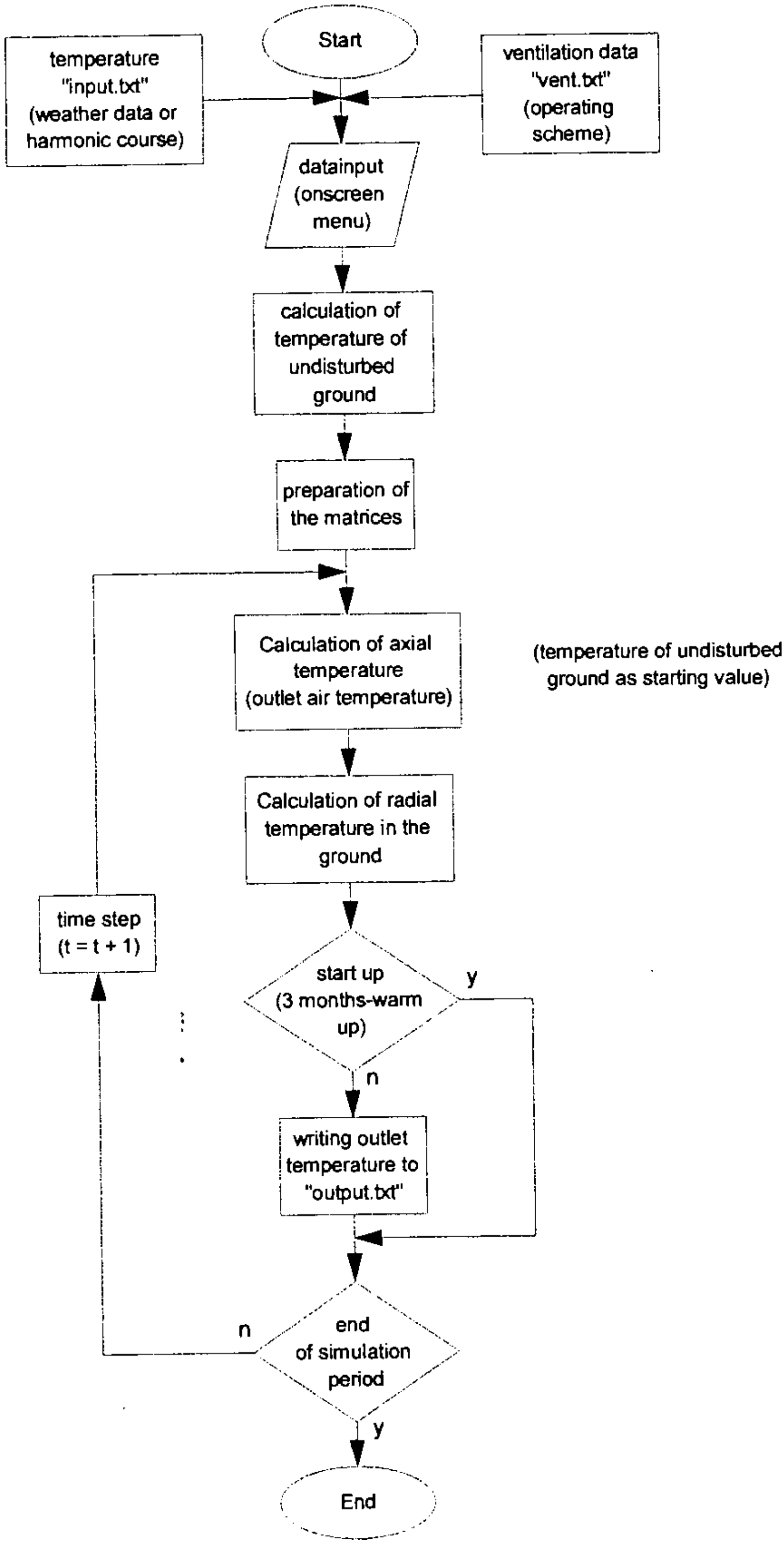
***Outer boundary condition:***

1. The temperature outside of the simulation area is the temperature of the undisturbed ground depending on the depth (share of ground)

$$T_e(x,t) = T_m + T_0 e^{-x\sqrt{\frac{\pi}{aT_a}}} \cos\left(\frac{2\pi}{T_a}t - x\sqrt{\frac{\pi}{aT_a}}\right) \tag{16}$$

2. The influences of a building may be considered by a constant temperature (corresponding to the average temperature of the basement) as boundary condition (share of basement)
3. Adiabatic case, the last heat conductance of an electrical branch is set Ln = 0 (adiabatic share)

# 6. Flowchart


Start
temperature
"input.txt"
(weather data or
harmonic course)
ventilation data
"vent.txt"
(operating
scheme)
datainput
(onscreen
menu)
calculation of
temperature of
undisturbed
ground
preparation of
the matrices
Calculation of axial
temperature
(outlet air temperature)
(temperature of undisturbed
ground as starting value)
Calculation of radial
temperature in the
ground
time step
(t = t + 1)
start up
(3 months-warm
up)
y
n
writing outlet
temperature to
"output.txt"
end
of simulation
period
n
y
End

# 7. Source code

```
Program WKM_LTe;

Uses crt;

Const  dim                       = 6;                  (* number of nodes/grid
*)
       pi                        = 3.14159;
       def_cp_air                = 1005;                          (* J/kgK
*)
       def_cp_pipe               = 2150;                          (* J/kgK
*)
       def_cp_earth              = 1400;                          (* J/kgK
*)
       def_rho_air               = 1.2;                           (* kg/m3
*)
       def_rho_pipe              = 2000;                          (* kg/m3
*)
       def_rho_earth             = 1400;                          (* kg/m3
*)
       def_nu_air                = 1.45E-5;                       (* m2/s
*)
       def_lambda_air            = 0.024;                         (* W/mK
*)
       def_LambdaPipe            = 0.35;                          (* W/mK
*)
       def_LambdaEarth           = 1.5;                           (* W/mK
*)
       def_D                     = 0.1922;             (* m , tube diameter
*)
       def_PipeLength            = 30;                            (* m
*)
       def_PipeWallThickness     = 0.0039;                        (* m
*)
       def_NumberPipes           = 15;                            (* -
*)
       def_depth                 = 1;                             (* m
*)
       def_DepthPenetration      = 0.5;                           (* m
*)
       grid_factor               = 2;                             (* -
*)
       def_VolumeFlowRate        = 3700;                          (* m3/h
*)
       TimeStep                  = 3600;                          (* s
*)
       def_inputfile             = 'input.txt';
       def_outputfile            = 'output.txt';
       def_VentilationService    = 'service.txt';
       def_ventdat               = 'N';             (* running time of system
*)
       def_TempBasement          = 15;                            (* øC
*)
       def_ShareAdabatic         = 0.5;
       def_ShareBasement         = 0;


type   HourlyValues = array[0..24,0..31] of real;
       vector       = array[0..dim+1] of real;
       matrix       = array[0..dim+1,0..dim+1] of real;
```

```
Var     T,Tadiabat,Tbasement,E,Ead,H,Had                        : matrix;
        Lrun,Lstop,Lrunad,Lstopad,C,r,rz                        : vector;
        AirTemp,SystemOutTemp,
        VentDat                                                 : HourlyValues;
        dl,Tin,AverageTemp,
        InTemp,GroundTemp,InTempOld,OutTemp,
        OutTempOld,cp_air,cp_pipe,cp_earth,rho_air,
        rho_pipe,rho_earth,nu_air,lambda_air,
        DepthPenetration,D,PipeLength,PipeWallThickness,v,
        LambdaPipe,LambdaEarth,VolFlowRate,
        depth,YearlyAverage,k1,k2,k3,
        ServiceTime,TempBasement,
        ShareAdiabat,ShareBasement,ShareGround                : real;
        i,j,hour,month,day,NumberPipes                        : integer;
        WarmUp,operating,VentService                          : boolean;
        inputfile,outputfile,s,filename2                      : string;
        MonthlyAverageGround,MonthlyAverage                   : array[0..12] of real;


Function BeginningMonth(Month : integer):integer;
                      (*relates the yearly day number to the first day of a month*)
   var  i: integer;
   begin
     case Month of  1 : i:= 0;
                    2 : i:= 31;
                    3 : i:= 59;
                    4 : i:= 90;
                    5 : i:= 120;
                    6 : i:= 151;
                    7 : i:= 181;
                    8 : i:= 212;
                    9 : i:= 243;
                   10 : i:= 273;
                   11 : i:= 304;
                   12 : i:= 334;  end;
     BeginningMonth:=1+24*i;
   end;


Function LengthMonth(Month : integer):integer;
                              (*relates the number of days to each month*)
   var  i: integer;
   begin
     case Month of  1 : i:= 31;
                    2 : i:= 28;
                    3 : i:= 31;
                    4 : i:= 30;
                    5 : i:= 31;
                    6 : i:= 30;
                    7 : i:= 31;
                    8 : i:= 31;
                    9 : i:= 30;
                   10 : i:= 31;
                   11 : i:= 30;
                   12 : i:= 31;  end;
     LengthMonth:=i;
   end;


Procedure ReadText (u : string);        (*reads a text file and shows        *)
   var   f  : text;                     (*it on the screen                   *)
         ch : char;
   begin
     clrscr;
     assign(f,u);
     reset(f);
     if IORESULT <> 0 then begin
       writeln;
```

```
      writeln ('File ',u,' not found. ');
      writeln ('Copy file ',u,' to current directory !');
      writeln;
      halt(1);
    end else begin
      while not eof(f) do begin
        read(f,ch);
        write(ch);
      end;
      close(f);
    end;
  end;


Function Variable( x,y                    : byte;    (*reads a variable or takes a *)
      default,Varmin,VarMax : real     ) : real;    (*default value              *)
  var value   : real;
      s       : string;
      i       : integer;
  begin
    if (default<VarMin) or (default>VarMax) then begin
      Variable:=default;
      gotoxy(x,y);
      writeln('Out of pre-set range ');
      readln;
    end else begin
      value:=Default;
      repeat
        gotoxy(x,y);
        write(value:5:5);
        write('          ');
        gotoxy(x+10,y);
        readln(s);
        if s<>'' then val(s,value,i)
        until (value>=VarMin) and (value<=VarMax);
      gotoxy(x,y);
      HighVideo;
      write(value:5:2);
      NormVideo;
      write('        ');
      Variable:=value;
    end;
  end;


Function VarInteger( x,y                      : byte;  (*reads a integer variable or*)
  default,Varmin,VarMax : integer     ) : integer;  (*takes a default value       *)
  var value  : integer;
      s      : string;
      i      : integer;
  begin
    if (default<VarMin) or (default>VarMax) then begin
      VarInteger:=default;
      gotoxy(x,y);
      writeln('Out of pre-set range');
      readln;
    end else begin
      value:=default;
      repeat
        gotoxy(x,y);
        write(value:5);
        write('          ');
        gotoxy(x+10,y);
        readln(s);
        if s<>'' then val(s,value,i)
      until (value>=VarMin) and (value<=VarMax);
      gotoxy(x,y);
      HighVideo;
```

```
      write(value:5);
      NormVideo;
      write('        ');
      VarInteger:=value;
    end;
  end;


Function VarString ( x,y                    : byte;     (*reads a string or takes      *)
  default,s1,s2,s3,s4 : string      ) : string;     (*a default string             *)
  var value   : string;
      s       : string;
      i       : integer;
  begin
    if not ((s1='-') or (default=s1) or (default=s2) or (default=s3)
        or (default=s4)) then begin
        VarString:=default;
        gotoxy(x,y);
        writeln('Out of pre-set range');
        readln;
      end else begin
      value:=Default;
      repeat
        gotoxy(x,y);
        write(value);
        write('          ');
        gotoxy(x+10,y);
        readln(s);
        if s<>'' then value:=s;
      until ((value=s1) or (value=s2) or (value=s3) or (value=s4) or (s1='-'));
      gotoxy(x,y);
      HighVideo;
      write(value);
      NormVideo;
      write('          ');
      VarString:=value;
    end;
  end;


Procedure New(filename,inputfile : string);             (* prepares an outputfile *)
  var f : text;
  begin
   assign(f,filename);
   rewrite(f);
   close(f);
  end;


Procedure Input                                         (* provides the data input
*)
   var str    : string;
      next   : boolean;
   begin
     ReadText('lerle.txt');
     readln;
     Repeat begin
       ReadText('ler2e-1.txt');
       cp_air           := Variable(40,3,def_cp_air,1,5000);
       cp_pipe          := Variable(40,4,def_cp_pipe,1,5000);
       cp_earth         := Variable(40,5,def_cp_earth,1,5000);
       rho_air          := Variable(40,6,def_rho_air,0.001,10);
       rho_pipe         := Variable(40,7,def_rho_pipe,0.001,10000);
       rho_earth        := Variable(40,8,def_rho_earth,0.001,10000);
       nu_air           := Variable(40,9,def_nu_air,0.000001,10);
       lambda_air       := Variable(40,10,def_lambda_air,0.00001,10);
       LambdaPipe       := Variable(40,11,def_LambdaPipe,0.1,5);
       LambdaEarth      := Variable(40,12,def_LambdaEarth,0.5,5);
```

```
      D                  := Variable(40,13,def_D,0.001,10);
      PipeLength         := Variable(40,14,def_PipeLength,1,200);
      PipeWallThickness:= Variable(40,15,def_PipeWallThickness,0.0001,0.05);
      NumberPipes        := VarInteger(40,16,def_NumberPipes,1,1000);
      VolFlowRate        := Variable(40,17,def_VolumeFlowRate,1,1000000);
      v                  := VolFlowRate/3600/pi/sqr(D)*4/NumberPipes;
                            gotoxy(40,1);  write(v:4:1);
      depth              := Variable(40,19,def_depth,0,20);
      DepthPenetration := Variable(40,20,def_DepthPenetration,0.01,10);
      str                := VarString(44,22,'N','y','Y','N','n');
        if ((str='n') or (str='N')) then next:=true else next:=false;
        end until next;
    Repeat begin
      ReadText('ler2e-2.txt');
      ShareAdiabat       := Variable(40,3,def_ShareAdabatic,0,1);
      ShareBasement      := Variable(40,4,def_ShareBasement,0,1-ShareAdiabat);
      ShareGround        := 1 - ShareAdiabat - ShareBasement;
        if ShareBasement > 0 then
      TempBasement       := Variable(40,5,def_TempBasement,-10,30);
      Inputfile          := VarString(40,6,def_inputfile,'-','-','-','-');
      Outputfile         := VarString(40,7,def_outputfile,'-','-','-','-');
      VentService          := false;
      str                := VarString(44,8,def_ventdat,'y','Y','n','N');
        if ((str='y') or (str='Y')) then VentService:=true else
VentService:=false;
        if VentService then
      filename2:=VarString(40,9,def_VentilationService,'-','-','-','-');
      str                := VarString(44,11,'N','y','Y','N','n');
        if ((str='n') or (str='N')) then next:=true else next:=false;
    end until next;
  end;

Procedure Output(filename:string; month:integer;         (*writes the inlet and    *)
                     TempIn,TempOut:HourlyValues);         (*outlet air  temperature *)
  var f,f2   : text;                                         (*to "output.txt" file
*)
      i,j,k  : integer;
      s      : string;
  begin
    assign(f,filename);  append(f);
    for i:=1 to LengthMonth(month) do
    for j:=1 to 24 do begin
      write(f,TempIn[j,i]:1:1);
      write(f,';');
      write(f,TempOut[j,i]:1:1);
      writeln(f);
    end;
  close(f);
 end;


Function expo (a,b : real) : real;                                      (* a^b : real
*)
  begin
    expo:=exp(b*ln(abs(a)));
  end;


Procedure multiply (M:matrix; w:vector; var y:matrix; k: integer);
                                                                  (* {y}  = [M] x {w}
*)
  var
    i,j   : integer;
  begin
    for i:=0 to dim+1 do begin
      y[k,i]:=0;
      for j:=0 to dim+1 do y[k,i]:=y[k,i] + M[i,j] * w[j];
    end;
```

```
  end;


Procedure multiMatrix (var Ainv,F,E : matrix);                     (* E = Ainv x F
*)
  var
    i,j,k    : integer;
  begin
    for k:=0 to dim+1 do begin
      for i:=0 to dim+1 do begin
        E[k,i]:=0;
        for j:=0 to dim+1 do E[k,i]:=E[k,i] + Ainv[k,j] * F[j,i];
      end;
    end;
  end;


Procedure invert (A : matrix; var Ain : matrix);                   (* Ainv = 1/A
*)
  var
    pivot    : real;
    i,j,g    : integer;
  begin
    for i:=0 to dim+1 do for j:=0 to dim+1 do Ain[i,j]:=A[i,j];
    for g:=0 to dim+1 do begin
       pivot:=Ain[g,g];
       for j:=0 to dim+1 do Ain[g,j]:=Ain[g,j] * (-1) / pivot;
       for i:=0 to dim+1 do begin
         for j:=0 to dim+1 do
           if (i<>g) and (j<>g) then Ain[i,j]:=Ain[g,j]*Ain[i,g]+Ain[i,j];
         Ain[i,g] := Ain[i,g] / pivot;
       end;
       Ain[g,g] := 1 / pivot ;
     end;
  end;


Function Alpha : real;                                  (* heat transfer coefficient
*)
  var  Re,Pr,Nu : real;
  begin
    Re := v * D / nu_air;
    Pr := nu_air * rho_air * cp_air / lambda_air;
    Nu := 0.021 * Sqrt(Pr) * expo(Re,0.8);
    Alpha := Nu * lambda_air / D;
  end;


Procedure DefMatrixA (L,C : vector; dt : real; var A : matrix );
 var i,j : integer;
 begin
   for i:=0 to dim+1 do for j:=0 to dim+1 do A[i,j]:=0;
   A[0,0] := 1;  A[dim+1,dim+1] := 1;
   for i:=1 to dim do begin
     A[i,i] := 2 * C[i] + dt * L[i] + dt * L[i+1];
     A[i,i-1] := -dt * L[i];
     A[i,i+1] := -dt * L[i+1];
   end;
 end;


Procedure DefMatrixF (L,C : vector; dt : real; var F : matrix );
 var i,j : integer;
 begin
   for i:=0 to dim+1 do for j:=0 to dim+1 do F[i,j]:=0;
   F[0,0] := 1;  F[dim+1,dim+1] := 1;
   for i:=1 to dim do begin
     F[i,i] := 2 * C[i] - dt * L[i] - dt * L[i+1];
```

```
    F[i,i-1] := dt * L[i];
    F[i,i+1] := dt * L[i+1];
  end;
end;


Procedure SetMatrices ( dt : real );                (* definition of the matrices
*)
 var  i,j                          : integer;
      factor                       : real;
      A,B,Aad,Bad,F,G,Fad,Gad,
      Ainv,Binv,Ainvad,Binvad      : matrix;
      L,Lrun,Lstop,Lrunad,Lstopad : vector;
 begin
   for i:=0 to dim+1 do for j:=0 to dim+1 do begin
      T[i,j]          := GroundTemp;                          (*starting
condition*)
      Tadiabat[i,j] := GroundTemp;                            (*starting
condition*)
      Tbasement[i,j]:= TempBasement;                          (*starting
condition*)
   end;
   r[0] := D / 2;   r[1] := r[0] + PipeWallThickness;
   factor := DepthPenetration * (1-grid_factor) / (1 - expo(grid_factor,dim-1));
   for i:=2 to dim do r[i] := r[i-1] + factor*expo(grid_factor,i-2);
   for i:=1 to dim do rz[i]:= sqrt((sqr(r[i])+sqr(r[i-1]))/2);
   L[0] := pi * sqr(r[0]) * v * rho_air * cp_air;
                                  (*heat transfer resistance in axial
direction*)
   L[1] := 1 / (1/2/pi/Alpha/r[0]/dl + 1/2/pi/LambdaPipe/dl*ln(rz[1]/r[0]));
                                                     (*radial heat transfer
*)
   L[2] := 1 / (1/2/pi/dl*(ln(r[1]/rz[1])/LambdaPipe+ln(rz[2]/r[1])/LambdaEarth));
   for i:=3 to dim do L[i]:=1/(ln(rz[i]/rz[i-1])/2/pi/LambdaEarth/dl);
                                            (*radial heat transfer
resistance*)
   C[1] := cp_pipe * rho_pipe * pi * (sqr(r[1])-sqr(r[0])) * dl;
                                               (* heat capacity of the tube
*)
   for i:=2 to dim do
   C[i] := cp_earth * rho_earth * pi * (sqr(r[i])-sqr(r[i-1])) * dl;
                                              (* heat capacity of the
ground*)

   Lrun[1] := L[1];                                  (* ventilation is running
*)
   for i:=2 to dim do Lrun[i]:=L[i];
   Lrun[dim+1] := L[dim];
   DefMatrixA(Lrun,C,dt,A);                           (*  A * Tneu = F * Talt
*)
   DefMatrixF(Lrun,C,dt,F);
   invert(A,Ainv);                                    (* Ainv   = 1/A
*)
   Multimatrix(Ainv,F,E);                             (* E      = Ainv * F
*)

   Lstop[1] := 0;                            (* ventilation is out of operation
*)
   for i:=1 to dim do Lstop[i]:=L[i];
   Lstop[dim+1] := L[dim];
   DefMatrixA(Lstop,C,dt,B);                          (*  B * Tneu = G * Talt
*)
   DefMatrixF(Lstop,C,dt,G);
   invert(B,Binv);                                    (* Binv   = 1/B
*)
   Multimatrix(Binv,G;H);                             (* H      = Binv * G
*)
```

```
   Lrunad[1] := L[1];                          (* ventilation is running, adiabatic
*)
   for i:=1 to dim do Lrunad[i]:=L[i];
   Lrunad[dim+1] := 0;
   DefMatrixA(Lrunad,C,dt,Aad);                          (*  Aad * Tneu = Cad * Talt
*)
   DefMatrixF(Lrunad,C,dt,Fad);
   invert(Aad,Ainvad);                                   (* Ainvad    = 1/Aad
*)
   Multimatrix(Ainvad,Fad,Ead);                          (* Ead       = Ainvad • Fad
*)

   Lstopad[1] := 0;                  (* ventilation is out of operation, adiabatic
*)
   for i:=1 to dim do Lstopad[i]:=L[i];
   Lstopad[dim+1] := 0;
   DefMatrixA(Lstopad,C,dt,Bad);                         (*  Bad * Tneu = Gad * Talt
*)
   DefMatrixF(Lstopad,C,dt,Gad);
   invert(Bad,Binvad);                                   (* Binvad = 1/Bad
*)
   Multimatrix(Binvad,Gad,Had);                          (* Had    = Binvad • Gad
*)

   k1 := L[0] / (L[0]+L[1]);
   k2 := L[1] / (L[0]+L[1]);
   operating:=true;  ServiceTime:=0;
 end;


Procedure Calculate ( dt : real; k : integer);     (*calculates the ground
*)
 var i,j    : integer;                             (*temperature field
•)
   Temp,Tempad,TempBasement : vector;
 begin
   for i:=0 to dim+1 do Temp[i] := T[k,i];
   for i:=0 to dim+1 do Tempad[i] := Tadiabat[k,i];
   for i:=0 to dim+1 do TempBasement[i] := Tbasement[k,i];
   if operating then multiply(E,Temp,T,k)                          (*  T = E *
Talt*)
                 else multiply(H,Temp,T,k);                        (*  T = H • Tal
*)
   if operating then multiply(E,TempBasement,Tbasement,k)
                 else multiply(H,TempBasement,Tbasement,k);
   if operating then multiply(Ead,Tempad,Tadiabat,k)         (*
Tadiabat=Ead*Tadalt*)
                 else multiply(Had,Tempad,Tadiabat,k);       (*
Tadiabat=Had*Tadalt*)
 end;


Function OutTemperature( Tin : real ): real;  (*performs the axial heat transport*)
 var i,j : integer;
 begin
   T[0,0] := Tin;
   for i:=1 to dim do begin
     T[i,0] := T[i-1,0] * k1 + T[i,1] * k2 • ShareGround
                             + Tadiabat[i,1] * k2 • ShareAdiabat
                             + Tbasement[i,1] • k2 * ShareBasement;
     Tadiabat[i,0] := T[i,0];
     Tbasement[i,0] := T[i,0];
   end;
   OutTemperature:=T[dim,0];
 end;


Procedure ReadInput(inputfile: string; month: integer; var AverageTemp:real;
```

```
 var AirTemp: HourlyValues );     (*reading the temperature data from "input.txt"*)
 var    f          : text;
       i,j      : integer;
begin
  AverageTemp:=0;
  assign(f,inputfile);
  reset(f);
  for i:=1 to (BeginningMonth(month)-1) do readln(f);
  for i:=1 to LengthMonth(month) do begin
    for j:=1 to 24 do begin
      readln(f,AirTemp[j,i]);
      AverageTemp      := AverageTemp + AirTemp[j,i]/24/LengthMonth(month);
    end;
  end;
  AverageTemp:=AverageTemp;
  close(f);
end;

Procedure ReadService(filename2: string; var VentDat: HourlyValues );
var    f          : text;        (*reading the ventilation data from
"service.txt"*)
       i,j        : integer;
begin
  assign(f,filename2);
  reset(f);
  for i:=1 to (BeginningMonth(month)-1) do readln(f);
  for i:=1 to LengthMonth(month) do begin
    for j:=1 to 24 do begin
       readln(f,VentDat[j,i]);
    end;
  end;
  close(f);
end;


Procedure Undisturbed;          (* determines the indisturbed ground temperature *)
 var TempMax,TempMin,AirTempAmp,SqRoot : real;
     time : integer;
 begin
   YearlyAverage:=0; TempMax:=-100; TempMin:=100;
   for month:=1 to 12 do begin
     ReadInput(inputfile,month,MonthlyAverage[month],AirTemp);
     YearlyAverage:=YearlyAverage+MonthlyAverage[month]*LengthMonth(month)/365;
     if MonthlyAverage[month]>TempMax then TempMax:=MonthlyAverage[month];
     if MonthlyAverage[month]<TempMin then TempMin:=MonthlyAverage[month];
     write('*');
   end;
   AirTempAmp := (TempMax-TempMin)/2;
   SqRoot := sqrt(rho_earth*cp_earth*pi/(LambdaEarth*365*24*3600));
   For month:=1 to 12 do begin
     time := (BeginningMonth(month)-1) div 24 + (LengthMonth(month) div 2);
     MonthlyAverageGround[month]:=YearlyAverage + AirTempAmp *
                            exp(-depth*SqRoot) *
                            cos(2*pi*(time-196)/365-depth*SqRoot);
   end;
   GroundTemp:=MonthlyAverageGround[10];
 end;


 Procedure BoundaryCond(month : integer);
  var   i : integer;
  begin
    for i:=1 to dim do begin
      T[i,dim+1] := MonthlyAverageGround[month];
      Tbasement[i,dim+1] := TempBasement;
    end;
  end;
```

```
 Procedure Init;                                          (* setting of starting
values*)
   begin
     dl := PipeLength / dim;
     OutTemp := GroundTemp;
     WarmUp:=true;
     New(outputfile,inputfile);
end;

(* ------------------ main program -------------------------------------------*)

Begin
 Repeat begin
        Input                                                             (*data input*)
    clrscr;
    write('WKM_LTe is running ...'); writeln; writeln;
    write('Undisturbed ground temperature  ');
    Init;                                               (*setting of starting values*)
    Undisturbed;                  (*determines the indisturbed ground temperature*)
    SetMatrices(TimeStep);                             (*definition of the matrices*)
    for month:=10 to 12 do begin
    if WarmUp and (month=10) then begin
      writeln; write('Warm-up                                   ');
    end;
    if (not WarmUp) and (month=1) then begin
      writeln; write ('Outlet air temperature           ');
    end;
    write ('*');
    ReadInput(inputfile,month,AverageTemp, AirTemp);
    if VentService then ReadService(filename2,VentDat);  (*reading the ventilation
*)
    GroundTemp := MonthlyAverageGround[month];          (* data file
*)
    InTemp:=AirTemp[1,1];
    For day:=1 to LengthMonth(month) do begin
      BoundaryCond(month);
      for hour:=1 to 24 do begin
        InTempOld := InTemp;
        InTemp := AirTemp[hour,day];
        OutTempOld  := OutTemp;
        if VentService then begin
          if VentDat[hour,day] > 0 then operating:=true else begin
            OutTemp:=OutTemperature(InTemp);
            operating:=false;
          end;
        end;
        if operating then if (not WarmUp) then ServiceTime:=ServiceTime+1;
(*h*)
        if operating then OutTemp:=OutTemperature(InTemp);
                                             (*continues the curves in different
states*)
        if (not operating and VentService) then OutTemp:=OutTempOld;
        if (not operating and not VentService) then OutTemp:=InTemp;
        for j:=1 to dim do Calculate(TimeStep,j);       (* calculates the ground
*)
        SystemOutTemp[hour,day] := OutTemp;             (* temperature field
*)
      end;
    end;
    if (not WarmUp) then
    Output(Outputfile,month,AirTemp,SystemOutTemp);
    if ((month=12) and WarmUp) then begin
      month:=0; WarmUp:=false;
    end;
  end;
  clrscr;
  ReadText('ler3e.txt');
```

```
  s := VarString(60,12,'Y','y','Y','n','N');
  end until ((s='y') or (s='Y'));
End.
```

