

HEAT TRANSFER:

NUMERICAL MODELLING WITH EES APPLICATIONS

ARMANDO CARLOS F. COELHO OLIVEIRA

HEAT TRANSFER:
NUMERICAL MODELLING WITH EES APPLICATIONS

1st edition

Published by the author

Copyright © 2024

No part of this publication may be reproduced or distributed in any form or by any means, or stored in a database or retrieval system, without the prior written consent of the author

This book is published electronically

ISBN 978-972-752-319-1

*“Remember that all models are wrong,
but some are useful.”*

G.E.P. Box

“The purpose of computing is insight, not numbers.”

R. Hamming

Forward

Most heat transfer problems are complex, in the sense that in most cases there are no analytical (exact) solutions to obtain temperature fields or heat transfer rates/coefficients. Thermal energy systems, and in general energy systems, involve heat transfer processes and, therefore, also require approximate solutions.

The approximate solution to non-linear systems of algebraic equations or differential equations is typically obtained with numerical methods, which became increasingly popular with the advent of affordable and fast computing. Second generation computer programs, such as Engineering Equation Solver (EES), do not even require conventional computer programming to solve mathematical problems, being based on the use of internal algorithms.

The simulation of heat transfer processes and thermal energy systems widely benefits from such tools, allowing fast approximate solutions, and also the study of alternative processes/systems through the use of parametric analyses. This book is dedicated to the numerical simulation of thermal energy systems, with the use of the EES software tool.

However, prior to the use of software tools, problem analysis is fundamental, in order to decide which type of model and degree of accuracy is acceptable. In this book, several heat transfer and thermal system problems are presented, coming from the energy engineering practice. After problem analysis and discussion, a numerical model is applied and solutions are obtained with the use of EES. Computed results are discussed, always trying to assess the effect of model assumptions on the results, and obtain conclusions which might be useful for the design of those systems.

The book starts by revisiting some well-known numerical methods to solve equations that appear on most practical cases (chapter 1). Global and distributed models are distinguished. In the case of distributed models, the finite volumes approach is favoured. A brief presentation of the EES software follows (chapter 2); however, this book does not replace the software manual, and it is advisable that the reader has some prior experience of EES. Chapters 3 and 4 present several examples of thermal system modelling, with chapter 3 dedicated to global modelling and chapter 4 to distributed modelling.

Porto, May 2024

Armando C. F. C. Oliveira

Contents

List of symbols

1	Numerical methods	1
1.1	Solution of non-linear equations	1
1.1.1	Simple-iteration and Newton-Raphson methods	1
1.1.2	Equation sets: Gauss-Seidel and Newton Raphson methods	3
1.2	Numerical integration in global models	4
1.2.1	Euler explicit method	5
1.2.2	Euler implicit method	6
1.2.3	Crank-Nicolson semi-implicit method	6
1.2.4	Other methods	7
1.3	Numerical integration in distributed models	7
1.3.1	Distributed steady-state modelling (1D, 2D, 3D)	7
1.3.2	Distributed dynamic modelling	10
2	Engineering Equation Solver (EES)	13
2.1	Writing equations in EES - equations window	13
2.1.1	Variables and calculation process	14
2.1.2	Physical properties and library data	15
2.2	Expressing varying conditions – functions and procedures	16
2.3	Tables	17
2.4	Graphical outputs	19
3	Global modelling examples	21
3.1	Air cooling system with thermostatically controlled valve	21
3.2	Thermal bottle heat transfer	24
3.3	Electric kettle	30
3.4	Electric water heater	36
3.5	Domestic hot water solar system	42
3.6	Swimming pool solar heating	49
4	Distributed and combined modelling examples	57
4.1	Steady-state conduction, convection and radiation in a rod	57
4.2	Dynamic heat transfer in a Trombe wall	61
4.3	Dynamic heat transfer in a ventilated Trombe wall	66
4.4	Car glass heating system (dynamic)	73
4.5	Dynamic cooling of a concrete beam	79
4.6	Dynamic heat transfer in a roof solar pond	84
4.6.1	Summer operation	85
4.6.2	Winter operation	90
4.7	Heat transfer in a laminar flow in a tube	93
4.8	Heat transfer in a laminar flow over a flat plate	100
4.9	Solar dryer heat and mass transfer	107
4.10	Dew point air cooler	113
4.11	Domestic hot water solar system with thermal storage stratification	120
	References	127

List of Symbols

Symbols are the simplified image of known physical quantities and are fundamental to write equations. In this text a set of symbols similar to those found in heat transfer literature was adopted. Relatively to most English language literature there are however a few differences: the main one is related to the heat flux and heat transfer rate, with the use of \dot{q} and \dot{Q} , respectively, and where the dot represents heat per unit time (thermal power).

Roman letters

Symbol	Designation	Unit
A	Area	m^2
c_p	Specific heat at constant pressure	$J/(kgK)$ or $J/(kg^\circ C)$
D	Diameter or Mass diffusion coefficient	m m^2/s
F	Function	-
f	Function or Friction factor	-
g	Gravity acceleration	m/s^2
\dot{g}	Heat rate source per unit volume	W/m^3
H	Height	m
$h; \bar{h}$	Convection coefficient or heat transfer coefficient or Enthalpy; Average coefficient over a given area/surface	$W/(m^2K)$ or $W/(m^2^\circ C)$ J/kg $W/(m^2K)$ or $W/(m^2^\circ C)$
I	Incident radiation	W/m^2
K	Local pressure loss coefficient	-
k	Thermal conductivity	$W/(mK)$ or $W/(m^\circ C)$
L	Length or reference dimension	m
M	Mass	kg
\dot{M}	Mass flow rate	kg/s
\dot{m}	Mass flow rate per unit area	$kg/s/m^2$
$Nu; \overline{Nu}$	Nusselt number; Nu average value	-
P	Power	W
p	Pressure	N/m^2 or Pa
Pr	Prandtl number	-
Q	Heat	J
\dot{Q}	Heat transfer rate	W
\dot{q}	Heat flux (heat transfer rate per unit area)	W/m^2
R	Thermal resistance	K/W or $^\circ C/W$
r	Radial space coordinate or Radius	m
Ra	Rayleigh number	-

Re	Reynolds number	-
S	Section area	m^2
s	Arc length or Spacing	m
$T; \bar{T}$	Temperature; Average Temperature	K or $^{\circ}C$
t	Time	s
U	Overall heat transfer coefficient	$W/(m^2K)$ or $W/(m^2^{\circ}C)$
V	Volume	m^3
v	Velocity	m/s
\bar{v}	Average Velocity	m/s
w	Width	m
wf	Weighting factor	-
x	Space coordinate or Independent variable in a function	m depending on associated variable
y	Space coordinate or Dependent variable in a function	m depending on associated variable
z	Space coordinate	m

Greek letters

Symbol	Designation	Unit
α	Thermal diffusivity or Absorption coefficient	m^2/s -
β	Thermal expansion coefficient	K^{-1} or $^{\circ}C^{-1}$
Δ	Variation	depending on associated variable
δ	Boundary layer thickness	m
ε	Error or Heat exchange efficiency	depending on associated variable - or %
η	Efficiency	- or %
θ	Angle, or Circumferential space coordinate	rad
μ	Dynamic viscosity	$kg/(s\ m)$
ν	Kinematic viscosity	m^2/s
ρ	Specific mass or Reflexion coefficient	kg/m^3 -
σ	Stefan-Boltzmann constant	$W/(m^2K^4)$
τ	Transmission coefficient	-
ϕ	Angle, or Circumferential space coordinate; Relative humidity	rad - or %
Ψ	Physical property	depending on specific property
ω	Absolute humidity	$kg_{water}/kg_{dry\ air}$

Subscripts* / Superscripts

Script	Designation
<i>abs</i>	relative to absorption of radiation
<i>amb</i>	ambient
<i>b</i>	base
<i>c</i>	characteristic, or corrected
<i>cond</i>	conductive
<i>conv</i>	convective
<i>em</i>	relative to emission of radiation
<i>evap</i>	evaporation
<i>ext</i>	external
<i>h</i>	hydraulic
<i>i</i>	initial or iteration number or node number associated to space coordinate
<i>in</i>	inlet
<i>inc</i>	relative to incident radiation
<i>int</i>	internal
<i>j</i>	node number associated to space coordinate
<i>l</i>	liquid
<i>m</i>	mass
<i>n</i>	normal (to a surface)
<i>out</i>	outlet
<i>rad</i>	radiative
<i>s</i>	section
<i>sat</i>	saturation
<i>sf</i>	solid-fluid interface
<i>sol</i>	relative to solar radiation
<i>sur</i>	surface
<i>t</i> (superscript)	at time <i>t</i>
<i>t + Δt</i> (superscript)	at time <i>t + Δt</i>
<i>tot</i>	total
<i>trans</i>	relative to radiation transmission
<i>v</i>	vapour
<i>x</i>	<i>x</i> coordinate/direction
<i>y</i>	<i>y</i> coordinate/direction
∞	far from surface (solid wall)

* these are general subscripts; in several practical examples more specific ones are used (not listed here)

1 Numerical methods

This chapter revisits some well-known numerical methods to solve equations that appear on thermal engineering practical cases. It starts with methods to solve non-linear equations. After distinguishing *global* and *distributed* models, some numerical integration methods are presented, applicable to steady-state and dynamic cases. For distributed models, the finite volumes method is favoured.

1.1 Solution of non-linear equations

The solution of a non-linear algebraic equation generally requires an iterative procedure. Two of the most used methods are the simple-iteration and the Newton-Raphson methods. The corresponding methods for non-linear equation sets are also presented.

1.1.1 Simple-iteration and Newton-Raphson methods

The simple-iteration method requires the equation to obtain the unknown value (x) in the form:

$$x = f(x) \quad (1.1)$$

The iterative process is started with an initial or guess value (x_0); then a better solution is obtained with

$$x_1 = f(x_0) \quad (1.2)$$

and the process is continued for more iterations:

$$x_{i+1} = f(x_i) \quad (1.3)$$

where i denotes the iteration number. The process is continued in the expectation that after some iterations the difference between x_{i+1} and x_i is very small. If this happens the process is stopped and x_{i+1} is considered to be the solution with an error smaller than $|x_{i+1} - x_i|$. However, in some cases the convergence to a solution does not happen. This depends both on the function (f) and on the initial guess (x_0).

Consider the two functions in Figure 1.1. In Figure 1.1(a) the function $f(x)$ has increasing y values – positive derivative – and in Figure 1.1(b) decreasing y values – negative derivative. In both cases the process is converging to a solution (x_{sol}). In the case of Figure 1.1(b) the iteration results are alternately located to the left and to the right of the solution, but always approaching (in module) the solution.

However, not all cases are successful. For the cases in Figure 1.2 no convergence is obtained, due to the function high derivative values (steep curves).

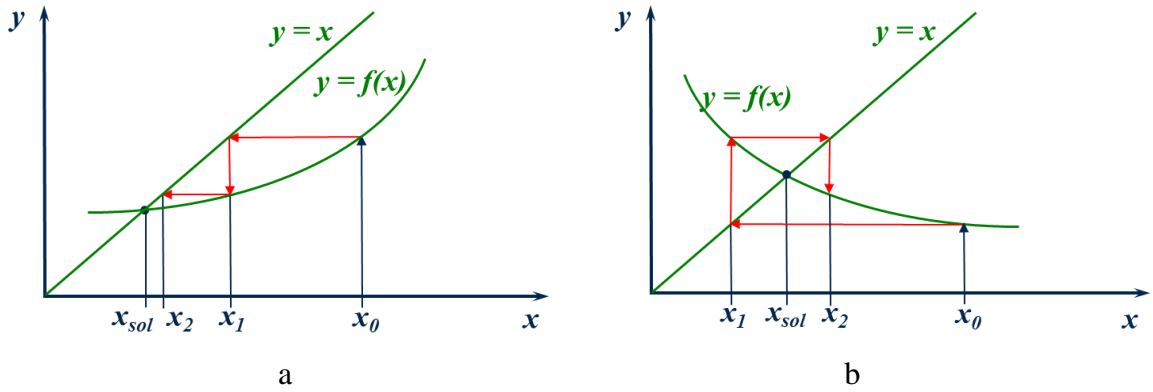


Figure 1.1 – Graphical representation of the simple-iteration method to obtain the solution of $x = f(x)$: (a) function with positive derivative (increasing y); (b) function with negative derivative (decreasing y).

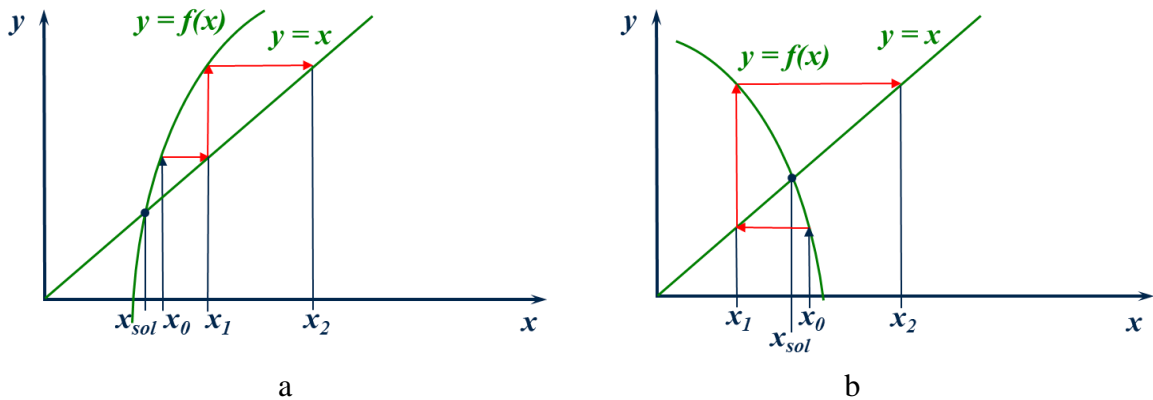


Figure 1.2 – Graphical representation of the simple-iteration method to obtain the solution of $x = f(x)$: (a) function with positive derivative (increasing y); (b) function with negative derivative (decreasing y).

In fact, it is a sufficient condition of convergence that

$$|f'(x_i)| < 1 \tag{1.4}$$

which means that low derivative values will lead to convergence. However, even with high values convergence may occur (the condition in equation (1.4) is not a necessary condition).

Another method to solve non-linear equations is the Newton-Raphson method. It is also an iterative method, with the equation to solve written in the form:

$$F(x) = 0 \tag{1.5}$$

Starting with an initial guess value, the value for the next iteration is obtained from the previous iteration one and the function derivative (F'). The relationship between 2 consecutive iteration values may be obtained through

$$\frac{F(x_i)}{F'(x_i)} \cong x_i - x_{i+1} \tag{1.6}$$

which is graphically represented in Figure 1.3 – the tangent to F at x_i is used.

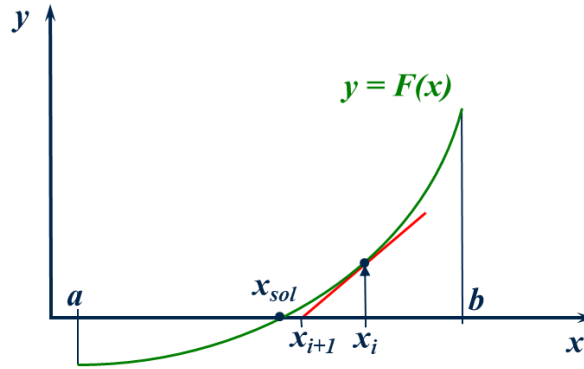


Figure 1.3 – Graphical representation of the iterative process in the Newton-Raphson method.

The same relationship of equation (1.6) may be obtained using the Taylor series expansion (with only the first order derivative):

$$F(x_{i+1}) \cong F(x_i) + \left. \frac{\partial F}{\partial x} \right|_{x_i} \cdot (x_{i+1} - x_i) = F(x_i) + F'(x_i) \cdot (x_{i+1} - x_i) \quad (1.7)$$

and imposing that $F(x_{i+1})$ becomes 0.

Although this method gives no guarantee of convergence, it is usually more efficient than the simple-iteration method, converging in more cases and with less iterations. But it requires the calculation of the derivative (F') in each iteration. Its success also depends on the initial guess value and function F .

1.1.2 Equation sets: Gauss-Seidel and Newton Raphson methods

Most energy/thermal systems have several components, and their numerical models involve several equations with several unknowns (independent variables – x_1, x_2, \dots, x_n). Generally, the equations are not linear. Therefore, the solution of a set of non-linear algebraic equations is required. Two methods are described, which correspond to the methods seen in 1.1.1, extended to more than one variable/equation.

The Gauss-Seidel method is an extension to equation sets of the simple-iteration method for single equations. The equation set is represented as

$$\begin{cases} x_1 = f_1(x_1, x_2, \dots, x_n) \\ x_2 = f_2(x_1, x_2, \dots, x_n) \\ \vdots \\ x_n = f_n(x_1, x_2, \dots, x_n) \end{cases} \quad (1.8)$$

and the successive iterations are defined by

$$\begin{cases} x_{1_{i+1}} = f_1(x_{1_i}, x_{2_i}, \dots, x_{n_i}) \\ x_{2_{i+1}} = f_2(x_{1_i}, x_{2_i}, \dots, x_{n_i}) \\ \vdots \\ x_{n_{i+1}} = f_n(x_{1_i}, x_{2_i}, \dots, x_{n_i}) \end{cases} \quad (1.9)$$

In each iteration the new values of the x_n variables are calculated independently, that is, the solution of independent non-linear single equations is repeated for the n equations. As in the simple-iteration method, the process is repeated until convergence of the values of all n variables.

The Newton-Raphson method for non-linear equation sets is also an extension of the method for non-linear single equations (keeping the same name). The equations are written as

$$\begin{cases} F_1(x_1, x_2, \dots, x_n) = 0 \\ F_2(x_1, x_2, \dots, x_n) = 0 \\ \vdots \\ F_n(x_1, x_2, \dots, x_n) = 0 \end{cases} \quad (1.10)$$

and the first order Taylor series expansion for each function is

$$\begin{aligned} F(x_{1_{i+1}}, \dots, x_{n_{i+1}}) &\cong F(x_{1_i}, x_{2_i}, \dots, x_{n_i}) + \left. \frac{\partial F}{\partial x_1} \right|_i \cdot (x_{1_{i+1}} - x_{1_i}) + \\ &+ \left. \frac{\partial F}{\partial x_2} \right|_i \cdot (x_{2_{i+1}} - x_{2_i}) + \dots + \left. \frac{\partial F}{\partial x_n} \right|_i \cdot (x_{n_{i+1}} - x_{n_i}) \end{aligned} \quad (1.11)$$

Using the expansion for all equations and imposing that $F(x_{1_{i+1}}, \dots, x_{n_{i+1}})$ becomes zero, the following set of equations is found

$$\begin{cases} -F_1(x_{1_i}, x_{2_i}, \dots, x_{n_i}) = \left. \frac{\partial F_1}{\partial x_1} \right|_i \cdot (x_{1_{i+1}} - x_{1_i}) + \dots + \left. \frac{\partial F_1}{\partial x_n} \right|_i \cdot (x_{n_{i+1}} - x_{n_i}) \\ -F_2(x_{1_i}, x_{2_i}, \dots, x_{n_i}) = \left. \frac{\partial F_2}{\partial x_1} \right|_i \cdot (x_{1_{i+1}} - x_{1_i}) + \dots + \left. \frac{\partial F_2}{\partial x_n} \right|_i \cdot (x_{n_{i+1}} - x_{n_i}) \\ \vdots \\ -F_n(x_{1_i}, x_{2_i}, \dots, x_{n_i}) = \left. \frac{\partial F_n}{\partial x_1} \right|_i \cdot (x_{1_{i+1}} - x_{1_i}) + \dots + \left. \frac{\partial F_n}{\partial x_n} \right|_i \cdot (x_{n_{i+1}} - x_{n_i}) \end{cases} \quad (1.12)$$

which allows the calculation of the x_{i+1} n values, since all previous iteration n values (x_i) are known; F_i and F'_i values may also be found with x_i values.

Therefore, the unknowns in the (1.12) equation set are the changes in the x_i values from iteration i to iteration $i+1$. The set can be solved as a linear set of equations, using standard methods such as the Gauss elimination method. In each iteration a different linear set has to be solved. The iterative process is stopped after convergence of all n variable values (x_i). As with single equations, this method is more efficient than the Gauss-Seidel method.

1.2 Numerical integration in global models

Besides algebraic equations, thermal energy models often contain differential equations. The type of differential equation encountered depends on the nature of the model: global or distributed. A global model is classified as one where properties of a system component are constant along its extension; this means that they do not change throughout a space coordinate. It is the case of a solid with a uniform temperature, or a tank filled with a fluid at uniform temperature. In a distributed model properties change along space coordinates.

In global modelling, if all components are in steady-state no time changes occur. Then, the equations that translate system operation are algebraic equations, and the model can be solved using the methods seen in section 1.1.2. But if time changes occur, there is a dynamic or unsteady situation, and to translate that, differential equations appear. Those equations express the change of a property as a function of time. The type of equation to solve in dynamic global modelling is

$$\frac{dy}{dt} = f(t, y) \quad (1.13)$$

where y is the property considered (dependent variable) and t (time) the independent variable. In many cases there is no possible analytical solution to equation (1.13), due to the nature of the function f . More than one property may be involved, and a set of differential equations needs to be solved.

For the (1.13) ordinary differential equation to be solved, numerical integration methods are presented in the next sections (1.2.1 to 1.2.4). They all transform the differential equation into an algebraic equation that may be solved with the methods seen in section 1.1.

1.2.1 Euler explicit method

The Euler explicit method allows the calculation of the value of y at different independent variable values (t). Those discrete time values are separated by Δt – the integration step. Starting with an initial value (y^t), the next is obtained with:

$$y^{t+\Delta t} = y^t + \Delta t \cdot f(t, y^t) = y^t + \Delta t \cdot \left. \frac{dy}{dt} \right|_t \quad (1.14)$$

Figure 1.4 illustrates the method. $F(t)$ is the exact solution of the differential equation. The derivative in the previous instant (t) is used to estimate the following value of the property ($y^{t+\Delta t}$). There is an error (ε) related to the difference to the exact solution $F(t + \Delta t)$; however, in the general case the exact solution is not known. Note that the smaller the Δt , the smaller the error. Also note that the error is cumulative: it increases step after step.

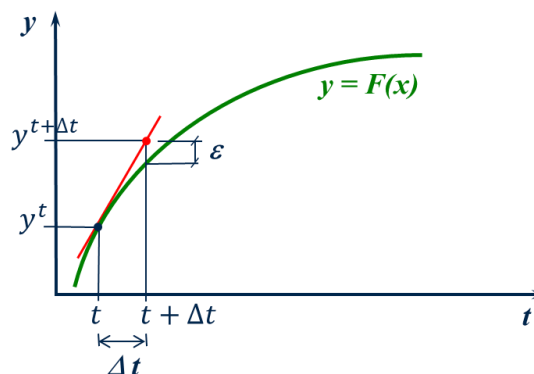


Figure 1.4 – Graphical representation of the Euler explicit method.

With this method, the calculation in each time step involves assessing only one derivative value. Due to this, it is classified as a first order integration method.

This method is not always stable, and depending on the F function may lead to unrealistic results. This also depends on the time step used (Δt), and it is advisable to use low values.

1.2.2 Euler implicit method

The Euler implicit method estimates the following value of the property ($y^{t+\Delta t}$) using the derivative in the next instant of time, through

$$y^{t+\Delta t} = y^t + \Delta t \cdot f(t + \Delta t, y^{t+\Delta t}) = y^t + \Delta t \cdot \left. \frac{dy}{dt} \right|_{t+\Delta t} \quad (1.15)$$

The process is represented in Figure 1.5. The segment used to estimate the next property value has the same slope as the derivative in the next point. The equation to obtain $y^{t+\Delta t}$ is implicit, and depending on the form of the derivative function (f) may require an iterative process to obtain the solution; the methods seen in section 1.1.1 may be used.

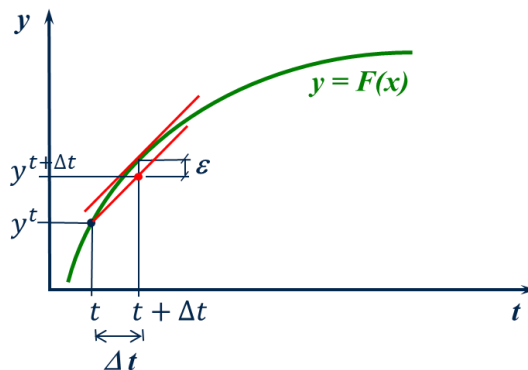


Figure 1.5 – Graphical representation of the Euler implicit method.

The Euler implicit method is also a first order integration method. Comparing Figures 1.4 and 1.5, one may conclude that both methods lead to an error with the same order of magnitude (module value). However, the implicit method is always stable, independently of the time step used.

1.2.3 Crank-Nicolson semi-implicit method

The Crank-Nicolson semi-implicit method is applicable when the derivative function (f) is only a function of y (and not explicitly t). Then

$$y^{t+\Delta t} = y^t + \Delta t \cdot f\left(\frac{y^t + y^{t+\Delta t}}{2}\right) \quad (1.16)$$

An example is the case of a body at uniform temperature under a cooling process: the change in temperature (derivative over time) is only a function of temperature and not time.

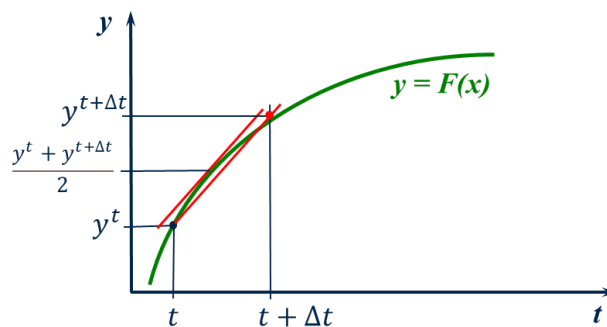


Figure 1.6 – Graphical representation of the Crank-Nicolson method.

Compared to the previous two methods, this one leads to lower errors (ε). Although it is only a first order integration method (one evaluation of f in each time step), it has a precision similar to second order integration methods. This semi-implicit method is always stable, independently of the time step used, and may require an iterative procedure in each time step.

1.2.4 Other methods

There are several other numerical integration methods for equation (1.13). Among them, the Euler modified method (second order), the Euler improved method (second order), or the Runge-Kutta method (fourth order). Of course, the calculation work and the precision increase with the increase in order.

However, due to the typical application of numerical integration methods to computing, the complexity of higher order methods is frequently replaced by simpler methods using a smaller integration step: more steps are needed, but with less calculations in each step and similar quality results.

1.3 Numerical integration in distributed models

In a distributed model, properties change along space coordinates. The spatial change of a given property depends on the physical process. But in transport phenomena, which occur in thermal engineering, the typical equations result from mass, momentum or energy balances for an infinitesimal volume. Those equations involve first and second order partial derivatives, where the independent variables are space coordinates, and also time in dynamic situations. We shall start by looking at steady-state cases.

1.3.1 Distributed steady-state modelling (1D, 2D, 3D)

In the case of a property (Ψ) varying along one space coordinate (x), the typical differential equation to solve has the following form

$$\frac{d^2\Psi}{dx^2} = f(x, \Psi) \quad (1.17)$$

We can take as an example the 1D heat conduction equation:

$$\frac{d^2T}{dx^2} = f(x, T) \quad (1.18)$$

which in the simplest case (1D, no internal heat sources, constant thermal conductivity) may be reduced to $d^2T/dx^2 = 0$, with a very simple analytical solution.

As a 2D example, the heat conduction equation in 2D cartesian coordinates, without sources and with constant conductivity, is

$$\frac{\partial^2 T}{\partial x^2} + \frac{\partial^2 T}{\partial y^2} = 0 \quad (1.19)$$

while the energy equation for a steady-state laminar flow in 2D may be simplified to

$$\rho c_p \left(v_x \frac{\partial T}{\partial x} + v_y \frac{\partial T}{\partial y} \right) = k \left(\frac{\partial^2 T}{\partial x^2} + \frac{\partial^2 T}{\partial y^2} \right) \quad (1.20)$$

There are different methods to transform those differential equations into sets of algebraic ones, that may then be solved with the methods seen in section 1.1.2.

These methods are named: finite differences, finite volumes, and finite elements.

The finite differences method replaces the first and second order derivatives by differences. If in a 1D situation, with variation of the Ψ property along x , we consider the 3 points separated by Δx in Figure 1.7:

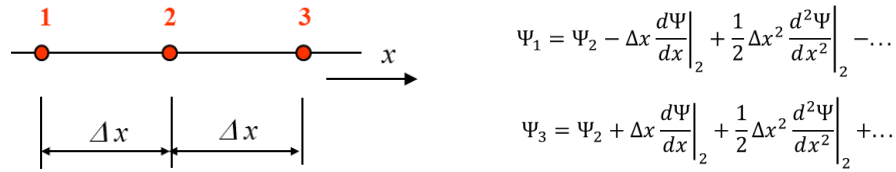


Figure 1.7 – Variation of the Ψ property along x and Taylor series expansion.

and using the expansion in Taylor series up to the second order term, and adding and subtracting the 2 equations in Figure 1.7, we will obtain for the first and second order derivatives in the mid-point (point 2):

$$\left. \frac{d\Psi}{dx} \right|_2 = \frac{\Psi_3 - \Psi_1}{2\Delta x} \tag{1.21}$$

and

$$\left. \frac{d^2\Psi}{dx^2} \right|_2 = \frac{\Psi_1 + \Psi_3 - 2\Psi_2}{\Delta x^2} \tag{1.22}$$

By dividing the spatial domain in discrete points (nodes), and replacing the first order and second order derivatives with expressions like (1.21) and (1.22), the differential equation to be solved will be replaced by a set of algebraic equations (one for each node), and its solution will lead to the values of the property Ψ in all nodes. In 2D and 3D cases, similar expressions to (1.21) and (1.22) are used to replace $\partial\Psi/\partial y$, $\partial\Psi/\partial z$, $\partial^2\Psi/\partial y^2$ and $\partial^2\Psi/\partial z^2$. In 2D each node will be identified by 2 numbers/subscripts (one for each coordinate) and in 3D by 3 numbers.

The finite volumes method, sometimes also known as control volume method, uses a physics approach instead of a mathematical one. As in the finite differences method, the definition of the shape of the volume elements to use depends on the system geometry. For instance, in 2D cartesian coordinates (plates) rectangular volumes are used, while in cylindrical coordinates circular sectors are used. Figure 1.8 represents the 2D volume elements used in the above cases. Each volume is located around a generic point in the material (P). This point is surrounded by another 4 points/volume elements in 2D (by 6 in 3D).

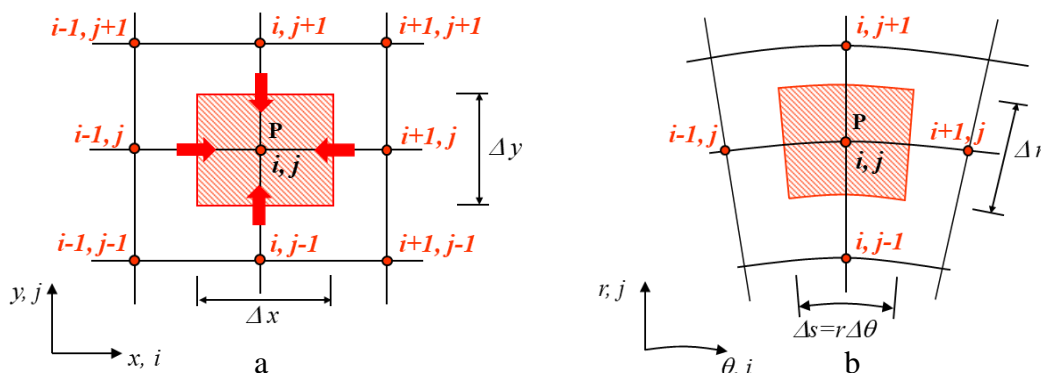


Figure 1.8 – Volume elements in the finite volumes method: (a) rectangular; (b) circular.

In the finite volumes method a balance of the property Ψ in the volume is made. For example, in the case of 2D heat conduction in a material ($\Psi = T$) there will be heat fluxes across the 4 volume borders; under steady-state the total balance of all fluxes multiplied by the respective areas will have to be zero. Considering that internal heat generation (\dot{g}) may occur, the equation for the general volume (i, j) in Figure 1.8(a), with constant thermal conductivity and constant Δx and Δy , will be

$$\begin{aligned} & \frac{k}{\Delta x} \Delta y (T_{i-1,j} - T_{i,j}) + \frac{k}{\Delta x} \Delta y (T_{i+1,j} - T_{i,j}) + \frac{k}{\Delta y} \Delta x (T_{i,j-1} - T_{i,j}) + \\ & + \frac{k}{\Delta y} \Delta x (T_{i,j+1} - T_{i,j}) + \dot{g} \Delta x \Delta y = 0 \end{aligned} \quad (1.23)$$

Taking into account a grid with several nodes and volume elements, there will be an equation analogous to (1.23) for each node/volume. The resulting algebraic equation set can be solved to calculate the temperatures in all nodes/volumes. The temperatures in intermediate points may be estimated by interpolation. The method is more accurate for smaller Δx and Δy values, that is, when more nodes/volumes are used. To obtain better solutions, the number of equations in the set is sometimes very high, which points to the use of computational means. The identification of symmetrical zones allows a reduction in the number of equations and related computational effort, as will be discussed in examples of chapter 4.

The nodes/volumes located in the domain frontiers (borders) require special attention. As recommended in [1], nodes should be placed at the borders, as represented in Figure 1.9(a). In this figure case, with a convective boundary in 2 sides (possibly with different external temperatures and heat transfer coefficients), the equation for the border volume will be

$$\begin{aligned} & \frac{k}{\Delta x} \frac{\Delta y}{2} (T_{i+1,j} - T_{i,j}) + \frac{k}{\Delta y} \frac{\Delta x}{2} (T_{i,j-1} - T_{i,j}) + \\ & + h_1 \frac{\Delta y}{2} (T_{ext,1} - T_{i,j}) + h_2 \frac{\Delta x}{2} (T_{ext,2} - T_{i,j}) + \dot{g} \frac{\Delta x \Delta y}{2} = 0 \end{aligned} \quad (1.24)$$

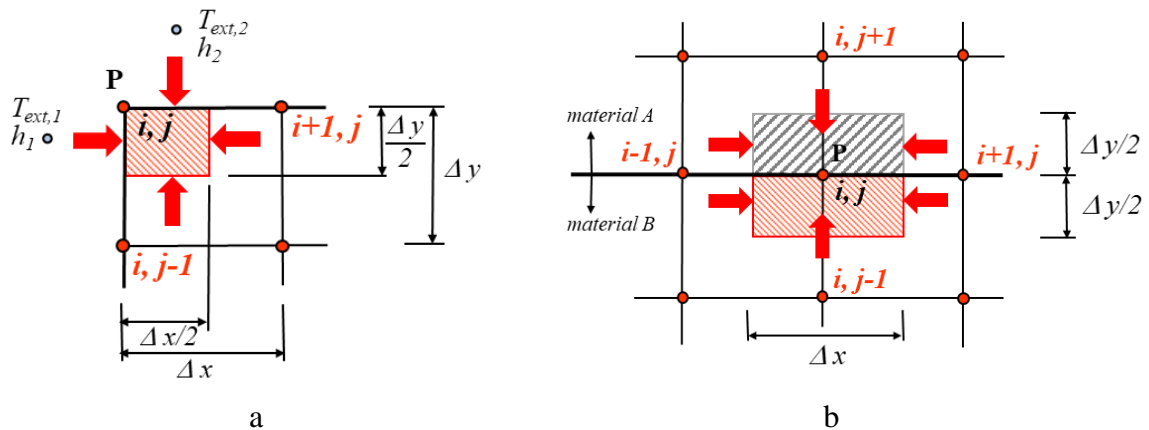


Figure 1.9 – Special situations in the finite volumes method: (a) boundary volume (corner); (b) volume with 2 different materials.

Another special case occurs in a boundary between 2 different materials. A single volume may be used, with the node in the boundary, and the volume including the 2 different materials – Figure 1.9(b). In this case the discretised equation will be

$$\frac{k_A \Delta y}{\Delta x} (T_{i-1,j} - T_{i,j}) + \frac{k_B \Delta y}{\Delta x} (T_{i-1,j} - T_{i,j}) + \frac{k_A \Delta y}{\Delta x} (T_{i+1,j} - T_{i,j}) + \frac{k_B \Delta y}{\Delta x} (T_{i+1,j} - T_{i,j}) + \frac{k_A \Delta x}{\Delta y} (T_{i,j+1} - T_{i,j}) + \frac{k_B \Delta x}{\Delta y} (T_{i,j-1} - T_{i,j}) + \dot{q}_A \Delta x \frac{\Delta y}{2} + \dot{q}_B \Delta x \frac{\Delta y}{2} = 0 \quad (1.25)$$

As for the finite elements method, it relies on a set of nodes and triangular elements (in 2D), distributed along the space domain. The boundaries are approximated by linear segments, which makes them easier to be adapted to curved boundary surfaces – see Figure 1.10.

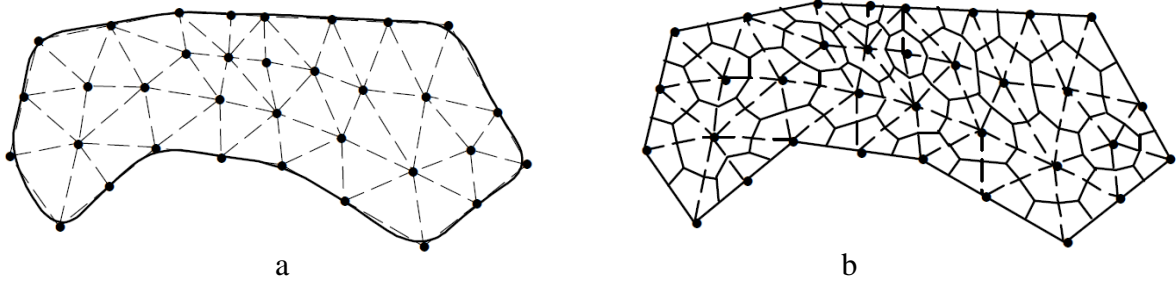


Figure 1.10 – 2D finite elements: (a) triangular elements and nodes; (b) nodal systems (closed frontiers).

In the case of heat conduction, in the elements of Figure 1.10(a) the temperature is supposed to vary linearly between the 3 nodes of each element. The problem consists in calculating the temperature in each node. Each triangular element has 3 degrees of freedom, because 3 nodal values are needed to calculate the temperature in any point inside the triangular element. Assuming a linear variation of the temperature in each side of the triangle, isothermal lines are normal to those sides, and volume elements with 6 isothermal segments around each internal node may be represented – Figure 1.10(b). An energy balance for each nodal system is performed, expressing the heat fluxes across the borders. For more mathematical details the reader is referred to [2].

In this book the finite volumes method will be favoured in all application examples of chapter 4. This is due to its more didactical and physics-based approach.

1.3.2 Distributed dynamic modelling

In this case the property to be assessed changes over space and time. We will consider the finite volumes method to integrate the differential equation, replacing it by a set of algebraic equations (discretised equations).

Besides the space coordinates, time is also an independent variable, and its first order derivative appears in transport equations. For example, in the 2D unsteady heat conduction equation:

$$\rho c_p \frac{\partial T}{\partial t} = k \left(\frac{\partial^2 T}{\partial x^2} + \frac{\partial^2 T}{\partial y^2} \right) \quad (1.26)$$

the temperature derivative over time appears in the left-hand side due to the change in energy contained in the volume over time.

Let us start by a simpler case: 1D unsteady heat conduction. The following equation applies:

$$\rho c_p \frac{\partial T}{\partial t} = k \frac{\partial^2 T}{\partial x^2} \quad (1.27)$$

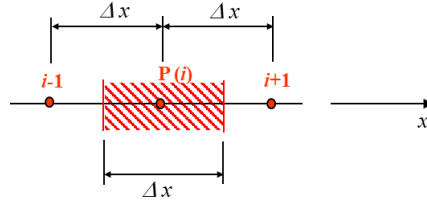


Figure 1.11 – Generic finite volume for 1D and dimensions.

Considering the finite volume in Figure 1.11 and assuming the temperature is uniform along its length (Δx) in every instant of time, the integration of the left-hand side leads to

$$\rho c_p \int_{\Delta x} \int_t^{t+\Delta t} \frac{\partial T}{\partial t} dt dx = \rho c_p \Delta x (T_i^{t+\Delta t} - T_i^t) \quad (1.28)$$

The right-hand side comes from the balance of heat fluxes across the volume borders, and may be integrated as

$$\int_t^{t+\Delta t} \int_{\Delta x} k \frac{\partial^2 T}{\partial x^2} dx dt = \int_t^{t+\Delta t} \left(\frac{k(T_{i+1} - T_i)}{\Delta x} - \frac{k(T_i - T_{i-1})}{\Delta x} \right) dt \quad (1.29)$$

To solve the time integral in (1.29) it is necessary to assume a variation of the temperatures along time. Among other possibilities, it is assumed that, for all temperatures

$$\int_t^{t+\Delta t} T dt = [wf T^{t+\Delta t} + (1 - wf) T^t] \Delta t \quad (1.30)$$

where wf is a weighting factor which may take values from 0 to 1. With its definition the discretised form of equation (1.27) for volume i is:

$$\begin{aligned} \rho c_p \frac{\Delta x}{\Delta t} (T_i^{t+\Delta t} - T_i^t) = & wf \left(\frac{k(T_{i+1}^{t+\Delta t} - T_i^{t+\Delta t})}{\Delta x} - \frac{k(T_i^{t+\Delta t} - T_{i-1}^{t+\Delta t})}{\Delta x} \right) + \\ & + (1 - wf) \left(\frac{k(T_{i+1}^t - T_i^t)}{\Delta x} - \frac{k(T_i^t - T_{i-1}^t)}{\Delta x} \right) \end{aligned} \quad (1.31)$$

This equation relates the temperature of node i at $t + \Delta t$ with its temperature at t and with the temperatures of the neighbour nodes/volumes in both instants ($t + \Delta t$ and t). The choice of the value of the weighting factor (wf) may affect the form of equation (1.31). Three typical wf values are considered in Figure 1.12. A value $wf = 0$ corresponds to assuming that the temperatures remain constant at the t values during Δt ; therefore $T_i^{t+\Delta t}$ will only depend on the temperatures in the previous instant; equation (1.31) allows the explicit calculation of $T_i^{t+\Delta t}$, as all temperatures at t are previously known.

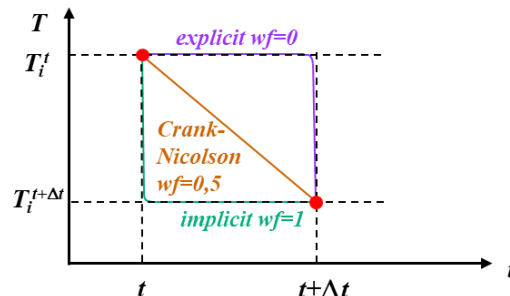


Figure 1.12 – Decreasing temperature time evolution according to the weighting factor value.

A value $wf = 0.5$ expresses a linear variation of the temperature between t and $t + \Delta t$ and, similarly to the method seen in section 1.2.3, it is called Crank-Nicolson or semi-implicit method. If $wf = 1$, the temperatures are assumed to immediately change to the $t + \Delta t$ level at t – implicit method.

Of the 3 values, Patankar [1] recommends the use of the implicit method, which allies simplicity with stability. The explicit method, as in the case of section 1.2.1, may lead to unstable results, requiring a relatively low time step.

With the use of the implicit formulation, equation (1.31) becomes

$$\rho c_p \frac{(T_i^{t+\Delta t} - T_i^t)}{\Delta t} = \frac{k}{\Delta x^2} (T_{i+1}^{t+\Delta t} + T_{i-1}^{t+\Delta t} - 2T_i^{t+\Delta t}) \quad (1.32)$$

Compare this equation with the application of the finite differences seen in equations (1.21) and (1.22) to equation (1.27). A similar result is obtained.

When treating higher space dimensions (2D, 3D) the same principles apply. More nodes are involved in each discretised equation – 4 nodes in 2D and 6 nodes in 3D, besides the central node P. For example, in the 2D unsteady heat conduction equation (cartesian coordinates) with the implicit formulation we will have:

$$\begin{aligned} \rho c_p \Delta x \Delta y \frac{(T_{i,j}^{t+\Delta t} - T_{i,j}^t)}{\Delta t} = & \frac{k}{\Delta x} \Delta y (T_{i-1,j}^{t+\Delta t} + T_{i+1,j}^{t+\Delta t} - 2T_{i,j}^{t+\Delta t}) + \\ & + \frac{k}{\Delta y} \Delta x (T_{i,j-1}^{t+\Delta t} + T_{i,j+1}^{t+\Delta t} - 2T_{i,j}^{t+\Delta t}) \end{aligned} \quad (1.32)$$

The difference between this equation and equation (1.23) for the 2D steady-state (excluding the heat source) lies on the left-hand side, which expresses the time variation.

Applications to 1D and 2D cases will be seen in chapter 4. The case of fluid flow, using the integration of equation (1.20), will also be discussed there.

2 Engineering Equation Solver (EES)

Engineering Equation Solver is used in this book as a computational tool to solve different modelling examples presented in chapters 3 and 4. As the name states, it is basically an equation solver tool, with an internal algorithm to solve sets of non-linear equations. This can be done with other existing software tools, such as MATLAB. However, EES has many advantages for thermal energy applications, due to an extensive database of fluid physical properties, and internal calculation tools related to heat transfer and fluid flow. It also allows to take into account changes in problem equations (conditions), through combined programming.

A very quick overview of EES is given in this chapter. It concentrates on its most important features, taking into account the objectives of this book. For further details the reader should use the software manual, [3], namely regarding software installation.

2.1 Writing equations in EES – equations window

The EES environment includes a main menu (Figure 2.1) and different windows are available. The main window is called *Equations Window*, where all equations and main problem conditions are defined.

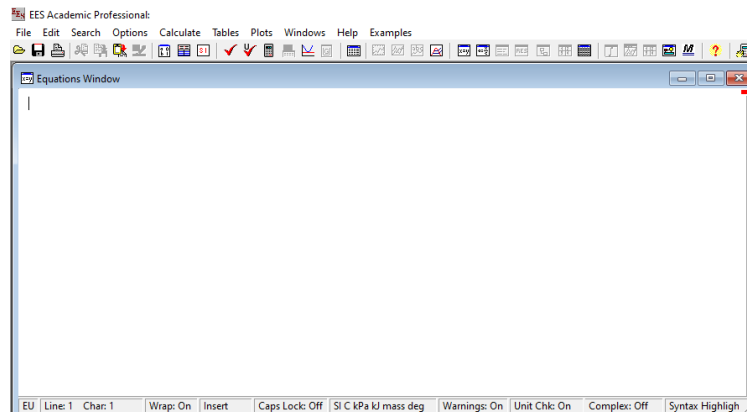


Figure 2.1 – EES menu and Equations Window.

The main menu includes different options (File, Edit, Search, Options, Calculate, Tables, Plots, Windows, Help and Examples – see Figure 2.1), and some sub-options can be directly assessed by the buttons located below.

The definition of the problem variables used in the equations, and the related calculation process is referred in the next sub-section. The definition of physical properties and library databases will be referred in sub-section 2.1.2.

2.1.1 Variables and calculation process

A set of equations to solve is defined in the Equations Window. Variables are identified as any combination of letters (capitals or smallcases without distinction) and numbers. In the example of Figure 2.2, 3 variables were set ($X1$, $X2$ and $X3$) plus the definition of $A=2$. EES identifies 4 equations and 4 variables, as although A has its value defined, it counts as a variable in the Equations Window. The $**$ or the $^$ symbols are used as the “to the power of” symbol.

Array variables may also be used, and they are very useful to express equations such as (1.32). They are written with square brackets around the indices (1, 2 or 3 indices), after the variable name, such as $X1[i]$, $T[i, j]$ or $Temp[i, j, k]$.



Figure 2.2 – Example of problem definition in the Equations Window.

A list of the variables in the Equations Window may be assessed by choosing in the menu Options, followed by Variable Info (Options → Variable Info). This will open the window in Figure 2.3. This window lists the 4 variables, and more information can be added in the other columns. The Guess column contains the defined values, such as $A=2$, plus the initial (guess) values used to start the iterative calculation process. By default all initial/guess values are set as 1, but this can (and sometimes must) be changed. EES applies the Newton-Raphson method seen in section 1.1.2, starting with the initial values. Note that the equations do not need to be written in the form $F(x_1, x_2, \dots, x_n) = 0$; any other form is adapted by EES; the order by which the equations appear is also totally flexible.

Lower and upper bounds for each variable may also be changed; by default all possible values (-infinity to +infinity) are considered. Units may also be assigned to each variable, and this information will be added to the results and outputs related to those variables (not mandatory).

Variable	Guess	Lower	Upper	Display	Units	Alt Units	Key	Comment
A	2	-infinity	infinity	A 3 N				
X1	1	-infinity	infinity	A 3 N				
X2	1	-infinity	infinity	A 3 N				
X3	1	-infinity	infinity	A 3 N				

Figure 2.3 – List of variables for the example of Figure 2.2.

After choosing in the menu Calculate → Solve, the iterative process is started, with the solution of the linear set of equations (1.12) in each iteration, as imposed by the Newton-Raphson method. The derivatives in (1.12) are calculated by numerical differentiation. The process continues until the established errors are met, or until no solution is found. A maximum error for all variables, and a maximum number of iterations are defined by default, but those values may be changed (Options → Stop Criteria). After conclusion of the calculations the Solution Window will pop up – Figure 2.4 – showing the results for all variables (and their units if previously defined).

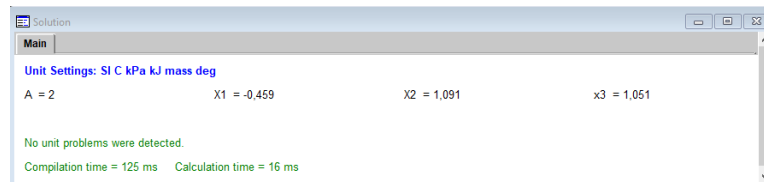


Figure 2.4 – Solution window for the example of Figure 2.2.

If the iterative calculation process is not successful, the initial/guess values may be changed and the calculation repeated. This depends on the equations and guess values, and will be discussed in the application examples of chapters 3 and 4.

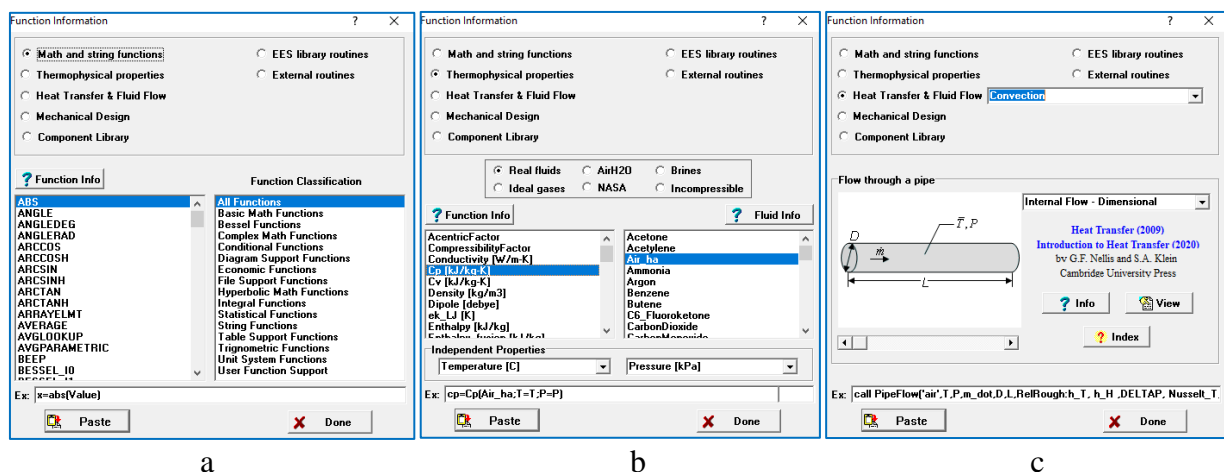
2.1.2 Physical properties and library data

EES includes a database for solid and fluid physical properties. They can be used in equations, where they are defined by internal functions. By choosing in the menu (Options → Function Info) a window appears where different options may be chosen. The first option is related to Math functions, such as absolute value, exponential function or trigonometric functions – see Figure 2.5(a). The second option is “Thermophysical properties” – Figure 2.5(b). There, different solids or fluids may be chosen; the database includes a wide range of materials and fluids found in engineering applications.

After choosing the fluid and the property in the Function Info window, information is given on the function name and arguments – Figure 2.5(b). As examples, for the specific heat of air and specific mass of water, these properties may be written in the Equations Window as:

$$c_p = Cp(\text{Air}; T = 20) * 1000 \quad (2.1)$$

$$\rho = \text{Density}(\text{Water}; T = T1; p = 100) \quad (2.2)$$



a

b

c

Figure 2.5 – Function information: (a) math functions; (b) properties; (c) heat and flow calculations.

In equation (2.1) the underscore sign (between c and p) that may be used in a variable name, allows p to become a subscript in future windows and variable outputs (appearing as c_p). If the fluid – *Air* – is considered as an ideal gas, only the temperature needs to be defined (it may be a number or a variable in another equation); if a real gas is considered – *Air_ha* – then the pressure also needs to be defined. The multiplication by 1000 is due to the default definition of units in EES (using the SI system). Default energy quantities come in kJ, and to keep consistency with the main unit (J) the multiplication is needed. Note that units and multiples may be changed in the menu Options → Unit System.

In equation (2.2) the name *rho* was used for the specific density of water. Writing the variable name with a Greek letter name, lately assigns the symbol of the Greek letter to the variable (ρ in this case), which will show up in results, tables or graphs. Here the temperature is defined as another variable (T1) and the pressure as 100 kPa (as the default pressure unit/multiple is kPa).

EES also contains library routines and internal functions that help in many problems. Heat Transfer & Fluid Flow calculation tools are particularly useful. As Figure 5.2(c) shows, they may be seen as the 3rd option in the Function Info window – in that figure the subject of convection was chosen, and a function to calculate the convection coefficient and pressure drop in a pipe flow is shown (“call PipeFlow”). When this function is pasted to the Equations Window, it will automatically calculate the required outputs, as a function of the inputs (in the case of PipeFlow: fluid name, geometrical dimensions, temperature, pressure, flow rate, tube roughness). Correlations for convective heat transfer found in the literature, [4, 5], are available, and the flow regime is also taken into account in the calculations. Many other functions, such as those for heat exchanger calculations, thermal radiation and mechanical design, are also available.

2.2 Expressing varying conditions – functions and procedures

In several numerical models the set of equations to solve varies according to different problem conditions or restrictions. For instance, considering a heating source, the energy delivered may be varied according to a given control algorithm, depending on a variable temperature which is calculated in the model (not fixed). EES allows expressing those variations and relationships through the use of FUNCTIONS and PROCEDURES. They allow combining conventional programming with the set of equations in the model.

They are defined in the Equations Window, but have to be declared/written before all other equations. If more than one FUNCTION or PROCEDURE exist, they should be placed in any order, but before the main equations. They may serve to repeat a calculation that occurs several times in the equations, or to express a logical condition. In this last case, the use of IF-THEN-ELSE instructions is frequent.

The difference between a FUNCTION and a PROCEDURE is that while a FUNCTION provides only one output to the equations, a PROCEDURE provides more than one output. Both may have receive one or more inputs from the main equations. Both must be given a name, but are called in a different manner by the main equations.

A FUNCTION is defined by its name, followed by the input variable(s) between brackets (if more than one, they must be separated by commas). And it is ended by the END instruction. A FUNCTION is called in a main equation by just writing its name and again the input variable(s) between brackets. A PROCEDURE is also defined by a name, followed by brackets that include

the input variable(s) and the output variables (variables must be separated by commas, and to separate inputs and outputs a colon must be used). It is also finished with the END instruction. The PROCEDURE is called in the main equations by a special instruction: CALL, followed by its name, followed by the inputs and outputs between brackets. This is similar to what Figure 2.5(c) shows for the PipeFlow calculation tool.

Several examples of FUNCTIONS and PROCEDURES will be shown in chapters 3 and 4.

2.3 Tables

There are 3 different types of tables in EES: *Lookup* tables, *Parametric* tables and *Array* Tables.

A *Lookup* table may be created or modified through the menu. To create one, choose Tables → New Lookup Table – see Figure 2.6. The *Lookup* table is used to define inputs to the equations that are in tabular form. Many tables may be created, and, additionally, using interpolation functions it is possible to obtain intermediate values (from the tabulated ones). Any number of rows and columns may be created in this table, or modified after (Insert/Delete Lookup Rows or Cols). Any value in a Lookup table may be read in the equations by using instructions like LOOKUP, LOOKUPROW, LOOKUPCOL. Applied examples will be shown in chapters 3 and 4.

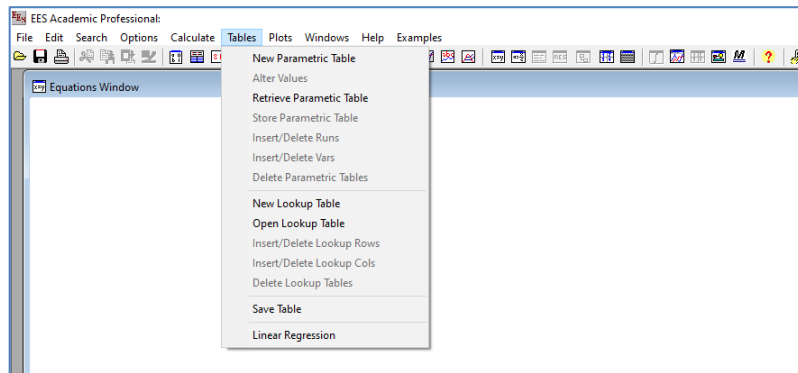


Figure 2.6 – Tables menu.

A *Parametric* Table may also be created or modified through the menu – see Figure 2.6. As the name suggests, it is appropriate to perform parametric analyses, allowing to vary one or more input variables and storing the results. Therefore, this table includes both inputs and output calculations. The inputs and outputs for each set of inputs is located in one row, and many rows may be created/added. Each row is identified with *Run* followed by the row number – see Figure 2.7. Several different tables may be created, with different names (“Table #” by default).

	\dot{Q} [W]	\dot{m} [kg/s]	T_1 [°C]	T_2 [°C]
Run 1	1500	0.05826	47.03	53.18
Run 2	1750	0.06144	51.71	58.51
Run 3	2000	0.06434	56.4	63.83
Run 4	2250	0.067	61.11	69.13
Run 5	2500	0.06948	65.83	74.42
Run 6	3000	0.07396	75.3	84.96
Run 7	3250	0.07601	80.04	90.22
Run 8	3500	0.07795	84.8	95.47
Run 9	3750	0.0798	89.56	100.7
Run 10	4000	0.1042	91.11	109.6

Figure 2.7 – Example of a Parametric Table (“Table 1”).

When creating the Parametric Table, the inputs and outputs are chosen from the available variables found in the Equations Window. When many variables occur, only the most significant outputs are usually included. By default, EES distinguishes with a different colour the inputs (in black) and the outputs (in blue) – see Figure 2.7. The colours may be changed in the menu (Options → Preferences → Display).

To fill a Parametric Table, EES is run several times. To achieve this, one must choose in the menu Calculate → Solve Table. Then the rows are calculated and filled in sequence. One run may use values from a previous run, using instructions such as TABLEVALUE. To use it, the full instruction, written in the Equations Window, should be TABLEVALUE('TableName', Row, Column), or TABLEVALUE('TableName', Row, 'VariableName').

Note that, according to the user, a different option related to the decimal point system used in the computer might occur. In the UK/USA system the dot is used as a decimal point, while in most European countries a decimal comma is used. The decimal comma is used throughout this book, and this may be noticed in Figure 2.7. Also, some instructions and functions use the semicolon (;) instead of the comma to separate variables.

An Array Table is automatically created by EES when array variables are defined in the Equations Window, and calculations are made. This table stores all the results related to the array variables. Figure 2.8 shows examples for one and two array variables. Note that the first index is displayed in rows and the second in columns. Therefore, when associating the indices with horizontal and vertical directions, it is preferable to use the first index as the one corresponding to the vertical direction.

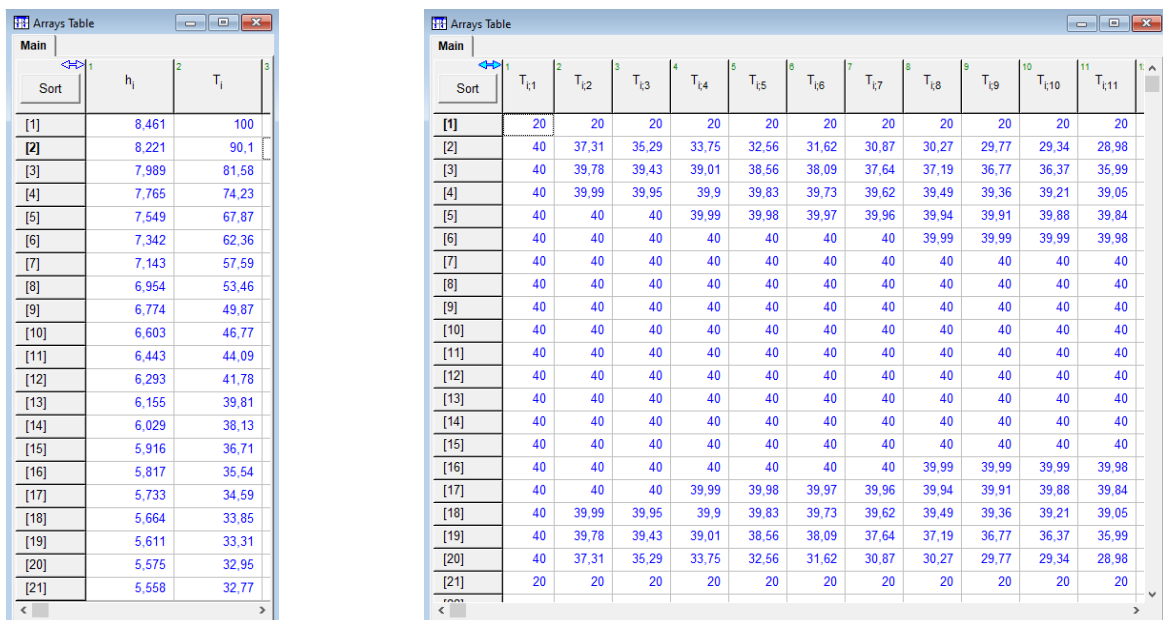


Figure 2.8 – Array Tables: (a) 1 index array variables - $h[i]$, $T[i]$; (b) 2 indices array variable - $T[i, j]$.

Although an array variable may have more than two indices, a table is only available for 1 or 2 indices, as shown in Figure 2.8.

Outputs from the Arrays Table may also be graphically represented, as explained in the next section.

2.4 Graphical outputs

With EES, multiple graphs may be created from any of the tables seen in the previous section. The menu option to create a graph is: Plots → New Plot Window. Different types of graphs may be generated, with one (X-Y Plot) or two (X-Y-Z) independent variables – see Figure 2.9. Thermodynamic property graphs, such as a pressure-enthalpy graph, may also be created (Property Plot).

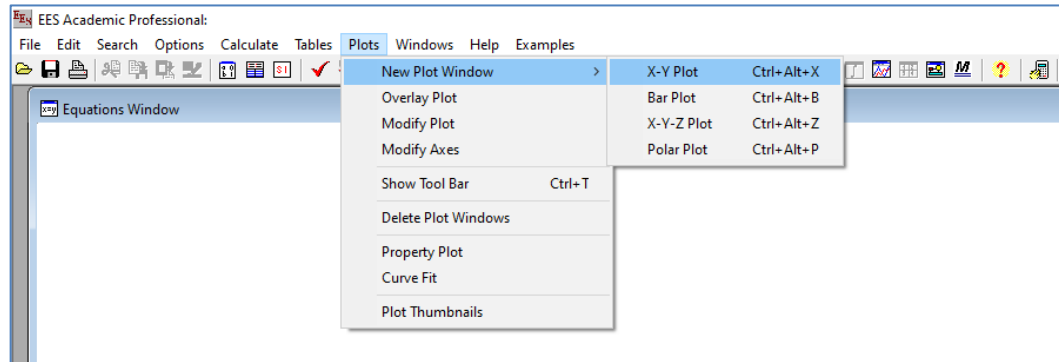


Figure 2.9 – Choice of plot type in the EES menu.

Existing graphs may be modified – see Figure 2.9. All points corresponding to the existing table cells can be represented, as well as connecting curves between the represented points. It is also possible to perform curve fitting, with different function types, using the least-squares method.

Linear and log scales may be used, and comments and drawings may be added later to the graphs. All scales and sizes are customizable. The next chapters will present many examples of 1D and 2D graphs, used to represent the results.

3 Global modelling examples

This chapter presents several examples of numerical models applied to thermal systems with a global approach. The models are first discussed, and then EES is used as a tool to obtain the solutions and perform parametric/sensitivity analyses. The EES equations/codes are also presented. Most of the examples are related to dynamic situations, with temperatures and other properties varying along time.

As remarked in chapter 2, in the UK/USA system the dot is used as a decimal point, while in most European countries a decimal comma is used. In this book, the decimal comma is used, and this affects the use/appearance of some instructions and functions, when compared with the EES software manual: as a comma replaces the dot, the sign for semicolon (;) is used to replace the comma.

3.1 Air cooling system with thermostatically controlled valve

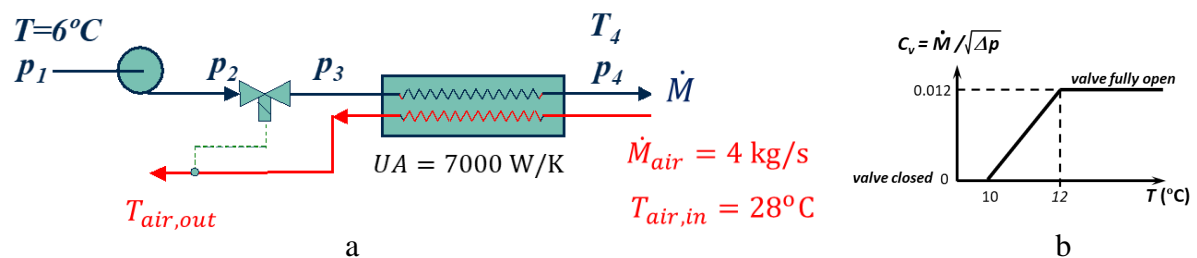


Figure 3.1.1 – Air cooling system: (a) air and cold water circuits; (b) valve control function.

Figure 3.1.1 represents an air cooling system, in which a stream of air is cooled through heat exchange to a stream of cold water. The water is moved by a pump with the following characteristic:

$$\Delta p_{pump} = p_2 - p_1 = 120000 - 15400 \dot{M}^2 \quad (3.1.1)$$

The pressure loss in the heat exchanger is given by:

$$\Delta p_{HX} = p_3 - p_4 = 9260 \dot{M}^2 \quad (3.1.2)$$

The valve is controlled through measurement of the air outlet temperature, with its pressure loss coefficient given in the above graph (Figure 3.1.1(b)).

The overall heat transfer coefficient in the air cooler (heat exchanger) may be considered independent of \dot{M} (approximation), with $UA = 7000 \text{ W/}^\circ\text{C}$.

To calculate the heat transferred in the heat exchanger, the following equation can be used:

$$\dot{Q} = UA \Delta T_{ln} = UA \frac{(T_{air,in} - T_4) - (T_{air,out} - 6)}{\ln\left(\frac{T_{air,in} - T_4}{T_{air,out} - 6}\right)} \quad (3.1.3)$$

Develop a numerical and a computational model to calculate \dot{M} , T_4 and $T_{air,out}$. Analyse the effect of varying $T_{air,in}$.

This air cooling system operates under steady-state when inlet air and inlet water temperatures are constant, as well as flow rates. However, the cooling water flow rate is controlled by the air outlet temperature, so that when the air inlet temperature becomes lower it will be possible to decrease the water flow rate, and therefore the cooling capacity. The water valve is fully open for outlet air temperatures above 12°C, and starts closing below that.

Under the conditions in Figure 3.1.1(a), with $T_{air,in} = 28^\circ\text{C}$, we may find \dot{M} , T_4 and $T_{air,out}$ using a set of algebraic equations. One expresses the pressure balance in the water stream:

$$\Delta p_{pump} = \Delta p_{HX} + \Delta p_{valve} \quad (3.1.4)$$

neglecting pressure losses in the connections, which gives

$$120000 - 15400 \dot{M}^2 = 9260 \dot{M}^2 + \dot{M}^2 / C_v^2 \quad (3.1.5)$$

and where

$$C_v = f(T_{air,out}) \quad (3.1.6)$$

according to the control function in Figure 3.1.1(b).

Another equation results from assuming there are no heat losses in the heat exchanger, meaning that the heat rate received by the water is equal to the heat rate lost by the air:

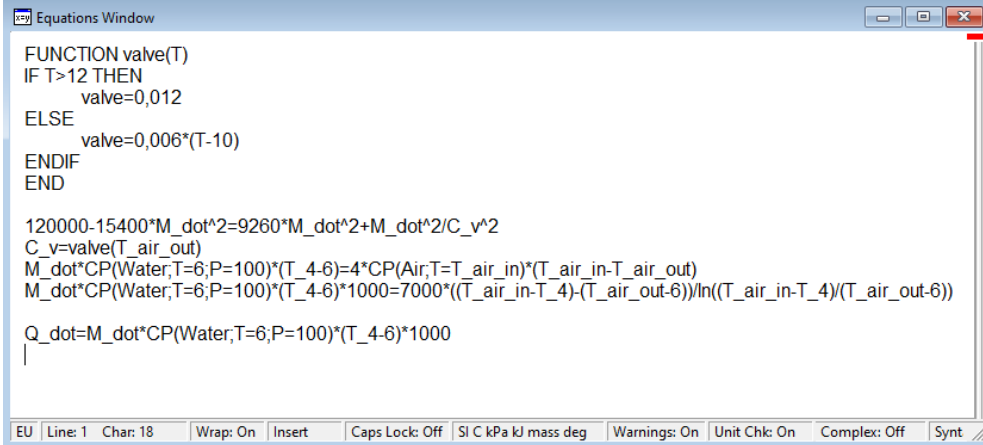
$$\dot{M} c_{p,w}(T_4 - 6) = \dot{M}_{air} c_{p,a}(28 - T_{air,out}) \quad (3.1.7)$$

and a final one from the *HX* performance:

$$\dot{M} c_{p,w}(T_4 - 6) = 7000 \frac{(28 - T_4) - (T_{air,out} - 6)}{\ln\left(\frac{28 - T_4}{T_{air,out} - 6}\right)} \quad (3.1.8)$$

The set of equations (3.1.5), (3.1.7) and (3.1.8), with the valve constant defined in (3.1.6), allows the calculation of the 3 unknown variables (\dot{M} , T_4 and $T_{air,out}$). It is a non-linear set which we will solve with EES. We may write the 3 equations, with the valve constant defined in a Function, placed in the Equations Window before the equations. To assess the specific heat of both fluids we will use EES property database; as those properties do not significantly change with temperature, we will fix them at the inlet temperatures; note that we could use c_p values calculated for the average inlet/outlet temperatures, but this would add another 2 variables to the equation model, as the outlet temperatures are not known.

Figure 3.1.2 shows the Equations Window to solve the model.



```

FUNCTION valve(T)
IF T>12 THEN
    valve=0,012
ELSE
    valve=0,006*(T-10)
ENDIF
END

120000-15400*M_dot^2=9260*M_dot^2+M_dot^2/C_v^2
C_v=valve(T_air_out)
M_dot*CP(Water,T=6,P=100)*(T_4-6)=4*CP(Air,T=T_air_in)*(T_air_in-T_air_out)
M_dot*CP(Water,T=6,P=100)*(T_4-6)*1000=7000*((T_air_in-T_4)-(T_air_out-6))/ln((T_air_in-T_4)/(T_air_out-6))

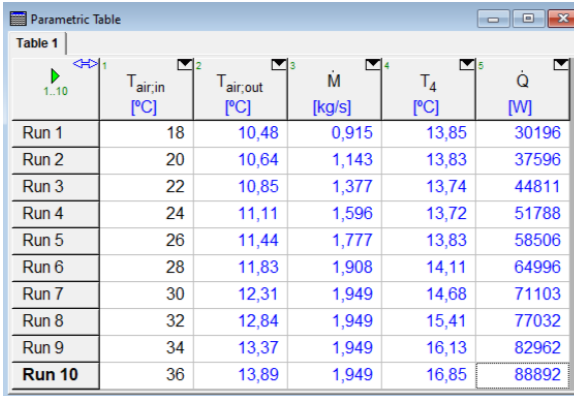
Q_dot=M_dot*CP(Water,T=6,P=100)*(T_4-6)*1000

```

Figure 3.1.2 – Equations Window for the air cooling system model.

Note that in the equivalent of equation (3.1.8) the specific heat of water was multiplied by 1000, as the c_p default units are kJ/kgK. Also note that, instead of fixing the inlet air temperature at 28°C, a variable was used ($T_{air,in}$) to investigate its effect on the results. Its value was then defined in a Parametric Table, and the solving menu choice (Calculate → Solve Table) led to the results in Figure 3.1.3.

In order to obtain convergence it was necessary to impose some initial/guess values for some variables, instead of using the default values of 1. The initial C_v was taken as 0.012, and $T_{air,out}$ and T_4 were changed to more realistic values ($T_{air,out} < T_{air,in}$ and $T_4 > 6$). Those guess values were only used in the first Table Run, as in later runs the solution from the previous run is used as initial guess.



Run	$T_{air,in}$ [°C]	$T_{air,out}$ [°C]	\dot{M} [kg/s]	T_4 [°C]	\dot{Q} [W]
Run 1	18	10,48	0,915	13,85	30196
Run 2	20	10,64	1,143	13,83	37596
Run 3	22	10,85	1,377	13,74	44811
Run 4	24	11,11	1,596	13,72	51788
Run 5	26	11,44	1,777	13,83	58506
Run 6	28	11,83	1,908	14,11	64996
Run 7	30	12,31	1,949	14,68	71103
Run 8	32	12,84	1,949	15,41	77032
Run 9	34	13,37	1,949	16,13	82962
Run 10	36	13,89	1,949	16,85	88892

Figure 3.1.3 – Parametric Table to assess the effect of changing $T_{air,in}$.

Note from the results in Figure 3.1.3 that when $T_{air,in} = 28^\circ\text{C}$ the valve is not totally open, as the outlet air temperature is below 12°C .

Figure 3.1.4 shows in a Plot Window the air outlet temperature and water flow rate when the air inlet temperature is changed, from 18 to 36°C . When the air inlet temperature is higher than 28°C the water flow rate is maximum; at lower values the flow rate decreases and cooling is not so intense as with the maximum flow rate (green line shown in the graph). Figure 3.1.5 represents the heat rate for the different air inlet temperatures considered, again comparing with the use of maximum flow rate.