# 8. Using the WKM_LTe

The WKM_LTe is implemented as Turbo-Pascal program. A compiled "WKM_LTe.exe" file should work on any DOS/Windows PC. In the development the WKM_LTe was successfully used under Windows 3.1 and Windows95 with 486 and Pentium processors.

To start the WKM_LTe the following files must be in the WKM_LTe directory:

input.txt

screen1.txt

screen2.txt

screen3.txt

screen4.txt

service.txt

wkm_lte.exe

The "input.txt" file has to contain the temperature data for a whole year (row of 8760 hourly temperature values).
The "service.txt" file is only needed if the ventilation should not operate continuously. This file has to contain a row of "1" (in operation) and "0" (out of operation) in accordance with the service scheme (operating schedule) of the ventilation (row of 8760 hourly values).

The WKM_LTe writes an "output.txt" file and places it in the WKM_LTe directory. This file contains a row with the inlet air temperatures (repeating the input data) and a row with the calculated outlet air temperatures separated by semicolons. This data can be analysed with a spreadsheet program e.g. Excel.

An onscreen menu allows the user to set important parameters. Other parameters, such as the number of nodes and the time step, may only be changed in the source code:

```
WKM_LTe (Simulation of air-earth heat exchangers)
*************************************************
Heat capacity air               [J/kgK]=
Heat capacity pipe              [J/kgK]=
Heat capacity earth             [J/kgK]=
Density air                     [kg/m3]=
Density pipe                    [kg/m3]=
Density earth                   [kg/m3]=
Kinematic viscostiy of air       [m2/s]=
Heat conductivity air            [W/mK]=
Heat conductivity pipe           [W/mK]=
Heat conductivity earth          [W/mK]=
Internal diameter of pipe           [m]=
Length of pipe                      [m]=
wall thickness of pipe              [m]=
number of pipes (parallel)             =
Air volume flow rate total       [m3/h]=
Air flow velocity                 [m/s]=
Setting depth of pipes              [m]=
Depth of penetration                [m]=

                       redo (Y/N) ?
```

```
WKM_LTe (Simulation of air-earth heat exchangers)
*************************************************
Adiabatic share                  (0..1)=
Share of basement                (0..1)=
Basement temperature               [øC]=
Name of input file                     =
Name of output file                    =
Alternating service               (Y/N)=
Name of service file                   =

                       redo (Y/N) ?
```

*Figure 5: Onscreen menu of the WKM_LTe program. The suggested values may be accepted (press RETURN) or replaced by another value.*

Notes:

- Depth of penetration: This is the distance from the surface to the depth where the amplitude of response to a temperature cycle is reduced to 1 %. (based on equation (16))

$$e^{-x\sqrt{\frac{\pi}{aT_a}}} = 0.01 \quad \rightarrow x \approx 4.6 \cdot \sqrt{\frac{\pi}{aT_a}} \qquad (17)$$

The depth of penetration defines the thickness of the observed layer around the pipe. The pre-set value corresponds to the depth of penetration of the daily temperature changes.

- **Adiabatic share:** At the circumference an adiabatic zone occurs due to neighbouring pipes. This effect may be modelled by a segment of the circumference with an adiabatic boundary condition (no heat flux at the border). The size of this segment depends on the distance of neighbouring pipes – no feedback based on experience is available yet.

- **Share of basement:** Analogous to the adiabatic case the influence of the basement may be considered by a separate segment. For this a constant temperature is taken as the boundary condition – no feedback based on experience is available yet.

- **Share of ground**: The share of the undisturbed ground is given through:

$$\text{share of ground} = 1 - \text{adiabatic share} - \text{share of basement}$$

The basic case is a single pipe in undisturbed ground. The adiabatic share as well as the share of basement are set to 0.
If the pipe is situated below a building it may be useful to consider an influence of the basement (share of basement > 0).
If the system consists of parallel pipes it may be useful to consider an influence of the neighbouring pipes (adiabatic share > 0).

In principle any piping system may be calculated as the basic case. This will give more accurate results the greater the distances to the sources of disturbance.

- **Alternating service:** It needs to be specified whether the ventilation should work continuously or not. If the ventilation should not work continuously a service file (e.g. "service.txt") is needed.

# IEA Annex 28

## Low Energy Cooling

### Subtask 2: Detailed Design Tools

# SLAB COOLING SYSTEM, WATER COOLED

## SLAB COOLING SYSTEM SIMULATION

Department Mechanical Engineering
University of Porto, Portugal
José L. Alexandre, E. Maldonado

# Contents Chapter K

# 1. Technology area

Slab cooling by water circulated through embedded pipework.

# 2. Developed by

**Name** : José Luis Alexandre and Eduardo Maldonado

**Organisation** : Dep. Mechanical Eng. University of Porto

**Address** : Dep. de Eng. Mecânica e Gestão Industrial - Faculdade de Eng. da Universidade do Porto - Rua dos Bragas - 4099 Porto Codex - PORTUGAL

**Phone** : + 351 2 2007455

**Fax** : + 351 2 312476

**E-mail** : jla@fe.up.pt or ebm@fe.up.pt

# 3. General description

The thermal inertia of buildings can be used with a slab cooling system to reject its own energy excess to the atmosphere with low energy consumption. The system considered consists of embedded pipes in the slab, where the water absorbs the excess heat gains in the space, and a cooling tower, where the energy is dissipated to the atmosphere, therefore controlling the building's indoor comfort temperature.

*This report describes a slab cooling system model that is able to dynamically simulate the behaviour of* the system described under real operating conditions. The model of slab cooling system contains two parts which interact dynamically, the slab with embedded pipes and the cooling tower.

Figure 1 illustrates the complete system and the key variables in the analysis.

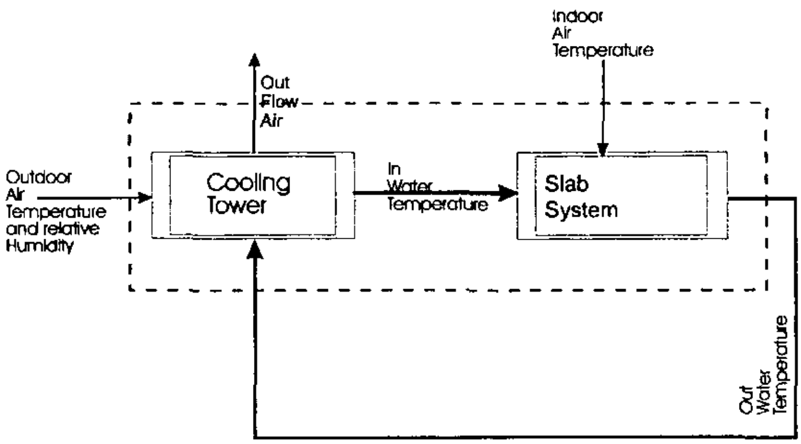



***Figure 1 - Complete Slab Cooling System Model***

**Inputs and outputs of the model:**

The inputs are:

- **SLAB**

1. Indoor air temperature - upper and/or lower indoor temperature [°C];

2. Global heat transfer coefficient ( convection + radiation) - upper and/or lower [$W/m^2K$];
3. Physical dimensions and thermal proprieties of the slab;
4. Water flow in the slab's embedded pipes.

- **COOLING TOWER**

5. Air flow [kg/s];
6. Water flow [kg/s];
7. Weather data (dry air temperature [°C], relative humidity [%]);
8. Two operating points of the cooling tower from the performance data supplied by its manufacturer.

The outputs are:

1. Energy stored within the slab;
2. Slab average surface temperatures;
3. Outlet water temperature for the cooling tower;
4. Heat flux absorbed by the slab;
5. Heat flux absorbed by the pipe network.

Figure 2 shows the inputs and outputs of the model.

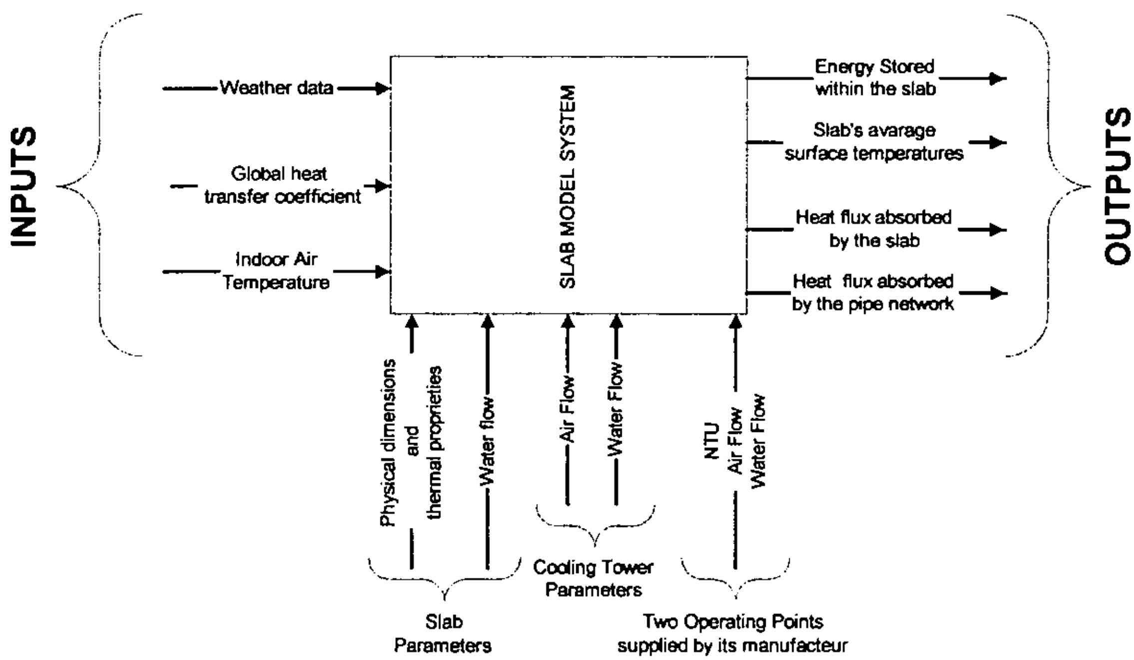



*Figure 2 - Information flow chart*

# 4. Nomenclature

The mathematical and the FORTRAN names of the variables used in Section 5 are written in the first and in the second columns respectively of the following table.

| Input Variables | | Description | Units |
|---|---|---|---|
| $\rho$ | rop(j) | Specific mass of the slab material | kg/m$^3$ |
| C | cpp(j) | Specific heat of the slab material | J/kg K |
| $\lambda$ | tk(j) | Thermal conductivity of the slab material | W/mK |
| $T_{water}$ | ttub(l) | Inlet water temperature to the slab | °C |
| $T_{indoor}$ | tin(l) | Indoor air temperature | °C |
| $\alpha$ | alfa | Global heat transfer coefficient ( conv. + rad) | W/m$^2$K |
| | ra | Radius of pipe | m |
| | cry | y position of the pipe center | m |
| | crx | x position of the pipe center | m |
| | boxh | Height of the slab | m |
| | boxl | Width between two pipes | m |
| Tin | Tagin | Outlet slab water temperature | °C |
| | Tarin | Outdoor dry-bulb temperature | °C |
| $\phi$ | Hr | Outdoor relative humidity | % |
| NTU | utn | Number of transfer units (cooling tower) | kW/(kJ/kg) |
| L | cw | Cooling tower water flow | kg/s |
| G | ca | Cooling tower air flow | kg/s |
| **Output Variables** | | | |
| $\Delta E_{total}$ | dener | Internal energy variation of the slab | J |
| qflux | qfluxo | Heat flux absorbed by the upper surface | W |
| $T_{med}$ | *tmed* | *Slab average temperature* | °C |
| Tout | Tagout | Inlet water temperature for the slab | °C |
| | Two | Outlet water temperature for the slab | °C |
| **Variables use in the mathematical description** | | | |
| $\Delta t$ | delt | Time step | s |
| a | a(i,j) | Coefficient of the e grid point temperature | W/mK |
| Su | su(i,j) | Source coefficient of the p grid point | W/m |
| Tp | t(i,j) | Temperature of the p grid point | °C |
| $a^o_p$ | apo(i,j) | Coefficient of the ap old | W/m°C |
| $T^o_p$ | told(i,j) | Old temperature of the p grid point temperature | |
| ha | | enthalpy of the air | kJ/kg d.a. |
| hi | | enthalpy of the air at the wetted-surface temperature | kJ/kg d.a. |
| hc | | Convection coefficient | W/m$^2$K |
| Cpm | | *specific heat of moist air* | kJ/kg K |
| Ps | | water saturation pressure | N/m$^2$ |
| P | | atmospheric pressure | N/m$^2$ |
| **Index** | | | |
| N | | North neighbour point | |
| S | | South neighbour point | |
| W | | West neighbour point | |
| E | | Est neighbour point | |
| P | | Central grid point | |

# 5. Mathematical description

## 5.1. Mathematical description for the slab

The model of the slab is based on the general conduction equation:

$$\frac{\partial T}{\partial t}\rho C_p = \frac{\partial}{\partial x}\left(\lambda_x \frac{\partial T}{\partial x}\right) + \frac{\partial}{\partial y}\left(\lambda_y \frac{\partial T}{\partial y}\right) + \frac{\partial}{\partial z}\left(\lambda_z \frac{\partial T}{\partial z}\right) + S \tag{1}$$

Although the system is three-dimensional, temperature differences are small and a 2-D representation is sufficiently accurate:

$$\frac{\partial T}{\partial t}\rho C_p = \frac{\partial}{\partial x}\left(\lambda_x \frac{\partial T}{\partial x}\right) + \frac{\partial}{\partial y}\left(\lambda_y \frac{\partial T}{\partial y}\right) + S \tag{2}$$

These equations can be solved by a finite-difference full implicit discretization. The application of this *discretization method to a control volume in the slab (Figure 3) results in equation (3) [Ref. 1].*

$$a_p T_P = a_e T_e + a_w T_w + a_n T_n + a_s T_s + S_u \tag{3}$$

Where:

$$a_e = \frac{\lambda_e \Delta y}{(\delta x)_e}$$

$$a_w = \frac{\lambda_w \Delta y}{(\delta x)_w}$$

$$a_s = \frac{\lambda_s \Delta x}{(\delta y)_s}$$

$$a_n = \frac{\lambda_n \Delta x}{(\delta y)_n}$$

$$a_p^0 = \frac{\rho C \Delta x \Delta y}{\Delta t}$$

$$S_u = a_p^0 T_p^0$$

$$a_p = a_e + a_w + a_s + a_n + a_p^0$$

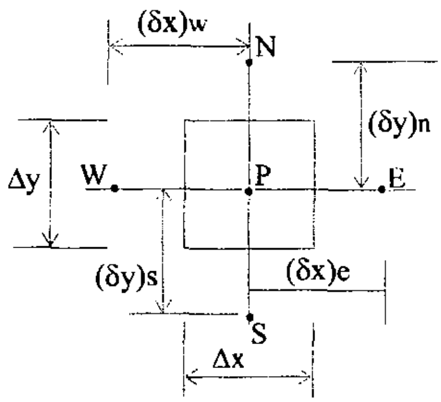



*Figure 3 - Elementary control volume in the interior of the slab*

For nodes on the boundary, different boudary conditions occur, as shown in fig.4:

- adiabatic wall surface (two planes of symmetry);
- upper and lower surfaces with convection and radiation, caracterized by a global heat transfer coefficient;
- pipe wall conditions.

a) **Adiabatic wall temperature (1 and 1a)** - In this case, the $a_w$ coefficient is equal to zero in the (1a) surface, and the $a_e$ coefficient is equal to zero in the (1) surface.

$$T_P = \frac{a_e}{a_p}T_e + \frac{a_w}{a_p}T_w + \frac{a_s}{a_p}T_s + \frac{a_n}{a_p}T_n + \frac{S_u}{a_p} \qquad (4)$$

where:

$a_e = a_w = 0.0$

b) **Convection and radiation wall** - Upper and lower surfaces

1 - Upper surface - *The coefficients su, ap and an are as follows;*

$$S_u = a_p^0 T_p^0 + \alpha_{upper}.\Delta x.T_{indoor}$$
$$a_p = a_e + a_w + a_s + a_n + a_p^0 + \alpha_{upper}.\Delta x \qquad (5)$$
$$a_n = 0$$

2 - Lower surface - The coefficients su, ap and as are as follows;

$$S_u = a_p^0 T_p^0 + \alpha._{lower}\,\Delta x.T_{indoor}$$
$$a_p = a_e + a_w + a_s + a_n + a_p^0 + \alpha_{lower}.\Delta x \qquad (6)$$
$$a_s = 0$$

c) **Pipe wall conditions** - The boundary condition near the pipe wall is obtained in the same way as the conditions above, with the convection factor evaluated by the classic equation for convection inside a pipe. The convection coefficient is function of the water flow rate (an input).

$$S_u = a_p^0 T_p^0 + \alpha_{inside\_pipe}.\Delta i.T_{water}$$
$$a_p = a_e + a_w + a_s + a_n + a_p^0 + \alpha_{inside\_pipe}.\Delta i \qquad (7)$$
$$a_j = 0$$

where :

$\Delta i$ = area of elementary volume (this value depends of the imaginary point position and can be $\Delta x$ or $\Delta y$)
$a_j$ = a coefficient in direction j (the j can be W, N or S depending on the imaginary point position)

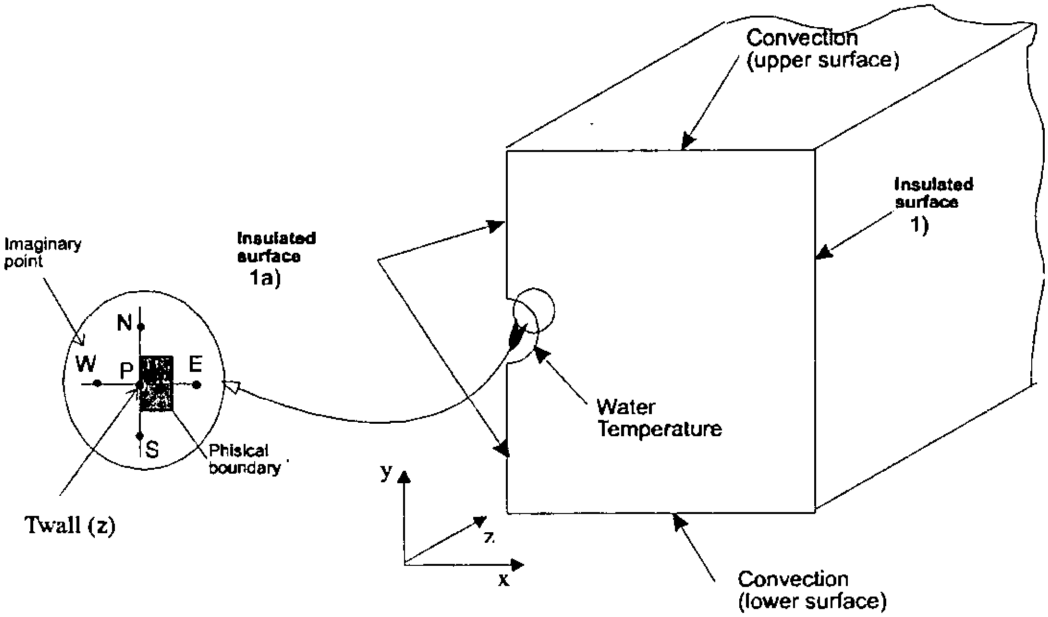



*Figure 4 - Slab's element boundary conditions*

Near the pipe boundary the energy balance is made by a sample approach as shown in Figure 5

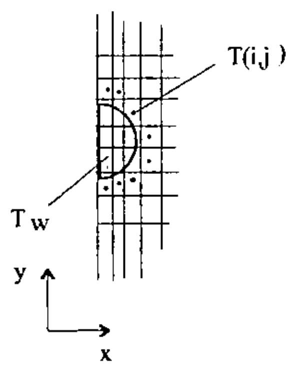



*Figure 5 - Grid boundary approach to the real pipe boundary*

Applying equation (3) a to all the interior nodes of the domain and equations (4) to (7) for the boundary nodes, yields a system of equations that can be solved by the line-by-line method, using a combination of the TDMA and Gauss-Seidel methods [Ref.1]. The line-by-line method gives good convergence, and is thus suitable for this kind of problem.

For each step (time step) of the simulation, the routine produces the temperature distribution in the slab, for two operating modes:

I. with water flow through the pipes
II. without water flow in the pipes

The routine calculates the real temperature distribution of the slab with water flow (I), as a default situation. However, if the pipe outlet water temperature is lower than the inlet water temperature the system is turned off and the new temperature distribution evaluated without water flow (II).

At the end of each time step, the routine gives the average slab temperature and the heat flux absorbed by the slab upper surface. In addition, the subroutine ENERGY gives the average slab temperature, calculated by:

$$T_{med} = \frac{\sum(T_p \Delta x . \Delta y)}{(boxh.boxl)} \tag{8}$$

The subroutine FLUXCAL evaluates the heat flux absorbed by the upper and lower slab surfaces, calculated by:

$$qflux = \sum\left(\alpha.\Delta x.\left(T_{indoor} - T_p\right)\right) \tag{9}$$

## 5.2 Mathematical description for the cooling tower

In a cooling tower, there is the heat exchange between the circulating water in the pipes, and the unsaturated atmospheric air. There are two driving forces for heat rejection; the difference in the dry-bulb temperatures for heat transfer, and the difference in the vapour pressures between the water surface and the air for the mass transfer. Figure 6 represents an elementary control volume of the tower section. G is the airflow that enters the lower surface of the tower and L is the water flow that enters at the top surface. For simplicity, the small quantity of water which evaporates is neglected, so the both water and air flow rate are assumed constant throughout the tower. Water enters the control volume at temperature T and leaves at T-dT, while air enters with enthalpy $h_a$ and leaves with $h_a$+$dh_a$. dA is the total area of the surface of the water drops plus the tower's wetted fill total surface. The energy balance can be expressed by equation 10.

$$dq = Gdh$$

$$Gdh = L.C_{ps}.dT \tag{10}$$

where:

$C_{ps}$ - specific heat capacity of water (4,19 kJ/kgK), assumed constant throughout the process

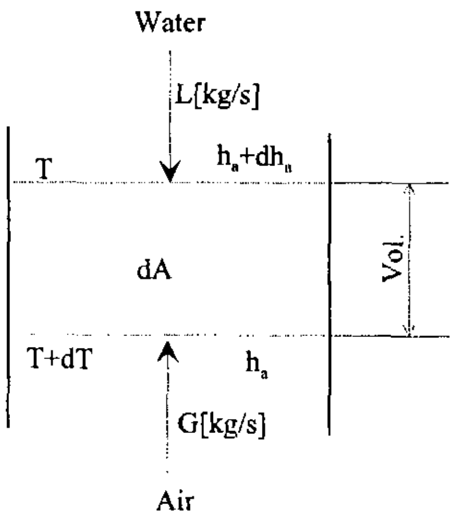



*Figure 6 - Elementary control volume of cooling tower*

As noted above, when a air flow past a wetted surface (Figure 7) there are two kind of heat transfer; latent and sensible heat. The heat exchange can be expressed by equation (11), which includes both heat transfer and mass transfer:

$$dq = dq_s + dq_l \tag{11}$$

Where:

$$dq_s = h_c.dA.(T_i - T_a)$$

$$dq_l = h_D.dA.(w_i - w_a).h_{lv}$$

$h_D$ - mass transfer coefficient [ Kg/m²s]

$h_c$ - convection coefficient [W/m²K]

$q_s$ - rate of sensible heat transfer [W]
$q_l$ - rate of latent heat transfer [W]
w - humidity ratio [(kg of water vapour)/(kg of dray air)]
$h_{lv}$ - latent heat of the water at $T_i$, [J/kg]

The mass coefficient transfer is proportional to $h_c$ and this proportionality can be expressed by equation 12:

$$h_D = \frac{h_c}{C_{pm}} \tag{12}$$

where:

$C_{pm}$ - Specific heat of moist air [J/kg K] and is equal to $C_p$ +w $C_{ps}$
$C_p$ - specific heat of the air [J/kgK]
$C_{ps}$ - specific heat of water steam [J/kg K]

Applying the definitions above the equation (11) becomes:

$$dq_t = \frac{h_c.dA}{C_{pm}}\left[\left(C_p.T_i + w_i h_{lv}\right) - \left(C_p.T_a + w_a.C_{ps}.T_a - w_a.C_{ps}.T_i + w_a.h_{lv}\right)\right] \tag{13}$$

Adding the expression ($w_i h_l$-$w_a h_l$) to equation 13, where hl is the enthalpy of saturated liquid water at temperature $T_i$ (this expression is almost negligible compared with the other terms in the equation) gives:

$$dq_t = \frac{h_c.dA}{C_{pm}}\left\langle\left[C_p.T_i + w_i\left(h_l + h_{lv}\right)\right] - \left[C_p.T_a + w_a.\left(h_l + h_{lv} + C_{ps}\left(T_a - T_i\right)\right)\right]\right\rangle \tag{14}$$

The first term in the equation (14) represents the enthalpy at the wetted-surface temperature ($h_i$) and the second is the enthalpy of the air in free stream($h_a$). Thus:

$$dq_t = \frac{h_c dA}{C_{pm}}\left(h_i - h_a\right) \tag{15}$$

The heat transfer between the air and water streams can be calculated as a function of the enthalpy potential, i.e. the difference between enthalpy of saturated air at the wetted-surface temperature ($h_i$) and the enthalpy of the air in the free stream ($h_a$) as expressed in equation 15.

The total heat flux transfer through the cooling tower can be obtained by integration of equations 10 or 15.

$$Q = \int_0^A \frac{h_c}{C_{pm}}.\frac{1}{\left(h_i - h_a\right)}.dA \tag{16}$$

$$Q = \int_{T_{in}}^{T_{out}} L.C_{ps}.dT \tag{17}$$

Combining the above equations (16 and 17) and rearranging gives the water temperatures entering and leaving the tower, and the NTU( $\mathrm{NTU} = \frac{h_c.A}{C_{pm}}$ ) of the cooling tower (equation 18). The value of NTU is often used to characterise a cooling tower and is the basis for predicting its performance at various inlet water and air (wet bulb) temperatures. The higher the value of NTU, the closer the temperature of the water leaving the cooling tower will be to the wet bulb temperature of the entering air.

$$\mathrm{NTU} = 4{,}19.\mathrm{L}.\Delta\mathrm{T}\sum\frac{1}{\left(h_i - h_a\right)_m} \tag{18}$$

Equation 18 is a numeric integration of equations 16 and 17 combination, where $(h_i$-$h_a)_m$ is the arithmetic-mean enthalpy difference for a incremental volume (see Figures 5 and 6).

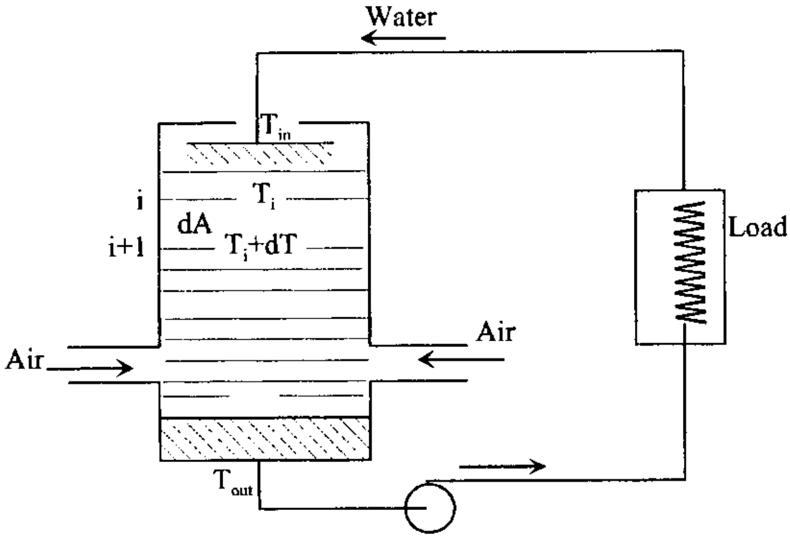



*Figure 7 - Division of the tower into incremental volumes*

The properties of air used in this model were obtained from classical psychometrics.

$$w = 0{,}6219\left(\frac{\phi P_s}{P - \phi P_s}\right) \tag{19}$$

$$h_a = 1{,}0.T_{air} + w\left(1{,}86.T_{air} + 2501{,}3\right) \tag{20}$$

$$P_s = \begin{cases} 0{,}067112 * 1{,}061845^T \Leftrightarrow 0 < T < 50°C \\ 0{,}135743 * 1{,}045663^T \Leftrightarrow 50 \le T < 80°C \end{cases} \tag{21}$$

$$h_i = 4{,}7926 + 2{,}568 * T - 0{,}029834 * T^2 + 0{,}0016657 * T^3 \tag{22}$$

# 6. References

[1] - Suhas V. Patankar - Numerical Heat Transfer and Fluid Flow, Mcgraw-Hill Book Company, 1980

[2] - William H. Press, at all - Numerical Recipes, The Art of Scientific Computing, Cambridge University Press, 1998

[3] - R. W. Lewis, K. Morgan and B. A. Schefler - Numerical Methods in Heat Transfer , Volume II, John Wiley &Sons, 1983

[4] - R. W. Lewis, K. Morgan - Numerical Methods in Heat Transfer, Volume III, John Wiley & Sons, 1985

[5] - Frank P. Incropera, David P. De Witt - Fundamentals of Heat and Mass Transfer, John Wiley & Sons, 1990

[6] - Wilbert F. Stoecker, Jerold W. Jones - Refrigeration & Air Conditioning

[7] - 1988 - ASHRAE Handbook , Fundamentals

[8] - El-Wakil, Powerplant Technology , McGraw-Hill

[9] - John R. Dormand -Numerical Methods for Differential Equations, CRC Press, Inc - 1996

[10] - Etude de la Régulation des Planchers Rafraîchissants - Project de fin d'etudes realise por Dominique Cunat and E. Michel - COSTIC France

[11] - Le Plancher Chauffant Rafraichissant - Réunion d'information Technique - COSTIC , Marseille Jun 1996 - France

# 7. Algorithm

I ) Input variables

II ) Calculate dimensions of the grid applied to the slab - Subroutine GRID

III ) Calculate physical proprieties of each elementary volume (ρ, C and λ)

Begin Loop 1 (Time step)

Begin Loop 2 (Slab System On-Off)

IV ) Initialise the temperature values for each point

V ) Calculate the different boundary conditions

VI ) Calculate the different coefficients of equation 3 - Subroutine COEF

VII ) Calculate the slab's temperature distribution - Subroutine LSOLVR

VIII ) Calculate the outlet water pipe temperature - Subroutine ENERGY

IX ) Compare outlet water pipe temperature with inlet water pipe temperature and evaluate the mode function of the system (with or without water flow)

End Loop 2

X ) Calculate the slab's average temperature

XI ) Calculate the heat flux absorbed by the slab (upper and lower heat flux)- subroutine FLUX-CAL

XII ) Output the different values into a file

End Loop1 (time step)