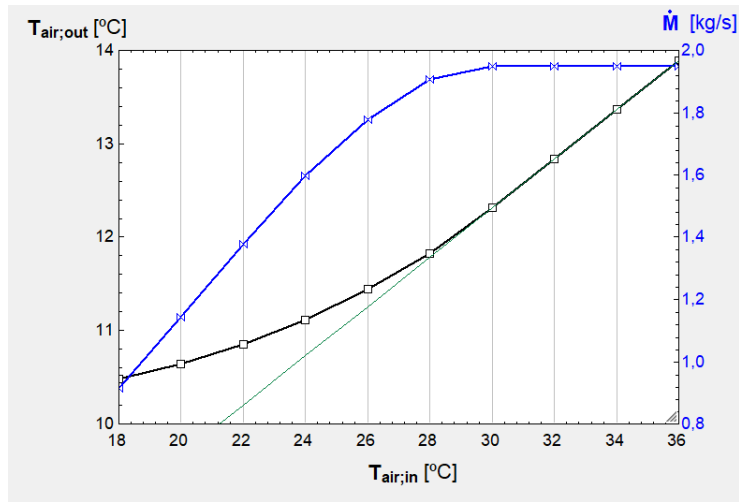


Figure 3.1.4 – Outlet air temperature and water flow rate when $T_{air,in}$ changes. Comparison with maximum flow rate for all air temperatures (green line).

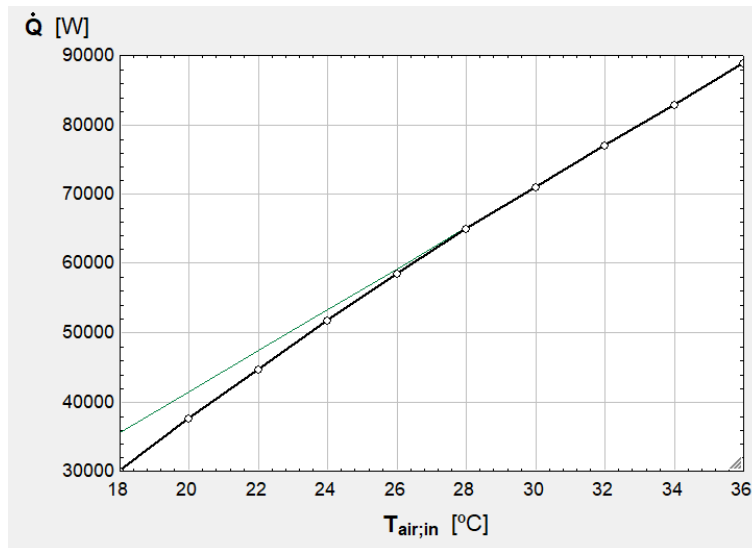


Figure 3.1.5 – Cooling rate when $T_{air,in}$ changes. Comparison with maximum flow rate for all air temperatures (green line).

3.2 Thermal bottle heat transfer

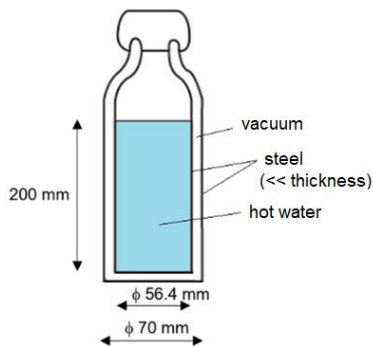


Figure 3.2.1 – Thermal bottle.

A thermal bottle is made with a stainless steel double wall, with vacuum inside the steel sheets (each with negligible thickness). When the bottle is closed it contains 0.5 litres of hot water at 90°C. The outside (calm) air and surfaces are at 20°C.

Heat exchanges at the top and bottom of the bottle may be neglected. All bottle surfaces are gray and diffuse, with an emissivity of 0.1.

Admitting that the water temperature is uniform, obtain its time evolution using EES. Analyse the effect of different time steps.

Evaluate also the thermal performance of the bottle when it is filled with cold water at 5°C, with all outside temperatures at 20°C.

The model will consider that the kettle walls have negligible thermal inertia, due to the small thickness and good conductivity of the walls. Therefore, it is assumed that the transfer of heat from the water (at T_w) to the outside (T_{ext}) is done in quasi-steady mode. Figure 3.2.2 represents the different heat transfer modes: internal free convection between the water and the inner wall surface (at T_{si}), thermal radiation through the vacuum between the two metal sheets, and convection and radiation from the external wall surface (at T_{se}) to the outside (air and outside surfaces at T_{ext}). Free convection is also assumed in the outside air.

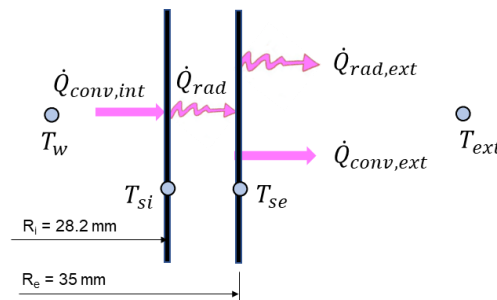


Figure 3.2.2 – Representation of heat transfer processes in the thermal bottle.

The variation of water energy is related to internal convection:

$$Mc_p \frac{dT_w}{dt} = -h_{int} A_{si} (T_w - T_{si}) \quad (3.2.1)$$

where h_{int} is a function of $(T_w - T_{si})$. With negligible wall inertia, internal convection equals the radiation balance between internal and external walls:

$$h_{int} A_{si} (T_w - T_{si}) = \frac{\sigma(T_{si}^4 - T_{se}^4)}{\frac{1-\epsilon_{si}}{A_{si}\epsilon_{si}} + \frac{1}{A_{si}1} + \frac{1-\epsilon_{se}}{A_{se}\epsilon_{se}}} \quad (3.2.2)$$

as the view factor between the two walls can be taken as 1. In a similar manner, the external wall balance will be:

$$\frac{\sigma(T_{si}^4 - T_{se}^4)}{\frac{1-\epsilon_{si}}{A_{si}\epsilon_{si}} + \frac{1}{A_{si}1} + \frac{1-\epsilon_{se}}{A_{se}\epsilon_{se}}} = \epsilon_{se} \sigma (T_{se}^4 - T_{ext}^4) A_{se} + h_{ext} A_{se} (T_{se} - T_{ext}) \quad (3.2.3)$$

The right-hand side of (3.2.3) includes the radiation exchange between the bottle external wall and external surfaces (with a much larger area), and free convection in the external air, where h_{ext} is a function of $(T_{se} - T_{ext})$.

Equations (3.2.1) to (3.2.3) allow the calculation of the 3 unknown temperatures (T_w, T_{si}, T_{se}). Equations (3.2.2) and (3.2.3) are non-linear algebraic equations, and (3.2.1) is a first order differential equation. However, due to the dependence of h_{int} on $(T_w - T_{si})$, and to the non-linearity of the radiation term, there is no exact solution to (3.2.1).

Therefore, a numerical solution is needed. Using the (Euler) implicit method seen in (1.2.2) the resulting discretised equation is:

$$T_w^{t+\Delta t} = T_w^t + \Delta t \cdot \left. \frac{dT_w}{dt} \right|_{t+\Delta t} \quad (3.2.4)$$

which is equivalent to

$$\frac{T_w^{t+\Delta t} - T_w^t}{\Delta t} = - \frac{h_{int}^{t+\Delta t} A_{si}}{M c_p} (T_w^{t+\Delta t} - T_{si}^{t+\Delta t}) \quad (3.2.5)$$

allowing the calculation of the water temperature after step Δt , depending on T_w^t (already known) and $T_{si}^{t+\Delta t} - h_{int}^{t+\Delta t}$ is a function of $(T_w^{t+\Delta t} - T_{si}^{t+\Delta t})$.

Equations (3.2.2) and (3.2.3) are valid for any $t + \Delta t$. Together with (3.2.5), and starting with an initial value of the water temperature, it is possible to obtain the time evolution of the 3 temperatures. The free convection relationships between h_{int} and h_{ext} and the temperatures will be defined in EES, using its heat transfer correlation database, [3].

The EES calculation procedure will be developed with a Parametric Table (Table 1) where the different time values are defined, step after step, or Run after Run. Figure 3.2.3 presents the Equations Window, with the definition of equations (3.2.5), (3.2.2) and (3.2.3), plus geometrical/problem inputs and the free convection functions. The time step was imposed at 60 s (1 minute), and its effect will be evaluated later.

For easier equation readability, Figure 3.2.4 presents the formatted equations.

```

D_i=0,0564
D_e=0,07
H=0,2
M=0,5
c_p=cp(Water,T=90,P=100)*1000 "in J/kgK"
epsilon=0,1 "emissivity of surfaces"
T_ext=20
DELTA=60 "in seconds"
line=time/DELTA+1 "line counter for Parametric Table"
hour=time/3600 "time in hours"
M*c_p*(T_w-T_w_old)/DELTA=-h_int*pi*D_i*H*(T_w-T_si)
T_w_old=TABLEVALUE(Table 1, line-1, #T_w)
h_int*pi*D_i*H*(T_w-T_si)=sigma#*((T_si+273,15)^4-(T_se+273,15)^4)/((1-epsilon)/(epsilon*pi*D_i*H)+1/(pi*D_i*H)+(1-epsilon)/(epsilon*pi*D_e*H))
sigma#*((T_si+273,15)^4-(T_se+273,15)^4)/((1-epsilon)/(epsilon*pi*D_i*H)+1/(pi*D_i*H)+(1-epsilon)/(epsilon*pi*D_e*H))=(h_ext*(T_se-T_ext)+epsilon*sigma#*((T_se+273,15)^4-(T_ext+273,15)^4))*(pi*D_e*H)
Call fc_vertical_cylinder(Air, T_se, T_ext, 100, H, D_e : h_ext, Nusselt_e, Ra_e)
Call fc_vertical_cylinder(Water, T_w, T_si, 100, H, D_i : h_rad_int, Nusselt_i, Ra_i)
Q_dot=sigma#*((T_si+273,15)^4-(T_se+273,15)^4)/((1-epsilon)/(epsilon*pi*D_i*H)+1/(pi*D_i*H)+(1-epsilon)/(epsilon*pi*D_e*H))
h_rad_int=Q_dot/(pi*D_i*H*(T_si-T_se))
U=Q_dot/(pi*D_i*H*(T_w-T_ext)) "based on A_int"

```

Figure 3.2.3 – Equations Window for the thermal bottle heat transfer example.

In the Equations Window all T variables correspond to $T^{t+\Delta t}$ values in the above equations. The previous value T_w^t is designated as T_{w_old} , or $T_{w,old}$. The time rows in the Parametric Table are identified by a counter (“line”), starting with the first row with initial values. The function TABLEVALUE recovers the previous temperatures (T_w^t or $T_{w,old}$) by searching them in the previous row (line-1). The heat transfer rate for each time (Q_dot , or \dot{Q}) was also calculated, as well as the equivalent radiation coefficient between walls (h_{rad_int} , or $h_{rad,int}$) and the overall heat transfer coefficient (U). The free convection functions were chosen from the EES database, choosing convection in a vertical cylinder, with water on the bottle internal surface and air on the bottle external surface.

Figure 3.2.5 shows a few of the initial Parametric Table rows. The numerical simulation was extended to a period of 24 hours, or 86440 s, which corresponds to 1441 rows with $\Delta t=60$ s. The calculations (Calculate → Solve Table) are started in Run number 2, as the first Run/line is used to impose the initial temperature of 90°C.

Formatted Equations

$D_i = 0,0564$
 $D_e = 0,07$
 $H = 0,2$
 $M = 0,5$
 $c_p = \text{Cp}(\text{WATER}; T = 90; P = 100) \cdot 1000 \text{ in J/kgK}$
 $\epsilon = 0,1 \text{ emissivity of surfaces}$
 $T_{\text{ext}} = 20$
 $\Delta t = 60 \text{ in seconds}$
 $\text{line} = \frac{\text{time}}{\Delta t} + 1 \text{ line counter for Parametric Table}$
 $\text{hour} = \frac{\text{time}}{3600} \text{ time in hours}$
 $M \cdot c_p \cdot \left[\frac{T_w - T_{w,\text{old}}}{\Delta t} \right] = -h_{\text{int}} \cdot \pi \cdot D_i \cdot H \cdot (T_w - T_{\text{si}})$
 $T_{w,\text{old}} = \text{TableValue}(\text{Table 1}; \text{line} - 1; T_w)$

$$h_{\text{int}} \cdot \pi \cdot D_i \cdot H \cdot (T_w - T_{\text{si}}) = 5,670\text{E-}08 \text{ [W/m}^2\text{-K}^4] \cdot \left[\frac{(T_{\text{si}} + 273,15)^4 - (T_{\text{se}} + 273,15)^4}{\frac{1 - \epsilon}{\epsilon \cdot \pi \cdot D_i \cdot H} + \frac{1}{\pi \cdot D_i \cdot H} + \frac{1 - \epsilon}{\epsilon \cdot \pi \cdot D_e \cdot H}} \right]$$

$$5,670\text{E-}08 \text{ [W/m}^2\text{-K}^4] \cdot \left[\frac{(T_{\text{si}} + 273,15)^4 - (T_{\text{se}} + 273,15)^4}{\frac{1 - \epsilon}{\epsilon \cdot \pi \cdot D_i \cdot H} + \frac{1}{\pi \cdot D_i \cdot H} + \frac{1 - \epsilon}{\epsilon \cdot \pi \cdot D_e \cdot H}} \right] = (h_{\text{ext}} \cdot (T_{\text{se}} - T_{\text{ext}}) + \epsilon \cdot 5,670\text{E-}08 \text{ [W/m}^2\text{-K}^4] \cdot ((T_{\text{se}} + 273,15)^4 - (T_{\text{ext}} + 273,15)^4)) \cdot \pi \cdot D_e \cdot H$$

 Call $\text{fc}_{\text{verticalcylinder}}(\text{AIR}; T_{\text{se}}; T_{\text{ext}}; 100; H; D_e; h_{\text{ext}}; \text{Nusselt}_e; \text{Ra}_e)$
 Call $\text{fc}_{\text{verticalcylinder}}(\text{WATER}; T_w; T_{\text{si}}; 100; H; D_i; h_{\text{int}}; \text{Nusselt}_i; \text{Ra}_i)$

$$\dot{Q} = 5,670\text{E-}08 \text{ [W/m}^2\text{-K}^4] \cdot \left[\frac{(T_{\text{si}} + 273,15)^4 - (T_{\text{se}} + 273,15)^4}{\frac{1 - \epsilon}{\epsilon \cdot \pi \cdot D_i \cdot H} + \frac{1}{\pi \cdot D_i \cdot H} + \frac{1 - \epsilon}{\epsilon \cdot \pi \cdot D_e \cdot H}} \right]$$

$$h_{\text{rad,int}} = \frac{\dot{Q}}{\pi \cdot D_i \cdot H \cdot (T_{\text{si}} - T_{\text{se}})}$$

$$U = \frac{\dot{Q}}{\pi \cdot D_i \cdot H \cdot (T_w - T_{\text{ext}})} \text{ based on } A_{\text{int}}$$

Figure 3.2.4 – Formatted Equations window for the thermal bottle heat transfer example.

Parametric Table

Table 1

	time [s]	hour [hours]	T_w [°C]	T_{si} [°C]	T_{se} [°C]	h_{int} [W/m ² C]	h_{ext} [W/m ² C]	$h_{\text{rad,int}}$ [W/m ² C]	U [W/m ² C - in]	\dot{Q} [W]
Run 1	0		90							
Run 2	60	0,01667	89,97	89,85	26,01	255,3	3,535	0,4816	0,4394	1,089
Run 3	120	0,03333	89,94	89,82	26	255,2	3,534	0,4815	0,4393	1,089
Run 4	180	0,05	89,91	89,79	26	255,1	3,534	0,4814	0,4393	1,088
Run 5	240	0,06667	89,88	89,75	26	255,1	3,533	0,4813	0,4392	1,088
Run 6	300	0,08333	89,84	89,72	25,99	255	3,533	0,4813	0,4391	1,087
Run 7	360	0,1	89,81	89,69	25,99	255	3,533	0,4812	0,439	1,086
Run 8	420	0,1167	89,78	89,66	25,99	254,9	3,532	0,4811	0,439	1,086
Run 9	480	0,1333	89,75	89,63	25,99	254,8	3,532	0,481	0,4389	1,085
Run 10	540	0,15	89,72	89,6	25,98	254,8	3,531	0,4809	0,4388	1,084
Run 11	600	0,1667	89,69	89,57	25,98	254,7	3,531	0,4809	0,4388	1,084
Run 12	660	0,1833	89,66	89,54	25,98	254,6	3,53	0,4808	0,4387	1,083
Run 13	720	0,2	89,63	89,51	25,97	254,6	3,53	0,4807	0,4386	1,082
Run 14	780	0,2167	89,6	89,48	25,97	254,5	3,529	0,4806	0,4386	1,082
Run 15	840	0,2333	89,56	89,44	25,97	254,4	3,529	0,4806	0,4385	1,081
Run 16	900	0,25	89,53	89,41	25,97	254,4	3,529	0,4805	0,4384	1,08
Run 17	960	0,2667	89,5	89,38	25,96	254,3	3,528	0,4804	0,4384	1,08
Run 18	1020	0,2833	89,47	89,35	25,96	254,3	3,528	0,4803	0,4383	1,079
Run 19	1080	0,3	89,44	89,32	25,96	254,2	3,527	0,4802	0,4382	1,078
Run 20	1140	0,3167	89,41	89,29	25,95	254,1	3,527	0,4802	0,4382	1,078
Run 21	1200	0,3333	89,38	89,26	25,95	254,1	3,526	0,4801	0,4381	1,077

Figure 3.2.5 – Parametric Table with results for the thermal bottle heat transfer example.

To obtain the results in Figure 3.2.5 it was necessary to alter the initial guess values (by default taken as 1) only for T_w and T_{si} , to avoid a zero temperature difference in the initial iteration. Figure 3.2.6 shows those guess values.

Variable	Guess	Lower	Upper	Display	Units
Nusselt_i	1	-infinity	infinity	A 3 N	
Q_dot	1	-infinity	infinity	A 1 N	W
Ra_e	1	-infinity	infinity	A 3 N	
Ra_i	1	-infinity	infinity	A 3 N	
time	1	-infinity	infinity	A 1 N	s
T_ext	20	-infinity	infinity	A 1 N	
T_se	1	-infinity	infinity	A 1 N	°C
T_si	89	-infinity	infinity	A 1 N	°C
T_w	90	-infinity	infinity	A 1 N	°C
T_w_old	1	-infinity	infinity	A 1 N	
U	1	-infinity	infinity	A 3 N	W/m ² C - int

Figure 3.2.6 – Variable Info window with guess values for the thermal bottle heat transfer example.

Figure 3.2.7 presents several results in graphical form for the 24 hour period. Due to the good wall insulation (vacuum), the water temperature (T_w) decreases slowly: after 12 hours it is still above 70°C, and after 24 hours it is still near 60°C. The temperature of the wall inside surface (T_{si}), represented by the green squares only, is almost equal to the water bulk temperature – the difference is usually only about 0.1°C. This is a consequence of the much higher free convection coefficient of the water, compared to the air one; while h_{int} varies between 255 and 188 W/m²C, h_{ext} varies between 3.5 and 3.0 W/m²C; the internal convection resistance is much lower; even with the addition of the external radiation coefficient ($h_{rad,ext}$), the total external coefficient is still much lower than the internal one. The bottle external surface temperature is always close to the outside temperature, varying between 26.0 and 23.2 °C. The graph also shows the evolution of the overall heat transfer coefficient (U), which varies between 0.44 and 0.37 W/m²C. The radiation coefficient inside the double wall varies from 0.48 to 0.41 W/m²C.

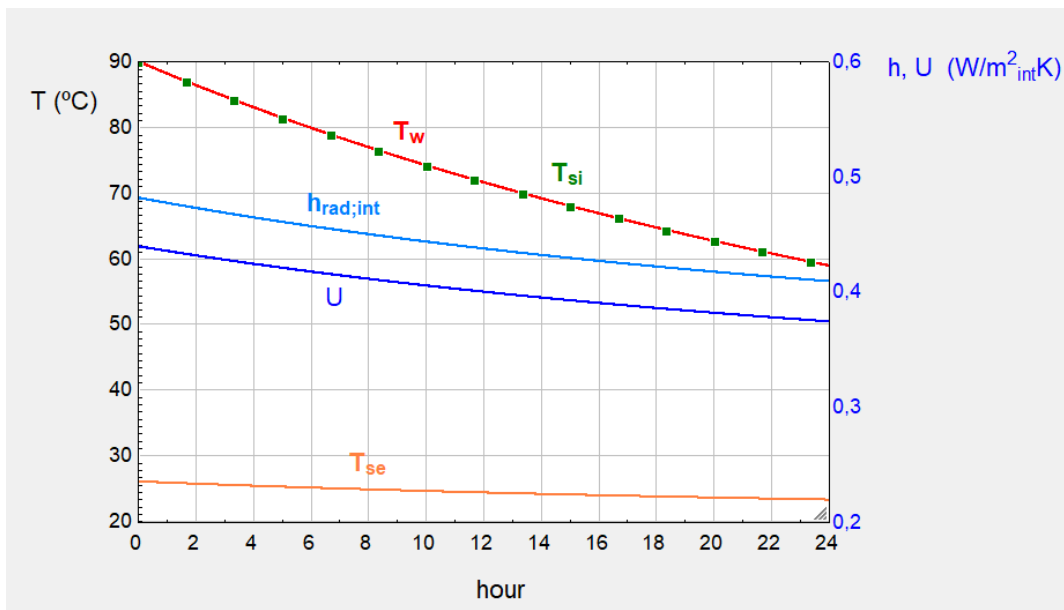


Figure 3.2.7 – Evolution of temperatures and heat transfer coefficients for the thermal bottle example.

Figure 3.2.8 shows the evolution of h_{int} along time.

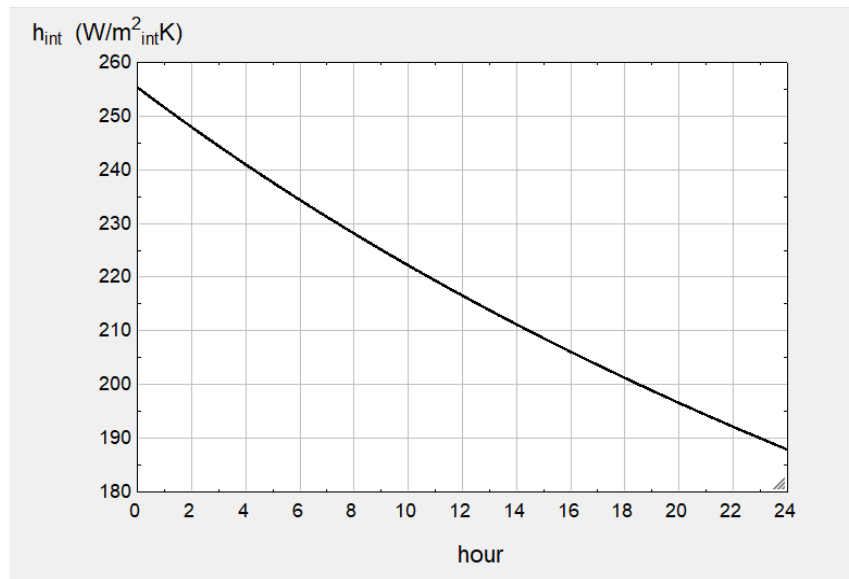


Figure 3.2.8 – Evolution of water free convection coefficient for the thermal bottle example.

To analyse the effects of different time steps, it is necessary to change DELTAt in the Equations Window and adapt the time in the Parametric Table to reflect the time step. Using 600 s instead of 60 s, the differences are very small: the water temperature after 12 hours becomes equal to 71.67°C instead of 71.63°C, while after 24 hours it becomes 59.02°C instead of 58.97°C. With a time step of 30 min the water temperature after 12 hours becomes equal to 71.75°C, while after 24 hours it becomes 59.13°C. Therefore, any of the values used is adequate in this example.

Now let us look at the cold water situation, when the water is introduced in the bottle at 5°C. To simulate this case it is only necessary to alter the first row of the Parametric Table to the initial water temperature of 5°C (instead of 90°C), and the 2 initial guess values T_w and T_{si} (for instance, to 5 and 6°C, respectively). This will converge to the solution represented in Figure 3.2.9. As can be noticed, the heat transfer coefficients have a smaller variation, as well as the temperatures; this is due to the lower driving force for the heat transfer between outside and inside. The water temperature is kept below 8°C during 12 hours and below 10°C after 24 hours.

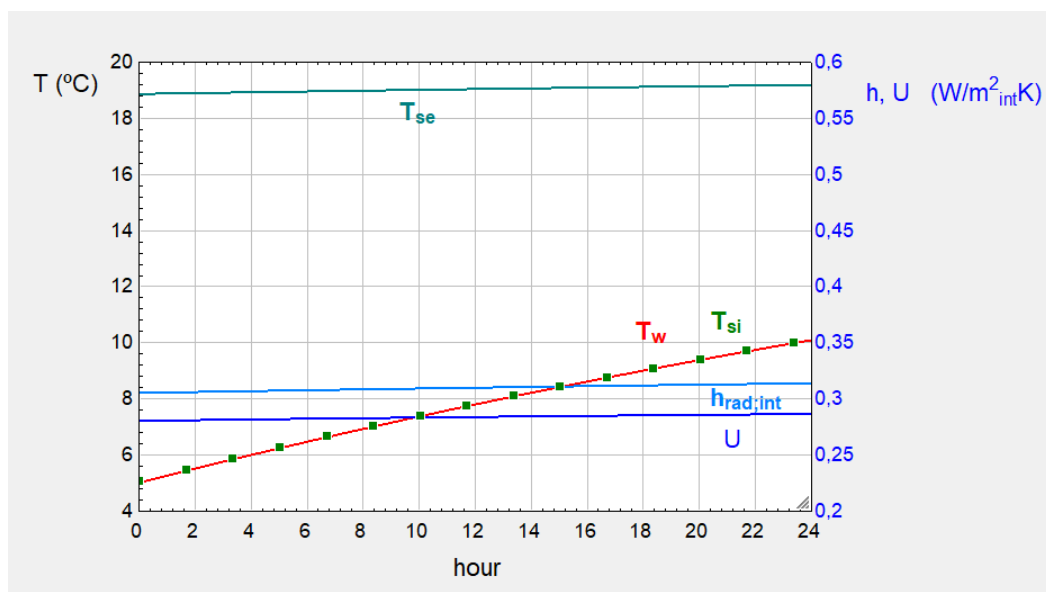


Figure 3.2.9 – Evolution of temperatures and heat transfer coefficients when the water is introduced at 5°C.

3.3 Electric kettle



Figure 3.3.1 – Kettle.

An electric kettle is filled with 1 litre of water at 20°C. The kettle is switched-on, heated by an electric resistance at the base, with a useful power of 750 W. However, the lid is kept open to the outside air, at 20°C and 40% relative humidity. Assume the kettle has a cylindrical shape, with a diameter of 10 cm, and its wall has a negligible thickness. Thermal radiation from the outer wall may be neglected, but the water surface has an emissivity of 0.9.

Admitting that the water temperature is uniform, obtain its time evolution until it reaches 100°C, taking into account heat losses at the top surface and side wall.

The model will consider the water at uniform temperature, which is reasonable due to the free convection currents, and that the kettle wall has negligible thermal inertia, due to the small thickness. There is heat transfer from the water top surface (at T_w) to the outside (at T_{ext}) by different modes: evaporation to the outside air, thermal radiation exchange with outside surfaces (also at T_{ext}), and free convection to outside air. Heat losses through the side wall include free convection between the water and the wall, and then free convection from the wall to the air (outside radiation neglected). Figure 3.3.2 represents the different heat transfer modes.

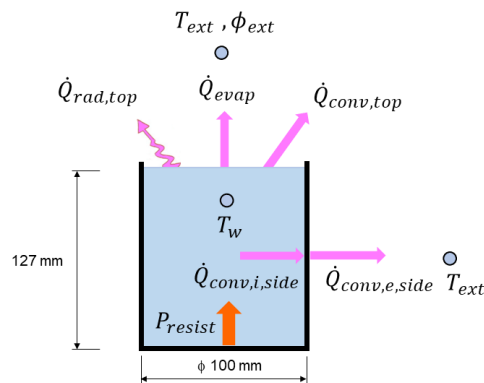


Figure 3.3.2 – Representation of heat transfer processes in the kettle.

Due to the top surface evaporation there is also a loss of liquid water. Although small, it will also be considered. The rate of evaporation (and mass loss) depends on the difference in water vapour concentration between the liquid surface (saturated air) and outside air (depending on its temperature and humidity), and on the mass transfer coefficient. Then, the differential equation to express the change in the liquid water mass is

$$\frac{dM_w}{dt} = -h_m A_{top} [\rho_{v,sat}(T_w) - \phi_{ext} \rho_{v,sat}(T_{ext})] \quad (3.3.1)$$

where ϕ_{ext} is the air relative humidity and $\rho_{v,sat}$ the vapour concentration of saturated air, which may be calculated with the EES property database as a function of temperature (T_w or T_{ext}). The mass transfer coefficient may be related to the heat convection coefficient with the Lewis relationship, [5]:

$$h_m \cong h D_{v,air}/k_{air} \quad (3.3.2)$$

where $D_{v,air}$ is the mass diffusivity for water vapour in air.

The variation of water energy (temperature) is related to the heat rates shown in Figure 3.3.2, with

$$\frac{d(M_w c_{p,w} T_w)}{dt} = P_{resist} - \dot{Q}_{conv,i,side} - \dot{Q}_{evap} - \dot{Q}_{rad,top} - \dot{Q}_{conv,top} \quad (3.3.3)$$

and using temperatures and coefficients:

$$\begin{aligned} \frac{d(M_w c_{p,w} T_w)}{dt} = & P_{resist} - h_{side,int} A_{side} (T_w - T_{wall}) - h_m A_{top} \Delta \rho_v \Delta h_{lv} - \\ & - \varepsilon_w \sigma (T_w^4 - T_{ext}^4) A_{top} - h_{top} A_{top} (T_w - T_{ext}) \end{aligned} \quad (3.3.4)$$

On the other hand, the kettle wall balance, neglecting its thickness, and external radiation, is

$$h_{side,int} A_{side} (T_w - T_{wall}) = h_{side,ext} A_{side} (T_{wall} - T_{ext}) \quad (3.3.5)$$

All h convection coefficients are a function of temperatures (free convection), and will be calculated with the EES heat transfer correlation database. Therefore, equations (3.3.1), (3.3.4) and (3.3.5), with the help of (3.3.2) allow the calculation of the water mass (M_w) and the 2 unknown temperatures (T_w, T_{wall}). Equations (3.3.1) and (3.3.4) are first order differential equations, that have to be solved simultaneously.

Therefore, a numerical solution is needed. Using the (Euler) implicit method seen in (1.2.2), with the derivatives calculated at $t + \Delta t$, the resulting discretised equations for M_w and T_w are:

$$\frac{M_w^{t+\Delta t} - M_w^t}{\Delta t} = -h_m^{t+\Delta t} A_{top} [\rho_{v,sat}(T_w^{t+\Delta t}) - \phi_{ext} \rho_{v,sat}(T_{ext})] \quad (3.3.7)$$

and

$$\begin{aligned} c_{p,w} \frac{M_w^{t+\Delta t} T_w^{t+\Delta t} - M_w^t T_w^t}{\Delta t} = & P_{resist} - h_{side,int}^{t+\Delta t} A_{side} (T_w^{t+\Delta t} - T_{wall}^{t+\Delta t}) - \\ & - h_m^{t+\Delta t} A_{top} \Delta \rho_v^{t+\Delta t} \Delta h_{lv} - \varepsilon_w \sigma (T_w^{t+\Delta t}{}^4 - T_{ext}^4) A_{top} - h_{top} A_{top} (T_w^{t+\Delta t} - T_{ext}) \end{aligned} \quad (3.3.8)$$

allowing the calculation of the water mass and temperature after step Δt , with $T_w^{t+\Delta t}$ depending on T_w^t (already known) and $T_{si}^{t+\Delta t} - h_{int}^{t+\Delta t}$ is a function of $(T_w^{t+\Delta t} - T_{si}^{t+\Delta t})$.

Equation (3.3.5) is valid for any $t + \Delta t$. Together with (3.3.7) and (3.3.8), and starting with the initial values of the water mass and temperature, it is possible to obtain the time evolution of the mass and temperature.

The EES calculation procedure will be developed with a Parametric Table (Table 1) where the different time values are defined, step after step, or Run after Run. Figure 3.3.3 presents the Equations Window, with the definition of all equations, plus geometrical/problem inputs, properties and the free convection functions. The time step was imposed at 5 s. In the Equations Window, M and all T variables correspond to $t + \Delta t$ values. The previous values are designated as M_w_old , or $M_{w,old}$, and T_w_old , or $T_{w,old}$. The time rows in the Parametric Table are identified by a counter (“line”), starting with the first row with initial values. The function TABLEVALUE recovers the previous mass and temperatures by searching them in the previous row (line-1). For easier equation readability, Figure 3.3.4 presents the formatted equations.

```

Equations Window

"KETTLE 750 W with losses on top by evaporation, convection and radiation and also side wall losses"
"initial mass 0,998 kg - 1 litre, variation of mass due to evaporation"
Diam=0,10:L=(Diam/2)^2/Diam:A_top=pi*Diam^2/4:Height=0,001/(pi*Diam^2/4)
T_ext=20: phi_air=0,4
P_resist=750
rho_w=density(Water,T=20:P=101)
c_p_w=cp(Water,T=20,P=101)*1000 "in J/kg°C"
M_init=1*rho_w/1000 "M_init=0,9982 kg"
(M_w-M_w_old)/DELTA_t=m_dot_evap*A_top "implicit method"
c_p_w*(M_w-T_w-M_w_old*T_w_old)/DELTA_t=P_resist-Q_dot_evap-Q_dot_conv-Q_dot_rad-Q_dot_side "implicit method"
DELTA_t=5
line=time/DELTA_t+1
M_w_old=tablevalue(Table 1;line-1;#M_w)
T_w_old=tablevalue(Table 1;line-1;#T_w)

Call fc_plate_horizontal2(Air; T_w; T_ext; 101; L: h_top; Nusselt_top; Ra_top) "calculate convection coeff in top surface"
D_v_air=2,6e-5
P_vs_sur=p_sat(Water,T=T_w) "in kPa"
R_v=R#/18 "in kJ/kgK"
rho_vs_sur=P_vs_sur/(R_v*(T_w+273,15))
P_vs_air=p_sat(Water,T=T_ext) "in kPa"
rho_vs_air=P_vs_air/(R_v*(T_ext+273,15))
h_m=h_top*D_v_air/k_air "mass transfer coeff"
k_air=conductivity(Air,T=T_ext)
m_dot_evap=h_m*(rho_vs_sur-phi_air*rho_vs_air) "in kg/s/m2"

Q_dot_evap=m_dot_evap*enthalpy_vaporization(Water,T=T_w)*1000*A_top "in W"
Q_dot_conv_top=h_top*(T_w-T_ext)*A_top
Q_dot_rad_top=0,9*sigma*((T_w+273,15)^4-(T_ext+273,15)^4)*A_top "outside surfaces at T_air"
Q_dot_side=h_side_int*pi*Diam*Height*(T_w-T_wall)
Q_dot_side_ext=h_side_ext*pi*Diam*Height*(T_w-T_ext) "negligible thermal radiation"

Call fc_vertical_cylinder(Water; T_w; T_wall; 101; Height; Diam : h_side_int; Nusselt_side_int; Ra_side_int)
Call fc_vertical_cylinder(Air; T_wall; T_ext; 101; Height; Diam : h_side_ext; Nusselt_side_ext; Ra_side_ext)
T_nolosses=T_ext+P_resist*time/(M_init*c_p_w)
    
```

Figure 3.3.3 – Equations Window for the kettle heat transfer example.

```

Formatted Equations

KETTLE 750 W with losses on top by evaporation, convection and radiation and also side wall losses
initial mass 0,998 kg - 1 litre, variation of mass due to evaporation

Diam = 0,1
L = (Diam^2)/4
A_top = pi * Diam^2 / 4
Height = 0,001 / (pi * Diam^2 / 4)

T_ext = 20
phi_air = 0,4

P_resist = 750

rho_w = rho(WATER; T = 20; P = 101)
c_p_w = Cp(WATER; T = 20; P = 101) * 1000 "in J/kg°C"

M_init = 1 * rho_w / 1000 "M_init=0,9982 kg"

(M_w - M_w_old) / dt = - m_dot_evap * A_top "implicit method"

c_p_w * (M_w * T_w - M_w_old * T_w_old) / dt = P_resist - Q_dot_evap - Q_dot_conv_top - Q_dot_rad_top - Q_dot_side "implicit method"

dt = 5

line = time / dt + 1

M_w_old = TableValue (Table 1; line - 1; M_w)
T_w_old = TableValue (Table 1; line - 1; T_w)

Call fc_plate_horizontal2 (AIR; T_w; T_ext; 101; L: h_top; Nusselt_top; Ra_top) "calculate convection coeff in top surface"

D_v_air = 0,000026
P_vs_sur = P_sat (WATER; T = T_w) "in kPa"
R_v = 8,314 / 18 "in kJ/kgK"

rho_vs_sur = P_vs_sur / (R_v * (T_w + 273,15))
P_vs_air = P_sat (WATER; T = T_ext) "in kPa"
rho_vs_air = P_vs_air / (R_v * (T_ext + 273,15))

h_m = h_top * D_v_air / k_air "mass transfer coeff"
k_air = k (AIR; T = T_ext)
m_dot_evap = h_m * (rho_vs_sur - phi_air * rho_vs_air) "in kg/s/m2"
Q_dot_evap = m_dot_evap * Enthalpy_vaporization (water; T = T_w) * 1000 * A_top "in W"
Q_dot_conv_top = h_top * (T_w - T_ext) * A_top
Q_dot_rad_top = 0,9 * 5,670E-08 [W/m^2.K^4] * ((T_w + 273,15)^4 - (T_ext + 273,15)^4) * A_top "outside surfaces at T_air"
Q_dot_side = h_side_int * pi * Diam * Height * (T_w - T_wall)
Q_dot_side_ext = h_side_ext * pi * Diam * Height * (T_w - T_ext) "negligible thermal radiation"

Call fc_vertical_cylinder (water; T_w; T_wall; 101; Height; Diam : h_side_int; Nusselt_side_int; Ra_side_int)
Call fc_vertical_cylinder (AIR; T_wall; T_ext; 101; Height; Diam : h_side_ext; Nusselt_side_ext; Ra_side_ext)

T_nolosses = T_ext + P_resist * time / (M_init * c_p_w)
    
```

Figure 3.3.4 – Formatted Equations window for the kettle heat transfer example.