XIII )End Programme

# 8. Flowchart

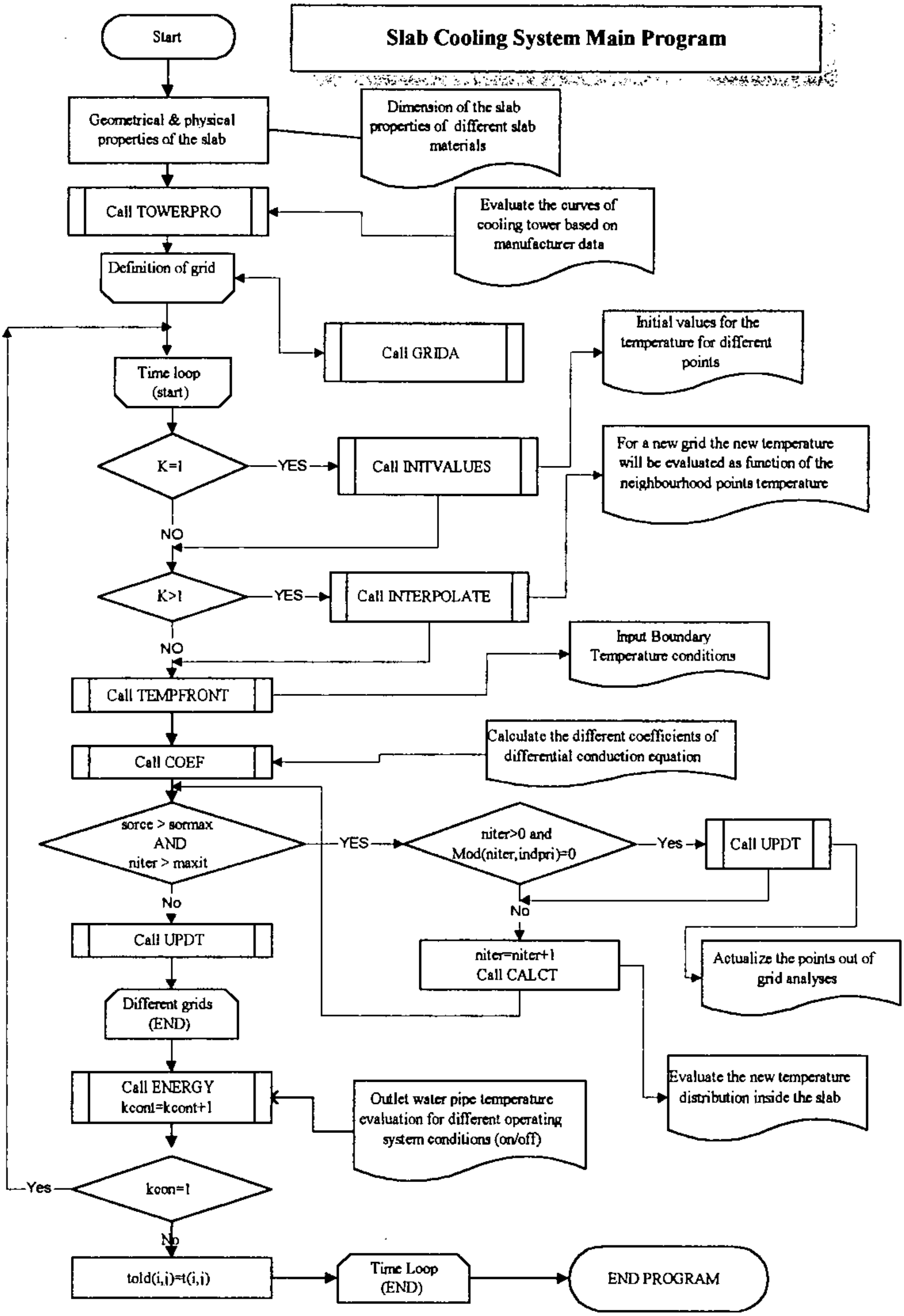



*Figure 8 - Flowchart for the model*

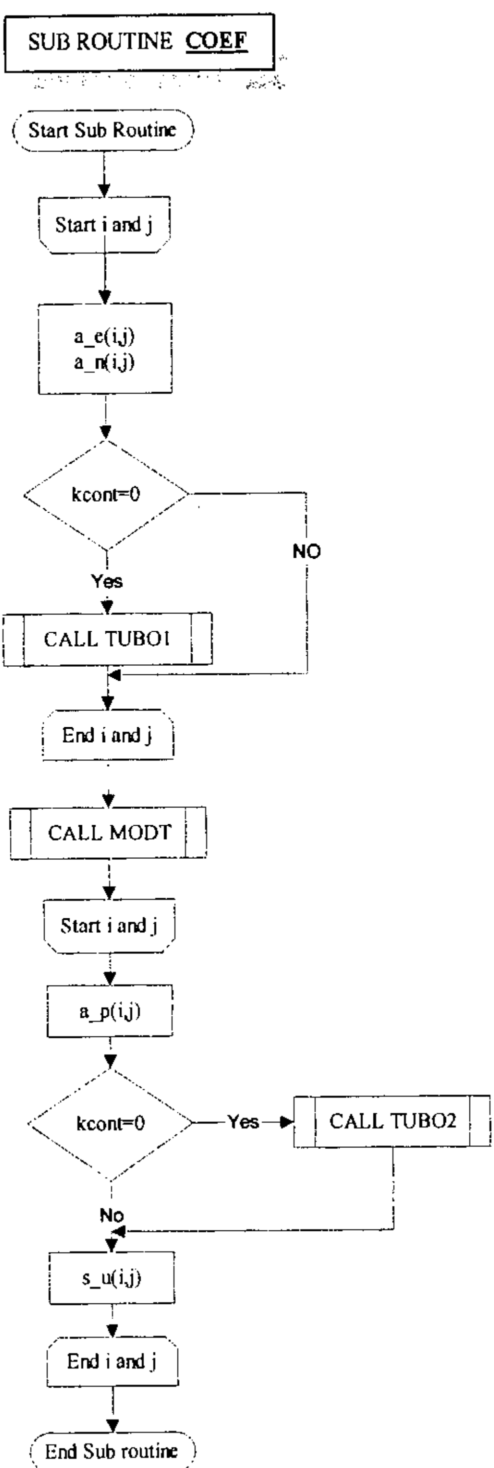



*Figure 9 -Routine grid and coef*

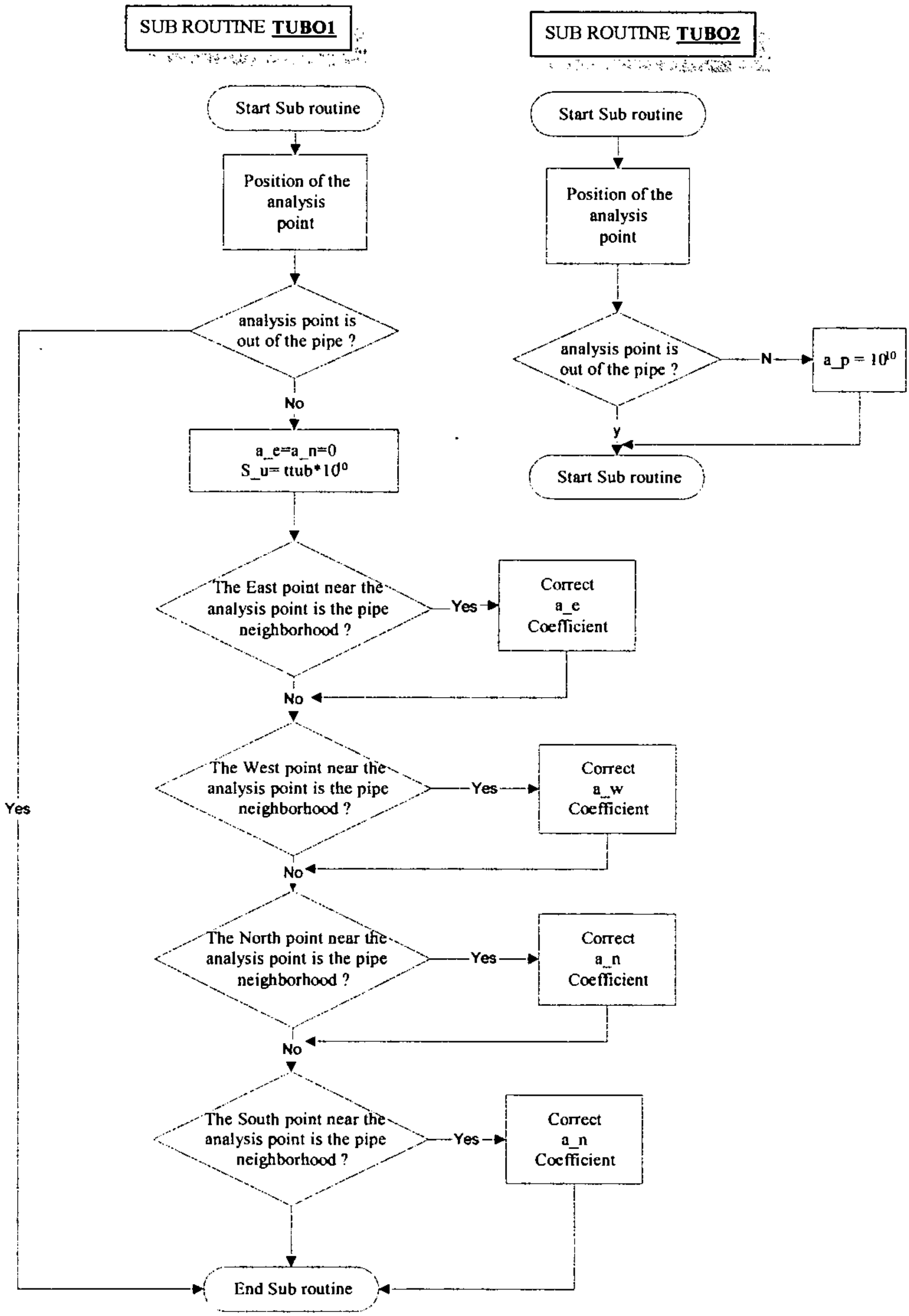



*Figure 10 - Routine TUBO1 and TUBO2*

# 9. Model validation

## 9.1 Introduction

This section describes a validation exercise for the slab model used in the study of the slab cooling system developed within Annex 28. Experimental data obtained in a test cell of COSTIC[®] - France was used for this purpose.

## 9.2 Experimental data

The tests done at COSTIC were designed to provide data to evaluate the effect of different variables on the comfort conditions in a space cooled by a slab cooling system.

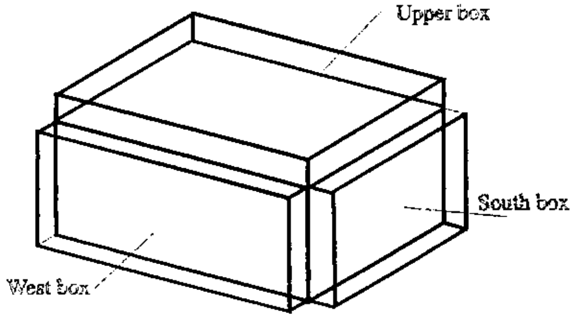



*Figure 11 - Test cell [Ref.10]*

The test cell has a double envelope as shown in Figure 11. It has six independent outside boxes that have a 1 meter spacing between the internal and the external walls. Air is supplied at different temperatures (between-5°C and 40°C) into the box, enabling the effects of outdoor air temperature and asymmetries to be tested. The north box is removable to allow for testing different wall types.

The cell's slab has a pipe network embedded similar to a radiant floor heating system which is used for cooling the cell. Figure 12 shows the pipework distribution in the floor slab, with two independent circuits of 58 m and 78 m in length.

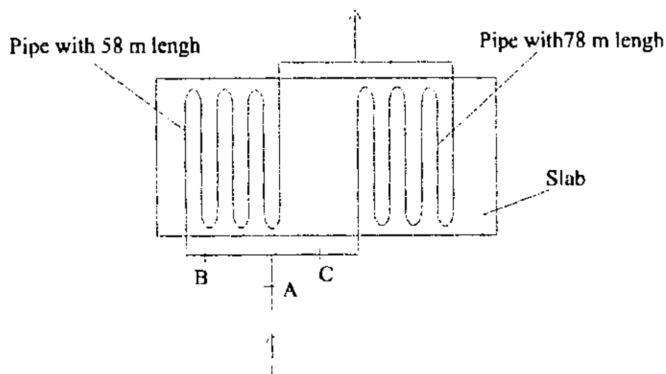



*Figure 12 - Pipework distribution in the slab*

---
[®] Special thanks are due to Mr. Feldman and Mr. Eric Michel (COSTIC - France) for supplying the data and allowing it to be used in this content.

The slab was made of different layers of materials, as shown in Figure 13.

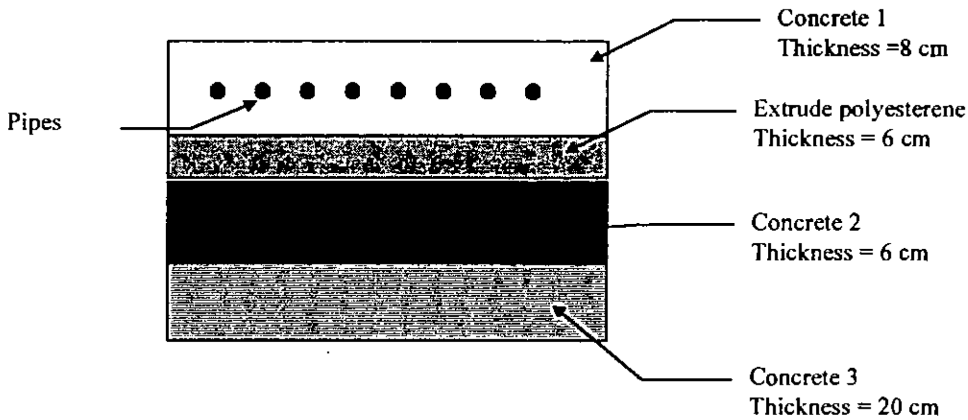



*Figure 13 - Slab composition*

The tests performed in the cell were as follows:

1. Influence of the inlet water temperature in the pipe network (12°C, 16°C and 20°C);
2. Influence of the cooling load of the cell (outdoor Air temperature - 40°C, 33°C and 26°C);
3. Influence of the surface finishing of the slab (types of materials used: plastic film, wooden floor and carpet);
4. Influence of the water flow rate (250, 140 and 90 l/h).

The temperature of the lower surface of the slab was kept constant and equal to 15°C for all tests.

The following data was measured:

Inlet water flow [l/h]
Inlet and outlet water temperature [°C]
Upper and lower surface heat flux absorbed by the slab [W/m²]
Average of the indoor air temperature [°C]
Upper slab surface global heat transfer coefficient [W/m² K]
Lower slab surface temperature [°C]

The procedure for evaluating global heat transfer coefficient and the indoor average air temperature is not very accurate for the experimental tests.

### 9.3 The model

The model of the slab used to simulate these experiments was the model described in section 3, but with inputs and outputs to conform with the test data:

- INPUT
  - average air temperature inside the cell;
  - global heat transfer coefficient between the upper layer of the slab and the indoor air;
  - inlet water temperature
  - water flow rate
  - slab information
    - physical properties of different materials
    - geometric characterisation of the different layers
    - physical properties of the pipework
- OUTPUT
  - average upper slab surface temperature
  - absorbed heat flux by the upper and lower slab surface
  - absorbed heat flux by the water

## 9.4 Comparison between experimental data and simulation results

The table and pictures below present the different results of the simulation and experimental data for different test conditions in the COSTIC cell. The values obtained with the model simulation are in very close agreement with the experimental data, as can be seen in the table below.

| | Average Temp. of upper surface (Ts) [°C] | | Inlet water Temp. [°C] | Outlet water Temp. of the slab (Tw_out) [°C] | | εQ(upper) | εQ(mCpΔT) |
|---|---|---|---|---|---|---|---|
| Teste n° | Model | Exp. | Tw_in | Model | Exp. | | |
| 1 | 18.69 | 19.46 | 13.1 | 16.41 | 16.2 | 14% | 7% |
| 2 | 21.09 | 21.83 | 17 | 19.42 | 19.2 | 16% | 10% |
| 3 | 23.56 | 24.13 | 21 | 22.47 | 22.3 | 15% | 13% |
| 4 | 18.60 | 19.59 | 13 | 16.36 | 16 | 16% | 12% |
| 5 | 19.34 | 20.33 | 13.1 | 18.41 | 17.8 | 17% | 13% |
| 6 | 20.59 | 21.46 | 13.1 | 20.63 | 19.8 | 17% | 12% |

*Table 1 - Experimental values versus model's output for the different tests*

where:

$T_{w\_in}$ - Inlet water temperature [°C]
$T_{w\_out}$ - Outlet water temperature [°C]
Q(upper) - heat flux absorbed by the upper slab surface [W/m²K]

$$\varepsilon Q(\text{upper}) = \frac{\left[\left(Q_{upper}\right)_{model} - \left(Q_{upper}\right)_{Exp.}\right]}{\left(Q_{upper}\right)_{Exp.}}$$

$$\varepsilon Q(mCp\Delta T) = \frac{\left[\left(Q_{mCp\Delta T}\right)_{model} - \left(Q_{mCp\Delta T}\right)_{Exp.}\right]}{\left(Q_{mCp\Delta T}\right)_{Exp.}}$$

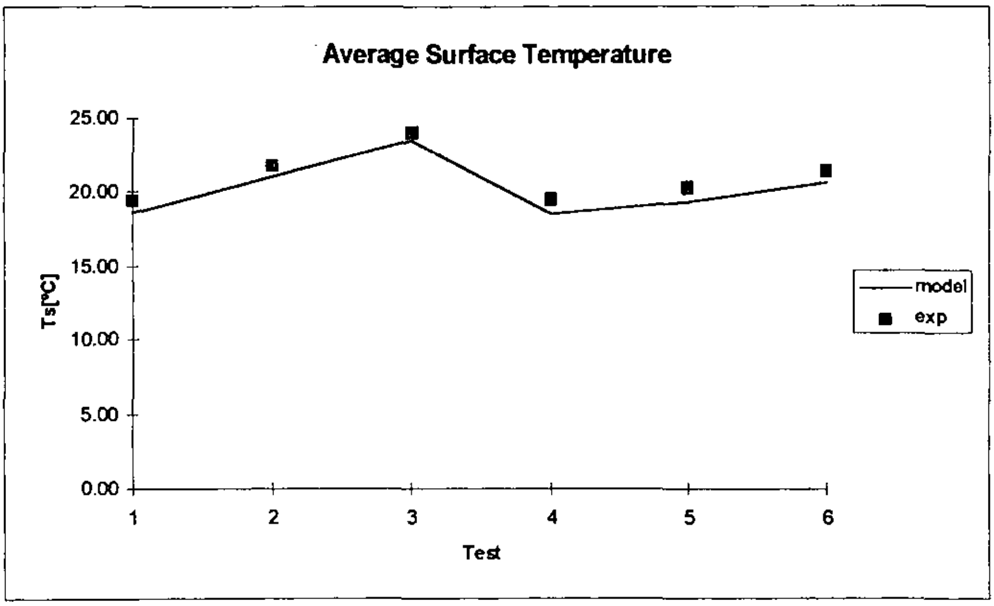



*Figure 14 - Average Surface Temperature*

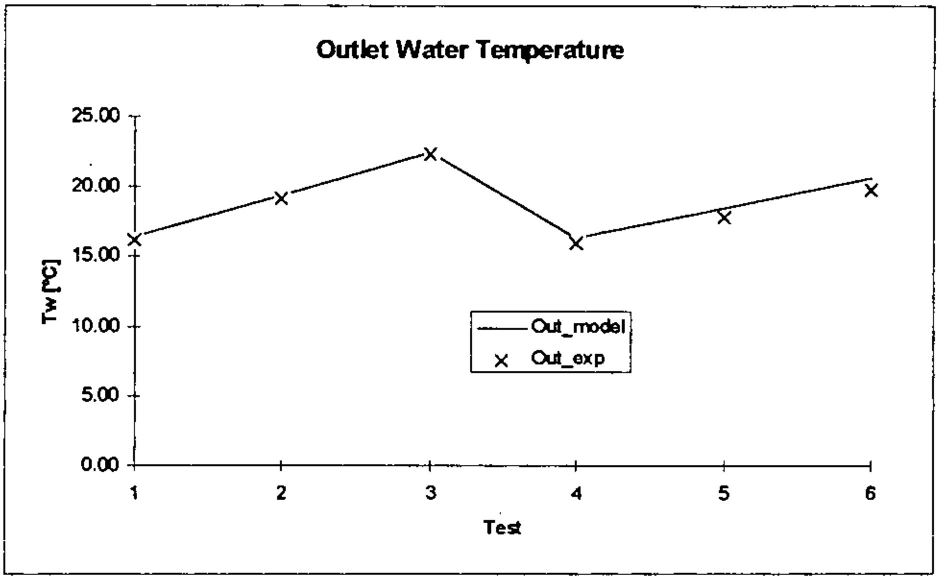



*Figure 15 - Outlet Water Temperature*

Differences between simulation (model) and the experimental values (exp) range from 7% to 17% for the two parameters analysed.

The procedure for evaluating the global heat transfer coefficient and the indoor/resultant temperature for the is not very accurate. Table 2 shows the effect of assuming small variations for the global heat transfer coefficient ($\alpha$) on the surfaces of the slab.

| | | T upper Surf. [°C] | | T water[°C] | | | Ts -ast | Tw out |
|---|---|---|---|---|---|---|---|---|
| T_air | alfa | model | exp | Tw_in | Out_model | Out_exp | ε ΔT1 | εΔT2 |
| 24.2 | 7.68 | 18.7 | 19.46 | 13.1 | 16.4 | 16.2 | 14% | 7% |
| 24.2 | 7.58 | 18.5 | 19.46 | 13.1 | 16.4 | 16.2 | 16% | 6% |
| 24.2 | 8 | 18.7 | 19.46 | 13.1 | 16.5 | 16.2 | 14% | 8% |
| 24.2 | 8.1 | 18.7 | 19.46 | 13.1 | 16.5 | 16.2 | 13% | 9% |
| 24.9 | 6.78 | 18.6 | 19.46 | 13.1 | 16.4 | 16.2 | 14% | 6% |
| 24.9 | 6.88 | 18.6 | 19.46 | 13.1 | 16.4 | 16.2 | 14% | 7% |
| 24.9 | 7.08 | 18.7 | 19.46 | 13.1 | 16.5 | 16.2 | 13% | 8% |
| 24.9 | 7.28 | 18.8 | 19.46 | 13.1 | 16.5 | 16.2 | 11% | 10% |

*Table 2 - Effect of the indoor /resultant temperature and the global heat coefficient*

Ts-ast - Average slab surface temperature [°C]
Tw_in - Inlet water temperature inside the pipe network [°C]
Tw_out - Outlet water temperature of the pipe network [°C]
ε - Relative error between experimental temperatures and output model temperatures

# 10. Sample results

Results have been generated for the following general conditions:

Weather data - Lisbon- from 1 June to 21 July
Cooling Tower (BALTIMORE- VTL045 H)
Water flow : 4.5 Kg/s

Figure 16 shows the input parameters used for the model simulation program.

**SLAB COOLING SIMULATION transient**

INPUTS

**Slab Properties**

| | |
|---|---|
| Slab thickness [mm] | 300.0 |
| Distance between pipes [mm] | 150.0 |

Number of lyers within the slab = 2 — update Layers nº

| | | | |
|---|---|---|---|
| Layer nº | 1 | 2 | end of slab |
| Thickness (es) [mm]= | 50.0 | 250.0 | |
| The layer start at = | 0.0 | 50.0 | 300.0 |
| Density [Kg/m3]= | 32.0 | 2185.0 | 2185.0 |
| Specific heat [J/kgK]= | 1412.0 | 880.0 | 880.0 |
| Thermal conductivity [W/mK]= | 0.0400 | 1.3300 | 1.3300 |

**Pipe: Dimensions and Properties**

| | |
|---|---|
| Outside diameter (D1) [mm]= | 17.00 |
| Wall thickness (et) [mm]= | 2.20 |
| Distance from botton of the slab (cry) [mm]= | 150.00 |
| Density [Kg/m3]= | 910.00 |
| Specific heat [J/kgK]= | 2300.00 |
| Thermal conductivity [W/mK]= | 40.000 |
| Pipe Length [m]= | 58.0 |
| Pipe length for each simulation [m]= | 10.0 |

(default value is 10 m)

**Other Parameters**

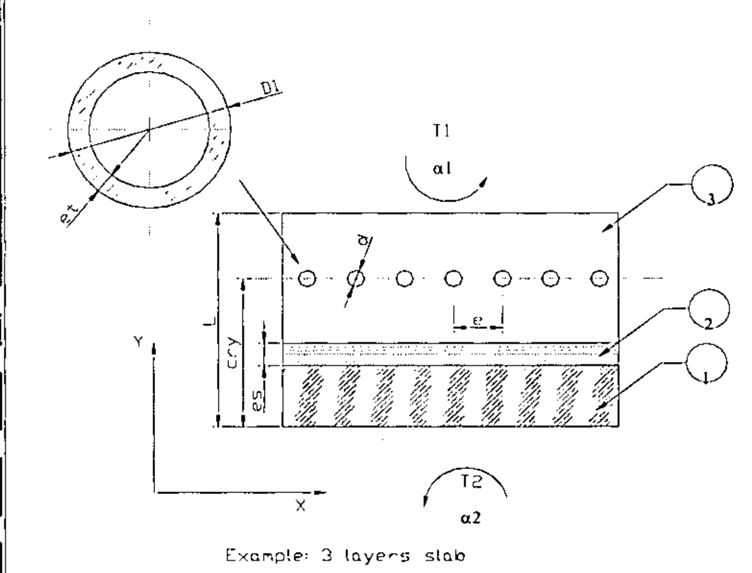
Surface heat transfer coeficients

| | |
|---|---|
| Upper surface ($\alpha$1) [W/m²K] = | 6.7 |
| Lower surface ($\alpha$2)* [W/m²K] = | 6.7 |

Indoor air temperature

| | |
|---|---|
| Upper surface (T1) [°C]= | 25 |
| Lower surface (T2) [°C] = | 25 |

| | |
|---|---|
| Inlet pipe water temperature[°C]= | 13 |
| Water flow (inside the pipe) [l/h] = | 58.94 |
| Time step simulation (max. =1464)= | 1464 |




* if you know the slab surface temperature the global heat transfer coefficient must be 1.0E+5

View Grid for simulation (1)

Cooling Tower (2)

Go to Weather (3)

*Figure 16 - Slab Cooling inputs program simulation*

The figures 17 and 18 show the final results of the simulation program as a time simulation

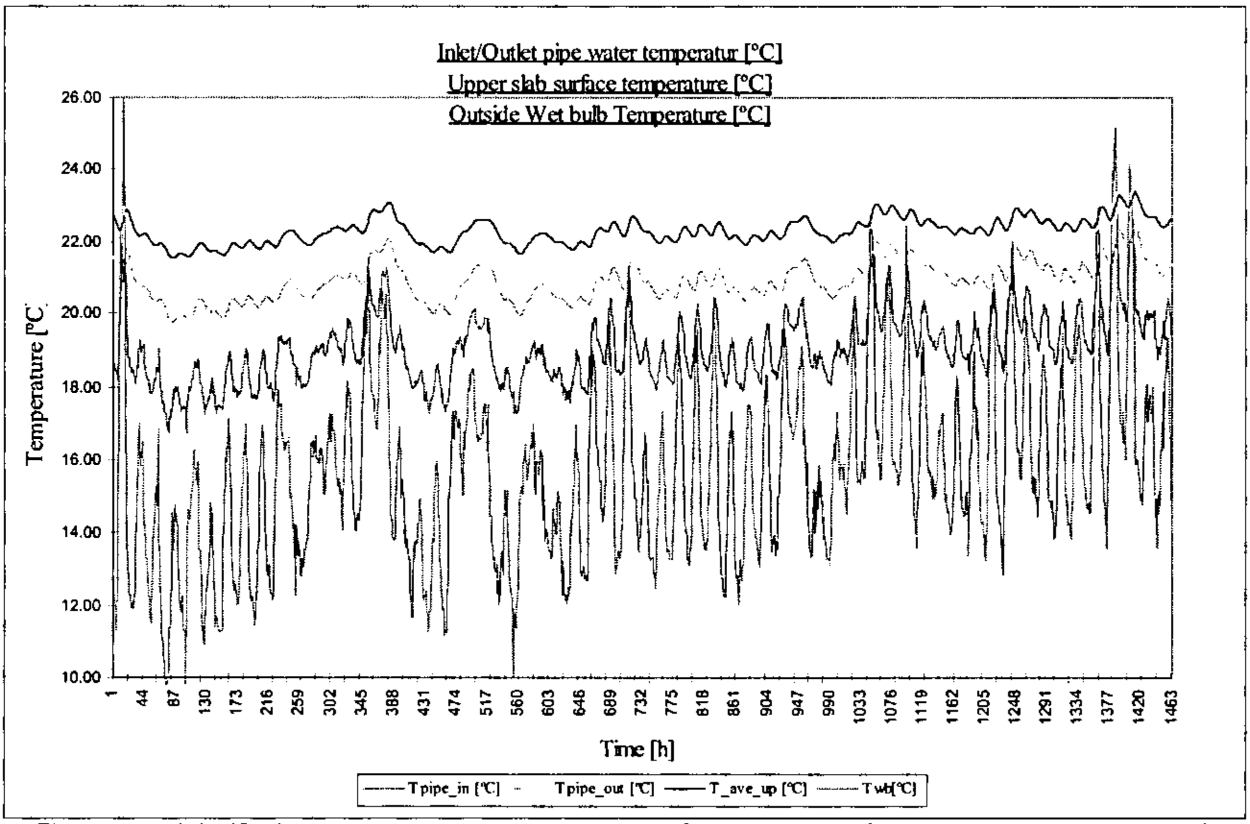



*Figure 17 – Inlet/Outlet water temperatures, average of upper slab surface temperature and wet bulb temperature*