1	2	3	4	5	6	7	8	9	10	
time [s]	$T_{\text{no losses}}$ [°C]	T_w [°C]	T_{wall} [°C]	$\dot{Q}_{\text{rad, top}}$ [W]	$\dot{Q}_{\text{conv, top}}$ [W]	\dot{Q}_{evap} [W]	\dot{Q}_{side} [W]	\dot{m}_{evap} [kg/(sm ²)]	M_w [kg]	
Run 1	0	20	20			0	0	0	0,9982	
Run 2	5	20,9	20,9	20,87	0,0364	0,01983	0,6331	0,0824	0,00003288	0,9982
Run 3	10	21,8	21,79	21,75	0,07311	0,04355	0,7542	0,1936	0,00003921	0,9982
Run 4	15	22,69	22,69	22,63	0,1101	0,06909	0,8631	0,32	0,00004491	0,9982
Run 5	20	23,59	23,59	23,5	0,1475	0,09592	0,9698	0,4576	0,0000505	0,9982
Run 6	25	24,49	24,48	24,38	0,1852	0,1238	1,078	0,6044	0,00005617	0,9982
Run 7	30	25,39	25,38	25,25	0,2232	0,1525	1,188	0,7588	0,00006198	0,9982
Run 8	35	26,29	26,27	26,13	0,2615	0,1819	1,302	0,92	0,000068	0,9982
Run 9	40	27,18	27,17	27,01	0,3002	0,2119	1,421	1,087	0,00007425	0,9982
Run 10	45	28,08	28,06	27,88	0,3392	0,2426	1,544	1,26	0,00008075	0,9982
Run 11	50	28,98	28,95	28,76	0,3785	0,2737	1,672	1,437	0,00008753	0,9982
Run 12	55	29,88	29,85	29,63	0,4182	0,3053	1,806	1,62	0,00009461	0,9982
Run 13	60	30,77	30,74	30,51	0,4582	0,3374	1,945	1,806	0,000102	0,9982
Run 14	65	31,67	31,63	31,39	0,4985	0,3699	2,09	1,997	0,0001097	0,9982
Run 15	70	32,57	32,52	32,26	0,5391	0,4027	2,242	2,191	0,0001178	0,9982
Run 16	75	33,47	33,41	33,14	0,5801	0,4359	2,4	2,389	0,0001262	0,9982
Run 17	80	34,37	34,3	34,01	0,6214	0,4695	2,565	2,59	0,000135	0,9981
Run 18	85	35,26	35,19	34,89	0,6631	0,5033	2,737	2,66	0,0001441	0,9981
Run 19	90	36,16	36,08	35,77	0,7051	0,5375	2,916	2,859	0,0001537	0,9981
Run 20	95	37,06	36,97	36,64	0,7474	0,572	3,102	3,061	0,0001637	0,9981
Run 21	100	37,96	37,86	37,52	0,7901	0,6067	3,296	3,266	0,0001741	0,9981
Run 22	105	38,86	38,75	38,39	0,8331	0,6417	3,499	3,473	0,0001849	0,9981
Run 23	110	39,75	39,64	39,27	0,8765	0,6769	3,709	3,683	0,0001962	0,9981
Run 24	115	40,65	40,52	40,14	0,9202	0,7124	3,928	3,895	0,000208	0,9981
Run 25	120	41,55	41,41	41,01	0,9642	0,7481	4,156	4,109	0,0002203	0,9981
Run 26	125	42,45	42,3	41,89	1,009	0,7841	4,393	4,326	0,000233	0,9981
Run 27	130	43,34	43,18	42,76	1,053	0,8202	4,64	4,545	0,0002463	0,9981
Run 28	135	44,24	44,07	43,63	1,098	0,8566	4,895	4,765	0,0002601	0,9981
Run 29	140	45,14	44,95	44,5	1,144	0,8931	5,161	4,988	0,0002745	0,9981
Run 30	145	46,04	45,83	45,37	1,19	0,9298	5,437	5,213	0,0002894	0,998
Run 31	150	46,94	46,72	46,24	1,236	0,9668	5,724	5,439	0,0003049	0,998
Run 32	155	47,83	47,6	47,11	1,282	1,004	6,021	5,667	0,000321	0,998
Run 33	160	48,73	48,48	47,98	1,329	1,041	6,329	5,897	0,0003378	0,998
Run 34	165	49,63	49,36	48,85	1,376	1,079	6,648	6,129	0,0003552	0,998
Run 35	170	50,53	50,24	49,72	1,424	1,116	6,98	6,362	0,0003732	0,998
Run 71	350	82,85	81,31	80,43	3,367	2,542	29,49	15,46	0,001629	0,9967
Run 72	355	83,75	82,15	81,27	3,427	2,583	30,49	15,73	0,001686	0,9966
Run 73	360	84,65	82,99	82,1	3,488	2,624	31,52	15,99	0,001745	0,9966
Run 74	365	85,54	83,83	82,93	3,549	2,665	32,57	16,26	0,001805	0,9965
Run 75	370	86,44	84,67	83,76	3,61	2,706	33,65	16,52	0,001866	0,9964
Run 76	375	87,34	85,5	84,59	3,672	2,747	34,76	16,79	0,001929	0,9963
Run 77	380	88,24	86,34	85,42	3,734	2,788	35,89	17,06	0,001994	0,9963
Run 78	385	89,14	87,17	86,24	3,796	2,829	37,05	17,32	0,00206	0,9962
Run 79	390	90,03	88	87,07	3,859	2,87	38,23	17,59	0,002128	0,9961
Run 80	395	90,93	88,83	87,89	3,922	2,911	39,45	17,86	0,002198	0,996
Run 81	400	91,83	89,66	88,71	3,985	2,952	40,69	18,12	0,002269	0,9959
Run 82	405	92,73	90,49	89,53	4,048	2,993	41,96	18,39	0,002342	0,9958
Run 83	410	93,63	91,32	90,35	4,112	3,034	43,25	18,66	0,002416	0,9957
Run 84	415	94,52	92,14	91,16	4,177	3,075	44,58	18,93	0,002493	0,9956
Run 85	420	95,42	92,96	91,98	4,241	3,117	45,93	19,2	0,002571	0,9955
Run 86	425	96,32	93,78	92,79	4,306	3,158	47,32	19,46	0,002651	0,9954
Run 87	430	97,22	94,6	93,6	4,371	3,199	48,73	19,73	0,002733	0,9953
Run 88	435	98,11	95,42	94,41	4,436	3,24	50,17	20	0,002816	0,9952
Run 89	440	99,01	96,24	95,22	4,502	3,281	51,65	20,27	0,002902	0,9951
Run 90	445	99,91	97,05	96,03	4,568	3,322	53,15	20,54	0,002989	0,995
Run 91	450	100,8	97,86	96,83	4,634	3,363	54,69	20,81	0,003078	0,9949
Run 92	455	101,7	98,67	97,64	4,701	3,404	56,25	21,07	0,003169	0,9947
Run 93	460	102,6	99,48	98,44	4,768	3,445	57,85	21,34	0,003262	0,9946
Run 94	465	103,5	100,3	99,24	4,835	3,486	59,48	21,61	0,003358	0,9945

Figure 3.3.5 – Parametric Table with results for the kettle heat transfer example.

The free convection functions were chosen from the EES database: convection in a vertical cylinder, with water on the internal surface (for $h_{side,int}$) and air on the external surface (for $h_{side,ext}$); and convection to air on the top of an horizontal plate (for h_{top}).

To calculate $\rho_{v,sat}(T)$, the saturation water pressure for T was obtained from the EES property database (P_sat function), and by calculating $\rho_{v,sat}(T) = P_{sat}/(R_v T)$. Alternatively, property functions for humid air could have been used (*AirH2O* mixed fluid). Further details about the use of EES in psychrometric calculations may be found in [6].

Figure 3.3.5 shows a few of the initial Parametric Table rows. The numerical simulation was extended until the water reached 100°C, corresponding to a period of 465 s (94 rows with $\Delta t=5$ s). The calculations (Calculate → Solve Table) are started in Run number 2, as the first Run/line is used to impose the initial temperature of 20°C. All initial/guess default values were used (taken as 1), except T_w and T_{wall} , to avoid a zero temperature difference in the initial iteration. Figure 3.3.6 shows the list of variables and the guess values.

Variable	Guess	Lower	Upper	Display	Units
A_top	0,007854	-infinity	infinity	A 3 N	
c_p_w	4184	-infinity	infinity	A 3 N	
DELTA _t	5	-infinity	infinity	A 3 N	
Diam	0,1	-infinity	infinity	A 3 N	
D_v_air	0,000026	-infinity	infinity	A 3 N	
Height	0,1273	-infinity	infinity	A 0 N	
h_m	0,003	-infinity	infinity	A 0 N	
h_side_ext	1	-infinity	infinity	A 0 N	
h_side_int	1	-infinity	infinity	A 0 N	
h_top	1	-infinity	infinity	A 0 N	
k_air	0,02514	-infinity	infinity	A 3 N	
L	0,025	-infinity	infinity	A 3 N	
line	1	-infinity	infinity	A 3 N	
m_dot_evap	0,001	-infinity	infinity	A 3 N	kg/(sm2)
M_init	0,9982	-infinity	infinity	A 3 N	
M_w	1	-infinity	infinity	A 3 N	kg
M_w_old	1	-infinity	infinity	A 3 N	
Nusselt_side_ext	1	-infinity	infinity	A 3 N	
Nusselt_side_int	1	-infinity	infinity	A 3 N	
Nusselt_top	1	-infinity	infinity	A 3 N	
phi_air	0,4	-infinity	infinity	A 0 N	
P_resist	750	-infinity	infinity	A 0 N	
P_vs_air	2,339	-infinity	infinity	A 0 N	
P_vs_sur	1	-infinity	infinity	A 0 N	
Q_dot_conv_top	1	-infinity	infinity	A 1 N	W
Q_dot_evap	10	-infinity	infinity	A 1 N	W
Q_dot_rad_top	1	-infinity	infinity	A 1 N	W
Q_dot_side	1	-infinity	infinity	A 1 N	W
Ra_side_ext	1	-infinity	infinity	A 3 N	
Ra_side_int	1	-infinity	infinity	A 3 N	
Ra_top	1	-infinity	infinity	A 3 N	
rho_vs_air	0,01728	-infinity	infinity	A 3 N	
rho_vs_sur	1	-infinity	infinity	A 3 N	
rho_w	998,2	-infinity	infinity	A 3 N	
R_v	0,4619	-infinity	infinity	A 3 N	
time	1	-infinity	infinity	A 1 N	s
T_ext	20	-infinity	infinity	A 1 N	°C
T_nolosses	1	-infinity	infinity	A 1 N	°C
T_w	25	-infinity	infinity	A 1 N	°C
T_wall	24	-infinity	infinity	A 1 N	°C
T_w_old	1	-infinity	infinity	A 1 N	

Figure 3.3.6 – Variable Info window with guess values for the kettle heat transfer example.

Figure 3.3.7 presents the evolution of water temperature (T_w), and the comparison with the ideal case when no losses occur (mass and energy). Note from the results in Figure 3.3.5 that the mass loss due to evaporation only represents 0.4% of the initial mass, and therefore could have been neglected. Taking into account the energy losses, the water takes 15 seconds more to reach 100°C (compared to no losses).

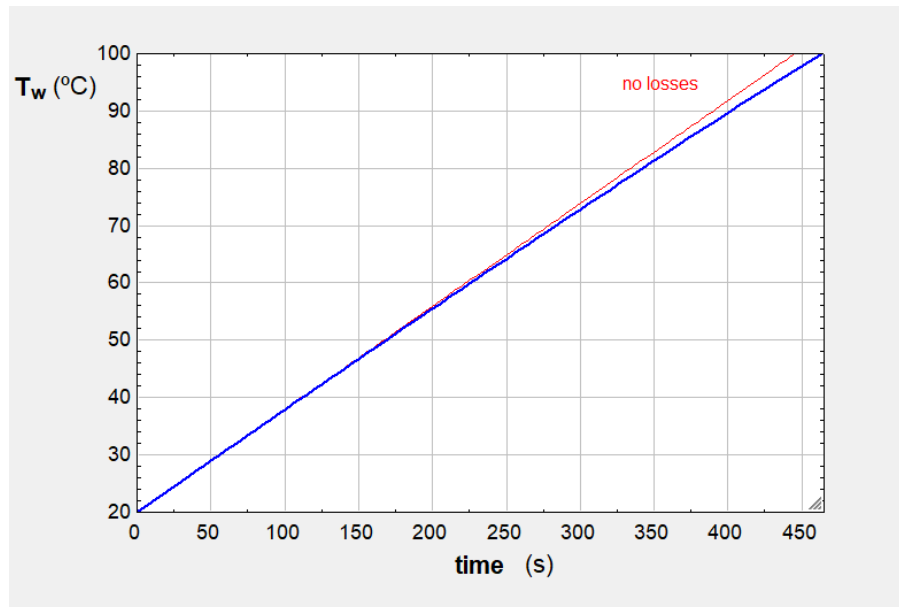


Figure 3.3.7 – Evolution of water temperature for the kettle example.

Figure 3.3.8 shows the evolution of the different heat rate losses. As can be seen, evaporative losses are the most important ones, especially when the water temperature becomes higher. Top surface losses also represent the highest share.

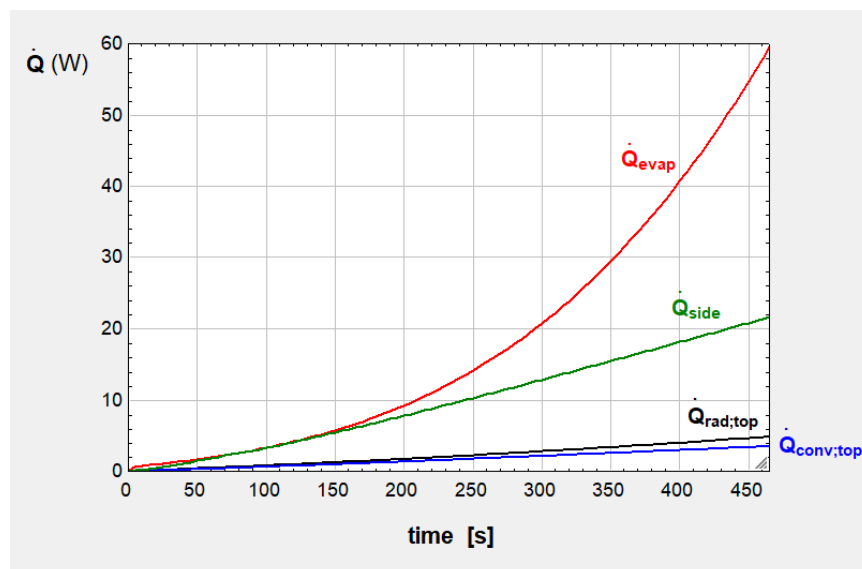


Figure 3.3.8 – Evolution of heat rate losses for the kettle example.

The temperature of the wall (T_{wall}) is always very close to the water bulk temperature. This is a consequence of the much higher free convection coefficient of the water, compared to the air one; while $h_{side,int}$ varies between 91 and 514 W/m²°C, $h_{side,ext}$ varies between 2 and 7 W/m²°C. The top coefficient varies from 3 to 6 W/m²°C.

3.4 Electric water heater

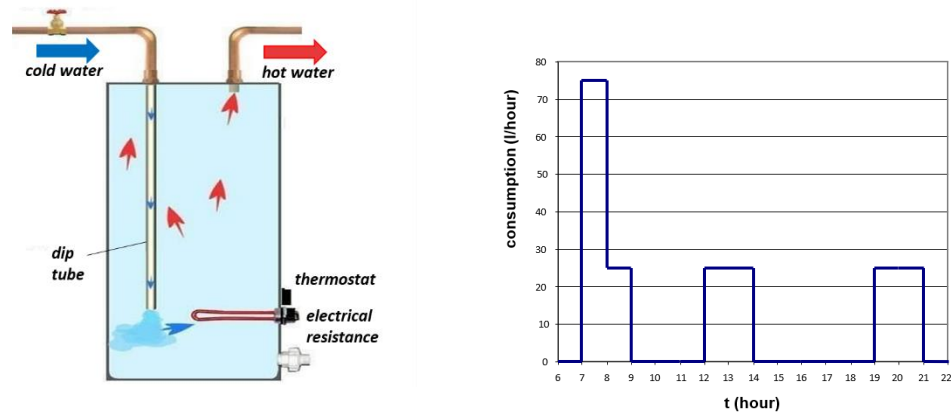


Figure 3.4.1 – Water heater and daily water consumption pattern.

A system that provides domestic hot water uses an electrically heated storage tank with a capacity of 100 litres. The total daily consumption is equal to 200 litres/day, according to the graph in Figure 3.4.1.

The tank has the following characteristics:

- electrical resistance power: 2 kW;
- “on-off” resistance control: on at 50°C, off at 55°C;
- tank heat loss coefficient: 1.5 W/°C;
- inlet cold water temperature (from the mains): 15°C;
- outside (air) temperature: 20°C.

Assuming a global tank model, simulate a system daily cycle, using EES software. Obtain the daily energy consumption and analyse the effect of varying the time step.

Compare the previous results with those for a proportional control of the resistance (proportional band between 50 and 55°C).

The model will be based on the calculation of the tank water temperature (T), considered to be uniform. The variation of water energy is related to all energy inputs (resistance, water inlet from mains) and outputs (water outlet, heat losses to the outside):

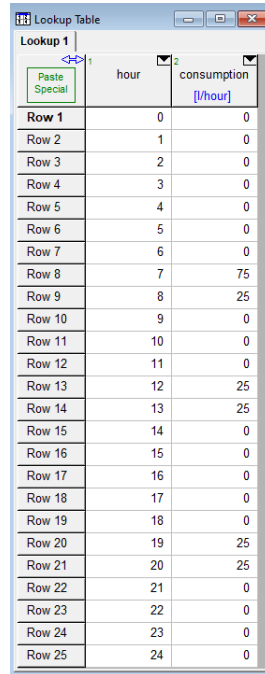
$$Mc_p \frac{dT}{dt} = P_{resist} + \dot{M}_{cons} c_p (T_{in} - T) - (UA)_{tank} (T - T_{ext}) \quad (3.4.1)$$

Since P_{resist} may assume 2 different values (0 or 2000 W), and \dot{M}_{cons} is not constant, there is no exact solution to the differential equation. Using the implicit method of numerical integration we will have:

$$\frac{T^{t+\Delta t} - T^t}{\Delta t} = \frac{1}{Mc_p} \left[P_{resist}^{t+\Delta t} + \dot{M}_{cons}^{t+\Delta t} c_p (T_{in} - T^{t+\Delta t}) - (UA)_{tank} (T^{t+\Delta t} - T_{ext}) \right] \quad (3.4.2)$$

To solve equation (3.4.2), the model needs to define $\dot{M}_{cons}^{t+\Delta t}$ – given in Figure 3.4.1 – and $P_{resist}^{t+\Delta t}$. The resistance heat input is a function of the temperature ($T^{t+\Delta t}$), but with an on-off control, it is also a function of its previous state (P_{resist}^t): if the temperature is somewhere between 50 and 55°C, the resistance will maintain its previous state – it will stay on if it was on before Δt , and

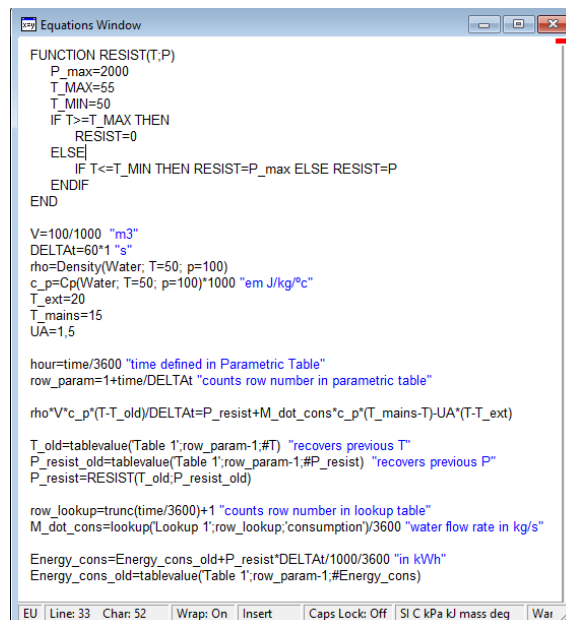
off if it was off before Δt . The behaviour of the resistance will be defined in EES with a FUNCTION. The water consumption flow rate will be defined in a Lookup Table (Lookup 1), translating the graph of Figure 3.4.1, and shown in Figure 3.4.2. The values of flow rate for a given time will be read with the Lookup function, according to the time.



	hour	consumption [l/hour]
Row 1	0	0
Row 2	1	0
Row 3	2	0
Row 4	3	0
Row 5	4	0
Row 6	5	0
Row 7	6	0
Row 8	7	75
Row 9	8	25
Row 10	9	0
Row 11	10	0
Row 12	11	0
Row 13	12	25
Row 14	13	25
Row 15	14	0
Row 16	15	0
Row 17	16	0
Row 18	17	0
Row 19	18	0
Row 20	19	25
Row 21	20	25
Row 22	21	0
Row 23	22	0
Row 24	23	0
Row 25	24	0

Figure 3.4.2 – Lookup table to define water consumption in the water heater example.

Then we may start with an initial value of the water temperature, and obtain the following values. The EES calculation procedure will be developed with a Parametric Table (Table 1) where the different time values are defined, step after step, or Run after Run. Figure 3.4.3 presents the Equations Window, with the definition of equations (3.4.2), and the control function (RESIST), plus a few of the other input values. The time step was imposed at 60 s (1 minute), and its effect will be evaluated later.



```

FUNCTION RESIST(T,P)
  P_max=2000
  T_MAX=55
  T_MIN=50
  IF T>=T_MAX THEN
    RESIST=0
  ELSE
    IF T<=T_MIN THEN RESIST=P_max ELSE RESIST=P
  ENDIF
END

V=100/1000 "m3"
DELTA=60*1 "s"
rho=Density(Water, T=50, p=100)
c_p=Cp(Water, T=50, p=100)*1000 "em J/kg°C"
T_ext=20
T_mains=15
UA=1,5

hour=time/3600 "time defined in Parametric Table"
row_param=1+time/DELTA "counts row number in parametric table"

rho*V*c_p*(T-T_old)/DELTA=P_resist+M_dot_cons*c_p*(T_mains-T)-UA*(T-T_ext)

T_old=tablevalue(Table 1;row_param-1;#T) "recovers previous T"
P_resist_old=tablevalue(Table 1;row_param-1;#P_resist) "recovers previous P"
P_resist=RESIST(T_old,P_resist_old)

row_lookup=trunc(time/3600)+1 "counts row number in lookup table"
M_dot_cons=lookup(Lookup 1;row_lookup;consumption)/3600 "water flow rate in kg/s"

Energy_cons=Energy_cons_old+P_resist*DELTA/1000/3600 "in kWh"
Energy_cons_old=tablevalue(Table 1;row_param-1;#Energy_cons)

```

Figure 3.4.3 – Equations Window for the water heater example.

In the Equations Window T corresponds to $T^{t+\Delta t}$ in equation (3.4.2). The previous value T^t is designated as T_{old} , or T_{old} . The time rows in the Parametric Table are identified by a counter (“row_param”), starting with the first row with initial values. The function TABLEVALUE recovers the previous temperatures (T^t or T_{old}) by searching them in the previous row (row_param -1).

Note that the resistance power at a given time (P_{resist}) is calculated using the previous values of power (P_{resist_old}) and temperature (T_{old}); the previous value of P is needed to define the state of the resistance between the on-off values, when the resistance keeps the previous value; but according to the implicit method chosen, P_{resist} should be calculated with T instead of T_{old} ; however, in this problem, using T would lead to numerical instability (no convergence of the solution); this is due to the on-off situation, and introduces an explicit method influence on the integration of equation (3.4.2).

The daily energy consumption is calculated by accumulating the resistance power over time, using the variable Energy_cons.

Figure 3.4.4 shows a few of the initial and final Parametric Table rows. The numerical simulation was extended to a period of 24 hours, or 86440 s, which corresponds to 1441 rows with $\Delta t=60$ s. The calculations (Calculate → Solve Table) are started in Run number 2, as the first Run/line is used to impose the initial water temperature. However, this temperature is not directly known. Therefore, we assumed that the same daily cycle is repeated every day; this way, the initial temperature will be equal to the temperature at the end of the day. As the simulation was run for 24 hours only, this involved an iterative use of the model, starting with an initial temperature value between 50 and 55°C (and $P_{resist} = 0$), and changing it to the final temperature value (24 h), until a close match between the 2 values was obtained; only 2 or 3 iterations were needed to reach the value of 53.7°C in Figure 3.4.4. Note that, to obtain the solution, it was not necessary to alter the default guess values of all variables (taken as 1), which are only used in Run 2, as the solution from the previous Run is used as guess for the next Run.

Figure 3.4.5 graphically presents the temperature results of the Parametric Table. The water temperature is kept between 50 and 55°C most of the time. There is however a morning period (between 7 and 8.7 hours), with the highest water consumption rate, when the resistance power is not enough to maintain the water above 50°C. During this period, the water temperature reaches a minimum of 43.5°C. We may note that in short periods the temperature exceeds 55°C; this is due to the delay in the model response to the resistance switch-off; the effect of the switch-off is only noticed after the end of the integration step (Δt) – the resistance is switched-on until the end of the integration step. We could reduce the step in those moments, but the error associated with energy quantities is very small. The calculated daily energy consumption is equal to 9.2 kWh.

Figure 3.4.6 presents the resistance operation and the water consumption rate during the daily cycle.

Figure 3.4.7 presents a comparison between the previous temperature evolution ($\Delta t=60$ s) and the evolution when a time step of 300 s is used. The previously noted resistance switch-off model delay is responsible for the most noticeable differences. They are however small, and the calculated daily energy consumption does not differ much: 9.17 kWh for a $\Delta t=300$ s, compared to 9.20 kWh for a $\Delta t=60$ s.

Run	time [s]	hour	T [°C]	P _{resist} [W]	\dot{M}_{cons} [kg/s]	Energy _{cons} [kWh]
Run 1	0		53,7	0		0
Run 2	60	0,01667	53,69	0	0	0
Run 3	120	0,03333	53,69	0	0	0
Run 4	180	0,05	53,68	0	0	0
Run 5	240	0,06667	53,67	0	0	0
Run 6	300	0,08333	53,66	0	0	0
Run 7	360	0,1	53,66	0	0	0
Run 8	420	0,1167	53,65	0	0	0
Run 9	480	0,1333	53,64	0	0	0
Run 10	540	0,15	53,63	0	0	0
Run 11	600	0,1667	53,63	0	0	0
Run 12	660	0,1833	53,62	0	0	0
Run 13	720	0,2	53,61	0	0	0
Run 14	780	0,2167	53,6	0	0	0
Run 15	840	0,2333	53,6	0	0	0
Run 16	900	0,25	53,59	0	0	0
Run 17	960	0,2667	53,58	0	0	0
Run 18	1020	0,2833	53,58	0	0	0
Run 19	1080	0,3	53,57	0	0	0
Run 1423	85320	23,7	53,91	0	0	9,2
Run 1424	85380	23,72	53,9	0	0	9,2
Run 1425	85440	23,73	53,89	0	0	9,2
Run 1426	85500	23,75	53,89	0	0	9,2
Run 1427	85560	23,77	53,88	0	0	9,2
Run 1428	85620	23,78	53,87	0	0	9,2
Run 1429	85680	23,8	53,86	0	0	9,2
Run 1430	85740	23,82	53,86	0	0	9,2
Run 1431	85800	23,83	53,85	0	0	9,2
Run 1432	85860	23,85	53,84	0	0	9,2
Run 1433	85920	23,87	53,83	0	0	9,2
Run 1434	85980	23,88	53,83	0	0	9,2
Run 1435	86040	23,9	53,82	0	0	9,2
Run 1436	86100	23,92	53,81	0	0	9,2
Run 1437	86160	23,93	53,8	0	0	9,2
Run 1438	86220	23,95	53,8	0	0	9,2
Run 1439	86280	23,97	53,79	0	0	9,2
Run 1440	86340	23,98	53,78	0	0	9,2
Run 1441	86400	24	53,77	0	0	9,2

Figure 3.4.4 – Parametric Table with results for the water heater example.

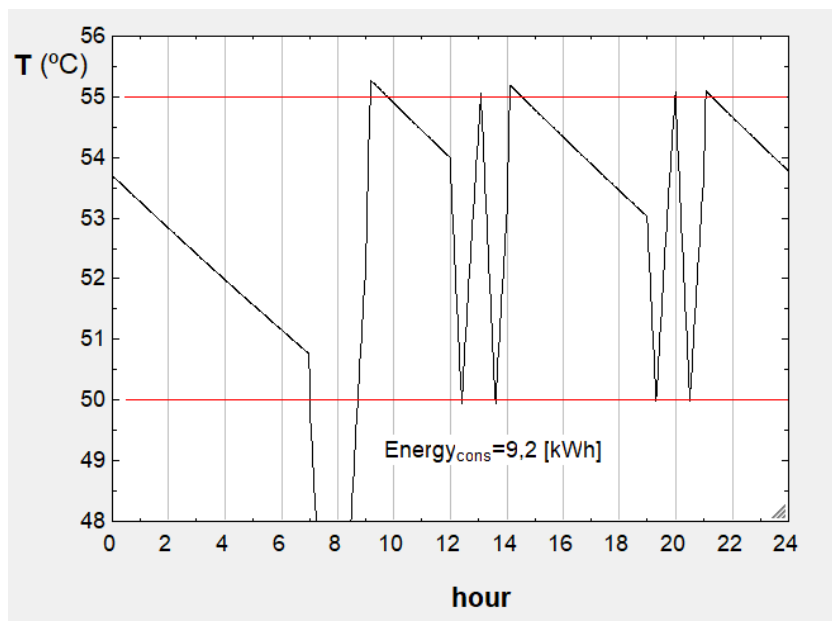


Figure 3.4.5 – Evolution of the water temperature with on-off control.

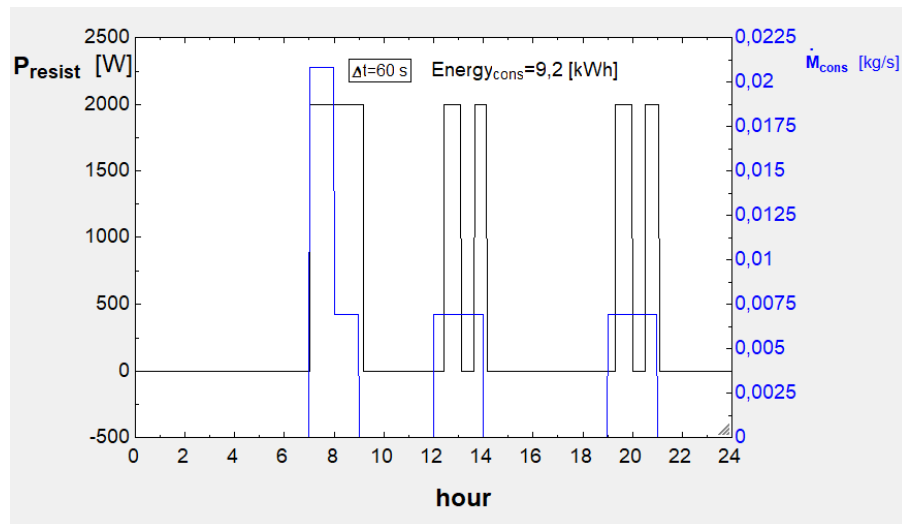


Figure 3.4.6 – Evolution of resistance power (on-off control) and water consumption rate.

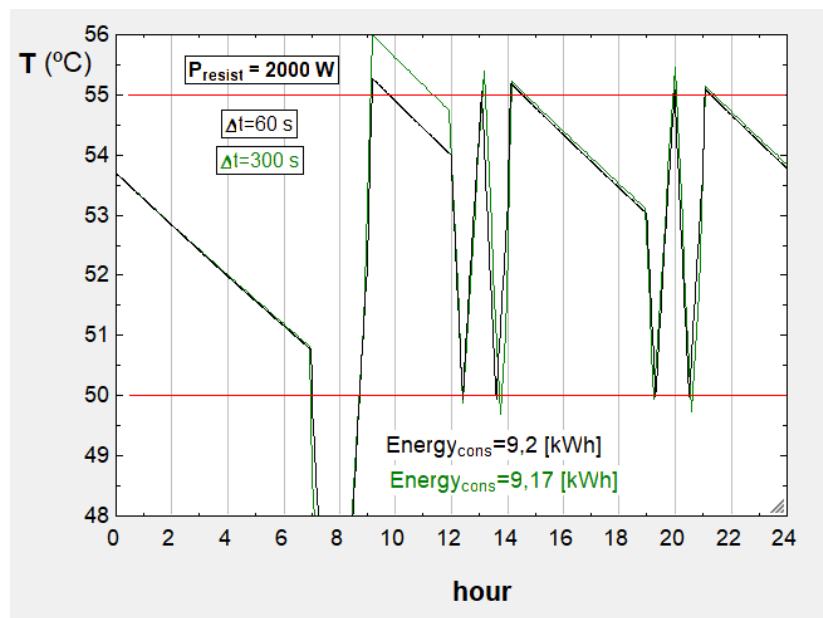


Figure 3.4.7 – Comparison of temperature profiles and daily energy values with 2 different time steps.

In order to limit the minimum water temperature, during the higher consumption period, it is possible to increase the resistance power. Figure 3.4.8 shows the water temperature profile with a 3000 W resistance. The minimum temperature increases to 49.4°C (an increase of almost 6°C), but with an increase in daily energy consumption to 9.65 kWh.

As an alternative to the on-off control, we also analyse the effect of using a proportional control between 50 and 55°C. In that case, the EES Function RESIST of Figure 3.4.3 will be only a function of temperature, and should be adapted. In the equations the resistance is defined with $P_{\text{resist}} = \text{RESIST}(T)$. And in this case, there is no need to use T_{old} , as there is no convergence instability, allowing a fully implicit formulation of the solution. Figure 3.4.9 shows the Equations Window for the proportional control case.

Figures 3.4.10 and 3.4.11 show the graphical results with proportional control, and a maximum power of 2000 W. The minimum water temperature is now 44.8°C (1.3°C higher), and in other consumption periods it is always higher than with the on-off control.

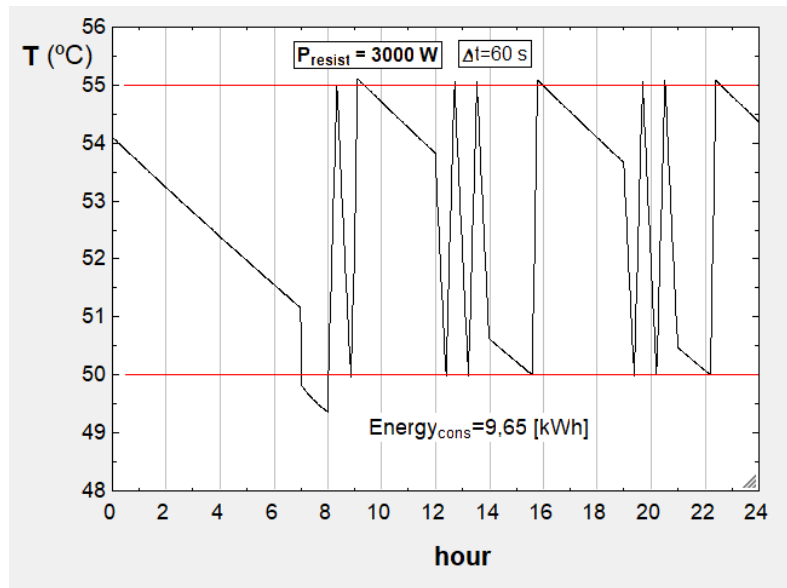


Figure 3.4.8 – Evolution of water temperature and daily energy consumption with $P_{\text{resist}} = 3000 \text{ W}$.

```

Equations Window
FUNCTION RESIST(T)
  P_max=2000
  T_max=55
  T_min=50
  IF T>=T_max THEN
    RESIST=0
  ELSE
    IF T<=T_min THEN RESIST=P_max ELSE RESIST=P_max*(T_max-T)/(T_max-T_min)
  ENDIF
END

V=100/1000 "m3"
DELTA=60*1 "s"
rho=Density(Water, T=50, p=100)
c_p=Cp(Water, T=50, p=100)*1000 "em J/kg/c"
T_ext=20
T_mains=15
UA=1,5

hour=time/3600 "time defined in Parametric Table"
row_param=1+time/DELTA "counts row number in parametric table"

rho*V*c_p*(T-T_old)/DELTA=P_resist+M_dot_cons*c_p*(T_mains-T)-UA*(T-T_ext)

T_old=tablevalue(Table 1;row_param-1,T) "recovers previous T"
P_resist=RESIST(T)

row_lookup=trunc(time/3600)+1 "counts row number in lookup table"
M_dot_cons=lookup(Lookup 1;row_lookup;consumption)/3600 "water flow rate in kg/s"

Energy_cons=Energy_cons_old+P_resist*DELTA/1000/3600 "in kWh"
Energy_cons_old=tablevalue(Table 1;row_param-1;Energy_cons)

```

Figure 3.4.9 – Equations Window for the water heater with proportional control.