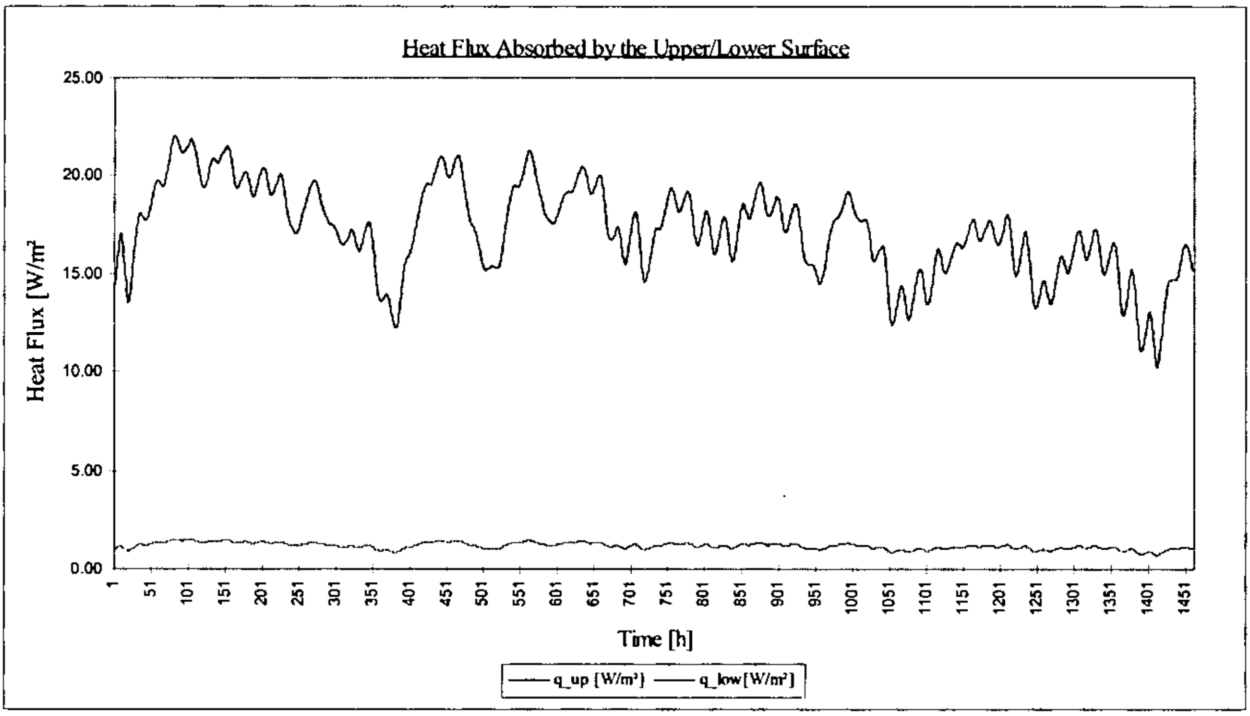



*Figure 18 - Heat flux absorbed by the upper and lower slab surfaces*

# 11 Source Codes

## 11.1 Source code for slab cooling

```
!-----------------------------------------------------------
-----
c
C         MODEL OF SALB COOLING SYS-
TEM
C        experimental data - COSTIC`s test cell
C
C
C-----------------------------------------------------------
C    This program evaluate as time function the
following variables:
c        slab's surfacestemperature
c        absorved heat flux  by
c                the slab
c                the pipe
c                water network
c
c        absorved energy by the slab
c-----------------------------------------------------------
C
      PROGRAM SLAB_COOLING
c
c   General variables
c
      INCLUDE 'PrgSub'
      INCLUDE 'PrgLsolv'
      INCLUDE 'Prg1'
c
      great = 1E+30
      nswpt = 1
c ___________input slab
prop.___________________#
      CALL INPUTPRO
c______________________________________________#
!          ttub=16.1  !  constant Water temperature
!
!          WRITE(*,'(A,\)') ' temperatura do tubo ->'
!          READ(*,*)twi1
!          WRITE(*,'(A,\)') ' Temperatura do ar interio -
>'
!          READ(*,*)tin1
!          WRITE(*,'(A,\)')' Temperatura da parte
inferior ->'
!          READ(*,*)tlower
!          WRITE(*,'(A,\)')' Delta comprimento ->'
!          READ(*,*)comp1
!          WRITE(*,'(A,\)') ' Coeficiente de transferen-
cia ->'
!          READ(*,*)alfa
          !ttub=ttub1
!          tin=tin1
 !    tin=24.89 ! indoor air temperature
!          tlower=16.8 ! lower surface temperature
c # #
       crx = 0.
c # #
       delt=60000.0
       maxit = 50000
       indpri = 10
       urft = 1.0
 !     CALL GRIDA (boxh,boxl,cry,x,y,ni,nj,ngrids)
          OPEN(UNIT=59,File='gridaxy.dat',
STATUS='Unknown')
          READ(59,*)ni
          READ(59,'(f8.6)')(x(i),i=1,ni)
          READ(59,*)nj
          READ(59,'(f8.6)')(y(i),i=1,nj)
          READ(59,*)ngrids
          CLOSE(59)
          OPEN(58,ACCESS='APPEND',FILE='outtim
e.dat',STATUS='UNKNOWN')
          Close(58,status='delete')
c
C
      kcont=0
C
      ijstep=1
      imin = 1 + ijstep
      jmin = 1 + ijstep
      nim1 = ni - ijstep
      njm1 = nj - ijstep
          CALL PROPRI
          kk1=0
          comp=comp1
          DO WHILE (compr.GE.comp)
          ttub=twi1
          kk1=kk1+1
      DO 467 l=1,6 !o valor max é de 1464
500   niter = 0
      !IF(l.le.5) THEN
      !  sormax = .00001
      ! ELSE
       sormax=.000001 ! .000001 - valor ideal
      !END IF
      sorce=0.
      DO 400 k=1,ngrids
       ijstep=2**(ngrids-k)
       imin = 1 + ijstep
       jmin = 1 + ijstep
       nim1 = ni - ijstep
       njm1 = nj - ijstep
       CALL GRID
       IF (k.EQ.1) CALL INITVALUES
       IF ((k.GT.1).AND.(l.EQ.1)) CALL INTERPO-
LATE
       CALL COEFT
       sorce = 2. * sormax
320    IF ((niter.LT.maxit) .AND. (sorce.GT.sormax))
THEN
        IF ((MOD(niter,indpri).EQ.0).AND.(niter.GT.0))
CALL UPDT
        !IF (MOD (niter,indpri).EQ.0) CALL PRTOUT
             ! CALL UPDT
        niter = niter + 1
        CALL CALCT
        nik = (ni - 1) / ijstep + 1
        njk = (nj - 1) / ijstep + 1
        sorce = resort
        GOTO 320
       END IF
      CALL UPDT
      write(*,9400) nik,njk,niter,l,resort
400   CONTINUE
c
```

```
c--------------------------------------------------
c
    !CALL ENERGY
c     IF(kcont.EQ.1) GOTO 500
    told = t
c
c--------------------------------------------------
467  CONTINUE
    CALL ENERGY
         comp=comp+comp1
         acomp=comp-comp1
         IF
((comp.GT.compr).AND.(acomp.Lt.compr)) then
          acomp=compr-(comp-comp1)
          comp=(comp-comp1)+acomp
         comp1=acomp
         END IF
         END DO
         CALL PRTOUT
    PRINT*,'Tarefa concluida'
    STOP
9200  FORMAT(3E11.4)
9250  FORMAT (/)
9400  FORMAT(4I5,1PE12.2)
    END PROGRAM
         !RETURN
         !END SUBROUTINE
C--------------------------------------------------
    SUBROUTINE INITVALUES
    INCLUDE 'PrgSub'
    INCLUDE 'PrgLsolv'
    INCLUDE 'Prg1'
     IF((I.NE.1).OR.(kk1.NE.1)) THEN
!.AND.(kk1.NE.1)
      t=told
     ELSE
      t=0.0
     END IF
    RETURN
    END
C--------------------------------------------------
    SUBROUTINE INTERPOLATE
    INCLUDE 'PrgSub'
    INCLUDE 'PrgLsolv'
    INCLUDE 'Prg1'
C
    DO 3010 i=1, ni, 2*ijstep
    DO 3010 j=jmin, njm1, 2*ijstep
     t(i,j)=dyps(j)/(dynp(j)+dyps(j))*t(i,j+ijstep)+
   :       dynp(j)/(dynp(j)+dyps(j))*t(i,j-ijstep)
3010  CONTINUE
    DO 3020 i=imin, nim1, 2*ijstep
    DO 3020 j=1, nj, ijstep
     t(i,j)=dxpw(i)/(dxep(i)+dxpw(i))*t(i+ijstep,j)+
   :        dxep(i)/(dxep(i)+dxpw(i))*t(i-ijstep,j)
3020  CONTINUE
    RETURN
    END
C--------------------------------------------------
    SUBROUTINE PRTOUT
    INCLUDE 'PrgSub'
    INCLUDE 'PrgLsolv'
    INCLUDE 'Prg1'
c----------------- saida de resultados p/ ficheiros---
    OPEN (52, file='temp.dat', STA-
TUS='UNKNOWN')
         DO  i=1,ni
                  DO  j=1,nj
                  WRITE(52,'(3f9.4)')x(i),y(j),t(i,j)
                  END DO
         !WRITE(52,'(/)')
         !write(52,'(/)')
         END DO
         !DO j=1,nj
         !        DO i=1,ni
         !        WRITE(52,'(3f9.4)')x(i),y(i),t(i,j)
         !        END DO
         !WRITE(52,'(/)')
         !WRITE(52,'(/)')
         !
         !END DO
    CLOSE(52)
c--------------------------------------------------

    PRINT*
    PRINT*
1750  FORMAT (1x,50(F5.1,\),1x)
    RETURN
    END
C--------------------------------------------------
    SUBROUTINE GRID
    INCLUDE 'PrgSub'
    INCLUDE 'PrgLsolv'
    INCLUDE 'Prg1'
C
    dxpw(1) = 0.
    dxep(ni) = 0.
    DO 1010 i=1, nim1, ijstep
     dxep(i) = x(i + ijstep) - x(i)
     dxpw(i + ijstep) = dxep(i)
1010  CONTINUE
    dyps(1) = 0.
    dynp(nj) = 0.
    DO 1020 j=1, njm1, ijstep
     dynp(j) = y(j + ijstep) - y(j)
     dyps(j + ijstep) = dynp(j)
1020  CONTINUE
    sew(1) = 0.
    sew(ni) = 0.
    DO 1030 i=imin, nim1, ijstep
     sew(i) = (dxep(i) + dxpw(i)) / 2.
1030  CONTINUE
    sew(imin) = sew(imin) + dxpw(imin) / 2.
    sew(nim1) = sew(nim1) + dxep(nim1) / 2.
    sns(1) = 0
    sns(nj) = 0
    DO 1040 j=jmin, njm1, ijstep
    sns(j) = (dynp(j) + dyps(j)) / 2.
1040  CONTINUE
    sns(jmin) = sns(jmin) + dyps(jmin) / 2.
    sns(njm1) = sns(njm1) + dynp(njm1) / 2.
    RETURN
    END
C--------------------------------------------------
    SUBROUTINE UPDT
    INCLUDE 'PrgSub'
    INCLUDE 'PrgLsolv'
    INCLUDE 'Prg1'
C
    DO 6400 j=1,nj,ijstep
     t(ni, j) = t(nim1,j)
     t(1,j)=t(imin,j)
6400  CONTINUE
    DO 6410 i=imin,nim1,ijstep
         arean=sew(i)
         an1=tk(i,njm1)*arean/dynp(njm1)
```

```
c         !t(i,nj)=t(i,njm1)*an(i,njm1)+ae(i-
ijstep,nj)*t(i-ijstep,nj)+
c!    :ae(i,nj)*t(i+ijstep,nj)+tin(l)*sew(i)*alfa
c         !ap(i,nj)=an(i,njm1)+alfa*sew(i)+ae(i-
ijstep,nj)+ae(i,nj)
c    t(i,nj)=19.46
     t(i,nj)=t(i,njm1)*an1+tin(l)*sew(i)*alfa
         ap(i,nj)=an1+alfa*sew(i)
         t(i,nj)=t(i,nj)/ap(i,nj)
c
         IF(alfal.LT.1.E5) THEN
                an1=tk(i,jmin)*arean/dynp(1)

         t(i,1)=t(i,jmin)*an1+tlower*sew(i)*alfal
                ap(i,1)=an1+alfal*sew(i)
                t(i,1)=t(i,1)/ap(i,1)
         ELSE
                t(i,1)=tlower
         END IF
6410  CONTINUE
      RETURN
      END
C------------------------------------------------
      SUBROUTINE COEFT
      INCLUDE 'PrgSub'
      INCLUDE 'PrgLsolv'
      INCLUDE 'Prg1'
C
      DO 4100 j=jmin, njm1, ijstep
       areaew = sns(j)
       ae(1, j) = tk(1,j)*areaew/dxep(1)
4100  CONTINUE
      DO 4200 i=imin, nim1, ijstep
      arean = sew(i)
      an(i,1)=tk(i,1)*arean/dynp(1)
4200  CONTINUE
!
!      DO 4300 i=imin,nim1,ijstep
!          ae(i,nj)=tk(i,nj)*dyps(nj)/(2.*dxep(i))
!4300  CONTINUE
!
      DO 5100 i=imin, nim1, ijstep
      DO 5100 j=jmin, njm1, ijstep
       arean=sew(i)
       areaew=sns(j)
       vol = sew(i) * sns(j)
       an(i, j) = tk(i,j)*arean/dynp(j)
       ae(i, j) = tk(i,j)*areaew/dxep(i)
       IF (l.EQ.1) THEN
         su(i,j)=0
         ap(i,j)=0
       ELSE
        su(i,j)=rop(i,j)*cpp(i,j)*sns(j)*sew(i)/delt*told(i,j)
        ap(i,j)=rop(i,j)*cpp(i,j)*sns(j)*sew(i)/delt
      END IF
      CALL TUBO1(i,j)
5100  CONTINUE
      CALL MODT
      DO 5300 i=imin, nim1, ijstep
      DO 5300 j=jmin, njm1, ijstep
       ap(i,j)=ap(i,j)+an(i,j)+an(i,j-ijstep)+ae(i,j)+ae(i-
ijstep,j)
       ap(i,j)=ap(i,j)/urft
       su(i,j)=su(i,j)+(1-urft)*ap(i,j)*t(i,j)
5300  CONTINUE
      RETURN
      END
C------------------------------------------------
```

```
      SUBROUTINE TUBO1 (i,j)
      INCLUDE 'PrgSub'
      INCLUDE 'PrgLsolv'
      INCLUDE 'Prg1'
c ///////////////// #
      circu=0.
      circu=(x(i)-crx)*(x(i)-crx)
      circu=circu+(y(j)-cry)*(y(j)-cry)
      IF (circu.LE.ra**2.) THEN
          su(i,j)=ttub(l)*1.e20
          ap(i,j)=1.e20
      END IF
      RETURN
      END
c------------------------------------------------
c------------------------------------------------
C------------------------------------------------
      SUBROUTINE MODT
      INCLUDE 'PrgSub'
      INCLUDE 'PrgLsolv'
      INCLUDE 'Prg1'
C
      alfau=alfa
C         alfal=1.e10
          tinu=tin(l)
          tinl=tlower
c
          DO 6100 i=imin,nim1,ijstep
          an(i,njm1)=0.0
          ap(i,njm1)=ap(i,njm1)+alfau*sew(i)
          su(i,njm1)=su(i,njm1)+tinu*sew(i)*alfau
c
          IF (alfal.LT.1E5) THEN
                 an(i,1)=0.0
                 ap(i,jmin)=ap(i,jmin)+alfal*sew(i)

          su(i,jmin)=su(i,jmin)+tinl*sew(i)*alfal
          END IF
6100  CONTINUE
      DO 6400 j=1, nj, ijstep
       ae(nim1,j) = 0.
       ae(1,j)=0.
6400  CONTINUE
      RETURN
      END
C------------------------------------------------
      SUBROUTINE CALCT
      INCLUDE 'PrgSub'
      INCLUDE 'PrgLsolv'
      INCLUDE 'Prg1'
C
      resort = 0.
      nrstep=2**(ngrids-1)
      DO 5400 i=imin, nim1, nrstep
      DO 5400 j=jmin, njm1, nrstep
       sur=su(i,j)-(1-urft)*ap(i,j)*t(i,j)
       apr=ap(i,j)*urft
       resor=(an(i,j)*t(i,j+ijstep)+an(i,j-ijstep)*t(i,j-
ijstep)+
     :  ae(i,j)*t(i+ijstep,j)+ae(i-ijstep,j)*t(i-ijstep,j)-
     :  apr*t(i,j)+sur)/apr
       resort=resort+ABS(resor)
5400   CONTINUE
      resort=resort/(ni/nrstep-1)/(nj/nrstep-1)/ijstep
      IF(resort.GT.sormax)CALL LSOL-
VR(nswpt,imin,jmin,ni,nj,ijstep,t)
      RETURN
      END
```

```
C-----------------------------------------------------
    SUBROUTINE LSOLVR(nsweep, imin, jmin, ni,
nj, ijstep, phi)
        IMPLICIT REAL*8 (A-H,O-Z)
    REAL*8 a(200), b(200), C(200),
d(200),phi(200,200)
    INCLUDE 'PrgLsolv'
c
    nim1 = ni - ijstep
    njm1 = nj - ijstep
    jminm1 = jmin - ijstep
    iminm1 = imin - ijstep
    DO 1000 n=1,nsweep
     a(jminm1) = 0.
     DO 100 i=imin, nim1, ijstep
      C(jminm1) = phi(i, jminm1)
      DO 50 j=jmin, njm1, ijstep
      a(j) = an(i, j)
      b(j) = an(i, j - ijstep)
      c(j)=ae(i,j)*phi(i+ijstep,j)+ae(i-ijstep,j)*
   :       phi(i-ijstep,j)+su(i,j)
      d(j) = ap(i, j)
c
      term = 1 / (d(j) - b(j) * a(j - ijstep))
      a(j) = a(j) * term
      C(j) = (C(j) + b(j) * C(j - ijstep)) * term
50      CONTINUE
     DO 70 j=njm1,jmin, -ijstep
      phi(i, j) = a(j)*phi(i,j+ijstep)+C(j)
70      CONTINUE
100     CONTINUE
    a(iminm1) = 0.
    DO 200 j=jmin, njm1, ijstep
     C(iminm1) = phi(iminm1, j)
     DO 150 i=imin, nim1, ijstep
      a(i) = ae(i, j)
      b(i) = ae(i - ijstep, j)
      c(i)=an(i,j)*phi(i,j+ijstep)+an(i,j-ijstep)*
   :       phi(i,j-ijstep)+su(i,j)
      d(i) = ap(i, j)
c
      term = 1 / (d(i) - b(i) * a(i - ijstep))
      a(i) = a(i) * term
      C(i) = (C(i) + b(i) * C(i - ijstep)) * term
150     CONTINUE
     DO 170 i=nim1, imin, -ijstep
      phi(i, j) = a(i) * phi(i + ijstep, j) + C(i)
170     CONTINUE
200    CONTINUE
1000  CONTINUE
    RETURN
    END
C-----------------------------------------------------
    SUBROUTINE PROPRI
    INCLUDE 'PrgSub'
c
c    initialisation of the properties
    tk=0.0
        cpp=0.0
        rop=0.0
!
    DO 820 i=1,ni,ijstep
          rop(i,1)=ropy(1)
          cpp(i,1)=cppy(1)
          tk(i,1)=tky(1)
          rop(i,nj)=ropy(ny)
          cpp(i,nj)=cppy(ny)
          tk(i,nj)=tky(ny)
820  CONTINUE
    Do 800 n=1,ny
          DO 810 i=imin,nim1,ijstep
          DO 810 j=jmin,njm1,ijstep
          y1=ye(n)+ey(n)
          IF ((y(j).GE.ye(n)).AND.(y(j).LT.(y1)))
THEN
           rop(i,j)=ropy(n)
           cpp(i,j)=cppy(n)
           tk(i,j)=tky(n)
          END IF
          IF (n.NE.ny) THEN
           IF((y(j).LE.y1).AND.(y(j+ijstep).GE.y1))
THEN
            rop(i,j)=(ropy(n)+ropy(n+1))/2.
            cpp(i,j)=(cppy(n)+cppy(n+1))/2.

tk(i,j)=tky(n)*tky(n+1)*2./(tky(n)+tky(n+1))
           END IF
          END IF
810   CONTINUE
800  CONTINUE
    DO 830 j=1,nj,ijstep
     rop(1,j)=rop(1+ijstep,j)
          cpp(1,j)=cpp(1+ijstep,j)
          tk(1,j)=tk(1+ijstep,j)
     rop(ni,j)=rop(ni-ijstep,j)
          cpp(ni,j)=cpp(ni-ijstep,j)
          tk(ni,j)=tk(ni-ijstep,j)
830  CONTINUE
!    Pipe's Properties
    DO 825 i=imin,nim1,ijstep
          DO 825 j=jmin,njm1,ijstep
          racri=(x(i)-crx)**2.+(y(j)-cry)**2.
          IF (racri.LE.rae**2.) THEN
           tk(i,j)=tkp
           cpp(i,j)=cppp
           rop(i,j)=ropp
           circn=(x(i)-crx)**2.+(y(j+ijstep)-cry)**2.
           circs=(x(i)-crx)**2.+(y(j-ijstep)-cry)**2.
           circe=(x(i+ijstep)-crx)**2.+(y(j)-cry)**2.
           IF(circe.GE.rae**2.) THEN
            cpp(i,j)=(cppp+cpp(i+ijstep,j))/2.
            rop(i,j)=(rop(i+ijstep,j)+ropp)/2.

tk(i,j)=tk(i+ijstep,j)*tkp*2./(tk(i+ijstep,j)+tkp)
        END IF
           IF (circn.GE.rae**2.)THEN
            cpp(i,j)=(cpp(i,j+ijstep)+cppp)/2.
            rop(i,j)=(rop(i,j+ijstep)+ropp)/2.

tk(i,j)=tk(i,j+ijstep)*tkp*2./(tk(i,j+ijstep)+tkp)
        END IF
           IF(circs.GE.rae**2.) THEN
            cpp(i,j-ijstep)=(cpp(i,j-ijstep)+cppp)/2.
            rop(i,j-ijstep)=(rop(i,j-ijstep)+ropp)/2.
            tk(i,j-ijstep)=tk(i,j-ijstep)*tkp*2./(tk(i,j-
ijstep)+tkp)
           END IF
          END IF
          IF (racri.LT.ra**2.) THEN
           tk(i,j)=tkp
           cpp(i,j)=cppp
           rop(i,j)=ropp
         END IF
825  CONTINUE

826     RETURN
```

```
    END
C-----------------------------------------------
    SUBROUTINE ENERGY
    INCLUDE 'PrgSub'
    INCLUDE 'PrgLsolv'
    INCLUDE 'Prg1'
        REAL ttubin,tmedj,qfluxu,qfluxl
        CHARACTER*10 aa
    ttubin=ttub(l)
    CALL FLUXOCAL(qfluxu,qfluxl)
    CALL FLUXPIPE
c ----evaluation of outlet water temperaure and print
it -----
        a1=3.6/(wflow*4.18)
        twi1=ttub(l)+qtub(l)*a1
        qtub1=qtub(l)
C ----------------------
        somx=0.
         tmed=0.
    DO 8030 i=1,ni
         somx=somx+sew(i)
         tmed=tmed+t(i,njm1)*sew(i)

8030    CONTINUE
        tmed=tmed/somx
        tmedj=tin(l)-qflux/(alfa*boxl)
        goto 19
        WRITE(*,'(A,\)')'Ficheiro de saida->'
        READ(*,'(A)')aa
c----
        OPEN(51,file=aa,STATUS='unknown')
        WRITE(51,'(1x,a)')'---------------output
program simulation-----
        :-------------------'
        WRI-
TE(51,10)'Inlet','Outlet','Absorved','Upper','Lower','Up
per'    WRITE(51,15)'Water temp.','Water
temp.','Heat by','flux','flux',
        :'average Temp.'
        WRI-
TE(51,16)'[ºC]','[ºC]','[W]','[W/m2]','[W/m2]','[ºC]'
        WRITE(51,'(1x,a)')'------------------------
        :-------------------'
        WRITE(51,17)ttub(l),twi1,qtub(l),alfa*(tin(l)-
tmed),qfluxl,tmed

10
        FORMAT(1x,T4,A,(9x,A),(7x,A),(5x,A),(5x,A
),(5x,A))
15
        FORMAT(1x,T4,A,3x,A,(2x,A),(6x,A),(6x,A),
(6x,A))
16
        FORMAT(1x,T4,A,11x,A,(9x,A),(9x,A),(4x,A)
,(6x,A))
17      FOR-
MAT(1x,f8.3,7x,f8.3,(4x,f8.3),(4x,f8.3),(3x,f8.3),(5x,f8
.3))
        CLOSE(51)
19      CONTINUE

OPEN(51,ACCESS='APPEND',FILE='outtime.dat',ST
ATUS='UNKNOWN')
        WRI-
TE(51,20)ttub(l),twi1,qtub(l),qfluxu,qfluxl,tmed,
    :qtub(l)/comp1,comp1,comp
        CLOSE(51)
18  FORMAT(1x,I4,4(1x,F8.3))
20  FORMAT(1x,9(1x,f9.4))
C
    RETURN
    END
C_______________________________________
    SUBROUTINE FLUXOCAL (qfluxu,qfluxl)
    INCLUDE 'PrgSub'
    INCLUDE 'PrgLsolv'
    INCLUDE 'Prg1'
        REAL qfluxu,qfluxl


c
    qflux=0.
        qfluxu=0.
        qfluxl=0.
        areau=0.
        areal=0.
    DO 8200 i=1,ni
     qfluxu=qfluxu+alfa*sew(i)*(tin(l)-t(i,nj))
8200 CONTINUE
    areau=0.
    DO 8250 i=1,ni
        qflux=qflux+tk(i,njm1-1)*sew(i)*(t(i,nj)-
t(i,njm1))
    : /dynp(njm1)
        areau=areau+sew(i)
8250 CONTINUE
    areal=0.
        qfluxl=0.
    DO 8260 i=1,ni
        qfluxl=qfluxl+tk(i,jmin)*sew(i)*(t(i,1)-t(i,jmin))
        : /dynp(1)
        areal=areal+sew(i)
8260 CONTINUE
    qfluxu=qflux/areau
        qfluxl=qfluxl/areal
        !print*, qflux,qfluxu,qfluxl,areal,areau
        !pause
C
    RETURN
    END
c-----------------------------------------------
    SUBROUTINE FLUXPIPE
    INCLUDE 'PrgSub'
    INCLUDE 'PrgLsolv'
    INCLUDE 'Prg1'
        REAL tpipe,qsoma
c-----------------------------------------------
    qtub(l)=0.
        tpipe=ttub(l)
        a1=3.6/(wflow*4.18)
        nstep=INT(compr)
        !DO 2200 ncomp=1,nstep ,4
        qsoma=0.
        area=0.
    DO 2100 j=jmin,njm1,ijstep
    DO 2100 i=imin,nim1,ijstep
    qtubs=0.
    qtubn=0.
    qtubw=0.
    qtube=0.
c
    racri=(x(i)-crx)**2.+(y(j)-cry)**2.
    IF(racri.LE.ra**2.)THEN
c
     circe=(x(i+ijstep)-crx)**2.+(y(j)-cry)**2.
     circw=(x(i-ijstep)-crx)**2.+(y(j)-cry)**2.
     circn=(x(i)-crx)**2.+(y(j+ijstep)-cry)**2.
```

```
    circs=(x(i)-crx)**2.+(y(j-ijstep)-cry)**2.
    areans=sew(i)
    areaew=sns(j)
    IF(circe.GT.ra**2.) qtube=(t(i+ijstep,j)-
tpipe)*tk(i,j)*
   :sns(j)/dxep(i)
    IF (circn.GT.ra**2.) qtubn=(t(i,j+ijstep)-
tpipe)*tk(i,j)*
   :sew(i)/dynp(j)
    IF(circs.GT.ra**2.) qtubs=(t(i,j-ijstep)-
tpipe)*tk(i,j-ijstep)*
   :sew(i)/dynp(j-ijstep)
        IF(circe.GT.ra**2.) area=area+sns(j)
!     IF(circw.GT.ra**2.) area=area+sns(j)
    IF (circn.GT.ra**2.) area=area+sew(i)
    IF(circs.GT.ra**2.) area=area+sew(i)
   qsoma=qsoma+(qtubs+qtubn+qtubw+qtube)
        !
   END IF
2100 CONTINUE
        !Print*,'W por m',qsoma,'   area=',area
   qsoma=qsoma/area ! average temperature
        !print*,'por m2',qsoma
        qtub(l)=qsoma*ra*(3.141592)*2*comp1
!+qtub(l)
        !tpipe=a1*2.*qsoma*ra*(3.141592)+tpipe
   !PRINT*,qsoma
2200 CONTINUE

!       qtub(l)=qsoma
c
   RETURN
   END
c_______________________________________
        SUBROUTINE INPUTPRO
        INCLUDE 'PrgSub'
   INCLUDE 'PrgLsolv'
   INCLUDE 'Prg1'
        CHARACTER *12,name
c
c
          name ='inpfile.dat'

        OPEN(59,FILE=name,STATUS='unknown')
      READ(59,*)ny
        DO n=1,ny
         READ(59,220)ye(n)
         READ(59,220)ey(n)
         READ(59,220)ropy(n)
         READ(59,220)cppy(n)
         READ(59,220)tky(n)
        END DO
               READ(59,220)rae
        READ(59,220)ra
               READ(59,220)ropp
        READ(59,220)cppp
               READ(59,220)tkp
 READ(59,220)cry
        READ(59,220)boxh
        READ(59,220)boxl
        READ(59,220)alfa
        READ(59,220)comp1
               READ(59,220)wflow
!
               read(59,220)twi1
               read(59,220)tin1
               read(59,220)tlower
               read(59,220)compr
               read(59,220)alfal
!
     CLOSE(59)
     tin=tin1
     ye=ye*1e-3  !Convert Units
     ey=ey*1e-3  !Convert Units
     rae=rae*1.e-3 !Convert Units
     ra=ra*1.e-3 !Convert Units
     cry=cry*1.e-3
     boxh=boxh*1.0e-3
     boxl=boxl*1.0e-3
     !comp1=compr
c
200  FORMAT(1x,A,\)
205  FORMAT(I2)
210  FORMAT(1x,A8,I2,A12,F8.2,A23,\)
220  FORMAT(F9.4)
     RETURN
     END
c_______________________________________
```

## 11.2 Source code for the cooling tower subroutine

```
SUBROUTINE TOWERPRO
   IMPLICIT REAL*8 (A-H,O-Z)
        COMMON /vars/twi1,cw1,dt,jl,car,two(1464),utn1
        COMMON
/arp/p,w(1464),tar(1464),fia(1464),twb(1464),jmax
        DIMENSION twi(2),cw(2),ca(2),utn(2),temp(1464)
        INTEGER temp ,jmax
c
        OPEN(1,FILE='c:\slab\trans\dadin.dat',STATUS='U
NKNOWN')
        DO 50 i=1,2
        READ(1,*)tar(i),twi(i),two(i),cw(i),ca(i),fia(i)
50      CONTINUE
        READ(1,*)cw1
        CLOSE(1)
        p=1.0
        dt=0.1
c_______________________________________
        DO 55 i=1,2
        fia(i)=fia(i)/100.
        CALL NTU
(tar(i),twi(i),two(i),cw(i),ca(i),p,fia(i),dt,utn(i))
55      CONTINUE
c________ NTU - fitting
_______________________________________
        y1=log10(utn(1))
        y2=log10(utn(2))
        x1=log10(cw(1)/ca(1))
        x2=log10(cw(2)/ca(2))
        a=((y2-y1)/(x2-x1))
        b=y1-a*x1
c
c       WRITE(*,*)utn(1),utn(2),a,b
c       WRITE(*,*)cw(1),cw(2),ca(1),ca(2)
c       WRITE(*,10)'Cooling Tower - water flow [kg/s]->'
c       READ(*,15)cw1

        car=ca(1)
        utn1=10**(b+a*log10(cw1/ca(1)))
        OPEN(UNIT=1,FILE='c:\slab\trans\dadout.dat',STA
TUS='UNKNOWN')
```

```
        WRITE(1,*)utn(1),utn(2),utn1,a,b
        CLOSE(1)
c
        OPEN(1,FILE='c:\slab\trans\weather.dat',STATUS='
UNKNOWN')
        DO 60 jl=1,jmax
        READ(1,*)temp(jl),tar(jl),fia(jl)
60      CONTINUE
        CLOSE(1)
c
        DO 65 jl=1,jmax
        ps=psat(tar(jl))/9.8
        fia(jl)=fia(jl)/100.
        w(jl)=0.6219*(fia(jl)*ps/(p-fia(jl)*ps))
65      CONTINUE
C
        CALL TEMPW
C
        OPEN(1,FILE='c:\slab\trans\twtar.dat',STATUS='un
known')
        DO 90 jl=1,jmax
        WRITE(1,16)jl,tar(jl),fia(jl),w(jl),twb(jl)
90      CONTINUE
        CLOSE(1)
10      FORMAT(1x,A,\)
15      FORMAT(f6.3)
16      FOR-
MAT(1X,i5,1X,F5.2,1X,F5.2,1X,EN9.1,1X,F5.2)
        RETURN
        END !    PROGRAM
c_____________________________________
        SUBROUTINE NTU (ta-
rin,tagin,tagout,cl,cg,pat,fi,dt,antu)
    IMPLICIT REAL*8 (A-H,O-Z)
        DIMENSION tag(200),ha(200)
        n=(tagin-tagout)/dt
        tag(1)=tagout
        ps=psat(tarin)/9.8
        w=0.6219*(fi*ps/(pat-fi*ps))
        ha(1)=1.0*tarin+w*(1.86*tarin+2501.3)
        sntu=0.0
c
        DO 200 jl=2,n+1
        tag(jl)=tag(jl-1)+dt
        tagm=(tag(jl)+tag(jl-1))/2.
        ha(jl)=ha(jl-1)+cl/cg*4.19*dt
        ham=(ha(jl)+ha(jl-1))/2.
        hi=hasat(tagm)
        sntu=sntu+1/(hi-ham)
c
200     CONTINUE
        antu=4.19*cl*dt*sntu
        RETURN
        END
c_____________________________________
        FUNCTION psat (te)
    IMPLICIT REAL*8 (A-H,O-Z)
        IF ((te.GT.0.).AND.(te.LT.50.)) THEN
                psat=0.067112*1.061854**(te)
    END IF
        IF ((te.GE.50.).AND.(te.LT.80.)) THEN
                psat=0.135743*1.045663**(te)
        END IF
        END FUNCTION
c_____________________________________
        FUNCTION hasat(te)
    IMPLICIT REAL*8 (A-H,O-Z)
        hasat=4.7926+2.568*te-
.029834*te*te+0.0016657*te**3
        END FUNCTION
c_____________________________________
        SUBROUTINE TEMPW
    IMPLICIT REAL*8 (A-H,O-Z)
        COMMON
/arp/p,w(1464),tar(1464),fia(1464),twb(1464),jmax
        PARAMETER (imax=50)
        EXTERNAL psat
        EXTERNAL dw
        xacc=0.001
        DO 300 jl=1,jmax
         x1=tar(jl)-20
        if (x1.lt.0) x1=0.0
         x2=tar(jl)
         fmid=dw(tar(jl),w(jl),x2)
         f=dw(tar(jl),w(jl),x1)
         IF(f*fmid.GE.0) then
                write(*,*)jl
                pause 'erro'
        END IF
         IF(f.LT.0) THEN
                rtbis=x1
                dx=x2-x1
          ELSE
                rtbis=x2
                dx=x1-x2
         END IF
         DO 310 i=1,imax
                dx=dx*.5
                xmid=rtbis+dx
                fmid=dw(tar(jl),w(jl),xmid)
                IF(fmid.LE.0)  rtbis=xmid
                IF(ABS(dx).LT.xacc) GOTO 315
310      CONTINUE
315      twb(jl)=rtbis
300     CONTINUE
        RETURN
        END
c_____________________________________
        FUNCTION dw (te,wa,twet)
    IMPLICIT REAL*8 (A-H,O-Z)
         EXTERNAL psat
         real ps,ws,w1,w2
         ps=psat(twet)/9.8
         ws=0.62198*(ps/(1.-ps))
         w1=(2501.0-2.381*twet)*ws-(te-twet)
         w2=2501.0+1.805*te-4.186*twet
         dw=(wa-(w1/w2))
        END FUNCTION
c_____________________________________
        SUBROUTINE TWATER
    IMPLICIT REAL*8 (A-H,O-Z)
        COMMON /vars/twi1,cw1,dt,jl,car,two(1464),utn1
        COMMON
/arp/p,w(1464),tar(1464),fia(1464),twb(1464),jmax
        PARAMETER (imax=100)
        EXTERNAL psat
        xacc=0.01
        x1=twb(jl)
        x2=twi1
        call NTU (tar(jl),twi1,x1,cw1,car,p,fia(jl),dt,utn2)
    f=utn1-utn2
        call NTU (tar(jl),twi1,x2,cw1,car,p,fia(jl),dt,utn2)
        fmid=utn1-utn2
        IF (f*fmid.gt.0) then
           terror=0.1
c          IF(ABS(x1-x2).LT.2.0) terror=0.01
c          IF(ABS(x1-x2).GT.10) terro=.5
           err=0.01
           mcont=0
420        call NTU (tar(jl),twi1,x2,cw1,car,p,fia(jl),dt,utn2)
      mcont=mcont+1
      x22=x2
                IF (utn2.GT.utn1) THEN
      IF(ABS(x1-x2).GT.10.) terror=1.
           IF((ABS(x1-x2).LE.10.).AND.(ABS(x1-
x2).GT.2.)) terror=.1
      IF(ABS(x1-x2).LT.2.0) terror=0.01
                x2=x2+terror
                ELSE
           IF(ABS(x1-x2).GT.10.) terror=1.
           IF((ABS(x1-x2).LE.10.).AND.(ABS(x1-
x2).GT.2.)) terror=.1
      IF(ABS(x1-x2).LT.2.0) terror=0.01
                x2=x2-terror
```

```
              END IF
          IF (mcont.GE.40) THEN
            WRITE(*,*)ABS(x22-x2)
            x1=x22
            x2=x2
            call NTU (tar(jl),twi1,x1,cw1,car,p,fia(jl),dt,utn2)
      f=utn1-utn2
            call NTU (tar(jl),twi1,x2,cw1,car,p,fia(jl),dt,utn2)
            fmid=utn1-utn2
            WRITE(*,*)f,fmid
            GOTO 430
      END IF
                 if (abs(utn2-utn1).ge.(0.5)) goto 420
          xmid=x2
          goto 410
        END IF
430     IF (f.lt.0) THEN
          rtbis=x1
    dx=x2-x1
   ELSE
          rtbis=x2
          dx=x1-x2
        END IF
        DO 400 i=1,imax
        dx=0.5*dx
        xmid=rtbis+dx
              call NTU (tar(jl),twi1,xmid,cw1,car,p,fia(jl),dt,utn2)
              fmid=utn1-utn2
              IF (fmid.le.0) rtbis=xmid
              IF ((ABS(dx).lt.xacc).or.(fmid.eq.0)) GOTO 410
400   CONTINUE
        PAUSE ' TOO MANY BISECTIONS IN NTU???'

410   two(jl)=xmid
              RETURN
              END
C_______________________________________
```

# IEA Annex 28

## Low Energy Cooling

### Subtask 2: Detailed Design Tools

# FALSE FLOOR SLAB, AIR COOLED

VTT Building Technology, Finland
A.Laitinen

# Contents Chapter L

# 1. Technology area

The basic idea behind slab cooling is the exploitation of the thermal inertia of the building mass for the purpose of energy storage. The technology is an integral part of the ventilation system since channels in the slabs serve as the ducting for the ventilation air. The technology requires mechanical supply and exhaust ventilation .

During the summer, the system can be run during the night (or whenever the outdoor temperature is lower than the indoor temperature) to store cool energy in the building mass. The "coolth" is then transferred during the day to the supply air, thus decreasing its temperature. Part of the conditioning process takes place in the ductwork, where the heat exchange takes place between the supply air and the building mass. Final conditioning may be left for terminal units.

# 2. Developed by



| | | |
|---|---|---|
| **Name** | : | Ari Laitinen |
| **Organisation** | : | VTT Building Technology |
| **Address:** | | PO Box 1804 , SF-02044 VTT |
| **Phone** | : | +358-0-456 4721 (direct) |
| **Fax** | : | +358-0-455 2408 |
| **E-mail** | : | Ari.Laitinen@vtt.fi |



# 3. General description

The model allows very simplified study of a false floor slab with night cooling in order to calculate:

- the cooling capacity of the slab
- the supply air temperature after the slab
- the temperatures of the slab structure
- the surface temperatures inside and outside the slab.

The principle of false floor slab cooling is shown in Figure 1.

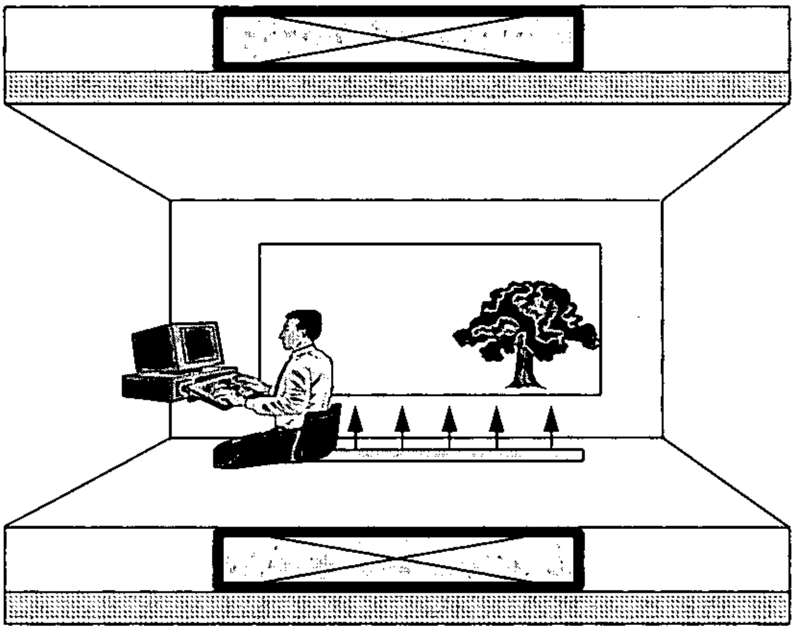


Figure 1 Slab Cooling

The general arrangement of the model is shown in Figure 2.

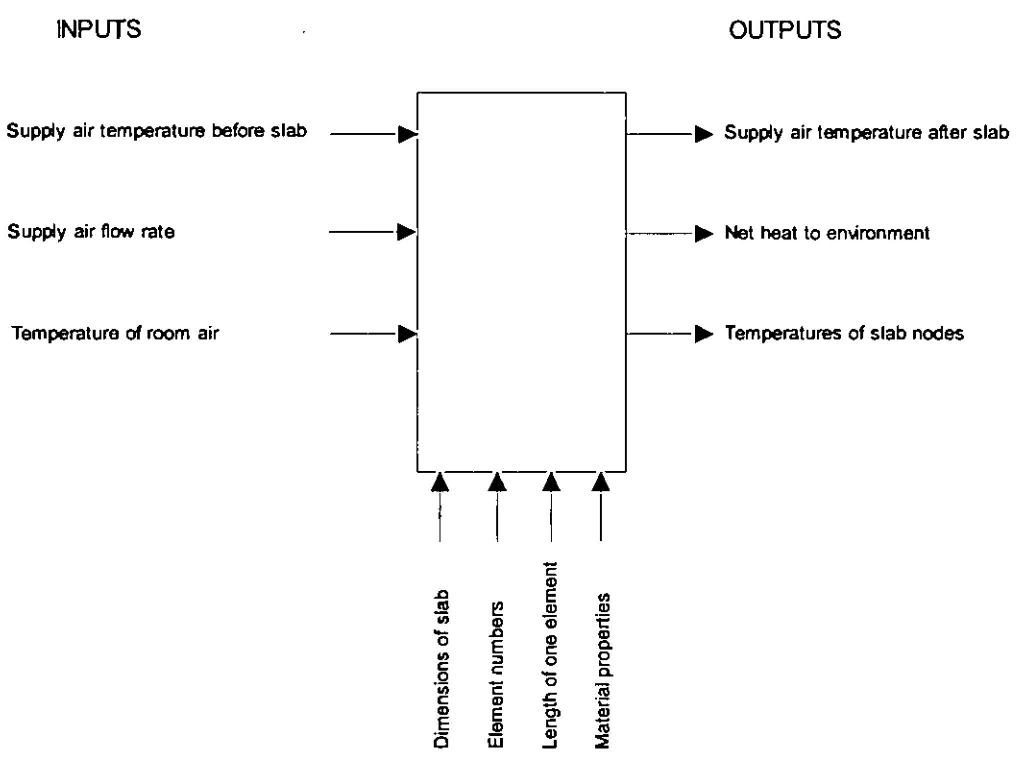



Figure 2 Information Flow Diagram

# 4. Nomenclature

Mathematical nomenclature

| | | |
|---|---|---|
| $\lambda$ | heat conductivity of the layer | (W/m,K) |
| d | thickness of the layer | (m) |
| $A_p$ | area between nodes | ($m^2$) |
| $c_p$ | specific heat capacity of the material | (J/kg,K) |
| $\rho$ | density of the material | ($kg/m^3$) |
| V | volume of the part of the structure | ($m^3$) |

Dimensionless numbers

| | | |
|---|---|---|
| Nu | Nusselt number ( $Nu = \frac{\alpha d_h}{\lambda}$ ) | (-) |
| Pr | Prandl number | (-) |
| Re | Reynolds number | (-) |
| f | friction factor | (-) |

# 5. Mathematical description

The thermal behavior of the false floor is modelled by using an RC-model which is analogous to electrical circuit modelling. For the calculation the false floor slab is divided lengthwise into elements as shown in Figure 3. The nodal network for one element in the simpliest case is shown in Figure 4. In every element the heat capacity of the structure is reduced to three nodes and heat balance is formulated between nodes. For more detailed studies the nodal network can be extended using the same approach.

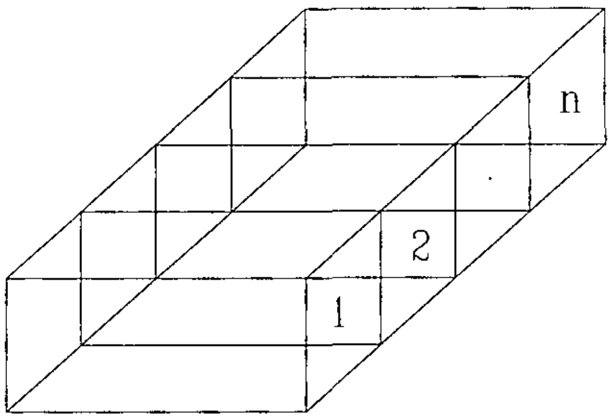



Figure 3 Principle of the element division

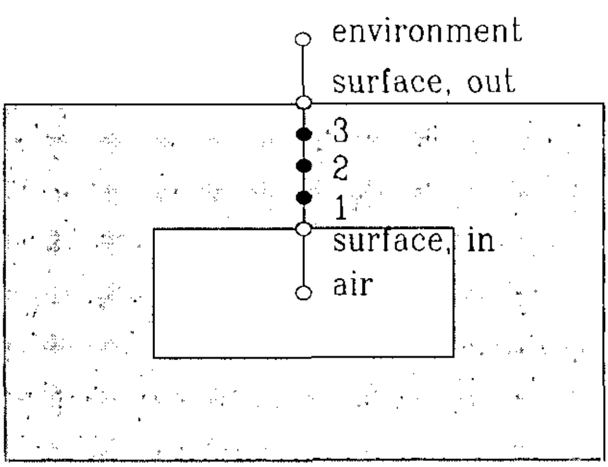



Figure 4 Nodal network for one element in the simpliest case

Heat conduction and capacity of the structure is calculated using eqns. 1 and 2:

$$G = \frac{\lambda}{d} A_p \qquad (1)$$

$$C = c_p \rho V \qquad (2)$$

The heat balance for the inside surface node of an element is:

$$\alpha_{in} (T_{air} - T_{surf,in}) - G_{surf,1} (T_{surf,in} - T_1) = 0 \qquad (3)$$

and respectively for the outside surface:

$$\alpha_{out} (T_{surf,out} - T_{envi}) - G_{3,surf} (T_3 - T_{surf,out}) = 0 \qquad (4)$$

Heat balances for the internal nodes are:

$$C_1 \frac{dT_1}{dt} = G_{surf,1}(T_{surf,in} - T_1) + G_{2,2}(T_1 - T_2) \qquad (5)$$

$$C_2 \frac{dT_2}{dt} = G_{1,2}(T_1 - T_2) + G_{2,3}(T_2 - T_3) \qquad (6)$$

$$C_3 \frac{dT_3}{dt} = G_{2,3}(T_2 - T_3) + G_{3,surf}(T_3 - T_{surf,out}) \qquad (7)$$

The heat balance between the air flow and the structure in every element is:

$$q_{mi} c_{pi} (T_{air,n-1} - T_{air,n}) = \alpha_{in} (T_{air} - T_{surf,in}) \qquad (8)$$

The convection heat transfer coefficient for the inside surface is calculated according to Kolar [1]:

$$Nu = 0{,}05814 * \sqrt{\Pr}\left(\mathrm{Re}\sqrt{\frac{f}{2}}\right)^{0{,}986} \qquad (9)$$

## 6. References

[1] Kolar, V., Heat transfer in turbulent flow of fluids through smooth and rough tubes, Int. J. Mass Tr., vol. 8, pages 639-653, 1965.

# 7. Algorithm

INITIALIZATION

- Set up data for the calculation

START TIME STEP

- Building model passes supply air temperature, air flow rate and room air temperatures to slab model
- Slab model performs a heat balance on the nodal network for current time step
- Slab model passes boundary heat fluxes and supply air temperature after the slab to the building model
- Building model uses heat fluxes and supply air temperature to calculate new room air temperatures etc.

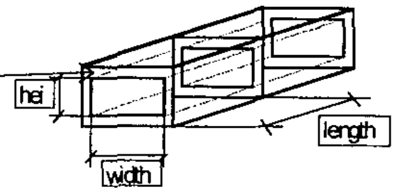
START NEW TIME STEP

# 9. Source code

The model has been programmed as an Excel workbook (on diskette) - printout below:

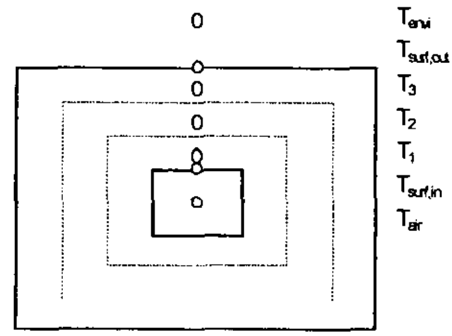
DIMENSIONS OF RECTANGULAR ELEMENT

| | |
|---|---|
| Height | 0,2 m |
| Width | 0,4 m |
| Construction thickness | 0,2 m |
| Length of one element use | 1,5 m |




CONSTRUCTION MATERIAL PROPERTIES

| Heat conductivity | Specific | Density |
|---|---|---|
| 1,7 | 940 | 2000 |




CALCULATED HEAT RESISTANCES AND CONDUCTANCES BETWEEN NODES

| $R_{surf\,in,1}$ | $R_{1,2}$ | $R_{2,3}$ | $R_{3,surf\,out}$ |
|---|---|---|---|
| 0,02 | 0,04 | 0,04 | 0,02 |
| 44,20 | 23,80 | 27,20 | 59,50 |
| 66,30 | 37,40 | 44,20 | 98,60 |

CALCULATED HEAT CAPACITIES OF NODES

| $C_1$ | $C_2$ | $C_3$ |
|---|---|---|
| 83556 | 100267 | 116978 |
| 93 | 111 | 130 |

HEAT TRANSFER COEFFICIENTS

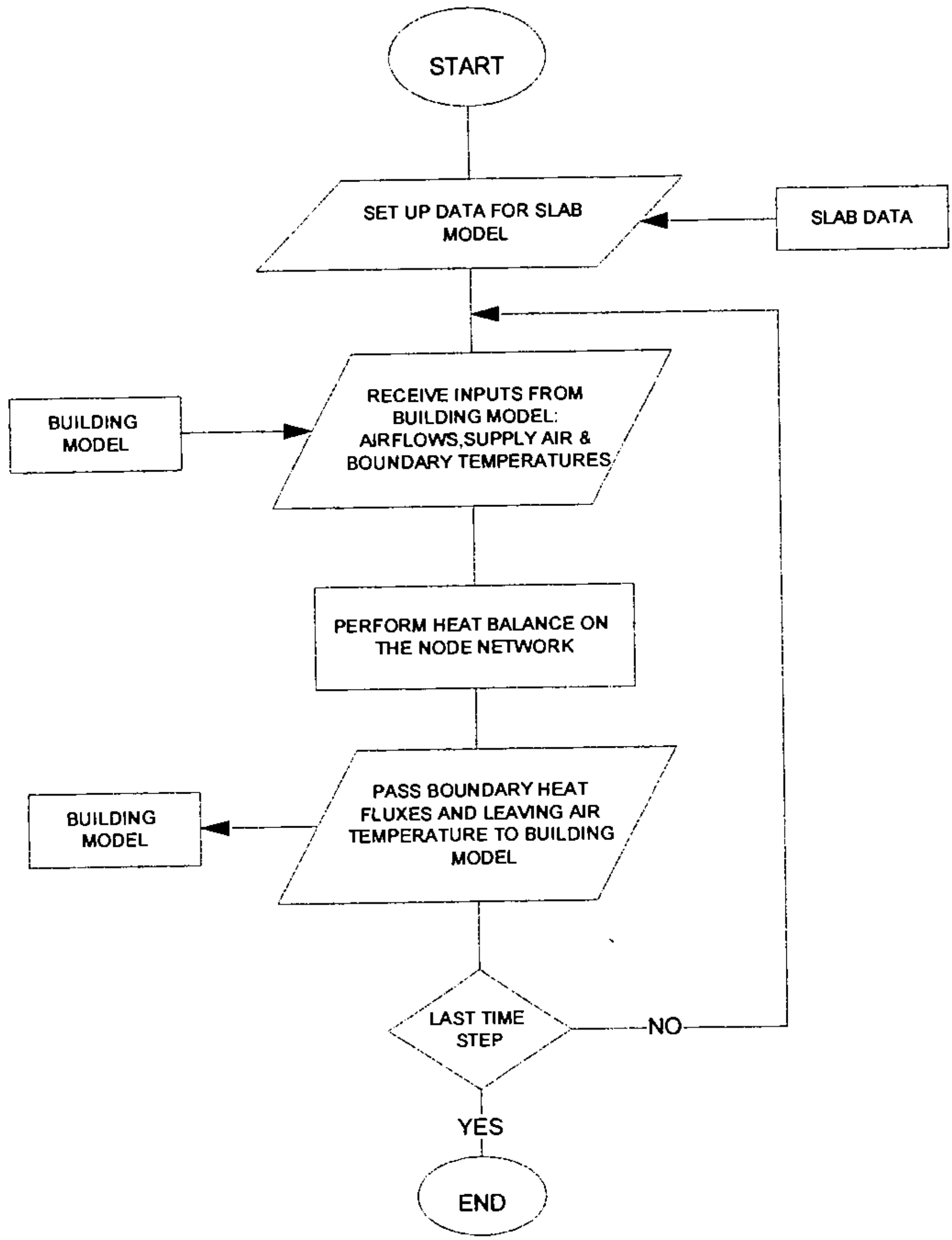
| | | Hydr.diam | Re | Nu | Friction fa | Nu |
|---|---|---|---|---|---|---|
| Velocity of air flow in the sla | 0,5 m/s | | | (Dittus-Boelter, AS | (Swamee- | (Kolar, TK 1 p.31) |
| Inside surface | 4,36 W/m².K | 0,27 | 12668 | 38,88 | 0,01 | 44,83 |
| Outside surface | 6 W/m²K | | | | | |
| Environment temp. | 23 °C | | | | | |

Calculation timestep 900 s

ROOM IDENTIFICATION

| | | Air capacity flowrate to the | Air capacity flowrate in the slab |
|---|---|---|---|
| Floor area | 10 m² | | |
| Supply air flowrate per floor | 3 dm³/s,m² | 36 W/K | 48 W/K |
| Room node capacity | 220000 J/K | | |
| Window conductance | 4,5 W/K | | |
| Inner wall conductance | 35 W/K | | |
| Inner wall node capacity | 250000 J/K | | |
| Outer wall conductance | 1,5 W/K | | |
| mass node capacity | 300000 J/K | | |
| conductance air-mass surfa | 315 W/K | | |
| conductance mass surface - | 350 W/K | | |

# 8. Flowchart


START
SET UP DATA FOR SLAB MODEL
SLAB DATA
BUILDING MODEL
RECEIVE INPUTS FROM BUILDING MODEL: AIRFLOWS,SUPPLY AIR & BOUNDARY TEMPERATURES
PERFORM HEAT BALANCE ON THE NODE NETWORK
BUILDING MODEL
PASS BOUNDARY HEAT FLUXES AND LEAVING AIR TEMPERATURE TO BUILDING MODEL
LAST TIME STEP
NO
YES
END

# 10. Sample results

As an example of the usage of the model, the thermal behavior of a typical office is presented. The simulated cases were heavy and light building structures combined with or without a false floor slab. The dimensions of the slab are shown in Chapter 9 in the printout of the Excel workbook. The outside temperature is defined as a cosine function, maximum temperature +26 °C (12 o'clock) and minimum temperature +18 °C. The heat gains and room air temperatures are shown in Figures 5 and 6.