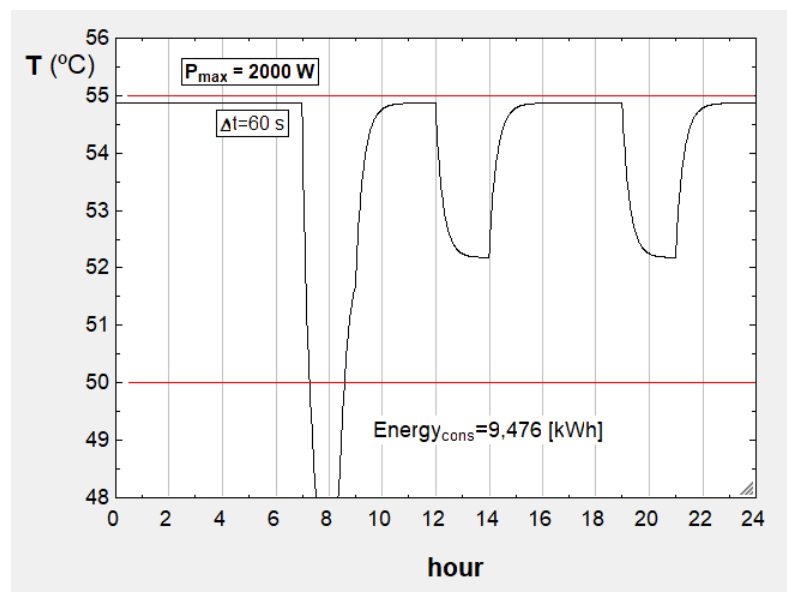


Figure 3.4.10 – Evolution of the water temperature with proportional control.

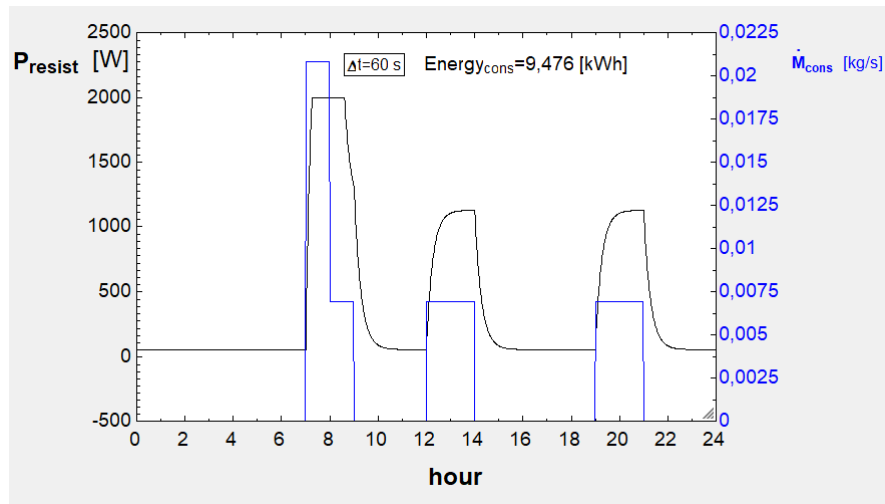


Figure 3.4.11 – Evolution of resistance power (proportional control) and water consumption rate.

With proportional control the resistance works most of the time at less than 1000 W. The daily energy consumption is a bit higher at 9.48 kWh (compared to 9.2 kWh with on-off).

3.5 Domestic hot water solar system

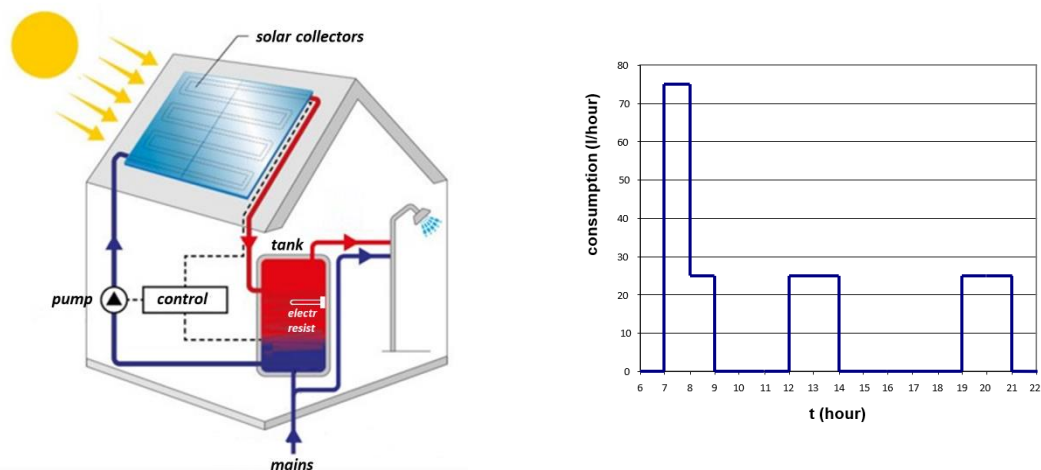


Figure 3.5.1 – Domestic hot water solar system and daily water consumption pattern.

Figure 3.5.1 represents a system that provides domestic hot water, using solar thermal collectors and a storage tank. Water is circulated in the collectors when there is an increase in the water temperature – collector outlet temperature higher than tank temperature. This is achieved with the differential controller that switches on or off the pump.

Consider the following system characteristics:

- solar collector area: 4 m²; collector water flow rate: 0.08 kg/s (when pump switched on);
- storage tank: volume – 300 l; heat loss coefficient – 1.8 W/°C; outside temperature – 20°C;
- inlet cold water temperature (from the mains): 15°C.

The collector thermal efficiency (useful heat divided by total incident solar radiation) depends on the collector inlet temperature, outside ambient air temperature, and incident solar radiation:

$$\eta_{col} = 0.8 - 5 (T_{col,in} - T_{amb}) / I_{sol} \quad (3.5.1)$$

Assuming a global tank model, and no water heating element besides the solar collectors, simulate the system during one day of March, with a total water consumption of 200 litres/day, according to the graph in Figure 3.5.1, and with the following climatic data:

<i>hour</i>	1	2	3	4	5	6	7	8	9	10	11	12
T_{amb} (°C)	10.7	10.5	10.4	10.2	10.2	10.0	9.9	10.3	11.4	12.5	13.4	14.1
I_{sol} (W/m ²)	0	0	0	0	0	0	0	0	40	168	319	436
<i>hour</i>	13	14	15	16	17	18	19	20	21	22	23	24/0
T_{amb} (°C)	14.7	15.2	15.2	15.3	14.7	14.0	12.8	12.2	11.7	11.3	11.1	10.8
I_{sol} (W/m ²)	564	677	556	409	351	246	148	14	0	0	0	0

Next, assume that the storage tank also provides auxiliary energy through an electrical resistance, with a maximum power of 1500 W and a proportional control (proportional band between 50°C – maximum power – and 55°C – zero power). Obtain the temperature evolution and the daily energy consumption, during the same day.

Finally, assume that auxiliary energy is provided in a separate (smaller) tank, placed after the larger storage tank, with a capacity of 50 litres and a heat loss coefficient of 0.6 W/°C, using an identical electrical resistance and control. Perform the new simulation and compare its results with the previous ones.

The model will be based on the calculation of the tank water temperature (T), considered to be uniform. The variation of water energy is related to all energy inputs (water inlet from mains and water inlet from the collector circuit) and outputs (water outlet for consumption, heat losses to the outside and water outlet to the collector circuit (collector inlet):

$$Mc_p \frac{dT}{dt} = \dot{M}_{cons} c_p (T_{mains} - T) + \dot{M}_{col} c_p (T_{col,out} - T) - (UA)_{tank} (T - T_{ext}) \quad (3.5.2)$$

\dot{M}_{cons} is not constant, and the collector input only exists if there is heat gain in the collectors ($T_{col,out} > T$). When this happens, $T_{col,out}$ may be related to T and to the climatic data through the collector efficiency – equation (3.5.1). We may write

$$\eta_{col} I_{sol} A_{col} = \dot{M}_{col} c_p (T_{col,out} - T) \quad (3.5.3)$$

or

$$0.8 I_{sol} A_{col} - 5(T - T_{amb}) A_{col} = \dot{M}_{col} c_p (T_{col,out} - T) \quad (3.5.4)$$

Equation (3.5.4) expresses a steady-state collector balance, valid for all instants, as the collector thermal inertia is negligible (short thermal response time). Then, this algebraic equation, together with the differential equation (3.5.2), define the values of T and $T_{col,out}$. Due to the changes in \dot{M}_{cons} and \dot{M}_{col} a numerical solution is required. Again, using the implicit method of integration, the algebraic/discretised equations to solve are:

$$\frac{T^{t+\Delta t} - T^t}{\Delta t} = \frac{1}{Mc_p} \left[\dot{M}_{cons}^{t+\Delta t} c_p (T_{mains} - T^{t+\Delta t}) + \dot{M}_{col}^{t+\Delta t} c_p (T_{col,out}^{t+\Delta t} - T^{t+\Delta t}) - (UA)_{tank} (T^{t+\Delta t} - T_{ext}) \right] \quad (3.5.5)$$

and

$$0.8 I_{sol}^{t+\Delta t} A_{col} - 5(T^{t+\Delta t} - T_{amb}^{t+\Delta t}) A_{col} = \dot{M}_{col}^{t+\Delta t} c_p (T_{col,out}^{t+\Delta t} - T^{t+\Delta t}) \quad (3.5.6)$$

Replacing (3.5.6) in (3.5.5) a single equation may be obtained:

$$\frac{T^{t+\Delta t} - T^t}{\Delta t} = \frac{1}{Mc_p} \left[\dot{M}_{cons}^{t+\Delta t} c_p (T_{mains} - T^{t+\Delta t}) + [0.8 I_{sol}^{t+\Delta t} - 5(T^{t+\Delta t} - T_{amb}^{t+\Delta t})]^+ A_{col} - (UA)_{tank} (T^{t+\Delta t} - T_{ext}) \right] \quad (3.5.7)$$

However, only positive values of the efficiency must be considered, as in other situations (e.g. at night) the collector circuit pump is stopped. This was noted in equation (3.5.7) with the $[]^+$ sign. Therefore, the pump operation needs to be defined in a function. The model also needs to define $\dot{M}_{cons}^{t+\Delta t}$. The water consumption flow rate will be defined in a Lookup Table (Lookup 1), translating the graph of Figure 3.5.1. The hourly values of ambient temperature and solar radiation, will also be included in the Lookup Table, and then used to calculate the values for a given time. Figure 3.5.2 shows the Lookup Table.

hour	T _{amb} [°C]	I _{sol} [W/m ²]	consumption [l/hour]
0	10.8	0	0
1	10.7	0	0
2	10.5	0	0
3	10.4	0	0
4	10.2	0	0
5	10.2	0	0
6	10	0	0
7	9.9	0	75
8	10.3	0	25
9	11.4	40	0
10	12.5	168	0
11	13.4	319	0
12	14.1	436	25
13	14.7	564	25
14	15.2	677	0
15	15.2	556	0
16	15.3	409	0
17	14.7	351	0
18	14	246	0
19	12.8	148	25
20	12.2	14	25
21	11.7	0	0
22	11.3	0	0
23	11.1	0	0
24	10.8	0	0

Figure 3.5.2 – Lookup Table for the hot water solar system example.

The EES calculation procedure will be developed with a Parametric Table (Table 1) where the different time values are defined, step after step, or Run after Run, during 24 hours. Starting with an initial value of the water temperature, the following values are obtained. Figure 3.5.3 presents the Equations Window, with the definition of equations (3.5.5) and (3.5.6), and the pump control function (PUMP), plus other input values. The control function defines a pump factor (f_{pump} , equal to 0 or 1) that, multiplied by the collector flow rate, takes into account the pump state in the tank balance. The time step was imposed at 60 s (1 minute).

In the Equations Window, T and other variables correspond to values at $t + \Delta t$ in the previous equations. The previous value T^t is designated as T_{old} , or T_{old} . The time rows in the Parametric Table are identified by a counter (“line”), starting with the first row with initial values. The function TABLEVALUE recovers the previous temperature (T^t or T_{old}) by searching in the previous row (line -1).

The values of consumption flow rate, ambient temperature and solar radiation, for a given time, were calculated from the Lookup Table using an interpolation function (Interpolate1 – a first degree interpolation).


```

Equations Window
"solar DHW system with storage"
FUNCTION PUMP(T_col_in,T_col_out)
  IF T_col_out>T_col_in THEN
    PUMP=1
  ELSE
    PUMP=0
  ENDIF
END

V=300/1000 "m3, storage tank"
DELTA_t=60*1 "s"
rho=Density(Water, T=50;p=100)
c_p=Cp(Water, T=50;p=100)*1000 "in J/kgK"
T_ext=20
T_mains=15
UA=1,8

eta_0_col=0,8
FU_col=5
A_col=4
M_dot_col=0,080 "kg/s"

hour=time/3600
line=1+time/DELTA_t

T_old=tablevalue(Table 1;line-1;#T) "recovers previous T"
T_amb=Interpolate1(Lookup 1;T_amb;hour;hour=time/3600)
I_sol=Interpolate1(Lookup 1;I_sol;hour;hour=time/3600)
M_dot_cons=Interpolate1(Lookup 1;consumption;hour;hour=trunc(time/3600))/3600

I_sol*A_col*eta_0_col-FU_col*A_col*(T-T_amb)=M_dot_col*c_p*(T_col_out-T)

rho*V*c_p*(T-T_old)/DELTA_t=M_dot_cons*c_p*(T_mains-T)+M_dot_col*f_pump*c_p*(T_col_out-T)-UA*(T-T_ext)
f_pump=PUMP(T_col_out)

```

Figure 3.5.3 – Equations Window for the hot water solar system example.

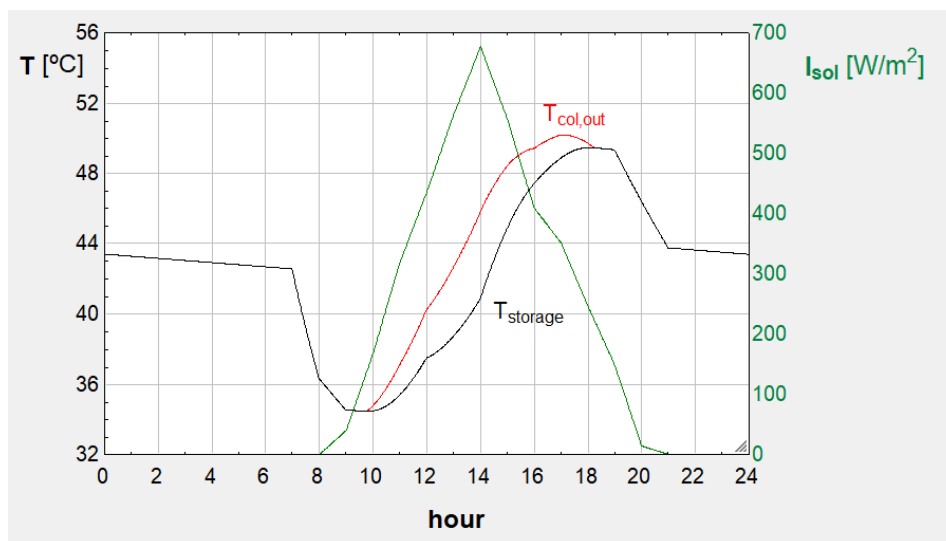


Figure 3.5.4 – Evolution of storage and collector temperatures for the hot water solar system example.

To find the initial water temperature (Run 1 of the Parametric Table) it was assumed that the same daily cycle is repeated every day; this way, the initial temperature will be equal to the temperature at the end of the day. As the simulation was run for 24 hours only, this involved an iterative use of the model, starting with an initial temperature value, and changing it to the final temperature value (24 h), until a close match between the 2 values was obtained; a value of 43.4°C was obtained.

Figure 3.5.4 presents the temperature results. The minimum water temperature is 34.5°C, and the maximum is 49.5°C. Note that the pump circulates water in the collectors between 9.8 and 18.2 (during 8.4 hours). The maximum water temperature increase in the solar collectors is equal to 4.9°C at 14:00, when the solar radiation is higher.

If higher water temperatures are needed, then auxiliary heating needs to be used. The first option to analyse is to use an electrical resistance in the storage tank, with a maximum power of 1500 W and a proportional control between 50 and 55°C. The corresponding model needs to include the resistance input, which depends on water temperature. The discretised equations become:

$$\frac{T^{t+\Delta t} - T^t}{\Delta t} = \frac{1}{M c_p} \left[P_{resist}^{t+\Delta t} + \dot{M}_{cons}^{t+\Delta t} c_p (T_{mains} - T^{t+\Delta t}) + \dot{M}_{col}^{t+\Delta t} c_p (T_{col,out}^{t+\Delta t} - T^{t+\Delta t}) - (UA)_{tank} (T^{t+\Delta t} - T_{ext}) \right] \quad (3.5.8)$$

and

$$0.8 I_{sol}^{t+\Delta t} A_{col} - 5(T^{t+\Delta t} - T_{amb}^{t+\Delta t}) A_{col} = \dot{M}_{col}^{t+\Delta t} c_p (T_{col,out}^{t+\Delta t} - T^{t+\Delta t}) \quad (3.5.9)$$

Figure 3.5.5 presents the Equations Window to solve the model. As in the example of section 3.4, the resistance input is calculated with a Function (RESIST), depending on the water temperature. Besides that, a new variable was introduced to calculate the daily energy consumption, accumulating the resistance power over time: variable Energy_cons.

```

solar DHW system with integrated storage and aux heater with proportional control

FUNCTION PUMP(T_col_in,T_col_out)
  IF T_col_out>T_col_in THEN
    PUMP=1
  ELSE
    PUMP=0
  ENDF
END

FUNCTION RESIST(T)
  P_max=1500
  T_max=55
  T_min=50
  IF T>=T_max THEN
    RESIST=0
  ELSE
    IF T<=T_min THEN RESIST=P_max ELSE RESIST=P_max*(T_max-T)/(T_max-T_min)
  ENDF
END

V=300/1000 "m3, storage tank"
DELTA=60*1 "s"
rho=Density(Water, T=50,p=100)
c_p=Cp(Water, T=50,p=100)*1000 "in J/kgK"
T_ext=20
T_mains=15
UA=1,8

V_aux=50/1000 "m3, auxiliary tank"
UA_aux=0,6

eta_col=0,8
FU_col=5
A_col=4
M_dot_col=0,080 "kg/s"

hour=time/3600
line=1+time/DELTA

T_old=tablevalue(Table 1,line-1,#T) "recovers previous T"
T_amb=Interpolate1(Lookup 1,T_amb,hour,hour=time/3600)
I_sol=Interpolate1(Lookup 1,I_sol,hour,hour=time/3600)
M_dot_cons=Interpolate1(Lookup 1,consumption,hour,hour=trunc(time/3600))/3600

I_sol*A_col*eta_col-FU_col*A_col*(T-T_amb)=M_dot_col*c_p*(T_col_out-T)

rho*V*c_p*(T-T_old)/DELTA=P_resist+M_dot_cons*c_p*(T_mains-T)+M_dot_col*f_pump*c_p*(T_col_out-T)-UA*(T-T_ext)
f_pump=PUMP(T_col_out)
P_resist=RESIST(T)

Energy_cons=Energy_cons_old+P_resist*DELTA/1000/3600 "em kWh"
Energy_cons_old=tablevalue(Table 1,line-1,#Energy_cons)

```

Figure 3.5.5 – Equations Window for the hot water solar system with integrated storage and electrical heating.

The graphical results are shown in Figures 3.5.6 and 3.5.7. The temperature levels are higher than in Figure 3.5.4: the minimum storage temperature is now 48.6°C and the maximum is now 60.5°C, due to the resistance input. But as the collector inlet temperature (storage temperature) is higher, the collectors operate with poorer efficiency and less time. The daily electrical energy consumption is equal to 5.753 kWh, while the solar collectors provide 4.617 kWh to the storage (45% of the total). The collected solar energy may be calculated by adding another accumulation variable in the Equations Window – Energy_sol (similar to Energy_cons):

$$\text{Energy}_{sol} = \text{Energy}_{sol,old} + f_{pump} * \dot{M}_{col} * c_p * (T_{col,out} - T) * \Delta t \quad (3.5.10)$$

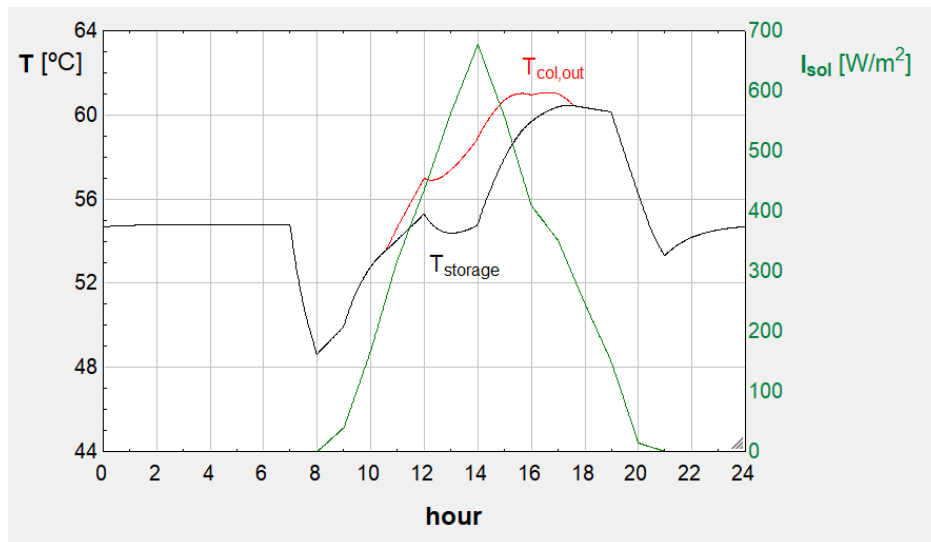


Figure 3.5.6 – Evolution of storage and collector temperatures for the hot water solar system with integrated storage and electrical heating.

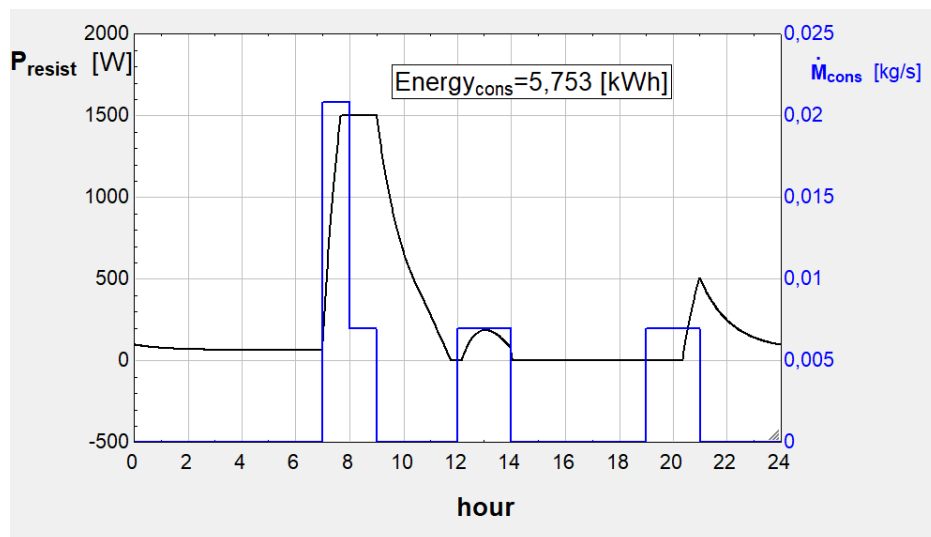


Figure 3.5.7 – Evolution of resistance power (proportional control) and water consumption rate for the hot water solar system with integrated storage and electrical heating.

The second alternative to analyse is the use of a separate auxiliary energy tank, with a smaller volume (50 litres), where the electrical resistance is placed; as before, the resistance has a maximum power of 1500 W and a proportional control between 50 and 55°C. Figure 3.5.8 represents this system configuration.

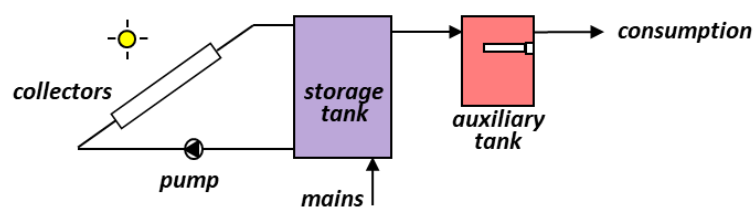


Figure 3.5.8 – Schematic representation of the hot water solar system with separate storage and auxiliary heating.

The Equations Window for this system configuration is shown in Figure 3.5.9. By comparison with Figure 3.5.5 (integrated storage and heating), another set of equations was added to include the auxiliary tank and calculate the variables T_{aux} and T_{aux_old} . The resistance input (P_{resist}) was removed from the storage tank and included in the smaller tank. Figures 3.5.10 and 3.5.11 show the EES results in graphical form.

```

"solar DHW system with storage and separate tank with aux heater with proportional control"

FUNCTION PUMP(T_col_in,T_col_out)
  IF T_col_out>T_col_in THEN
    PUMP=1
  ELSE
    PUMP=0
  ENDIF
END
END

FUNCTION RESIST(T)
  P_max=1500
  T_max=55
  T_min=50
  IF T>=T_max THEN
    RESIST=0
  ELSE
    IF T<=T_min THEN RESIST=P_max ELSE RESIST=P_max*(T_max-T)/(T_max-T_min)
  ENDIF
END

V=300/1000 "m3, storage tank"
DELTA=60*1 "s"
rho=Density(Water, T=50,p=100)
c_p=Cp(Water, T=50,p=100)*1000 "in J/kgK"
T_ext=20
T_mains=15
UA=1.8

V_aux=50/1000 "m3, auxiliary tank"
UA_aux=0.6

eta_0_col=0.8
FU_col=5
A_col=4
M_dot_col=0,080 "kg/s"

hour=time/3600
line=1+time/DELTA

T_old=tablevalue(Table 1;line-1;#T) "recovers previous T"
T_amb=Interpolate1(Lookup 1;T_amb;hour;hour=time/3600)
I_sol=Interpolate1(Lookup 1;I_sol;hour;hour=time/3600)
M_dot_cons=Interpolate1(Lookup 1;consumption;hour;hour=trunc(time/3600))/3600

I_sol*A_col*eta_0_col-FU_col*A_col*(T-T_amb)=M_dot_col*c_p*(T_col_out-T)

rho*V*c_p*(T-T_old)/DELTA=M_dot_cons*c_p*(T_mains-T)+M_dot_col*f_pump*c_p*(T_col_out-T)-UA*(T-T_ext)
f_pump=PUMP(T_col_out)
P_resist=RESIST(T_aux)

T_aux_old=tablevalue(Table 1;line-1;#T_aux) "recovers previous T_aux"
rho*V_aux*c_p*(T_aux-T_aux_old)/DELTA=M_dot_cons*c_p*(T-T_aux)+P_resist-UA_aux*(T_aux-T_ext)

Energy_cons=Energy_cons_ant+P_resist*DELTA/1000/3600 "em kWh"
Energy_cons_ant=tablevalue(Table 1;line-1;#Energy_cons)
    
```

Figure 3.5.9 – Equations Window for the hot water solar system with separate storage and auxiliary heating.

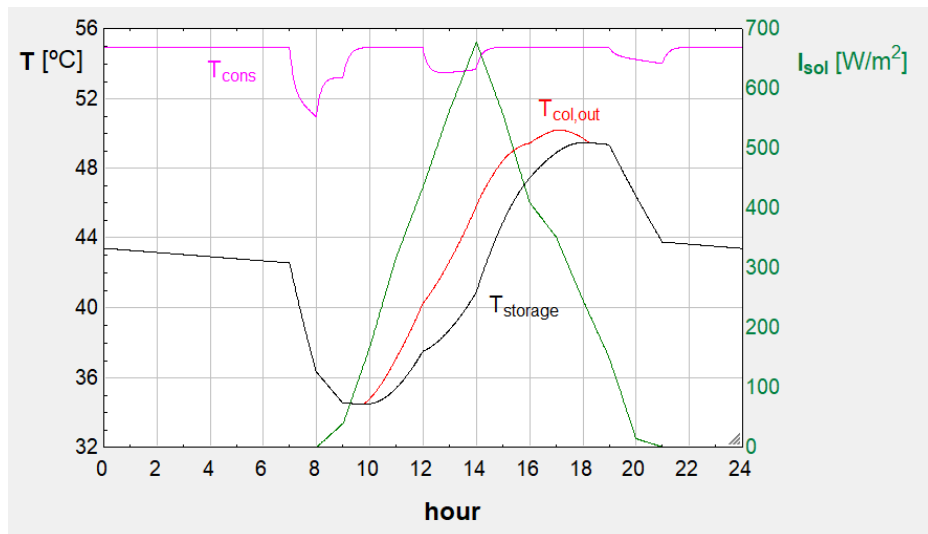


Figure 3.5.10 – Evolution of storage, consumption and collector temperatures for the hot water solar system with separate storage and auxiliary heating.

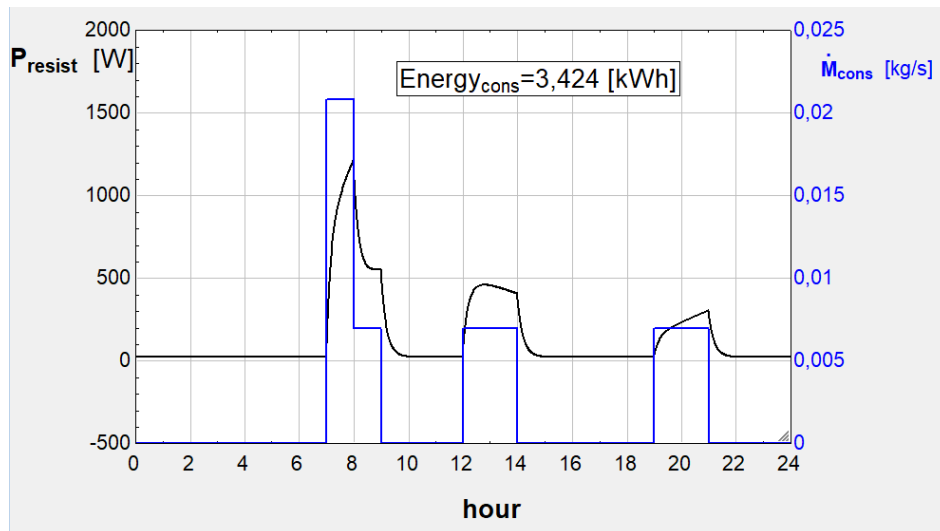


Figure 3.5.11 – Evolution of resistance power (proportional control) and water consumption rate for the hot water solar system with separate storage and auxiliary heating.

With separate storage and auxiliary heating, the storage and collector outlet temperatures have exactly the same values as in Figure 3.5.4, as no electrical heating is provided to the storage. Therefore, the inlet collector temperatures are kept at the minimum level, with higher collector efficiencies. The auxiliary tank temperature never goes below 51°C, even when the water consumption is higher, and in other water consumption periods it is always above 53.5°C.

In this configuration the daily electrical energy consumption is equal to 3.424 kWh, with a reduction of 40% compared to the integrated storage/heating configuration. The solar collectors provide now 6.883 kWh to the storage (67% of the total).

3.6 Swimming pool solar heating

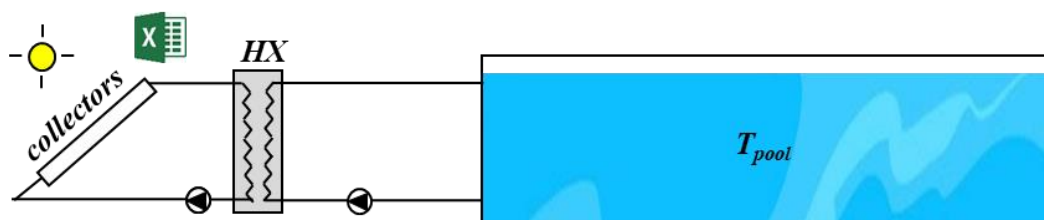


Figure 3.6.1 – Swimming pool solar heating system.

An outdoor swimming pool is heated with a solar thermal system, as shown in the figure. There is a heat exchanger (HX) between the collector circuit and the pool circuit (with water circulating in both). We would like to evaluate the time required to heat the pool water from the mains temperature of 16°C to the required operating temperature of 28°C. The pool is filled with water on the 1st of May, and the hourly climatic data are available in an Excel file (solar radiation on horizontal and collector surfaces, ambient air temperature and humidity, and wind speed). The pool water has no significant shading and its surface is covered during the night period, from 18:00 to 8:00 (during this period all pool thermal losses may be neglected). Consider the pool at uniform temperature.

The solar collector area is equal to 125 m², equal to 50% of the pool area (250 m²). The pool average depth is equal to 2 m. The collectors have the following efficiency characteristic curve:

$$\eta_{col} = 0.72 - 5 (T_{col,in} - T_{amb})/I_{sol,col} \quad (3.6.1)$$

The collector circuit pump works whenever there is heat gain. The pool circuit pump works when there is collector circulation, and the fluids and flow rates of both circuits are the same (0.020 kg/s/m²_{col} in the collector circuit). Under these conditions the heat exchanger efficiency is equal to 0.7.

Build a numerical model for the pool water temperature evolution, and solve it using the EES software.

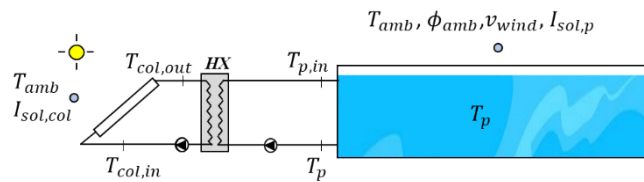


Figure 3.6.2 – Temperatures and climatic variables for the swimming pool model.

The main model equation to calculate the pool water temperature (T_p) is related to the time variation of water energy, which depends on all energy inputs and outputs. We will assume that, although some of the water evaporates to the ambient air, its mass is compensated by introducing new water; therefore, the pool water mass is constant. But the effect of this new water on the energy balance is neglected. When the pumps of Figure 3.6.2 are in operation, the pool receives water at a higher temperature ($T_{p,in}$). On the other hand, if the water surface is not covered, it absorbs solar radiation and loses heat by convection, radiation and evaporation. Other pool heat losses or gains are neglected. The following equation expresses the energy balance:

$$M_p c_{p,p} \frac{dT_p}{dt} = [\dot{M}_p c_{p,p} (T_{p,in} - T_p)]^+ + [\dot{Q}_{sol,abs} - (\dot{Q}_{evap} + \dot{Q}_{conv} + \dot{Q}_{rad})]^{++} \quad (3.6.2)$$

The $[]^+$ notation means that this term is considered only when positive, that is, only when there is heat gain in the solar collectors ($T_{col,out} > T_{col,in}$ implies that $T_{p,in} > T_p$). And the $[]^{++}$ notation means that the term only applies if the pool surface is not covered (otherwise it is considered as zero).

The absorbed solar radiation depends on the absorptance coefficient of water and incident solar radiation on horizontal surfaces:

$$\dot{Q}_{sol,abs} = \alpha_p I_{sol,p} A_p \quad (3.6.3)$$

The evaporation losses depend on the evaporated mass, which depends on the mass transfer coefficient, difference in water vapour concentration and enthalpy of vaporisation:

$$\dot{Q}_{evap} = h_m [\rho_{v,sat}(T_p) - \phi_{amb} \rho_{v,sat}(T_{amb})] \Delta h_{lv} A_p \quad (3.6.4)$$

The convective heat losses depend on the convective heat transfer coefficient:

$$\dot{Q}_{conv} = h (T_p - T_{amb}) A_p \quad (3.6.5)$$

The heat and mass transfer coefficients may be related through the Lewis relationship – equation (3.3.2).

For a surface exposed to outside air, the coefficient may be related to the wind speed through a simple linear relationship for a varying flow over a rough flat surface:

$$h = 6.19 + 4.29 v_{wind} \quad (3.6.6)$$

One must note that the wind speed available in climatic files corresponds to values measured at a given height above the ground level (usually 10 m). The wind speed to be used in equation (3.6.6) should be the speed at ground level. It is recommended to divide the standard measured values by a factor between 5 and 10, depending on the wind protection at the pool site.

The radiative losses, assuming that all outside surfaces and sky are at the same temperature (equal to T_{amb})

$$\dot{Q}_{rad} = \varepsilon_p \sigma (T_p^4 - T_{amb}^4) A_p \quad (3.6.7)$$

The relationship between $T_{col,out}$ and $T_{col,in}$ depends on collector efficiency and climatic data. Using equation (3.6.1) we may write (assuming steady-state in the collectors):

$$0.72 I_{sol,col} A_{col} - 5(T_{col,in} - T_{amb}) A_{col} = \dot{M}_{col} c_{p,c} (T_{col,out} - T_{col,in}) \quad (3.6.8)$$

The relationship between $T_{p,in}$ and the other relevant temperatures depends on the heat exchanger efficiency:

$$\dot{M}_p c_{p,p} (T_{p,in} - T_p) / [C_{min} (T_{col,out} - T_p)] = 0.7 \quad (3.6.9)$$

where C_{min} is the minimum flow heat capacity ($\dot{M} c_p$) of the 2 streams; in this example the 2 capacities will be considered as equal (same fluid and same flow rate).

A final equation states the equality of the heat received by the colder stream and the heat lost by the warmer stream in the heat exchanger:

$$\dot{M}_p c_{p,p} (T_{p,in} - T_p) = \dot{M}_{col} c_{p,col} (T_{col,out} - T_{col,in}) \quad (3.6.10)$$

The model main equations are then equations (3.6.2), (3.6.8), (3.6.9) and (3.6.10), assisted by equations (3.6.3) to (3.6.7). Equations (3.6.8-10) are algebraic equations, valid for any instant $t + \Delta t$. Equation (3.6.2) must be numerically integrated; with the implicit method it becomes:

$$\dot{M}_p c_{p,p} \frac{T_p^{t+\Delta t} - T_p^t}{\Delta t} = [\dot{M}_p c_{p,p} (T_{p,in} - T_p)]^{t+\Delta t,+} + [\dot{Q}_{sol,abs} - (\dot{Q}_{evap} + \dot{Q}_{conv} + \dot{Q}_{rad})]^{t+\Delta t,++} \quad (3.6.11)$$

To implement the model equations in EES, we defined the pump operation ($[]^+$) and cover placement ($[]^{++}$) conditions, using FUNCTIONS, and 2 multiplying factors that are either 0 or 1: f_{pump} and f_{cover} . The values of the climatic variables were defined in a Lookup Table (“Lookup 1”) and used with the equations to sequentially calculate the temperatures, step after step. The Parametric Table (“Table 1”) will start with row/run number 1, with the initial pool water temperature of 16°C, and continue with time steps that were defined equal to 300 s (5 min). Figure 3.6.3 shows only the first rows of the Lookup Table, since the hourly data for the full month of May were introduced (for an average year).

hour [TSV]	Rad _{sol, horizontal} [W/m ²]	Rad _{sol, col} [W/m ²]	T _{amb} [°C]	φ _{amb} [%]	v _{wind} [m/s]	
Row 1	1	0	0	15,6	52	2,8
Row 2	2	0	0	17,2	41	2,8
Row 3	3	0	0	18	37	2,2
Row 4	4	0	0	18,2	35	2,2
Row 5	5	0	0	14,8	45	2,2
Row 6	6	44	48	14,4	50	1,1
Row 7	7	203	189	16,2	41	3,3
Row 8	8	425	405	16	46	3,6
Row 9	9	606	612	17	39	4,4
Row 10	10	744	775	18	41	4,4
Row 11	11	869	922	19,6	41	4,4
Row 12	12	919	980	21,6	42	3,9
Row 13	13	917	975	23	33	5
Row 14	14	853	893	24	29	2,2
Row 15	15	742	754	24,4	25	2,2
Row 16	16	583	557	25	27	3,3
Row 17	17	392	364	25,6	28	1,7
Row 18	18	189	248	25,4	26	2,8
Row 19	19	36	0	23,8	34	3,3
Row 20	20	0	0	21,4	39	5,6
Row 21	21	0	0	19,4	42	5
Row 22	22	0	0	18,6	40	6,1
Row 23	23	0	0	17,8	42	5
Row 24	24	0	0	16,4	46	3,3

Figure 3.6.3 – Lookup Table with climatic variables for the swimming pool example.

```

FUNCTION PUMP(T_col_in,T_col_out)
IF T_col_out>T_col_in THEN
    PUMP=1
ELSE
    PUMP=0
ENDIF
END

FUNCTION COVER(hour)
schedule=(hour/24-TRUNC(hour/24))*24
IF (schedule<8) OR (schedule>18) THEN
    COVER=0 "cover ON"
ELSE
    COVER=1 "cover OFF"
ENDIF
END

A_p=250 [m2] "Pool surface area"
V_p=A_p*2 [m3] "Pool volume"
A_col=125 [m2] "Collector area"
cp_p=Cp(Water,T=20,P=100)*1000 [J/kgK]
rho_p=Density(Water,T=20,P=100)
alpha_p=0,9
epsilon_p=0,9
FRtaualfa_col=0,72 "collector efficiency parameter"
FRU_col=5 "collector efficiency loss factor"
epsilon_HX=0,7 "HX efficiency"
M_dot_col=0,02*A_col
M_dot_p=M_dot_col
cp_col=cp_p
C_min=MIN(M_dot_p*cp_p,M_dot_col*cp_col)

DELTA_t=60*5 [s] "5 min"
hour=time/3600
line=1+time/DELTA_t

"interpolation of climatic variables"
I_sol=Interpolate1(Clima;Rad_sol_horizontal;hour;hour=hour)
I_sol_col=Interpolate1(Clima;Rad_sol_col;hour;hour=hour)
T_amb=Interpolate1(Clima;T_amb;hour;hour=hour)
phi_amb=Interpolate1(Clima;phi_amb;hour;hour=hour)
v_wind=Interpolate1(Clima;v_wind;hour;hour=hour)/5 "air velocity at water surface level"

"pool balance: implicit method"
rho_p*v_p*cp_p*(T_p-T_p_old)/DELTA_t=M_dot_p*cp_p*(T_p-in-T_p)*f_pump+(Q_dot_sol_amb-Q_dot_rad-Q_dot_conv-Q_dot_evap)*f_cover
T_p_old=TABLEVALUE(Table 1; line-1,#T_p)
"solar collectors"
M_dot_col*cp_col*(T_col_out-T_col_in)=(FRtaualfa_col*I_sol_col-FRU_col*(T_col_in-T_amb))*A_col
"HX balance"
M_dot_p*cp_p*(T_p-in-T_p)=epsilon_HX*C_min*(T_col_out-T_p)
M_dot_col*cp_col*(T_col_out-T_col_in)=M_dot_p*cp_p*(T_p-in-T_p)

"pump operation"
f_pump=PUMP(T_col_in,T_col_out)
"cover utilisation factor: according to schedule in FUNCTION COVER"
f_cover= COVER (hour)

"pool solar gains"
Q_dot_sol_amb=alpha_p*I_sol_p*A_p
"convection losses"
h_conv=6,19+4,29*v_wind
Q_dot_conv=h_conv*A_p*(T_p-T_amb)
"radiation losses; T_sky assumed = T_amb"
Q_dot_rad=epsilon_p*sigma*(T_p+273,15)^4-(T_amb+273,15)^4*A_p
"evaporation losses"
h_m=h_conv*2,82E-5/Conductivity(Air;h_a;T=20,P=100)
rho_vs_Tp=HumRat(AirH2O;T=T_p,r=1,P=100)/Volume(AirH2O;T=T_p,r=1,P=100)
rho_vs_Tamb=HumRat(AirH2O;T=T_amb,r=1,P=100)/Volume(AirH2O;T=T_amb,r=1,P=100)
Q_dot_evap=h_m*(rho_vs_Tp-rho_vs_Tamb*phi_amb/100)*enthalpy_vaporization(Water;T=T_p)*1000*A_p "in W"
    
```

Figure 3.6.4 – Equations Window for the swimming pool example.

The PUMP Function defines if the pump operates ($f_{pump} = 1$, when $T_{col,out} > T_{col,in}$), and is similar to the one seen in example/section 3.5.

The COVER Function defines when the pool cover is placed ($f_{cover} = 0$, between 18:00 and 8:00). This will eliminate solar gains and heat losses in the pool surface in that period. Because several days were simulated, a “schedule” variable was used to take into account the same period every day.

Figure 3.6.5 presents results from a few rows/runs of “Table 1”. In those rows different combinations of f_{pump} and f_{cover} values occurred: pump not operating with cover on, followed by pump operating with cover, and then followed by pump operating without cover. Note that solar gains and evaporation losses that appear on “Table 1” when the cover is on, are calculated but do not affect the pool balance, as they are multiplied by $f_{cover} = 0$.

Run	time [s]	hour	$I_{sol,p}$ [W/m ²]	$I_{sol,col}$ [W/m ²]	T_{amb} [°C]	A_{amb} [%]	V_{wind} [m/s]	T_p [°C]	$T_{col,in}$ [°C]	$T_{col,out}$ [°C]	$T_{p,in}$ [°C]	f_{pump}	f_{cover}	$\dot{Q}_{sol,abs}$ [W]	\dot{Q}_{evap} [W]
Run 2088	626100	173.9	58.67	71.5	11.4	69	0.7883	25.83	25.73	25.49	25.59	0	0	13200	108359
Run 2089	626400	174	64	78	11.4	69	0.78	25.83	25.75	25.56	25.64	0	0	14400	107954
Run 2090	626700	174.1	78.83	90.75	11.37	68.42	0.7667	25.83	25.8	25.72	25.75	0	0	17737	107776
Run 2091	627000	174.2	93.67	103.5	11.33	67.83	0.7533	25.83	25.84	25.87	25.85	1	0	21075	107591
Run 2092	627300	174.3	108.5	116.3	11.3	67.25	0.74	25.83	25.89	26.02	25.96	1	0	24413	107399
Run 2093	627600	174.3	123.3	129	11.27	66.67	0.7267	25.83	25.93	26.17	26.07	1	0	27750	107202
Run 2094	627900	174.4	138.2	141.8	11.23	66.08	0.7133	25.83	25.98	26.32	26.17	1	0	31088	106998
Run 2095	628200	174.5	153	154.5	11.2	65.5	0.7	25.83	26.02	26.47	26.28	1	0	34425	106789
Run 2096	628500	174.6	167.8	167.2	11.17	64.92	0.6867	25.83	26.07	26.62	26.38	1	0	37762	106573
Run 2097	628800	174.7	182.7	180	11.13	64.33	0.6733	25.83	26.11	26.77	26.49	1	0	41100	106352
Run 2098	629100	174.8	197.5	192.8	11.1	63.75	0.66	25.84	26.16	26.92	26.59	1	0	44438	106124
Run 2099	629400	174.8	212.3	205.5	11.07	63.17	0.6467	25.84	26.21	27.07	26.7	1	0	47775	105891
Run 2100	629700	174.9	227.2	218.3	11.03	62.58	0.6333	25.84	26.25	27.22	26.81	1	0	51113	105651
Run 2101	630000	175	242	231	11	62	0.62	25.84	26.3	27.37	26.91	1	0	54450	105406
Run 2102	630300	175.1	259.6	246.9	11.12	61.67	0.6383	25.84	26.36	27.58	27.06	1	0	58406	106285
Run 2103	630600	175.2	277.2	262.8	11.23	61.33	0.6567	25.84	26.42	27.78	27.2	1	0	62363	107167
Run 2104	630900	175.3	294.8	278.8	11.36	61	0.675	25.85	26.49	27.98	27.34	1	0	66319	108050
Run 2105	631200	175.3	312.3	294.7	11.47	60.67	0.6933	25.85	26.55	28.18	27.48	1	0	70275	108935
Run 2106	631500	175.4	329.9	310.6	11.58	60.33	0.7117	25.85	26.61	28.39	27.63	1	0	74231	109823
Run 2107	631800	175.5	347.5	326.5	11.7	60	0.73	25.85	26.67	28.59	27.77	1	0	78188	110712
Run 2108	632100	175.6	365.1	342.4	11.82	59.67	0.7483	25.86	26.74	28.79	27.91	1	0	82144	111604
Run 2109	632400	175.7	382.7	358.3	11.93	59.33	0.7667	25.86	26.8	29	28.06	1	0	86100	112499
Run 2110	632700	175.8	400.3	374.3	12.05	59	0.785	25.86	26.86	29.2	28.2	1	0	90056	113396
Run 2111	633000	175.8	417.8	390.2	12.17	58.67	0.8033	25.87	26.93	29.4	28.34	1	0	94012	114296
Run 2112	633300	175.9	435.4	406.1	12.28	58.33	0.8217	25.87	26.99	29.61	28.49	1	0	97969	115198
Run 2113	633600	176	453	422	12.4	58	0.84	25.87	27.05	29.8	28.62	1	1	101925	116024
Run 2114	633900	176.1	468.5	439.6	12.67	56.25	0.83	25.86	27.11	30.03	28.78	1	1	105413	116052
Run 2115	634200	176.2	484	457.2	12.93	54.5	0.82	25.86	27.18	30.26	28.94	1	1	108900	116116

Figure 3.6.5 – Extract of the Parametric Table “Table 1” for the swimming pool example.

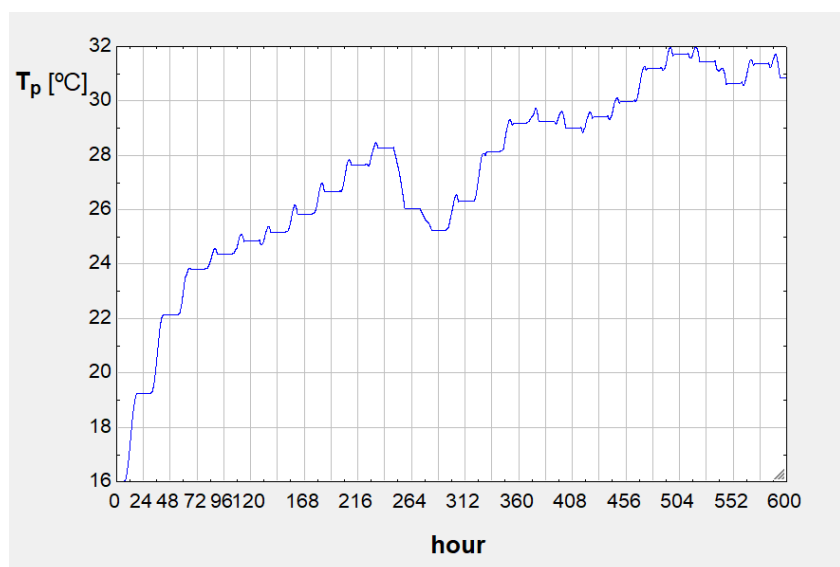


Figure 3.6.6 – Evolution of pool water temperature for the swimming pool example.

Figure 3.6.6 presents a graphical evolution of the pool water temperature during the first 25 days of May. Initially the increase in temperature is higher, as the water is colder, and steps may be noticed in the curve, corresponding to daily increases in water temperature. At the end of each day there is a slight decrease in water temperature. After 15 days the pool reaches a stable value above 28°C (after a decrease), and even increases above 30°C during the last week.

The interpretation of the daily evolution is clearer if we look at the evolution during 2 days – Figure 3.6.7 – where the collector inlet and outlet temperatures are also represented. At the start of the sunlight period the collectors start collecting solar radiation, and with the increase in incident solar radiation the pool water heats up. The temperature rise in the collectors achieves a maximum of about 8°C. Then, at the end of the sunlight period, the collector contribution reduces, and pool heat losses increase, which leads to the slight pool temperature reduction. This reduction could be avoided (or minimised) if the pool cover was placed sooner than 18:00. Figure 3.6.8 shows the schedule of cover and collector pump use. It would be beneficial to better match the 2 schedules, removing and placing the cover at about the same time as the pumps switch on and off (respectively).

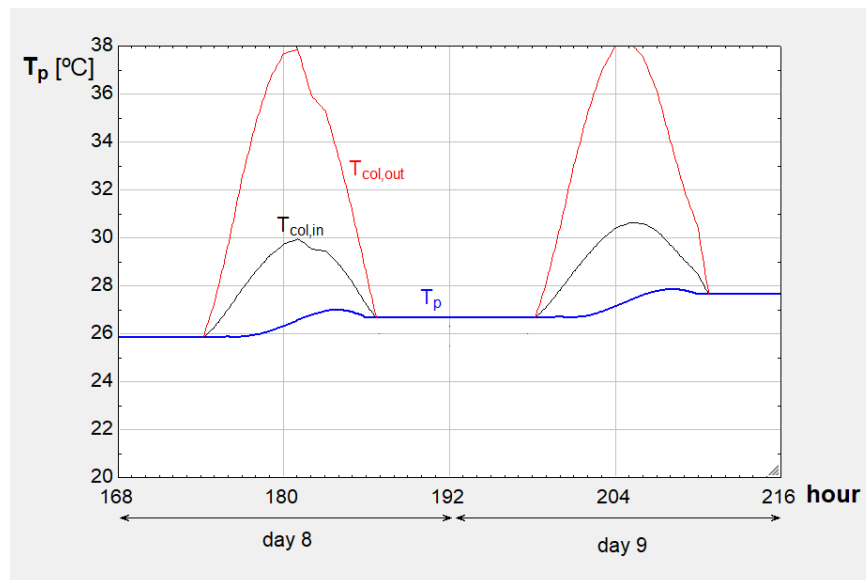


Figure 3.6.7 – Evolution of pool water and collector temperatures during two days.

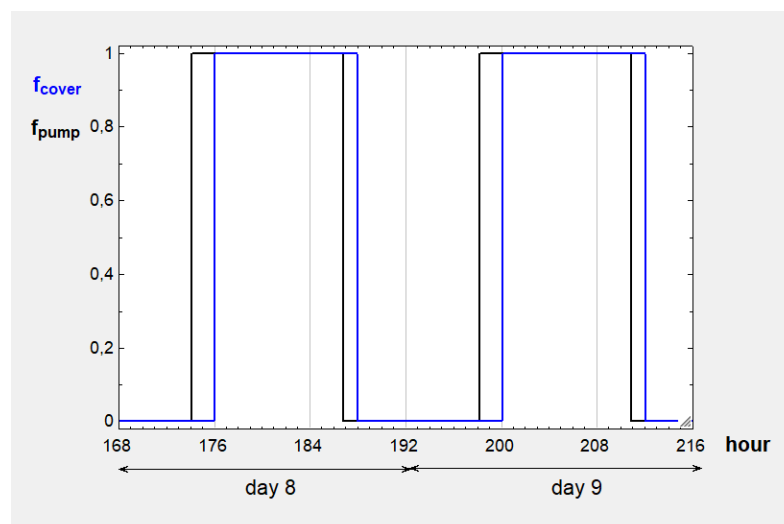


Figure 3.6.8 – Evolution of cover and pump factors during two days.

Figure 3.6.9 shows the different pool heat losses (radiative, convective and evaporative), compared with the solar radiation directly absorbed by the pool surface. Note that in certain periods the direct absorption exceeds the sum of all losses, meaning that the pool water is then heated, even without the collector contribution.

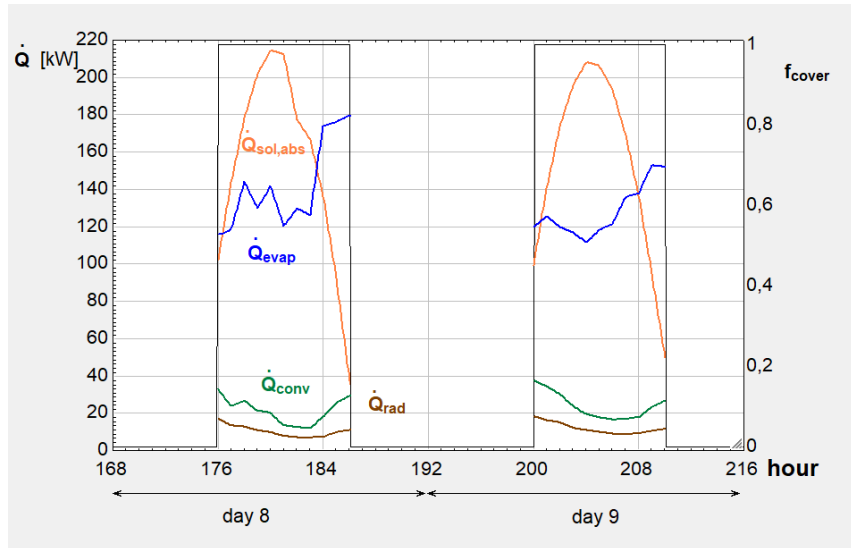


Figure 3.6.9 –Evolution of pool absorbed solar radiation and pool heat losses during two days.

It is also evident that evaporation losses represent the highest share of losses. To estimate the total loss of liquid water we may add to the Equations Window an accumulating variable, summing all the evaporation flowrates over time, when the cover is removed:

$$M_{p, \text{evap}} = M_{p, \text{evap}, \text{old}} + f_{\text{cover}} * \dot{Q}_{\text{evap}} / \Delta h_{lv} * \Delta t \quad (3.6.12)$$

Adding this variable to the Equations Window and Parametric Table, and defining $M_{p, \text{evap}, \text{old}}$ as the previous run value (starting at zero), the result for the 25 simulated days is that 9.4% of the initial 500 tons of water evaporate, and should be compensated by introducing fresh water. The impact on the total energy balance is not very significant: water evaporation is compensated by adding fresh water, at the mains temperature, which represents an energy input; however, its enthalpy is much lower than the enthalpy of vaporisation, and therefore negligible.

4 Distributed and combined modelling examples

This chapter presents several examples of numerical models applied to thermal systems, using a distributed approach. In some cases a distributed model is used for some system components, and a global model for other components, and thus a combination of the two is used. The models are first discussed, and then EES is used as a tool to obtain the solutions and perform parametric/sensitivity analyses. The EES equations/codes are also presented. Most of the examples are related to dynamic situations, with temperatures and other properties varying along time.

As remarked in chapter 2, in the UK/USA system the dot is used as a decimal point, while in most European countries a decimal comma is used. In this book, the decimal comma is used, and this affects the use/appearance of some instructions and functions, when compared with the EES software manual: as a comma replaces the dot, the sign for semicolon (;) is used to replace the comma.

4.1 Steady-state conduction, convection and radiation in a rod

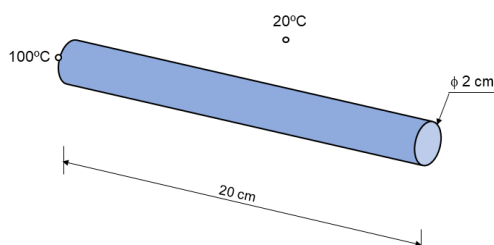


Figure 4.1.1 – Rod and dimensions.

An horizontal steel rod ($k=15$ W/mK) with the dimensions shown in the picture has one extremity kept at 100°C . The rod surface transfers heat to the surrounding air at 20°C by free convection, and exchanges radiation with indoor surfaces, which are kept at 20°C . The rod surface may be assumed as gray and diffuse, with $\alpha=\varepsilon=0.8$.

Assuming the rod temperature only varies along its length, calculate the temperature with the finite volume method with $\Delta x=1$ cm. Also calculate the dissipated heat.

Assuming the temperature variation along the rod radius is negligible, due to its small diameter and good conductivity, the steady-state temperature distribution is only a function of the length coordinate (x). The problem would have an analytical solution if the outside heat transfer coefficient was constant along x . However, both free convection and thermal radiation lead to a variable coefficient: the free convection coefficient depends on the temperature difference between the rod and outside air, and radiation exchanges depend on temperatures to the power of 4. Therefore, a numerical solution is needed to obtain $T(x)$.

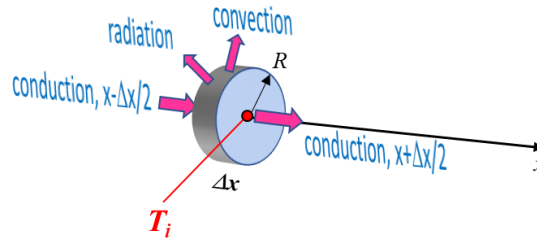


Figure 4.1.2 – Finite volume and its heat exchanges.

Using the finite volumes method, the equation for an internal volume takes into account all heat transfer fluxes across its boundaries – see Figure 6.1.2. The energy balance is given by the equation

$$\frac{k}{\Delta x} A_s (T_{i-1} - T_i) = \frac{k}{\Delta x} A_s (T_i - T_{i+1}) + h_i P \Delta x (T_i - T_{ext}) + \varepsilon \sigma P \Delta x (T_i^4 - T_{ext}^4) \quad (4.1.1)$$

where A_s is the conduction (section) area and $P \Delta x$ the convection and radiation (surface) area (P is the perimeter of the section, equal to $2\pi R$).

The above equation is repeated for all internal volumes, that is, for $i = 2$ until 20. The first and last volumes have special conditions. For the first volume/node ($i = 1$) the temperature is imposed at 100°C . In the last volume/node ($i = 21$) there is conduction with the previous volume and also convection and radiation exchanged by the top circular surface. The equation for this volume/node, with a length equal to $\Delta x/2$, is:

$$\begin{aligned} \frac{k}{\Delta x} A_s (T_{20} - T_{21}) = h_i P \frac{\Delta x}{2} (T_{21} - T_{ext}) + \varepsilon \sigma P \frac{\Delta x}{2} (T_{21}^4 - T_{ext}^4) + \\ + h_{top} A_s (T_{21} - T_{ext}) + \varepsilon \sigma A_s (T_{21}^4 - T_{ext}^4) \end{aligned} \quad (4.1.2)$$

When implementing the set of 21 equations to calculate the 21 temperatures, the *Duplicate* instruction was used to repeat equation (4.1.1), from $i = 2$ to 20. The EES heat transfer correlation database was used to calculate the 21 values of h_i , as a function of $(T_i - T_{ext})$. For that, an horizontal cylinder geometry exposed to quiet air was assumed, with varying surface temperature, that is, neglecting the influence that neighbour boundary layers of the free convection flows might have.

Figure 4.1.3 shows the Equation Windows that defines the problem conditions and set of equations. Array variables were used for the x coordinate ($x[i]$), the temperature ($T[i]$), and the convection coefficient ($h[i]$). For the convection coefficient on the top surface (h_{top} , at $x=L$) a different correlation from the literature was used, [7], as the vertical disk geometry is not available in the EES database. Figure 4.1.4 presents the Formatted Equations.

Other array variables used were $\dot{Q}_{conv}[i]$, $\dot{Q}_{rad}[i]$ and $\dot{Q}[i]$, to calculate the convective, radiative, and total heat transfer rates of each volume. Then, using the SUM function, the total heat transfer rates in all volumes were calculated ($\dot{Q}_{conv,total}$, $\dot{Q}_{rad,total}$ and \dot{Q}_{total})

Figure 4.1.5 presents the results in the Arrays Table, for the 21 nodes considered. Figure 4.1.6 presents the temperature distribution and heat transfer coefficients, while Figure 4.1.7 presents the distribution of heat transfer rates. The temperature varies significantly from 100°C down to 32.58°C at the top. Note that the convective coefficient at the top circular surface is significantly higher, even with a lower temperature. Concerning the heat rates, they follow the temperature variation, with the lower values in the first volume/node due to its smaller size ($\Delta x/2$).

```

Equations Window
"rod in steady-state with surface free convection and radiation"
L=0.2
k=15
epsilon=0.8
D=0.02 "diameter 2 cm"
T_ext=20
A_s=pi*D^2/4
P=pi*D
T1=100

N=21
DELTAx=L/(N-1)
x[1]=0

Duplicate i=1; N
Call fc_horizontal_cylinder(Air; T[i]; T_ext; 100; D; h[i]; Nusselt[i]; Ra[i])
End

Q_dot_conv[1]=h[1]*P*DELTAx/2*(T1-T_ext)
Q_dot_rad[1]=epsilon*sigma#P*DELTAx/2*((T1)+273.15)^4-(T_ext+273.15)^4
Q_dot[1]=Q_dot_conv[1]+Q_dot_rad[1]

Duplicate i=2; N-1
x[i]=(i-1)*DELTAx
k/DELTAx*A_s*(T[i]-T[i-1])=k/DELTAx*A_s*(T[i]-T[i+1])+h[i]*P*DELTAx*(T[i]-T_ext)+epsilon*sigma#P*DELTAx*((T[i]+273.15)^4-(T_ext+273.15)^4)
Q_dot_conv[i]=h[i]*P*DELTAx*(T[i]-T_ext)
Q_dot_rad[i]=epsilon*sigma#P*DELTAx*((T[i]+273.15)^4-(T_ext+273.15)^4)
Q_dot[i]=Q_dot_conv[i]+Q_dot_rad[i]
End

x[N]=(N-1)*DELTAx
k/DELTAx*A_s*(T[N]-T[N-1])=(h[N]*P*DELTAx/2+h_top*A_s)*(T[N]-T_ext)+epsilon*sigma#(P*DELTAx/2+A_s)*((T[N]+273.15)^4-(T_ext+273.15)^4)
Q_dot_conv[N]=(h[N]*P*DELTAx/2+h_top*A_s)*(T[N]-T_ext) "different h in perimeter and circular top surface"
Q_dot_rad[N]=epsilon*sigma#(P*DELTAx/2+A_s)*((T[N]+273.15)^4-(T_ext+273.15)^4)
Q_dot[N]=Q_dot_conv[N]+Q_dot_rad[N]

Q_dot_total=sum(Q_dot[i];i=1;N)
Q_dot_conv_total=sum(Q_dot_conv[i];i=1;N)
Q_dot_rad_total=sum(Q_dot_rad[i];i=1;N)
h_top=Nusselt_top*k_air/D
k_air=conductivity(Air,T=20)
Nusselt_top=1.759*Ra[N]^0.15 "for the top (disk) Ra also defined with D, therefore equal to the previous Ra, because of the Ra value, the guess for T[N] was set at 30"

```

Figure 4.1.3 – Equations Window for the rod heat transfer example.

```

Formatted Equations
rod in steady-state with surface free convection and radiation

L = 0.2
k = 15
ε = 0.8
D = 0.02 diameter 2 cm
T_ext = 20

A_s = π · D² / 4
P = π · D
T_1 = 100
N = 21
Δx = L / (N - 1)
x_1 = 0

Call fc_horizontal_cylinder(AIR; T_i; T_ext; 100; D; h_i) (for i = 1 to N)

Q_dot_conv,1 = h_1 · P · Δx / 2 · (T_1 - T_ext)
Q_dot_rad,1 = ε · 5.670E-08 [W/m²·K⁴] · P · Δx / 2 · ((T_1 + 273.15)⁴ - (T_ext + 273.15)⁴)
Q_dot_1 = Q_dot_conv,1 + Q_dot_rad,1
x_i = (i - 1) · Δx (for i = 2 to N-1)

k / Δx · A_s · (T_{i-1} - T_i) = k / Δx · A_s · (T_i - T_{i+1}) + h_i · P · Δx · (T_i - T_ext) + ε · 5.670E-08 [W/m²·K⁴] · P · Δx · ((T_i + 273.15)⁴ - (T_ext + 273.15)⁴) (for i = 2 to N-1)
Q_dot_conv,i = h_i · P · Δx · (T_i - T_ext) (for i = 2 to N-1)
Q_dot_rad,i = ε · 5.670E-08 [W/m²·K⁴] · P · Δx · ((T_i + 273.15)⁴ - (T_ext + 273.15)⁴) (for i = 2 to N-1)
Q_dot_i = Q_dot_conv,i + Q_dot_rad,i (for i = 2 to N-1)
x_{21} = (N - 1) · Δx

k / Δx · A_s · (T_{20} - T_{21}) = [h_{21} · P · Δx / 2 + h_top · A_s] · (T_{21} - T_ext) + ε · 5.670E-08 [W/m²·K⁴] · [P · Δx / 2 + A_s] · ((T_{21} + 273.15)⁴ - (T_ext + 273.15)⁴)
Q_dot_conv,21 = [h_{21} · P · Δx / 2 + h_top · A_s] · (T_{21} - T_ext) different h in perimeter and circular top surface
Q_dot_rad,21 = ε · 5.670E-08 [W/m²·K⁴] · [P · Δx / 2 + A_s] · ((T_{21} + 273.15)⁴ - (T_ext + 273.15)⁴)
Q_dot_{21} = Q_dot_conv,21 + Q_dot_rad,21
Q_dot_tot = ∑_{i=1}^N (Q_dot_i)
Q_dot_conv,tot = ∑_{i=1}^N (Q_dot_conv,i)
Q_dot_rad,tot = ∑_{i=1}^N (Q_dot_rad,i)

h_top = Nusselt_top · k_air / D
k_air = k (AIR; T = 20)
Nusselt_top = 1.759 · Ra_{21}^{0.15} "for the top (disk) Ra also defined with D, therefore equal to the previous Ra, because of the Ra value, the guess for T_{21} was set at 30"

```

Figure 4.1.4 – Formatted Equations Window for the rod heat transfer example.

Sort	x_i [m]	h_i [W/m ² K]	T_i [°C]	Nusselt _i	Ra _i	$\dot{Q}_{conv,i}$ [W]	$\dot{Q}_{rad,i}$ [W]	\dot{Q}_i [W]
[1]	0	8,461	100	6,026	36777	0,2126	0,1711	0,3837
[2]	0,01	8,221	90,1	5,931	34511	0,3621	0,2858	0,6479
[3]	0,02	7,989	81,58	5,829	32185	0,3091	0,2408	0,5499
[4]	0,03	7,765	74,22	5,722	29863	0,2645	0,2045	0,469
[5]	0,04	7,549	67,85	5,611	27594	0,227	0,1749	0,4019
[6]	0,05	7,341	62,34	5,498	25415	0,1953	0,1506	0,3459
[7]	0,06	7,142	57,57	5,384	23353	0,1686	0,1305	0,2991
[8]	0,07	6,952	53,43	5,271	21425	0,146	0,1137	0,2597
[9]	0,08	6,772	49,84	5,16	19641	0,127	0,09969	0,2266
[10]	0,09	6,601	46,73	5,052	18009	0,1109	0,08792	0,1988
[11]	0,1	6,44	44,04	4,948	16529	0,09728	0,07802	0,1753
[12]	0,11	6,29	41,73	4,848	15201	0,08586	0,06969	0,1555
[13]	0,12	6,151	39,74	4,755	14022	0,07629	0,06269	0,139
[14]	0,13	6,024	38,05	4,668	12987	0,06832	0,05683	0,1252
[15]	0,14	5,909	36,63	4,589	12093	0,06173	0,05197	0,1137
[16]	0,15	5,808	35,44	4,518	11334	0,05636	0,04798	0,1043
[17]	0,16	5,722	34,48	4,457	10707	0,05206	0,04478	0,09684
[18]	0,17	5,651	33,72	4,407	10207	0,04873	0,04228	0,09101
[19]	0,18	5,596	33,16	4,367	9831	0,04628	0,04043	0,0867
[20]	0,19	5,558	32,78	4,34	9575	0,04464	0,03918	0,08383
[21]	0,2	5,538	32,58	4,326	9439	0,05638	0,03853	0,0949

Figure 4.1.5 – Arrays Table for the rod heat transfer example.

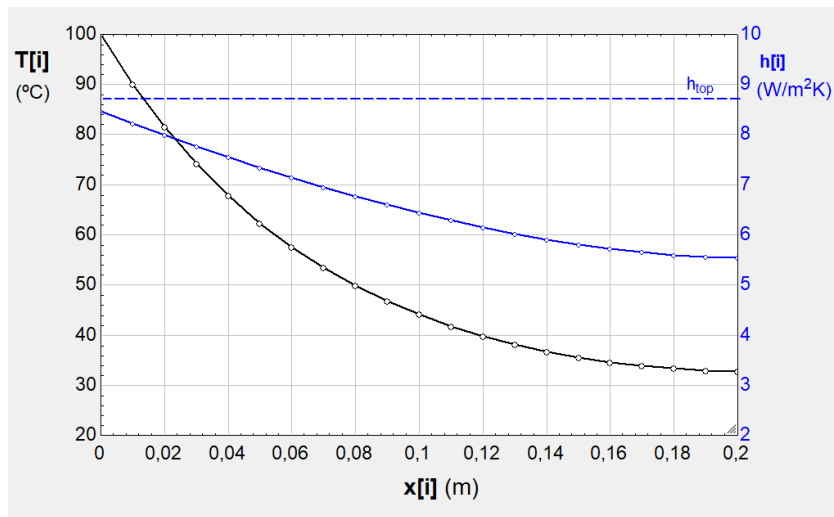


Figure 4.1.6 – Temperature distribution and convective coefficients for the rod heat transfer example.

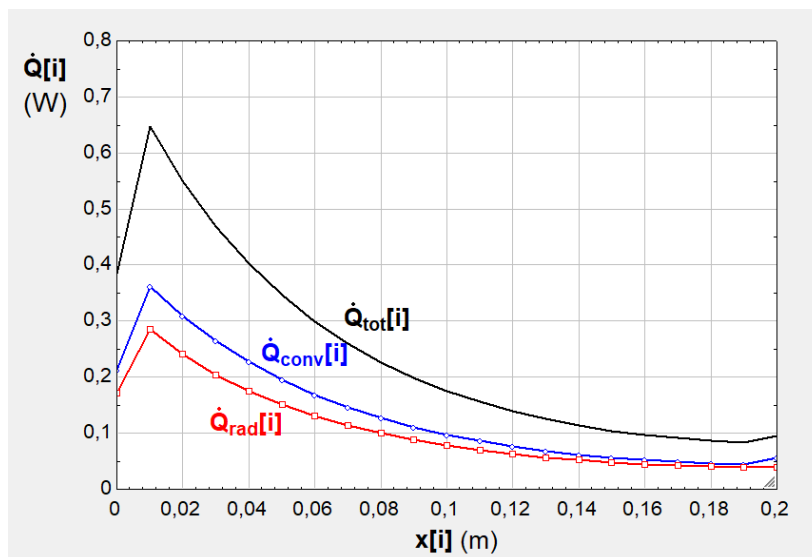


Figure 4.1.7 – Heat transfer rates for the rod heat transfer example.

4.2 Dynamic heat transfer in a Trombe wall

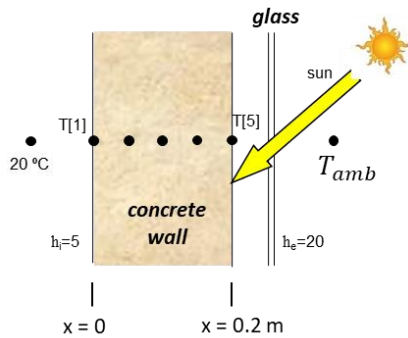


Figure 4.2.1 – Trombe wall.

The figure represents a wall that also acts as a solar collector (Trombe wall). Solar radiation reaches the outside surface, protected by a glass pane with a transmittance of 0.85 and an absorptance of 0.1 for solar radiation wavelengths. The glass is opaque to longwave radiation, and has an emissivity of 0.8 (equal to the absorption coefficient for longer wave radiation). The concrete wall has an absorptance and an emissivity equal to 0.9 (all wavelengths). The distance between the glass pane and the wall is equal to 10 cm (filled with air), and the wall thickness and height are 0.2 and 2.5 m.

The indoor temperature is kept at 20°C, and the indoor and external convective heat transfer coefficients are 5 and 20 W/m²°C, respectively.

During a Winter day, the incident solar radiation on a vertical surface and the outdoor ambient temperature vary according to the table below.