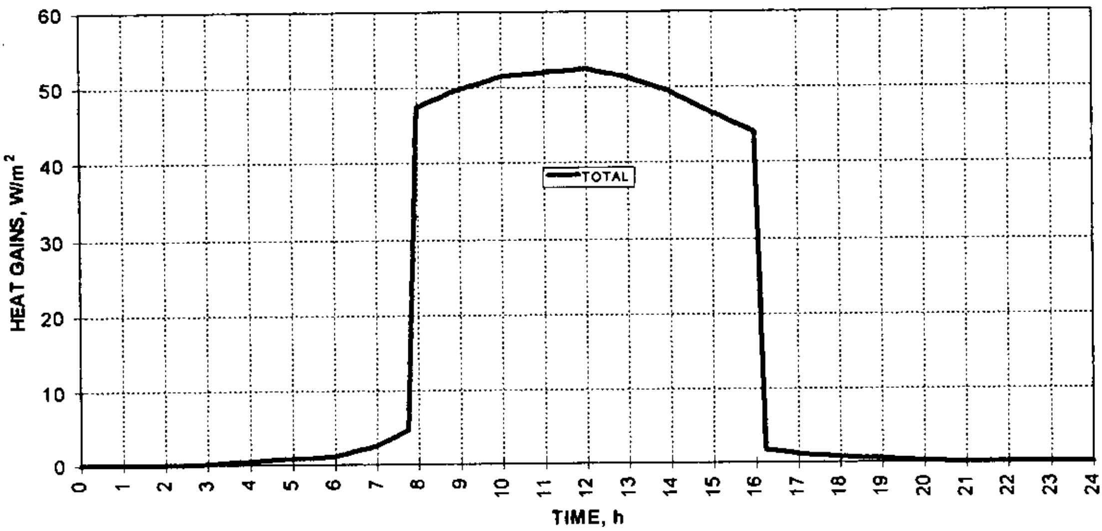



Figure 5 Total heat gain for office.

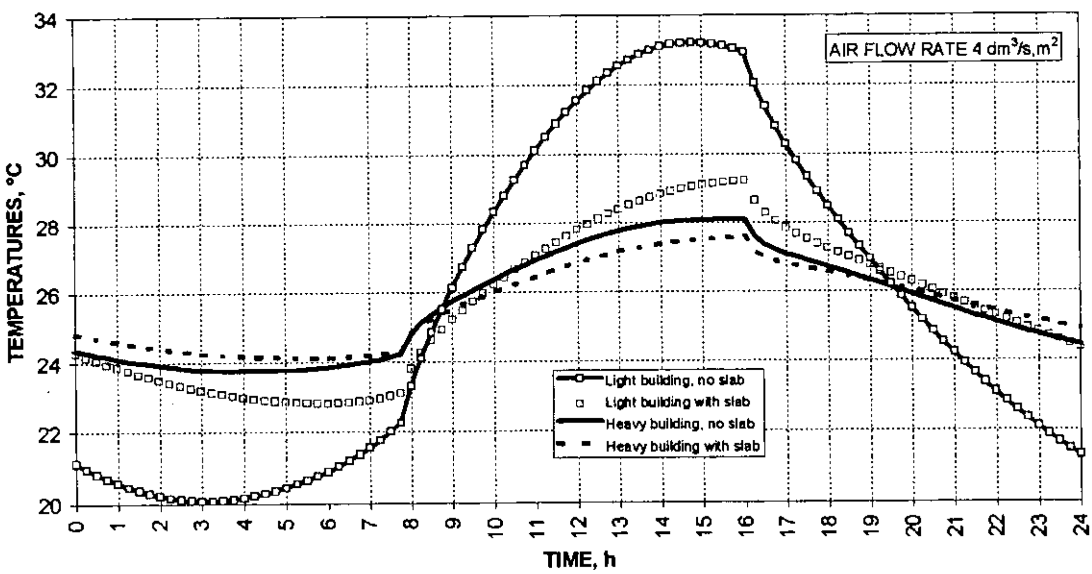



Figure 6 Room air temperatures.

# IEA Annex 28

## Low Energy Cooling

### Subtask 2: *Detailed Design Tools*

## Design Tool for an absorption cooling machine

## Computer programme for the Calculation of Heating Capacity

Institut für Luft-und Kältetechnik, Gemeinützige GmbH.
U.Franzke, A. Pietsch, C.Seifert

# Contents Chapter M

## 1. Technology area

**Computer programme for the calculation of heating capacity of an absorption cooling machine (ABSORPK).**

## 2. Developed by


**Name** : Dr.-Ing. Uwe Franzke, Dr.-Ing. Alexander Pietsch, Dipl.-Ing. Christian Seifert

**Organisation** : Institut für Luft- und Kältetechnik, gemeinnützige Gesellschaft mbH

**Address** : Germany, 01309 Dresden, Bertolt-Brecht-Allee 20

**Phone** : 49 (0) 351 / 4081650

**Fax** : 49 (0) 351 / 4081655

**E-mail** :


## 3. General description

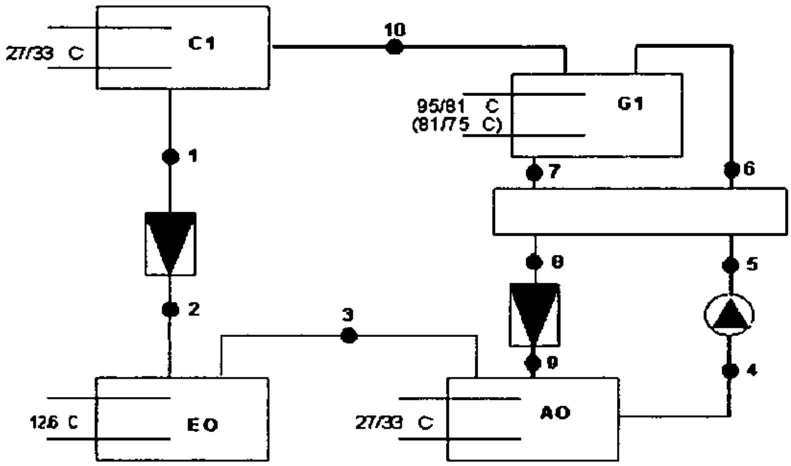
The algorithm calculates the required heating capacity of an absorption cooling machine. The machine is manufactured by York, type WFC - 10. For chilled ceiling applications the temperature of cold water varies only over a limited range, between 13 and 17 °C. In this range the COP of the cooling machine is approximately constant.

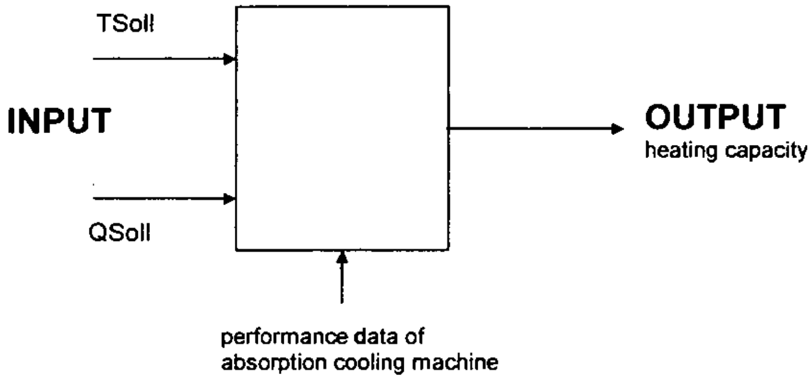
The condensing and absorption heat is rejected via a cooling tower. For middle european climates the cold water from the cooling tower to the condenser has a temperature between 27 and 33 °C. The refrigerant is water, the solvent is lithium bromide.

Heat recovery operates between the rich and poor side, which increasing the temperature of the rich and decreasing the temperature of the poor solvent.

Figure 1 shows the absorption cycle. Under design conditions the temperatures are as follows:

- inlet /outlet water temperature in condenser C1: 27 / 33 °C
- temperature difference between outlet water and condenser temperature: 2 K
- condenser temperature: 35 °C
- cold water temperature in evaporator E0: outlet / inlet: 6 / 12 °C
- the absorber A0 operates in absorption cycle mode
- heating temperature in generator G1: inlet / outlet : 95 / 81 °C




The task is to determine the power consumption of the absorption cooler as a function of the required cooling capacity. The power consumption is made up of the components heat energy requirement, the pump drive power and the drive power for the cooling tower fan. The heat energy requirement can be determined from the performance diagram through interpolation of the cooling capacity and the COP value for different temperature conditions. For pump power, constant water volumes are assumed for all operating conditions. The power consumption of the cooling tower fan was specified for the speed settings 480/960 rpm as being 60/340 W.

## 4. Nomenclature

| | Description | Units | Range |
|---|---|---|---|
| Input Variables | | | |
| TSoll | inlet water temperature in condenser | °C | 24 up to 31 °C |
| QSoll | needed cooling capacity | kW | 0 up to 60 kW |
| Output Variables | | | |
| QH | needed heating capacity | kW | |

## 5. Mathematical description

When providing cold water for chilled ceilings, the evaporation temperature varies within a limited range. The dissipation of the condensing and absorption heat is normally by way of cooling towers. In accordance with the air temperatures prevalent in the Central European region, cooling water supply temperatures of 27 to 33°C are assumed. With these assumptions the operating conditions of the absorption cooling system are effectively predetermined. Only the heating medium temperature will vary in accordance with the given heat source. For this application, the absorption liquid cooler WFC-10 manufactured by York was selected.

*From the technical data [1] it can be seen that with the parameters:*

| | |
|---|---|
| Heating water inlet | 95 °C |
| Cooling water inlet | 29,5 °C |
| Cold water outlet | 9 °C |

the cooler provides a cooling capacity of 46 kW. In the performance diagram it can be observed that the cooling capacity and the coefficient of performance (COP) are dependent on the cooling water inlet temperature, the heating water inlet temperature and the cold water outlet temperature. The performance *coefficient is identical to the thermal ratio* $Q_O/Q_H$.

It is assumed, in order to simplify the interpolation program, that the cold water outlet temperature for chilled ceilings need not be lower than 13 °C. The cooling capacities and COP values for all cooling water and heating water inlet temperatures are approximately constant for cold water outlet temperatures of 13 °C - 17 °C. (In future, it is intended to extend the range of application down to a cold water temperature of 7 °C.)

# 6. References

[1] Performance data of company York for the absorption liquid cooler WFC-10.

[2] Pietsch, A.: Modellbaustein Absorptionskälteanlage für „Stille Kühlung“
Forschungsbericht Nr.: ILK-B-5/95-116 vom 1.11.1995

[3] Seifert, C.: Lösungswege zur Berechnung des Betriebsverhaltens und Energieverbrauch der Komponenten der „Stillen Kühlung“.
Forschungsbericht Nr.: ILK-B-4/96-2500 vom 12.01.1996

# 7. Algorithm

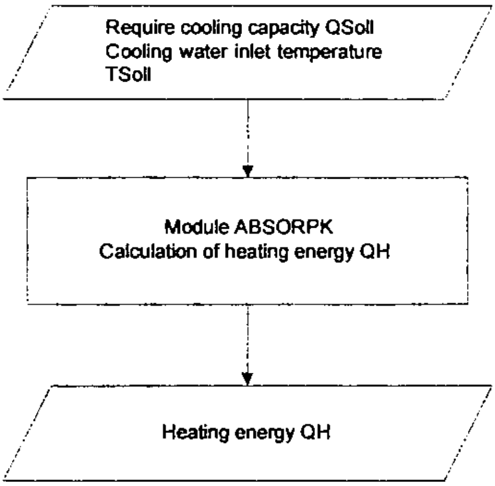
Input values for the simulation program are:

- 4 constants for the cooling capacity and COP value for each of 3 cooling water inlet temperatures:

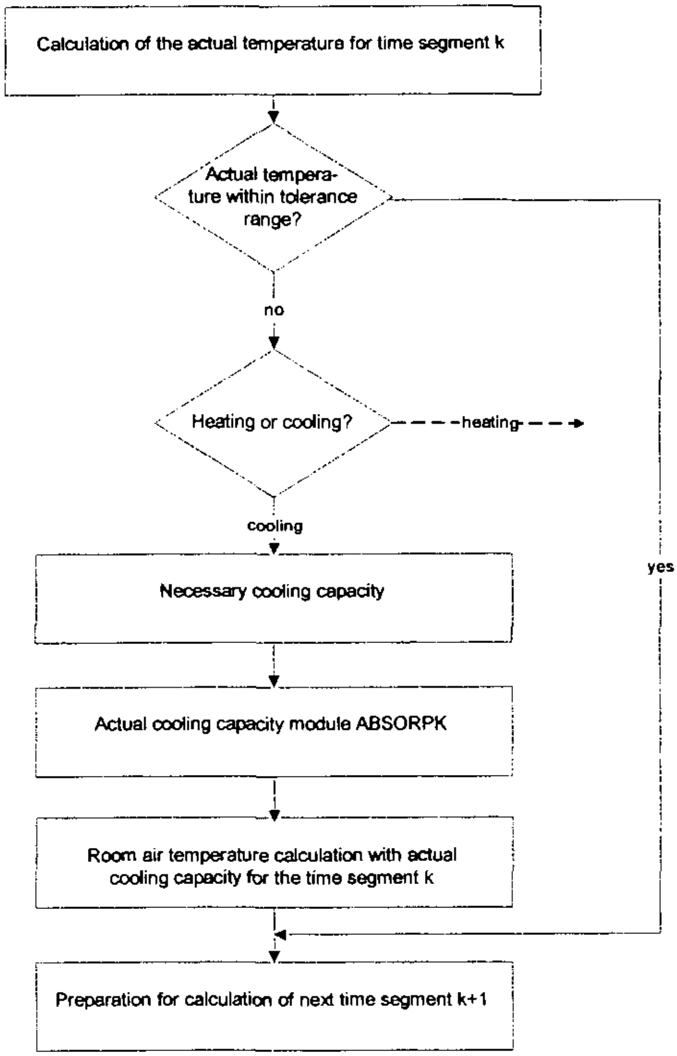
  K1 - K12, C1 - C12,

- the required cooling capacity QSOLL,
- the cooling water inlet temperature TSOLL

.



# 8. Flowchart


Calculation of the actual temperature for time segment k
Actual temperature within tolerance range?
no
yes
Heating or cooling?
heating
cooling
Necessary cooling capacity
Actual cooling capacity module ABSORPK
Room air temperature calculation with actual cooling capacity for the time segment k
Preparation for calculation of next time segment k+1

# 9. Source code

```
' Modul ABSORPK    ILK-Dr.Pietsch/Seifert-12/95
' Calculation of a absorption cooling machine
' -----------------------------------------------
'
' Input variables:     TSoll cooling water inlet temperature /øC/
'                      QSoll needed cooling capacity     /kW/
'
' Output variable:     QH    needed heating capacity    /kW/
'
' Constants for York WFC 10:

  DATA 24,24.5,35.5,47.5,59,0.7,0.72,0.77,0.78
  DATA 29.5,22,28,37.5,46.5,0.68,0.7,0.73,0.73
  DATA 31,15,24,31.5,44,0.52,0.615,0.7,0.715
  READ T1, K1, K2, K3, K4, C1, C2, C3, C4
  READ T2, K5, K6, K7, K8, C5, C6, C7, C8
  READ T3, K9, K10, K11, K12, C9, C10, C11, C12
  QH = 1000

  TSoll = 24  'cold water temperature from cooling tower

Plausibility:

  IF TSoll < 24 AND TSoll > 31 THEN QH = 0: END

Calculation:

  IF TSoll > T1 THEN GOTO A3

A:
  IF QSoll <= K1 THEN
    QSoll = K1
    QH = K1 / C1
    GOTO EndSorp
  ELSE
    IF QSoll > K2 THEN GOTO A1
    CSoll = C1 + (QSoll - K1) * (C2 - C1) / (K2 - K1)
    QH2 = QSoll / CSoll
    Test = QH2 - QH
    IF Test < -100 THEN
      QH = QH2
      GOTO EndSorp
    ELSE
      QH = QH2 + (QH - QH2) * (TSoll - T1) / (T2 - T1)
      GOTO EndSorp
    END IF
  END IF
```

```
A1:
  IF QSoll < K3 THEN
    CSoll = C2 + (QSoll - K2) * (C3 - C2) / (K3 - K2)
    QH3 = QSoll / CSoll
    Test = QH3 - QH
    IF Test < -100 THEN
      QH = QH3
      GOTO EndSorp
    ELSE
      QH = QH3 + (QH - QH3) * (TSoll - T1) / (T2 - T1)
      GOTO EndSorp
    END IF
  END IF


A2:
  IF QSoll < K4 THEN
    CSoll = C3 + (QSoll - K3) * (C4 - C3) / (K4 - K3)
    QH4 = QSoll / CSoll
    Test = QH4 - QH
    IF Test < -100 THEN
      QH = QH4
      GOTO EndSorp
    ELSE
      QH = QH4 + (QH - QH4) * (TSoll - T1) / (T2 - T1)
      GOTO EndSorp
    END IF
  ELSE
    QSoll = K4
    QH = K4 / C4
    GOTO EndSorp
  END IF

A3:
  IF TSoll > T2 THEN GOTO B2

B:
                         schleifb = schleifb + 1
                         IF schleifb > 100 THEN STOP
  IF QSoll > K5 THEN
    IF QSoll > K6 THEN GOTO B1
    CSoll = C5 + (QSoll - K5) * (C6 - C5) / (K6 - K5)
    QH6 = QSoll / CSoll
    Test = QH6 - QH
    IF Test < -100 THEN
      QH = QH6
      GOTO A
    ELSE
      QH = QH6 + (QH - QH6) * (TSoll - T2) / (T3 - T2)
      GOTO EndSorp
    END IF
  ELSE
    QSoll = K5
    QH = K5 / C5
 END IF
```

```
B1:
 IF QSoll < K7 THEN
    CSoll = C6 + (QSoll - K6) * (C7 - C6) / (K7 - K6)
    QH7 = QSoll / CSoll

   Test = QH7 - QH
    IF Test < -100 THEN
      QH = QH7
      GOTO A
    ELSE
      QH = QH7 + (QH - QH7) * (TSoll - T2) / (T3 - T2)
      GOTO EndSorp
    END IF
  END IF

  IF QSoll < K8 THEN
    CSoll = C7 + (QSoll - K7) * (C8 - C7) / (K8 - K7)
    QH8 = QSoll / CSoll
    Test = QH8 - QH
    IF Test < -100 THEN
      QH = QH8
      GOTO A
    ELSE
      QH = QH8 + (QH - QH8) * (TSoll - T2) / (T3 - T2)
      GOTO EndSorp
  ELSE
    END IF
    CSoll = C4 - (C4 - C8) * (TSoll - T1) / (T2 - T1)
    QH = QSoll / CSoll
    GOTO EndSorp
  END IF

B2:
   IF QSoll < K9 THEN
    QSoll = K9
    QH = K9 / C9
    GOTO EndSorp
  ELSE
    IF QSoll > K10 THEN
      GOTO B3
    ELSE
      CSoll = C9 + (QSoll - K9) * (C10 - C9) / (K10 - K9)
      QH = QSoll / CSoll
      GOTO B
    END IF
  END IF
B3:
  IF QSoll < K11 THEN
    CSoll = C10 + (QSoll - K10) * (C11 - C10) / (K11 - K10)
    QH = QSoll / CSoll
    GOTO B
  END IF
   IF QSoll < K12 THEN
    CSoll = C11 + (QSoll - K11) * (C12 - C11) / (K12 - K11)
    QH = QSoll / CSoll
    GOTO B
  ELSE
    CSoll = C8 - (C12 - C8) * (TSoll - T2) / (T3 - T2)
    QH = QSoll / Csoll
    GOTO EndSorp
  END IF
EndSorp
```

# 10. Sample results

**Inputs**

Tsoll = 29,5 °C

Qsoll = 46 kW

**Output**

QH = 63 kW

**Temperature range**

*cold water temperature = 13 ... 17 °C*