<i>hour</i>	1	2	3	4	5	6	7	8	9	10	11	12
T_{amb} (°C)	10.7	10.5	10.4	10.2	10.2	10.0	9.9	10.3	11.4	12.5	13.4	14.1
I_{sol} (W/m ²)	0	0	0	0	0	22	50	75	158	237	300	325

<i>hour</i>	13	14	15	16	17	18	19	20	21	22	23	24/0
T_{amb} (°C)	14.7	15.2	15.2	15.3	14.7	14.0	12.8	12.2	11.7	11.3	11.1	10.8
I_{sol} (W/m ²)	316	276	187	73	55	43	20	0	0	0	0	0

Considering an initial wall temperature distribution to be defined, obtain the wall temperature evolution during one day, by discretising the wall with the equally spaced 5 volumes/nodes represented in the figure. Calculate the variation of the heat transfer rate to the indoor space. Analyse the effect of varying the time step and number of nodes on the results.

A distributed model will be applied to the wall, modelling the heat transfer along its thickness. The glass pane will be considered at a uniform temperature (global approach). All temperatures will vary along time (dynamic situation).

As seen in section 1.3.2, using the implicit formulation, the discretised equation for the internal wall volumes (2 to 4) is

$$\rho c_p \Delta x \frac{(T_i^{t+\Delta t} - T_i^t)}{\Delta t} = \frac{k}{\Delta x} (T_{i+1}^{t+\Delta t} + T_{i-1}^{t+\Delta t} - 2T_i^{t+\Delta t}) \quad (4.2.1)$$

while for $i = 1$ we will have conduction and convection with indoor air:

$$\rho c_p \frac{\Delta x}{2} \frac{(T_1^{t+\Delta t} - T_1^t)}{\Delta t} = h_i (T_{int} - T_1^{t+\Delta t}) + \frac{k}{\Delta x} (T_2^{t+\Delta t} - T_1^{t+\Delta t}) \quad (4.2.2)$$

For the wall external volume ($i = 5$) we need to consider the transfer by conduction, the free convection in the air gap between the wall and the glass (rectangular enclosure), the longwave

thermal radiation balance between those two surfaces, and the solar radiation absorbed by the wall (transmitted through the glass):

$$\rho c_p \frac{\Delta x}{2} \frac{(T_5^{t+\Delta t} - T_5^t)}{\Delta t} = \frac{k}{\Delta x} (T_4^{t+\Delta t} - T_5^{t+\Delta t}) - h_{gw}^{t+\Delta t} (T_5^{t+\Delta t} - T_g^{t+\Delta t}) - \frac{\sigma (T_5^{t+\Delta t^4} - T_g^{t+\Delta t^4})}{\frac{1}{\epsilon_w} + \frac{1}{\epsilon_g} - 1} + \alpha_w \tau_g I_{sol}^{t+\Delta t} \quad (4.2.3)$$

For the glass energy balance, its thermal inertia is neglected (small thickness), and therefore we have a quasi-steady state equation:

$$h_{gw}^{t+\Delta t} (T_5^{t+\Delta t} - T_g^{t+\Delta t}) + \frac{\sigma (T_5^{t+\Delta t^4} - T_g^{t+\Delta t^4})}{\frac{1}{\epsilon_w} + \frac{1}{\epsilon_g} - 1} + \alpha_g I_{sol}^{t+\Delta t} = h_e (T_g^{t+\Delta t} - T_{amb}^{t+\Delta t}) + \epsilon_g \sigma (T_g^{t+\Delta t^4} - T_{amb}^{t+\Delta t^4}) \quad (4.2.4)$$

Starting with an initial temperature distribution, equations (4.2.1) – representing 3 equations – to (4.2.4) allow the calculation of the 5 wall temperatures and glass temperature, for all instants ($t + \Delta t$). The wall-glass enclosure convective coefficient (h_{gw}) is a function of the temperature difference ($T_5 - T_g$), and will also be calculated along time.

Figure 4.2.2 presents the Equations Window and Figure 4.2.3 the Formatted Equations Window. Besides the dimensions and properties, and the previous equations, the climatic variables (ambient temperature and solar radiation) for each time are interpolated from the hourly values introduced in the Lookup Table (“Lookup 1” – see Figure 4.2.4). The Duplicate instruction is used to write the equations for the 3 internal nodes. The EES Procedure Tilted_Rect_Enclosure, from its heat transfer database, is used to calculate h_{gw} , as a function of ($T_5 - T_g$). The heat flux exchanged between the wall and the indoor space is also calculated (q_{dot_int} , or \dot{q}_{int}). A Parametric Table (“Table 1”) is created with several rows, each for a given instant of time (step DELTAt), and the previous temperature values are read from the previous row (row number “row-1”).

```

" Trombe Wall
N=5 "nodes along wall thickness"
L=0.20 "wall thickness"
DELTAX=L/(N+1)
x[1]=0
duplicate i=2:N
  x[i]=x[i-1]+DELTAX
end
k=k_(Concrete_stone_mix; 26)
c_p=c_(Concrete_stone_mix; 26)*1000 "in J/kgK"
rho=rho_(Concrete_stone_mix; 26)
alpha=0.9 "wall solar absorptivity"
epsilon=0.9 "wall emissivity"
tau_g=0.85 "glass transmissivity, constant value - approximation"
alpha_g=0.1 "glass solar absorptivity"
epsilon_g=0.8 "glass emissivity - long wavelength"
T_int=20
h_i=5
h_e=20

DELTAt=60
row=1+time/DELTAt
hour=time/3600
I_sol=Interpolate1(Lookup 1;"Vertical Rad";hour;hour=time/3600)
T_amb=Interpolate1(Lookup 1;"External Temp";hour;hour=time/3600)

"Energy balance of internal nodes: nodes 2 to N-1"
duplicate i=2,N-1
  rho*c_p*DELTAX*(T[i]-TableValue(Table 1;row-1;#T[i]))/DELTAt=k*DELTAX*(T[i-1]+T[i+1]-2*T[i])
end
"Boundary conditions: nodes 1 and N"
rho*c_p*DELTAX/2*(T[1]-TableValue(Table 1;row-1;#T[1]))/DELTAt=h_i*(T_int-T[1])-k*DELTAX*(T[1]-T[2]) "internal wall surface"
q_dot_int=h_i*(T[1]-T_int) "internal heat flux, W/m2"
tau_g*alpha*I_sol+k*DELTAX*(T[N]-T_g)-h_gw*(T[N]-T_g)-(sigma*(T[N]+273.15)^4*(T_g+273.15)^4)/(1/epsilon_w+1/epsilon_g-1)=rho*c_p*DELTAX/2*(T[N]-TableValue(Table 1;row-1;#T[N]))/DELTAt "ext wall surface"
Call Tilted_Rect_Enclosure(Air; T[5]; T_g; 100;2.5; 0;10; 90 : h_gw; Nusselt; Ra) "here T[5] has to be specified"
"Glass energy balance - negligible inertia"
alpha_g*I_sol+h_gw*(T[N]-T_g)+(sigma*(T[N]+273.15)^4*(T_g+273.15)^4)/(1/epsilon_w+1/epsilon_g-1)=epsilon_g*sigma*(T_g+273.15)^4*(T_amb+273.15)^4+h_e*(T_g-T_amb)

```

Figure 4.2.2 –Equations Window for the Trombe wall example.

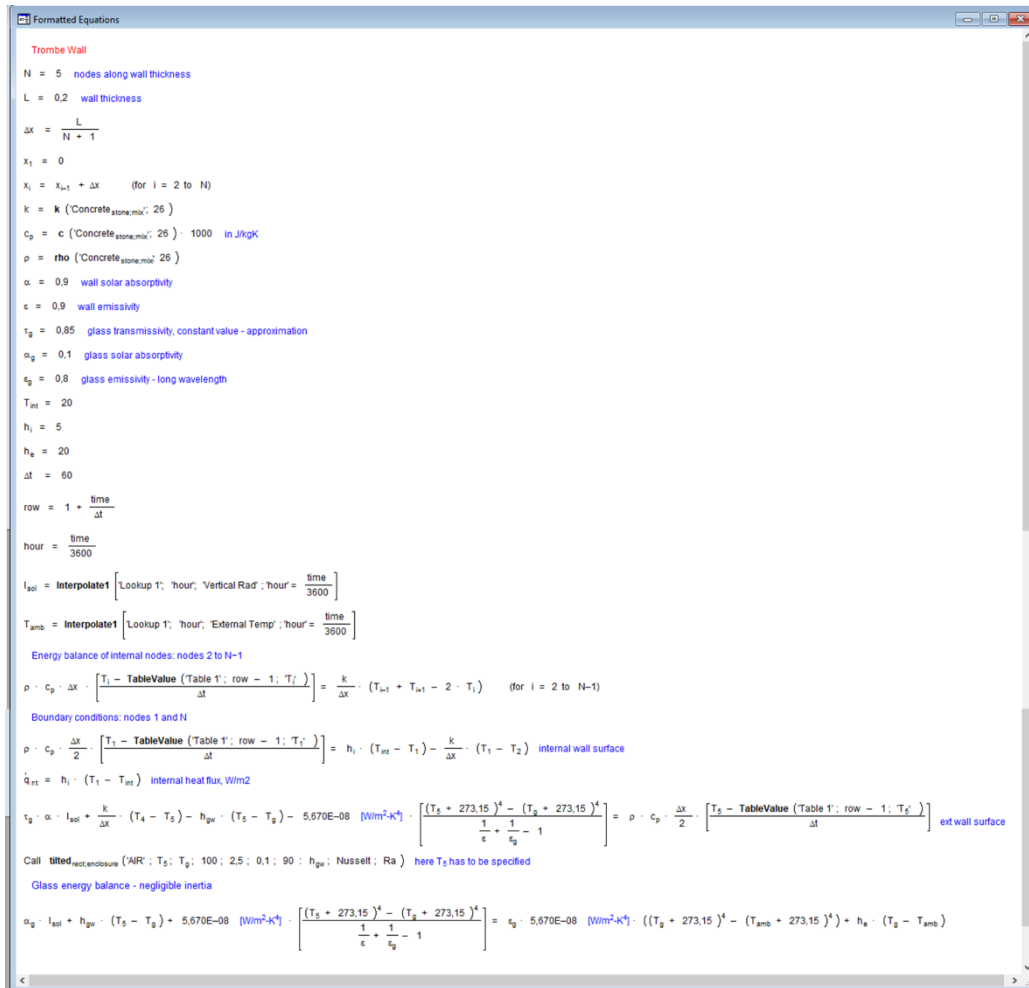


Figure 4.2.3 – Formatted Equations Window for the Trombe wall example.

hour	Vertical Rad [W/m2]	External Temp [°C]
0	0	10,7
1	0	10,5
2	0	10,4
3	0	10,2
4	0	10,2
5	0	10
6	22	9,9
7	50	10,3
8	75	10,4
9	158	12,5
10	237	13,4
11	300	14,1
12	325	14,7
13	316	15,2
14	276	15,2
15	187	15,3
16	73	14,7
17	55	14
18	43	12,8
19	20	12,2
20	0	11,7
21	0	11,3
22	0	11,1
23	0	10,8
24	0	10,7

Figure 4.2.4 – Lookup Table with hourly climatic data for the Trombe wall example.

Figure 4.2.5 shows the first rows of “Table 1”. For a Δt of 60 s, a total of 1441 rows were created (24 hour simulation). The initial (first row) wall temperature values were obtained after a couple of simulations starting with a fixed initial value of 22°C; they represent a daily cycle, and are equal to the temperatures at the end of the day.

Run	time [s]	hour	T_{amb} [°C]	I_{sol} [W/m ²]	T_1 [°C]	T_2 [°C]	T_3 [°C]	T_4 [°C]	T_5 [°C]	T_g [°C]	h_{gw} [W/m ² K]	q_{int} [W/m ²]
Run 1	0				22.48	22.62	22.42	21.87	20.97			
Run 2	60	0,01667	10,7	0	22,47	22,61	22,41	21,86	20,96	12,54	1,232	12,35
Run 3	120	0,03333	10,69	0	22,46	22,6	22,39	21,84	20,95	12,53	1,232	12,29
Run 4	180	0,05	10,69	0	22,45	22,58	22,38	21,83	20,93	12,53	1,231	12,23
Run 5	240	0,06667	10,69	0	22,43	22,57	22,37	21,82	20,92	12,52	1,231	12,17
Run 6	300	0,08333	10,68	0	22,42	22,56	22,36	21,8	20,91	12,52	1,231	12,11
Run 7	360	0,1	10,68	0	22,41	22,55	22,34	21,79	20,9	12,51	1,231	12,06
Run 8	420	0,11667	10,68	0	22,4	22,53	22,33	21,78	20,88	12,51	1,23	12
Run 9	480	0,13333	10,67	0	22,39	22,52	22,32	21,77	20,87	12,5	1,23	11,94
Run 10	540	0,15	10,67	0	22,38	22,51	22,3	21,75	20,86	12,5	1,23	11,89
Run 11	600	0,16667	10,67	0	22,37	22,5	22,29	21,74	20,85	12,49	1,23	11,83
Run 12	660	0,18333	10,66	0	22,35	22,48	22,28	21,73	20,84	12,49	1,229	11,77
Run 13	720	0,2	10,66	0	22,34	22,47	22,27	21,72	20,82	12,48	1,229	11,72
Run 14	780	0,21667	10,66	0	22,33	22,46	22,25	21,7	20,81	12,48	1,229	11,66
Run 15	840	0,23333	10,65	0	22,32	22,45	22,24	21,69	20,8	12,47	1,229	11,6
Run 16	900	0,25	10,65	0	22,31	22,43	22,23	21,68	20,79	12,47	1,228	11,55
Run 17	960	0,26667	10,65	0	22,3	22,42	22,21	21,66	20,78	12,46	1,228	11,49
Run 18	1020	0,28333	10,64	0	22,29	22,41	22,2	21,65	20,77	12,46	1,228	11,43
Run 19	1080	0,3	10,64	0	22,28	22,4	22,19	21,64	20,75	12,45	1,228	11,38
Run 20	1140	0,31667	10,64	0	22,26	22,38	22,18	21,63	20,74	12,45	1,227	11,32
Run 21	1200	0,33333	10,63	0	22,25	22,37	22,16	21,61	20,73	12,44	1,227	11,27
Run 22	1260	0,35	10,63	0	22,24	22,36	22,15	21,6	20,72	12,44	1,227	11,21
Run 23	1320	0,36667	10,63	0	22,23	22,35	22,14	21,59	20,71	12,43	1,227	11,15
Run 24	1380	0,38333	10,62	0	22,22	22,34	22,12	21,58	20,69	12,43	1,226	11,1
Run 25	1440	0,4	10,62	0	22,21	22,32	22,11	21,56	20,68	12,42	1,226	11,04

Figure 4.2.5 – First rows for Parametric Table “Table 1” in the Trombe wall example.

Figure 4.2.6 shows a graph with the evolution of wall temperatures (internal and external surfaces) and glass temperature. Note the higher temperature swing of the external wall surface, which reaches about 34°C, and that at night the external surface becomes colder than the indoor surface. The glass temperature closely follows the outdoor ambient temperature. Figure 4.2.7 shows the evolution of the h_{gw} free convection coefficient.

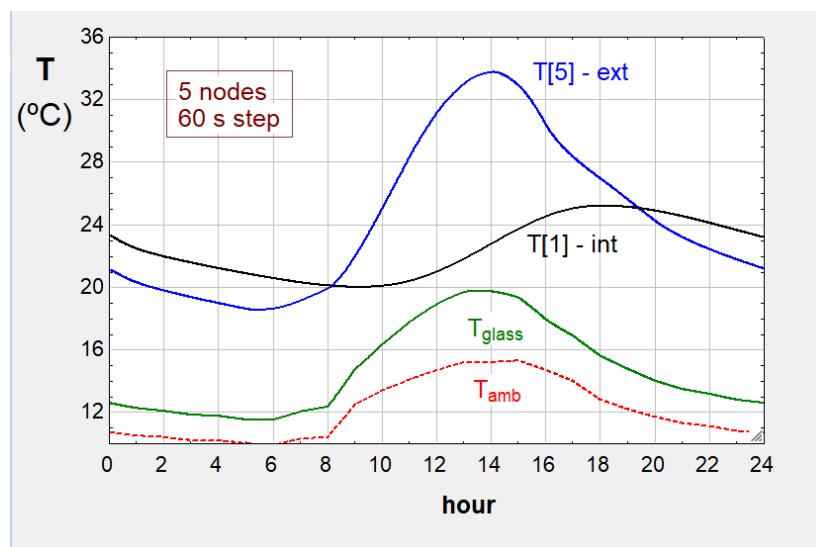


Figure 4.2.6 – Time evolution of several temperatures in the Trombe wall example.

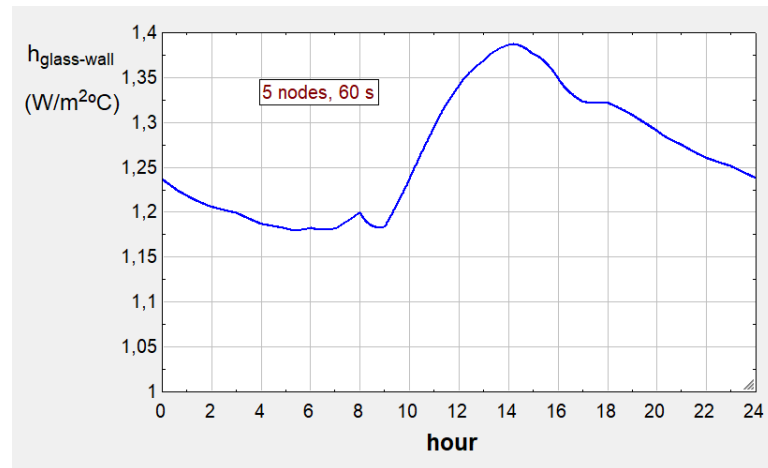


Figure 4.2.7 – Evolution of the free convection coefficient (wall-glass enclosure).

The effect of the time step, and also number of volumes/nodes, were analysed by increasing Δt five times to 300 s, and decreasing the number of nodes to only 3. Figure 4.2.8 shows the effect in the wall temperature profiles. The larger time step leads to a slight time delay in following the temperature changes, especially in the external wall surface, but differences are not very significant. Even the use of only 3 nodes leads to acceptable results. In this case, there is no practical interest in increasing the number of nodes. That is the reason why some software tools dedicated to building envelope thermal simulation only use 3 nodes in each wall.

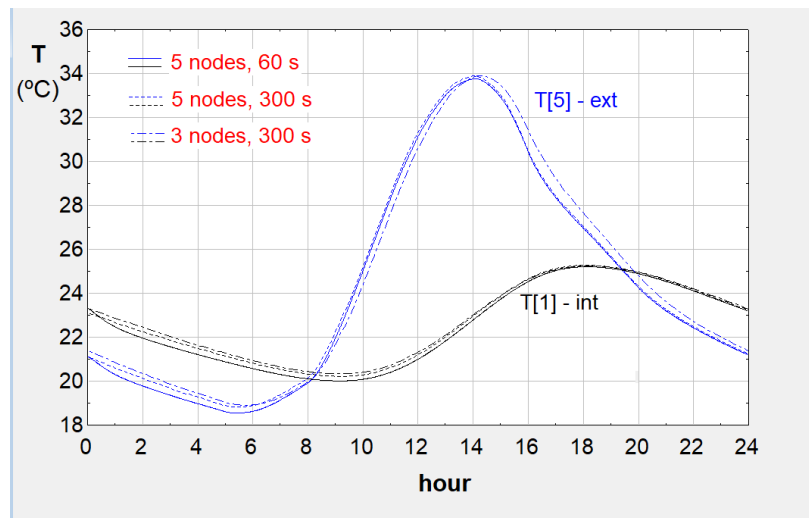


Figure 4.2.8 – Wall temperature profiles (internal and external surfaces) for different time steps and number of nodes.

Figure 4.2.9 analyses the evolution of the heat flux exchanged between the wall and the indoor space. It is always positive, that is, the wall heats the indoor space even with lower outside ambient temperatures. It is noticeable that the maximum flux occurs at the end of the evening, several hours after the maximum temperature on the wall external surface. This is the consequence of the wall thermal inertia. Therefore, the use of a Trombe wall is adequate in indoor spaces used in the evenings and early night periods. Again, the differences when using only 3 nodes and a larger time step of 300 seconds are small, and more significant during the early hours of the day.

Figure 4.2.9 also shows a comparison with the indoor heat flux for a “normal” wall, without the outside glazed surface. In that case, the heat flux would always be negative, meaning that the wall and the indoor space would lose heat throughout the day to the outside ambient air.

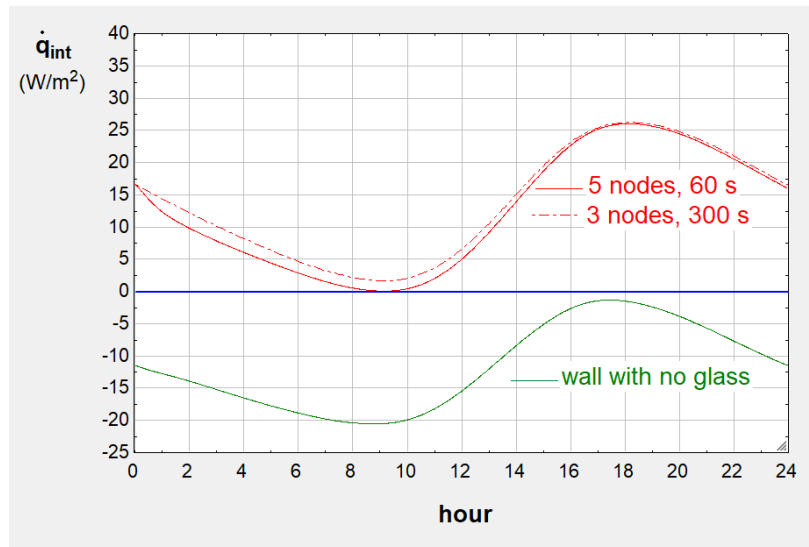


Figure 4.2.9 – Evolution of the heat flux to the indoor space for different discretisations and a non-glazed wall.

The wall thermal performance can be improved with the use of a selective coating on its outside surface. In this example the external wall surface had an emissivity for longwave radiation equal to its solar absorptance (0.9). The use of a selective coating may reduce the emissivity associated with the loss of thermal radiation to much lower values (<0.1), while maintaining the same solar absorptance, increasing the wall temperatures and indoor heat flux.

4.3 Dynamic heat transfer in a ventilated Trombe wall

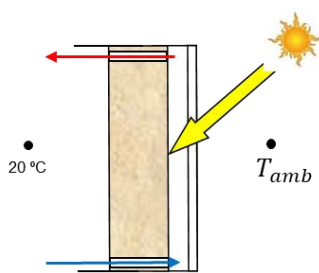


Figure 4.3.1 – Ventilated Trombe wall.

Consider a ventilated Trombe wall, as shown in the picture. The indoor air circulates by thermosyphon effect in the space between a glazing and a concrete wall, in a rectangular channel with a thickness of 10 cm. The concrete wall has a thickness of 20 cm, a width of 3 m and a height (between air inlet/outlet) of 2 m. Indoor air is always at 20°C, with an indoor heat transfer coefficient of 5 W/m²K.

The outside glazing is a double pane glazing, with a global transmission coefficient for solar radiation of 0.72 (constant), and with an overall heat transfer coefficient from the internal pane to the outside (including radiation and external convection) of 2.2 W/m²K. The glazing solar absorptivity is equal to 0.1 and its emissivity for longwave radiation is 0.8 (equal to the absorption coefficient); the glazing is opaque to longwave radiation. The concrete wall has an absorptance and an emissivity equal to 0.9 (all wavelengths).

Using an EES model, obtain the wall temperature evolution during a Winter day (the same as defined in example 4.2), by discretising the wall with 5 equally spaced nodes along its thickness, and also the heat transfer rate from the wall to the indoor space. Use a global model for the air

flow in the ventilated space, considering only inlet and outlet temperatures; for that flow, use friction losses only in the channel between wall and glass, and assume an overall local pressure loss coefficient of 3 (referred to the channel velocity).

The concrete wall will be modelled by assuming its temperature varies only along its thickness. This is a simplification, as in reality the temperature also changes in the vertical direction, as the circulating air temperature also changes. For the air flow, a global model approach will be followed, considering in each flow section the mixed mean (average) flow temperature. And the heat exchange between the air and channel surfaces (wall and glass) will be treated by considering the efficiency of heat transfer (and convection coefficient), without the need of obtaining the temperature evolution in the flow direction; this means that the average outlet air temperature will be related to the inlet temperature, flow rate and surface temperatures. And the air flow rate will be obtained after an hydraulic balance between the pressure drop and the buoyancy effect cause by its heating.

The distributed wall model is similar to the one in the previous section, with equations (4.2.1) and (4.2.2) expressing the temperatures of nodes 1 to 4. However, the equation for the external wall surface (volume/node 5) needs to be modified to take into account the channel air flow. The same happens with the glazing balance (internal pane).

The global approach to the air flow heat transfer follows the scheme in Figure 4.3.2.

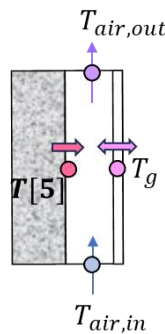


Figure 4.3.2 – Channel flow heat transfer.

Using the efficiencies of the steady-state transfer of heat with the 2 channel surfaces, we may calculate each heat transfer rate as a function of surface temperature and inlet air temperature. For the transfer between wall and air, using the forced convection coefficient (h_{FC}), we may write

$$\dot{Q}_{w-air} = \varepsilon_{w-air} \dot{Q}_{ideal} = \left[1 - \exp\left(-\frac{h_{FC}A_w}{\dot{M}c_{p,air}}\right) \right] \dot{M}c_{p,air}(T[5] - T_{air,in}) \quad (4.3.1)$$

and for the transfer between glass and air, assuming the same convection coefficient in both surfaces, which is a simplification, we may write

$$\dot{Q}_{g-air} = \varepsilon_{g-air} \dot{Q}_{ideal} = \left[1 - \exp\left(-\frac{h_{FC}A_g}{\dot{M}c_{p,air}}\right) \right] \dot{M}c_{p,air}(T_g - T_{air,in}) \quad (4.3.2)$$

Then, the heat rate exchanged by the air with the wall and glass will change the air temperature between inlet and outlet:

$$\dot{Q}_{air} = \dot{Q}_{w-air} + \dot{Q}_{g-air} = \dot{M}c_{p,air}(T_{air,out} - T_{air,in}) \quad (4.3.3)$$

The air flow rate is also unknown, and will be calculated after an hydraulic balance, stating that there is a (quasi-steady state) balance between the air flow pressure drop and the buoyancy effect due to air heating. We may write:

$$\bar{\rho}_{air} g \beta (T_{air,out} - T_{air,in}) \Delta H = \left(\sum_i K_i + f \frac{\Delta H}{D_h} \right) \bar{\rho}_{air} \frac{\bar{v}_{air}^2}{2} \quad (4.3.4)$$

with the average air velocity (\bar{v}) easily related to the flow rate. The sum of local pressure loss coefficients (K_i) was given as equal to 3, and the friction factor (f) will be calculated through the available EES procedure (the same to be used to calculate the heat transfer coefficient).

Equations (4.3.3) and (4.3.4) will be added to the model, to be able to calculate $T_{air,out}$ and \dot{M} , and equations (4.3.1) and (4.3.2) will be used in the equations for node 5 and glass internal pane, which will become:

$$\rho c_p \frac{\Delta x}{2} \frac{(T_5^{t+\Delta t} - T_5^t)}{\Delta t} = \frac{k}{\Delta x} (T_4^{t+\Delta t} - T_5^{t+\Delta t}) - \frac{\dot{Q}_{g-air}^{t+\Delta t}}{A_w} - \frac{\sigma (T_5^{t+\Delta t^4} - T_g^{t+\Delta t^4})}{\frac{1}{\epsilon_w} + \frac{1}{\epsilon_g} - 1} + \alpha_w \tau_g I_{sol}^{t+\Delta t} \quad (4.3.5)$$

and

$$\frac{\sigma (T_5^{t+\Delta t^4} - T_g^{t+\Delta t^4})}{\frac{1}{\epsilon_w} + \frac{1}{\epsilon_g} - 1} + \alpha_g I_{sol}^{t+\Delta t} - \frac{\dot{Q}_{g-air}^{t+\Delta t}}{A_g} = U_g (T_g^{t+\Delta t} - T_{amb}^{t+\Delta t}) \quad (4.3.6)$$

The complete model includes then equations (4.2.1) - 3 equations, (4.2.3), (4.3.3), (4.3.4) (4.3.5) and (4.3.6). A total of 8 equations to calculate $T[1]$ to $T[5]$, T_g , $T_{air,out}$ and \dot{M} . The convective coefficient (h_{FC}) is also required and will be calculated from the EES heat transfer correlation database, assuming there is forced convection in the channel. Actually, this is a situation where, due to the low air velocities, mixed convection may occur. But, by calculating the coefficients for separate forced and free convection, it was found that they have the same order of magnitude; therefore, the mixed convection coefficient will be similar. It will however change over time, due to the change in air flow rates.

Figure 4.3.3 presents the Equations Window and Figure 4.3.4 the Formatted Equations Window associated to this model.

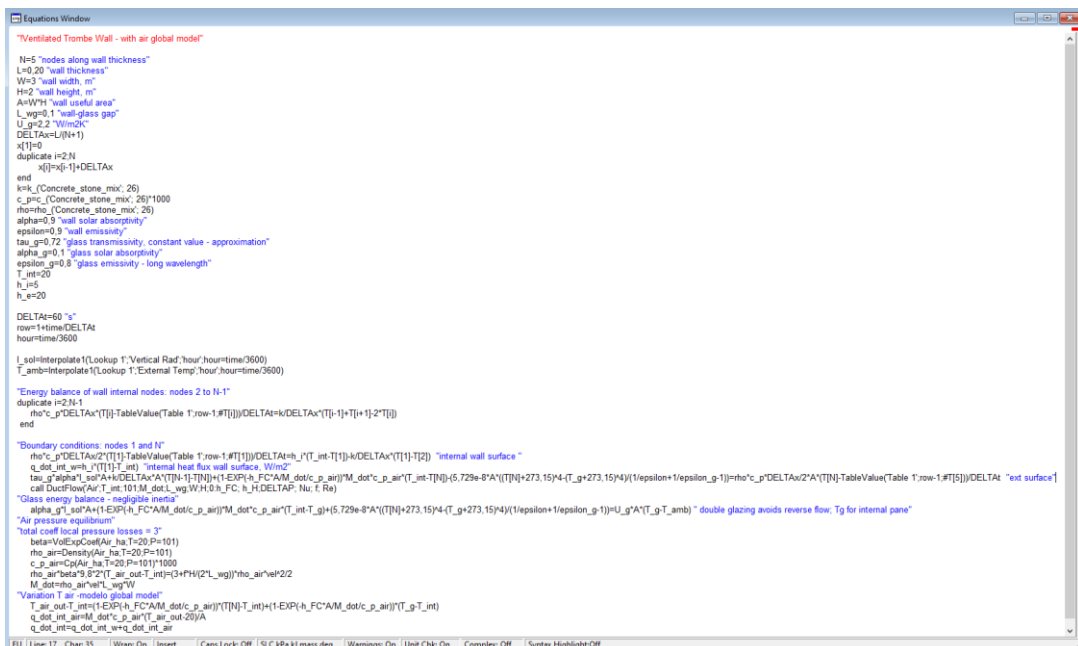


Figure 4.3.3 –Equations Window for the ventilated Trombe wall example.

Formatted Equations

Ventilated Trombe Wall - with air global model

N = 5 nodes along wall thickness

L = 0.2 wall thickness

W = 3 wall width, m

H = 2 wall height, m

A = W · H wall useful area

L_{wg} = 0.1 wall-glass gap

U_g = 2.2 W/m²K

$$\Delta x = \frac{L}{N + 1}$$

x₁ = 0

x_i = x_{i-1} + Δx (for i = 2 to N)

k = k ('Concrete', stone, m2, 26)

c_p = c ('Concrete', stone, m2, 26) · 1000

ρ = rho ('Concrete', stone, m2, 26)

α = 0.9 wall solar absorptivity

ε = 0.9 wall emissivity

T_g = 0.72 glass transmissivity, constant value - approximation

α_g = 0.1 glass solar absorptivity

ε_g = 0.8 glass emissivity - long wavelength

T_{ext} = 20

h_i = 5

h_e = 20

Δt = 60 s

row = 1 + $\frac{\text{time}}{\Delta t}$

hour = $\frac{\text{time}}{3600}$

I_{sol} = Interpolate1 [Lookup 1; 'hour', 'Vertical Rad'; 'hour' = $\frac{\text{time}}{3600}$]

T_{amb} = Interpolate1 [Lookup 1; 'hour', 'External Temp'; 'hour' = $\frac{\text{time}}{3600}$]

Energy balance of wall internal nodes: nodes 2 to N-1

$$\rho \cdot c_p \cdot \Delta x \cdot \left[\frac{T_i - \text{TableValue}(\text{Table 1}; \text{row} - 1; T_i)}{\Delta t} \right] = \frac{k}{\Delta x} \cdot (T_{i+1} + T_{i-1} - 2 \cdot T_i) \quad (\text{for } i = 2 \text{ to } N-1)$$

Boundary conditions: nodes 1 and N

$$\rho \cdot c_p \cdot \frac{\Delta x}{2} \cdot \left[\frac{T_1 - \text{TableValue}(\text{Table 1}; \text{row} - 1; T_1)}{\Delta t} \right] = h_i \cdot (T_{int} - T_1) - \frac{k}{\Delta x} \cdot (T_1 - T_2) \quad \text{internal wall surface}$$

$$\dot{q}_{ext,w} = h_e \cdot (T_1 - T_{ext}) \quad \text{internal heat flux wall surface, W/m}^2$$

$$T_2 \cdot \alpha \cdot I_{sol} \cdot A + \frac{k}{\Delta x} \cdot A \cdot (T_4 - T_2) + \left[1 - \exp\left(-h_{FC} \cdot \frac{A}{c_{p,air}}\right) \right] \cdot \dot{M} \cdot c_{p,air} \cdot (T_{int} - T_2) + 5.729 \times 10^{-8} \cdot A \cdot \left[\frac{(T_2 + 273.15)^4 - (T_g + 273.15)^4}{\frac{1}{\epsilon} + \frac{1}{\epsilon_g} - 1} \right] = \rho \cdot c_p \cdot \frac{\Delta x}{2} \cdot A \cdot \left[\frac{T_2 - \text{TableValue}(\text{Table 1}; \text{row} - 1; T_2)}{\Delta t} \right] \quad \text{ext surface}$$

Call ductflow (Air; T_{int}; 101; M; L_{wg}; W; H; 0; h_{FC}; h_u; ΔP; v; f; Re)

Glass energy balance - negligible inertia

$$\alpha_g \cdot I_{sol} \cdot A + \left[1 - \exp\left(-h_{FC} \cdot \frac{A}{c_{p,air}}\right) \right] \cdot \dot{M} \cdot c_{p,air} \cdot (T_{int} - T_g) + 5.729 \times 10^{-8} \cdot A \cdot \left[\frac{(T_g + 273.15)^4 - (T_2 + 273.15)^4}{\frac{1}{\epsilon} + \frac{1}{\epsilon_g} - 1} \right] = U_g \cdot A \cdot (T_g - T_{amb}) \quad \text{double glazing avoids reverse flow, Tg for internal pane}$$

Air pressure equilibrium

total coeff local pressure losses = 3

β = β (Air_{ha}; T = 20; P = 101)

p_{air} = p (Air_{ha}; T = 20; P = 101)

c_{p,air} = Cp (Air_{ha}; T = 20; P = 101) · 1000

$$p_{air} \cdot \beta \cdot 9.8 \cdot 2 \cdot (T_{arc,ext} - T_{int}) = \left[3 + f \cdot \frac{H}{2 \cdot L_{wg}} \right] \cdot \rho_{air} \cdot \frac{\text{Vel}^2}{2}$$

M = ρ_{air} · Vel · L_{wg} · W

Variation T air -modelo global model

$$T_{arc,out} - T_{int} = \left[1 - \exp\left(-h_{FC} \cdot \frac{A}{c_{p,air}}\right) \right] \cdot (T_2 - T_{int}) + \left[1 - \exp\left(-h_{FC} \cdot \frac{A}{c_{p,air}}\right) \right] \cdot (T_g - T_{int})$$

$$\dot{q}_{int,air} = \dot{M} \cdot c_{p,air} \cdot \left[\frac{T_{arc,out} - 20}{A} \right]$$

$$\dot{q}_{int} = \dot{q}_{ext,w} + \dot{q}_{int,air}$$

Figure 4.3.4 –Formatted Equations Window for the ventilated Trombe wall example.

The EES Procedure Ductflow, from its Heat Transfer & Fluid Flow database, was chosen to calculate both the forced convection coefficient in the air channel, and the flow friction factor, as a function of air flow rate and (physical and geometrical) properties.

The heat flux exchanged between the wall and the indoor space is also calculated (\dot{q}_{int} , or \dot{q}_{int}), by summing 2 different contributions: the flux transferred from the internal wall surface by convection ($\dot{q}_{int,w}$), and the heat rate transported by the heated air that re-enters the indoor space ($\dot{q}_{int,air}$). The climatic variables (ambient temperature and solar radiation) for each time are interpolated from the hourly values introduced in the Lookup Table (“Lookup 1” – the same as in Figure 4.2.4).

Figure 4.3.5 shows the first rows of “Table 1”. For a Δt of 60 s, a total of 1441 rows were created (24 hour simulation). The initial (first row) wall temperature values were obtained after

a couple of simulations starting with a fixed initial value; they represent a daily cycle, and are equal to the temperatures at the end of the day.

Run	time [s]	hour	T _{amb} [°C]	I _{sol} [W/m ²]	T ₁ [°C]	T ₂ [°C]	T ₃ [°C]	T ₄ [°C]	T ₅ [°C]	T _g [°C]	T _{air,out} [°C]	M [kg/s]	Re	f	h _{FC} [W/m ² °C]	q _{int} [W/m ²]	q _{int,w} [W/m ²]	q _{int,ar} [W/m ²]	
Run 1	0				24.68	25.11	25.24	25.06	24.58										
Run 2	60	0.01667	10.7	0	24.67	25.1	25.22	25.05	24.57	19.92	20.63	0.05162	1825	0.1096	1.419	28.8	23.37	5.429	
Run 3	120	0.03333	10.69	0	24.66	25.08	25.21	25.04	24.56	19.91	20.63	0.05153	1822	0.1097	1.419	28.72	23.32	5.404	
Run 4	180	0.05	10.69	0	24.65	25.07	25.2	25.03	24.55	19.9	20.62	0.05145	1819	0.1099	1.418	28.65	23.27	5.379	
Run 5	240	0.06667	10.69	0	24.64	25.06	25.19	25.01	24.54	19.9	20.62	0.05136	1815	0.11	1.417	28.57	23.22	5.354	
Run 6	300	0.08333	10.68	0	24.63	25.05	25.18	25	24.53	19.89	20.62	0.05128	1812	0.1101	1.416	28.5	23.17	5.328	
Run 7	360	0.1	10.68	0	24.62	25.04	25.17	24.99	24.52	19.88	20.62	0.05119	1809	0.1102	1.416	28.42	23.12	5.303	
Run 8	420	0.11667	10.68	0	24.61	25.03	25.16	24.98	24.5	19.88	20.62	0.05111	1806	0.1103	1.415	28.35	23.07	5.278	
Run 9	480	0.13333	10.67	0	24.6	25.02	25.15	24.97	24.49	19.87	20.61	0.05102	1803	0.1104	1.414	28.27	23.02	5.253	
Run 10	540	0.15	10.67	0	24.59	25.01	25.13	24.96	24.48	19.86	20.61	0.05094	1800	0.1105	1.413	28.2	22.97	5.228	
Run 11	600	0.16667	10.67	0	24.58	25	25.12	24.95	24.47	19.85	20.61	0.05085	1797	0.1106	1.412	28.12	22.92	5.203	
Run 12	660	0.18333	10.66	0	24.57	24.99	25.11	24.94	24.46	19.85	20.61	0.05076	1794	0.1108	1.412	28.05	22.87	5.178	
Run 13	720	0.2	10.66	0	24.56	24.98	25.1	24.92	24.45	19.84	20.61	0.05068	1791	0.1109	1.411	27.98	22.82	5.154	
Run 14	780	0.21667	10.66	0	24.55	24.97	25.09	24.91	24.44	19.83	20.6	0.05059	1788	0.1111	1.41	27.9	22.77	5.129	
Run 15	840	0.23333	10.65	0	24.54	24.96	25.08	24.9	24.43	19.83	20.6	0.05051	1785	0.1111	1.409	27.83	22.72	5.104	
Run 16	900	0.25	10.65	0	24.54	24.94	25.07	24.89	24.42	19.82	20.6	0.05042	1782	0.1112	1.408	27.76	22.68	5.079	

Figure 4.3.5 – First rows for Parametric Table “Table 1” in the ventilated Trombe wall example.

Figure 4.3.6 shows a graph with the evolution of wall temperatures (internal and external surfaces), internal glass temperature and air outlet temperature. The variation of wall surface temperatures is similar to Figure 4.2.6 (non-ventilated wall), but, due to the use of a more insulating double glazing, the temperatures are higher. A maximum of 36°C occurs on the external wall surface. The inside glazing temperature is also significantly higher, compared to the non-ventilated single glazed wall; it is most of the time higher than the air inlet temperature (20°C). The air is heated in the channel during the whole 24 hour period, achieving a maximum of 22.7°C at 14:00.

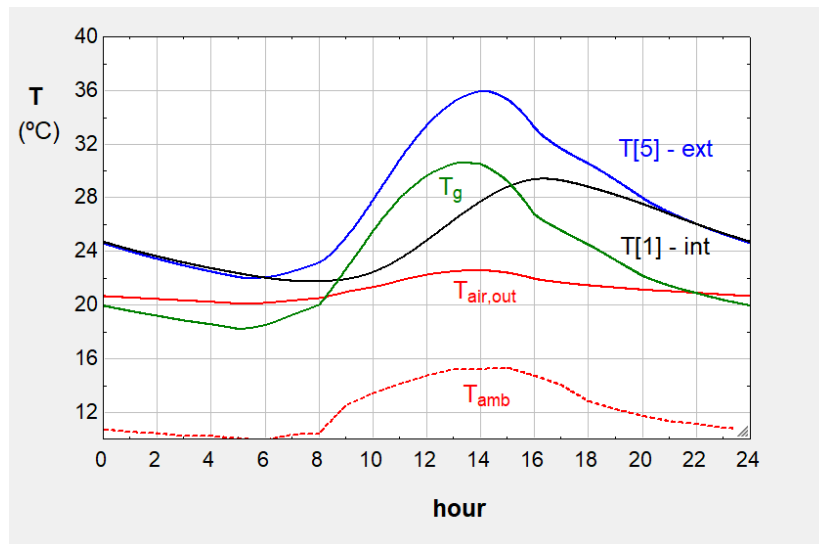


Figure 4.3.6 –Time evolution of several temperatures in the ventilated Trombe wall example.

Figure 4.3.7 shows the evolution of the air flow rate (\dot{M}), as well as the forced convection coefficient (h_{FC}) in the channel. As can be seen, the flow rate is always positive (upward direction), with a maximum value at 14:00. The convection coefficient more or less follows the same pattern. However, there is an instability associated with the EES Ductflow calculation, which is related with the transition zone from laminar to turbulent flow. This is better understood with the representation of Figure 4.3.8, where the Reynolds number is shown. However, this does not modify the main results and conclusions.

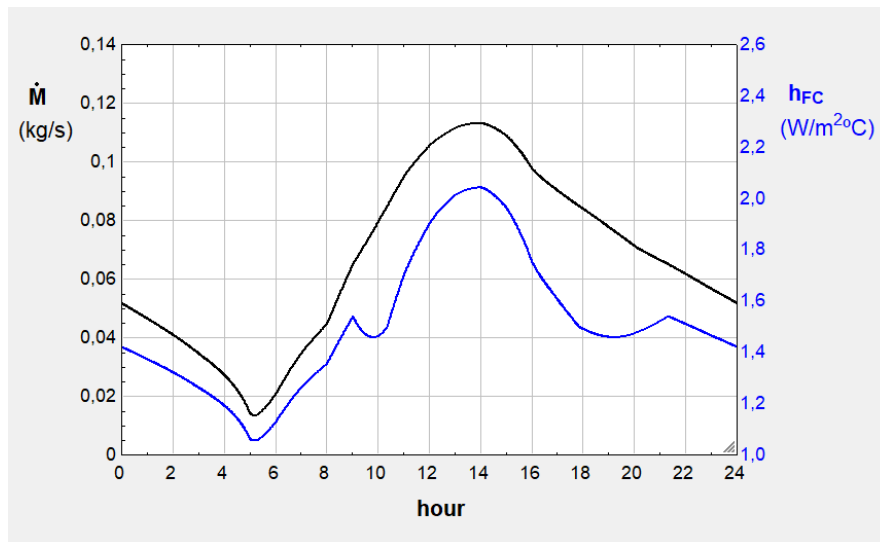


Figure 4.3.7 – Evolution of channel air flow rate and convective heat transfer coefficient.

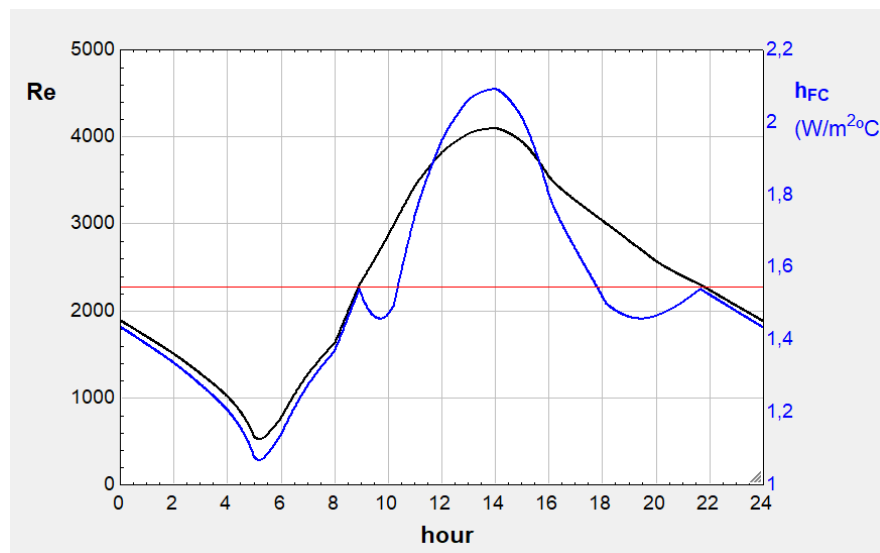


Figure 4.3.8 – Evolution of Reynolds number and convective heat transfer coefficient. Red line for $Re=2300$.

Figure 4.3.9 analyses the evolution of the heat flux exchanged between the wall and the indoor space. It also represents the 2 different components: surface convection and airflow. In this wall the maximum heat input (about 90 W/m^2) occurs at about 15:00, sooner than in the case of the unventilated wall of section 4.2, where the maximum occurred at 18:00. This is due to the faster removal of the heat stored in the wall when air circulation is used. The ventilation contribution is higher at 14:00. Therefore, this ventilated Trombe wall is more adequate to indoor spaces that are used in the afternoon.

Figure 4.3.10 compares the heating contribution of this ventilated wall with a non-ventilated wall, also using double glazing. The ventilated wall has a larger heat flux swing, with an earlier peak. A comparison is also made with an unglazed wall, which as seen in section 4.2 has a negative performance, losing heat to the outside environment.

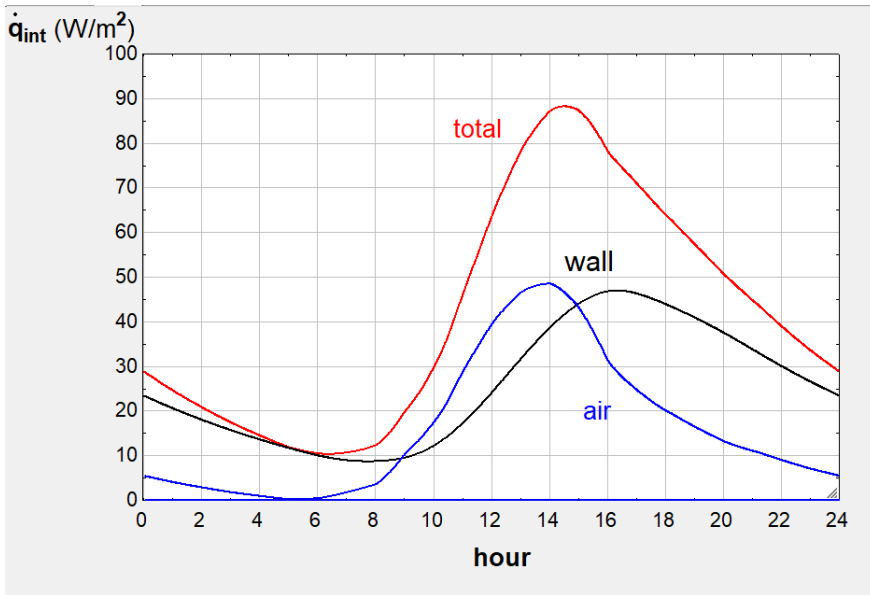


Figure 4.3.9 – Evolution of the heat flux to the indoor space in the ventilated Trombe wall.

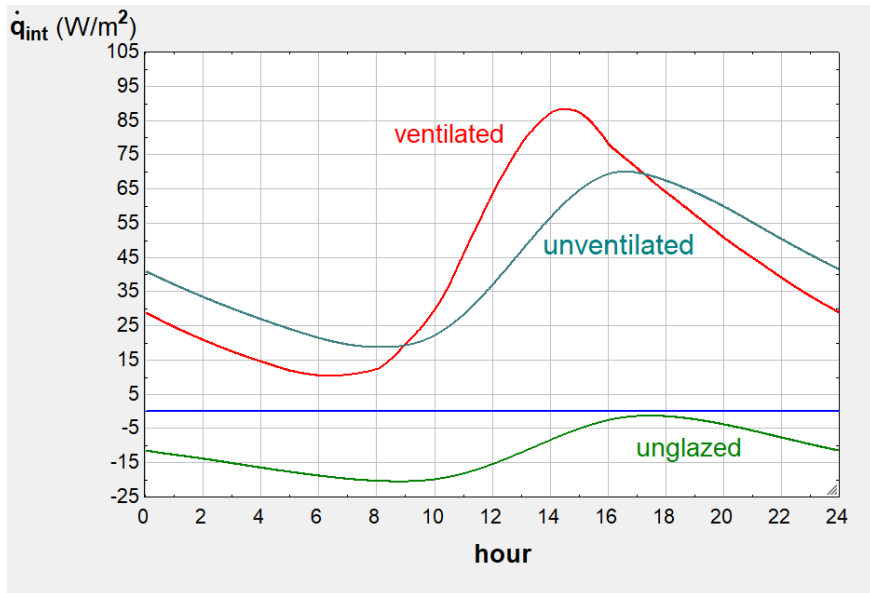


Figure 4.3.10 – Evolution of the heat flux to the indoor space for the ventilated wall, an unventilated wall with double glazing and an unglazed wall.

4.4 Car glass heating system (dynamic)

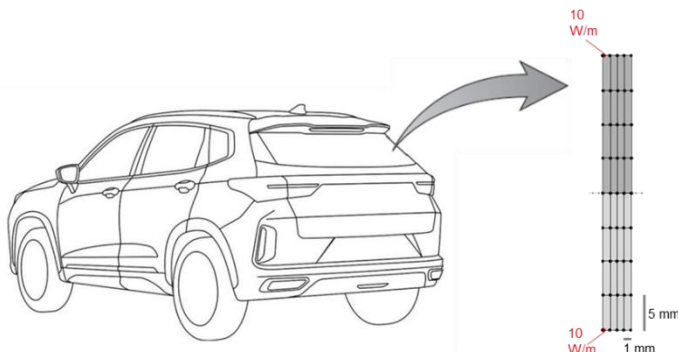


Figure 4.4.1 – Car glass discretisation.

To eliminate vapour condensation in the rear glass of a car, very small electric wires are connected to the glass inner surface. The wires have a spacing of 4 cm and generate a heating rate of 10 W per meter of glass width when electric current runs through them.

The glass is 4 mm thick and has a thermal conductivity $k=0.84 \text{ W/m}^\circ\text{C}$ and a thermal diffusivity $\alpha = k/\rho/c_p = 0.39 \times 10^{-6} \text{ m}^2/\text{s}$.

Starting from an initial situation when the glass, internal and external temperatures are all equal to 5°C , obtain the glass temperature distribution after 15 minutes. The inside and outside heat transfer coefficients remain constant at 6 and $20 \text{ W/m}^2\text{C}$, respectively. Indoor and outdoor temperatures also remain constant. Use a grid with nodes spaced 1 mm along the glass thickness and 5 mm along the vertical direction. The presence of condensate on the glass external surface may be neglected.

Repeat the calculation if the indoor temperature is kept at 15°C , with a constant external temperature of 5°C and an initial glass temperature of 5°C .

Using the finite volumes method, the glass will be discretised in 2D. Variations along the width direction are neglected. Figure 4.4.2 shows the region to consider, according to the values of Δx and Δy imposed. Only half of the distance between electric wires needs to be considered, as there is symmetry at mid-distance between wires ($i = 5$). There is also symmetry at the $i = 1$ surface, with half of the input power (5 W/m) distributed upwards and half downwards. Therefore, only 5×5 nodes/volumes need to be considered.

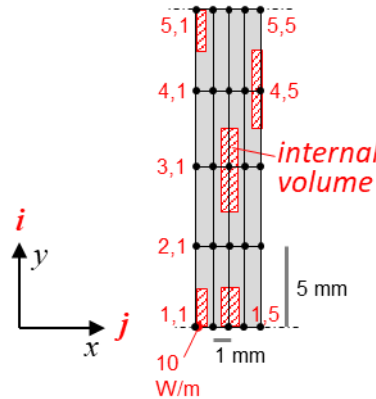


Figure 4.4.2 – Nodes and different types of volumes.

There are 9 internal volumes, 6 surface half-volumes with convection (at $j = 1$ and 5), 6 half-volumes in symmetry planes (at $i = 1$ and 5), and 4 corner volumes.

Using the implicit method, the discretised equations for the internal volumes ($i = 2$ to 4 , and $j = 2$ to 4) are

$$\rho c_p \Delta x \Delta y \frac{(T_{i,j}^{t+\Delta t} - T_{i,j}^t)}{\Delta t} = \frac{k}{\Delta x} \Delta y (T_{i,j-1}^{t+\Delta t} + T_{i,j+1}^{t+\Delta t} - 2T_{i,j}^{t+\Delta t}) + \frac{k}{\Delta y} \Delta x (T_{i-1,j}^{t+\Delta t} + T_{i+1,j}^{t+\Delta t} - 2T_{i,j}^{t+\Delta t}) \quad (4.4.1)$$

or

$$\rho c_p \frac{(T_{i,j}^{t+\Delta t} - T_{i,j}^t)}{\Delta t} = \frac{k}{\Delta x^2} (T_{i,j-1}^{t+\Delta t} + T_{i,j+1}^{t+\Delta t} - 2T_{i,j}^{t+\Delta t}) + \frac{k}{\Delta y^2} (T_{i-1,j}^{t+\Delta t} + T_{i+1,j}^{t+\Delta t} - 2T_{i,j}^{t+\Delta t}) \quad (4.4.2)$$

For the surface half-volumes with convection ($j = 1$ and 5 , with $i = 2$ to 4):

$$\begin{aligned} \rho c_p \frac{\Delta x}{2} \Delta y \frac{(T_{i,1}^{t+\Delta t} - T_{i,1}^t)}{\Delta t} &= \frac{k}{\Delta x} \Delta y (T_{i,2}^{t+\Delta t} - T_{i,1}^{t+\Delta t}) + \\ &+ \frac{k}{\Delta y} \frac{\Delta x}{2} (T_{i-1,1}^{t+\Delta t} + T_{i+1,1}^{t+\Delta t} - 2T_{i,1}^{t+\Delta t}) + h_{int} \Delta y (T_{int} - T_{i,1}^{t+\Delta t}) \end{aligned} \quad (4.4.3)$$

and

$$\begin{aligned} \rho c_p \frac{\Delta x}{2} \Delta y \frac{(T_{i,5}^{t+\Delta t} - T_{i,5}^t)}{\Delta t} &= \frac{k}{\Delta x} \Delta y (T_{i,4}^{t+\Delta t} - T_{i,5}^{t+\Delta t}) + \\ &+ \frac{k}{\Delta y} \frac{\Delta x}{2} (T_{i-1,5}^{t+\Delta t} + T_{i+1,5}^{t+\Delta t} - 2T_{i,5}^{t+\Delta t}) + h_{ext} \Delta y (T_{ext} - T_{i,5}^{t+\Delta t}) \end{aligned} \quad (4.4.4)$$

For the surface half-volumes in symmetry planes ($i = 1$ and 5 , with $j = 2$ to 4):

$$\begin{aligned} \rho c_p \Delta x \frac{\Delta y}{2} \frac{(T_{1,j}^{t+\Delta t} - T_{1,j}^t)}{\Delta t} &= \frac{k}{\Delta x} \frac{\Delta y}{2} (T_{1,j-1}^{t+\Delta t} + T_{1,j+1}^{t+\Delta t} - 2T_{1,j}^{t+\Delta t}) + \\ &+ \frac{k}{\Delta y} \Delta x (T_{2,j}^{t+\Delta t} - T_{1,j}^{t+\Delta t}) \end{aligned} \quad (4.4.5)$$

and

$$\begin{aligned} \rho c_p \Delta x \frac{\Delta y}{2} \frac{(T_{5,j}^{t+\Delta t} - T_{5,j}^t)}{\Delta t} &= \frac{k}{\Delta x} \frac{\Delta y}{2} (T_{5,j-1}^{t+\Delta t} + T_{5,j+1}^{t+\Delta t} - 2T_{5,j}^{t+\Delta t}) + \\ &+ \frac{k}{\Delta y} \Delta x (T_{4,j}^{t+\Delta t} - T_{5,j}^{t+\Delta t}) \end{aligned} \quad (4.4.6)$$

Finally, for the 4 corners we have:

$$\begin{aligned} \rho c_p \frac{\Delta x}{2} \frac{\Delta y}{2} \frac{(T_{1,1}^{t+\Delta t} - T_{1,1}^t)}{\Delta t} &= \frac{k}{\Delta x} \frac{\Delta y}{2} (T_{1,2}^{t+\Delta t} - T_{1,1}^{t+\Delta t}) + \frac{k}{\Delta y} \frac{\Delta x}{2} (T_{2,1}^{t+\Delta t} - T_{1,1}^{t+\Delta t}) + \\ &+ h_{int} \frac{\Delta y}{2} (T_{int} - T_{1,1}^{t+\Delta t}) + 5 \end{aligned} \quad (4.4.7)$$

$$\begin{aligned} \rho c_p \frac{\Delta x}{2} \frac{\Delta y}{2} \frac{(T_{1,5}^{t+\Delta t} - T_{1,5}^t)}{\Delta t} &= \frac{k}{\Delta x} \frac{\Delta y}{2} (T_{1,4}^{t+\Delta t} - T_{1,5}^{t+\Delta t}) + \frac{k}{\Delta y} \frac{\Delta x}{2} (T_{2,5}^{t+\Delta t} - T_{1,5}^{t+\Delta t}) + \\ &+ h_{ext} \frac{\Delta y}{2} (T_{ext} - T_{1,5}^{t+\Delta t}) \end{aligned} \quad (4.4.8)$$

$$\begin{aligned} \rho c_p \frac{\Delta x}{2} \frac{\Delta y}{2} \frac{(T_{5,1}^{t+\Delta t} - T_{5,1}^t)}{\Delta t} &= \frac{k}{\Delta x} \frac{\Delta y}{2} (T_{5,2}^{t+\Delta t} - T_{5,1}^{t+\Delta t}) + \frac{k}{\Delta y} \frac{\Delta x}{2} (T_{4,1}^{t+\Delta t} - T_{5,1}^{t+\Delta t}) + \\ &+ h_{int} \frac{\Delta y}{2} (T_{int} - T_{5,1}^{t+\Delta t}) \end{aligned} \quad (4.4.9)$$

$$\begin{aligned} \rho c_p \frac{\Delta x}{2} \frac{\Delta y}{2} \frac{(T_{5,5}^{t+\Delta t} - T_{5,5}^t)}{\Delta t} &= \frac{k}{\Delta x} \frac{\Delta y}{2} (T_{5,4}^{t+\Delta t} - T_{5,5}^{t+\Delta t}) + \frac{k}{\Delta y} \frac{\Delta x}{2} (T_{4,5}^{t+\Delta t} - T_{5,5}^{t+\Delta t}) + \\ &+ h_{ext} \frac{\Delta y}{2} (T_{ext} - T_{5,5}^{t+\Delta t}) \end{aligned} \quad (4.4.10)$$

These 25 equations are shown in Figures 4.4.3 – Equations Window – and 4.4.4 – Formatted Equations. We could also define the coordinates of the different nodes (x, y), using array variables $x[j]$ and $y[i]$, but if we represent the temperatures as a function of j and i , as the nodes are equally spaced in each direction, there is no special advantage in adding those variables.

```

Equations Window
k=0,84; alpha=0,39E-6
rhocp=k/alpha
h_int=6; h_ext=20
T_int=5; T_ext=5
DELTAx=0,001; DELTAy=0,005
DELTA=15
min=time/60
row=1+ time/DELTA

"Internal Nodes"
Duplicate i=2,4
  Duplicate j=2,4
    rhocp*DELTAx*DELTAy*(T[i,j]-tablevalue(Table 1; row-1; #T[i,j]))/DELTA= k/DELTAx*DELTAy*(T[i,j-1]+T[i,j+1]-2*T[i,j]) + k/DELTAy*DELTAx*(T[i-1,j]+T[i+1,j]-2*T[i,j])
  End
End

"surface half-volumes with convection (j=1 and 5, with i=2 to 4)"
Duplicate i=2,4
  rhocp*DELTAx/2*DELTAy*(T[i,1]-tablevalue(Table 1; row-1; #T[i,1]))/DELTA= k/DELTAx*DELTAy/2*(T[i,2]-T[i,1]) + k/DELTAy*DELTAx/2*(T[i-1,1]+T[i+1,1]-2*T[i,1])+h_int*DELTAy*(T_int-T[i,1])
  rhocp*DELTAx/2*DELTAy*(T[i,5]-tablevalue(Table 1; row-1; #T[i,5]))/DELTA= k/DELTAx*DELTAy/2*(T[i,4]-T[i,5]) + k/DELTAy*DELTAx/2*(T[i-1,5]+T[i+1,5]-2*T[i,5])+h_ext*DELTAy*(T_ext-T[i,5])
End

"surface half-volumes in symmetry planes (i=1 and 5, with j=2 to 4)"
Duplicate j=2,4
  rhocp*DELTAx*DELTAy/2*(T[1,j]-tablevalue(Table 1; row-1; #T[1,j]))/DELTA= k/DELTAx*DELTAy/2*(T[2,j]-T[1,j]) + k/DELTAy*DELTAx/2*(T[1,j-1]+T[1,j+1]-2*T[1,j])
  rhocp*DELTAx*DELTAy/2*(T[5,j]-tablevalue(Table 1; row-1; #T[5,j]))/DELTA= k/DELTAx*DELTAy/2*(T[4,j]-T[5,j]) + k/DELTAy*DELTAx/2*(T[5,j-1]+T[5,j+1]-2*T[5,j])
End

"bottom internal corner - wire"
rhocp*DELTAx/2*DELTAy/2*(T[1,1]-tablevalue(Table 1; row-1; #T[1,1]))/DELTA= k/DELTAx*DELTAy/2*(T[1,2]-T[1,1]) + k/DELTAy*DELTAx/2*(T[2,1]-T[1,1])+h_int*DELTAy/2*(T_int - T[1,1])+5
"bottom external corner"
rhocp*DELTAx/2*DELTAy/2*(T[1,5]-tablevalue(Table 1; row-1; #T[1,5]))/DELTA= k/DELTAx*DELTAy/2*(T[1,4]-T[1,5]) + k/DELTAy*DELTAx/2*(T[2,5]-T[1,5])+h_ext*DELTAy/2*(T_ext - T[1,5])
"top internal corner"
rhocp*DELTAx/2*DELTAy/2*(T[5,1]-tablevalue(Table 1; row-1; #T[5,1]))/DELTA= k/DELTAx*DELTAy/2*(T[5,2]-T[5,1]) + k/DELTAy*DELTAx/2*(T[4,1]-T[5,1])+h_int*DELTAy/2*(T_int - T[5,1])
"top external corner"
rhocp*DELTAx/2*DELTAy/2*(T[5,5]-tablevalue(Table 1; row-1; #T[5,5]))/DELTA= k/DELTAx*DELTAy/2*(T[5,4]-T[5,5]) + k/DELTAy*DELTAx/2*(T[4,5]-T[5,5])+h_ext*DELTAy/2*(T_ext - T[5,5])

```

Figure 4.4.3 – Equations Window for the car glass heating example.

Formatted Equations

$k = 0,84 \quad \alpha = 3,9 \times 10^{-7}$
 $\text{rhocp} = \frac{k}{\alpha}$
 $h_{\text{int}} = 6 \quad h_{\text{ext}} = 20$
 $T_{\text{int}} = 5 \quad T_{\text{ext}} = 5$
 $\Delta x = 0,001 \quad \Delta y = 0,005$
 $\Delta t = 15$
 $\text{min} = \frac{\text{time}}{60}$
 $\text{row} = 1 + \frac{\text{time}}{\Delta t}$

Internal Nodes

$$\text{rhocp} \cdot \Delta x \cdot \Delta y \cdot \left[\frac{T_{i,j} - \text{TableValue}(\text{Table 1}; \text{row} - 1; T_{i,j}')}{\Delta t} \right] = \frac{k}{\Delta x} \cdot \Delta y \cdot (T_{i,j-1} + T_{i,j+1} - 2 \cdot T_{i,j}) + \frac{k}{\Delta y} \cdot \Delta x \cdot (T_{i-1,j} + T_{i+1,j} - 2 \cdot T_{i,j}) \quad (\text{for } i = 2 \text{ to } 4; j = 2 \text{ to } 4)$$

surface half-volumes with convection (j=1 and 5, with i=2 to 4)

$$\text{rhocp} \cdot \frac{\Delta x}{2} \cdot \Delta y \cdot \left[\frac{T_{i,1} - \text{TableValue}(\text{Table 1}; \text{row} - 1; T_{i,1}')}{\Delta t} \right] = \frac{k}{\Delta x} \cdot \Delta y \cdot (T_{i,2} - T_{i,1}) + \frac{k}{\Delta y} \cdot \frac{\Delta x}{2} \cdot (T_{i-1,1} + T_{i+1,1} - 2 \cdot T_{i,1}) + h_{\text{int}} \cdot \Delta y \cdot (T_{\text{int}} - T_{i,1}) \quad (\text{for } i = 2 \text{ to } 4)$$

$$\text{rhocp} \cdot \frac{\Delta x}{2} \cdot \Delta y \cdot \left[\frac{T_{i,5} - \text{TableValue}(\text{Table 1}; \text{row} - 1; T_{i,5}')}{\Delta t} \right] = \frac{k}{\Delta x} \cdot \Delta y \cdot (T_{i,4} - T_{i,5}) + \frac{k}{\Delta y} \cdot \frac{\Delta x}{2} \cdot (T_{i-1,5} + T_{i+1,5} - 2 \cdot T_{i,5}) + h_{\text{ext}} \cdot \Delta y \cdot (T_{\text{ext}} - T_{i,5}) \quad (\text{for } i = 2 \text{ to } 4)$$

surface half-volumes in symmetry planes (i=1 and 5, with j=2 to 4)

$$\text{rhocp} \cdot \Delta x \cdot \frac{\Delta y}{2} \cdot \left[\frac{T_{1,j} - \text{TableValue}(\text{Table 1}; \text{row} - 1; T_{1,j}')}{\Delta t} \right] = \frac{k}{\Delta y} \cdot \Delta x \cdot (T_{2,j} - T_{1,j}) + \frac{k}{\Delta x} \cdot \frac{\Delta y}{2} \cdot (T_{1,j-1} + T_{1,j+1} - 2 \cdot T_{1,j}) \quad (\text{for } j = 2 \text{ to } 4)$$

$$\text{rhocp} \cdot \Delta x \cdot \frac{\Delta y}{2} \cdot \left[\frac{T_{5,j} - \text{TableValue}(\text{Table 1}; \text{row} - 1; T_{5,j}')}{\Delta t} \right] = \frac{k}{\Delta y} \cdot \Delta x \cdot (T_{4,j} - T_{5,j}) + \frac{k}{\Delta x} \cdot \frac{\Delta y}{2} \cdot (T_{5,j-1} + T_{5,j+1} - 2 \cdot T_{5,j}) \quad (\text{for } j = 2 \text{ to } 4)$$

bottom internal corner - wire

$$\text{rhocp} \cdot \frac{\Delta x}{2} \cdot \frac{\Delta y}{2} \cdot \left[\frac{T_{1,1} - \text{TableValue}(\text{Table 1}; \text{row} - 1; T_{1,1}')}{\Delta t} \right] = \frac{k}{\Delta x} \cdot \frac{\Delta y}{2} \cdot (T_{1,2} - T_{1,1}) + \frac{k}{\Delta y} \cdot \frac{\Delta x}{2} \cdot (T_{2,1} - T_{1,1}) + h_{\text{int}} \cdot \frac{\Delta y}{2} \cdot (T_{\text{int}} - T_{1,1}) + 5$$

bottom external corner

$$\text{rhocp} \cdot \frac{\Delta x}{2} \cdot \frac{\Delta y}{2} \cdot \left[\frac{T_{1,5} - \text{TableValue}(\text{Table 1}; \text{row} - 1; T_{1,5}')}{\Delta t} \right] = \frac{k}{\Delta x} \cdot \frac{\Delta y}{2} \cdot (T_{1,4} - T_{1,5}) + \frac{k}{\Delta y} \cdot \frac{\Delta x}{2} \cdot (T_{2,5} - T_{1,5}) + h_{\text{ext}} \cdot \frac{\Delta y}{2} \cdot (T_{\text{ext}} - T_{1,5})$$

top internal corner

$$\text{rhocp} \cdot \frac{\Delta x}{2} \cdot \frac{\Delta y}{2} \cdot \left[\frac{T_{5,1} - \text{TableValue}(\text{Table 1}; \text{row} - 1; T_{5,1}')}{\Delta t} \right] = \frac{k}{\Delta x} \cdot \frac{\Delta y}{2} \cdot (T_{5,2} - T_{5,1}) + \frac{k}{\Delta y} \cdot \frac{\Delta x}{2} \cdot (T_{4,1} - T_{5,1}) + h_{\text{int}} \cdot \frac{\Delta y}{2} \cdot (T_{\text{int}} - T_{5,1})$$

top external corner

$$\text{rhocp} \cdot \frac{\Delta x}{2} \cdot \frac{\Delta y}{2} \cdot \left[\frac{T_{5,5} - \text{TableValue}(\text{Table 1}; \text{row} - 1; T_{5,5}')}{\Delta t} \right] = \frac{k}{\Delta x} \cdot \frac{\Delta y}{2} \cdot (T_{5,4} - T_{5,5}) + \frac{k}{\Delta y} \cdot \frac{\Delta x}{2} \cdot (T_{4,5} - T_{5,5}) + h_{\text{ext}} \cdot \frac{\Delta y}{2} \cdot (T_{\text{ext}} - T_{5,5})$$

Figure 4.4.4 – Formatted Equations Window for the car glass heating example.

As seen before, the Parametric Table (“Table 1”) expresses the different time steps (one Run for each time step), and contains the calculation results – Figure 4.4.5. A time step of 15 s was used in the simulations. As an array variable was used for the temperature, the results for the final simulation/Run are also available in the Arrays Table – Figure 4.4.6.

Run	time [s]	min	T _{1,1}	T _{1,2}	T _{1,3}	T _{1,4}	T _{1,5}	T _{2,1}	T _{2,2}	T _{2,3}	T _{2,4}	T _{2,5}	T _{3,1}	T _{3,2}	T _{3,3}	T _{3,4}	T _{3,5}	T _{4,1}	T _{4,2}	T _{4,3}	T _{4,4}	T _{4,5}	T _{5,1}	T _{5,2}	T _{5,3}	T _{5,4}	T _{5,5}
Run 126	1875	31.25	25.75	23.86	22.5	21.57	21.01	17.25	17.24	17.11	16.9	16.64	13.62	13.64	13.6	13.49	13.33	11.73	11.76	11.73	11.65	11.52	11.15	11.17	11.14	11.07	10.95
Run 127	1890	31.5	25.76	23.86	22.5	21.57	21.01	17.25	17.24	17.11	16.9	16.64	13.62	13.64	13.6	13.5	13.33	11.74	11.76	11.73	11.65	11.52	11.15	11.17	11.14	11.07	10.95
Run 128	1905	31.75	25.76	23.86	22.5	21.57	21.02	17.25	17.24	17.11	16.9	16.65	13.62	13.65	13.6	13.5	13.33	11.74	11.76	11.73	11.65	11.52	11.15	11.17	11.14	11.07	10.95
Run 129	1920	32	25.76	23.87	22.5	21.57	21.02	17.25	17.24	17.11	16.9	16.65	13.62	13.65	13.61	13.5	13.33	11.74	11.76	11.73	11.65	11.52	11.15	11.17	11.15	11.07	10.95
Run 130	1935	32.25	25.76	23.87	22.51	21.58	21.02	17.25	17.24	17.11	16.91	16.65	13.62	13.65	13.61	13.5	13.33	11.74	11.76	11.73	11.65	11.52	11.15	11.17	11.15	11.07	10.95
Run 131	1950	32.5	25.76	23.87	22.51	21.58	21.02	17.25	17.24	17.11	16.91	16.65	13.62	13.65	13.61	13.5	13.33	11.74	11.77	11.74	11.66	11.53	11.15	11.18	11.15	11.08	10.96
Run 132	1965	32.75	25.76	23.87	22.51	21.58	21.02	17.26	17.25	17.12	16.91	16.65	13.63	13.65	13.61	13.5	13.34	11.74	11.77	11.74	11.66	11.53	11.16	11.18	11.15	11.08	10.96
Run 133	1980	33	25.76	23.87	22.51	21.58	21.02	17.26	17.25	17.12	16.91	16.65	13.63	13.65	13.61	13.51	13.34	11.75	11.77	11.74	11.66	11.53	11.16	11.18	11.15	11.08	10.96
Run 134	1995	33.25	25.77	23.87	22.51	21.58	21.03	17.26	17.25	17.12	16.91	16.66	13.63	13.65	13.61	13.51	13.34	11.75	11.77	11.74	11.66	11.53	11.16	11.18	11.15	11.08	10.96
Run 135	2010	33.5	25.77	23.88	22.51	21.58	21.03	17.26	17.25	17.12	16.91	16.66	13.63	13.66	13.61	13.51	13.34	11.75	11.77	11.74	11.66	11.53	11.16	11.18	11.15	11.08	10.96
Run 136	2025	33.75	25.77	23.88	22.51	21.58	21.03	17.26	17.25	17.12	16.91	16.66	13.63	13.66	13.62	13.51	13.34	11.75	11.77	11.74	11.66	11.53	11.16	11.18	11.16	11.08	10.96
Run 137	2040	34	25.77	23.88	22.52	21.59	21.03	17.26	17.25	17.12	16.92	16.66	13.63	13.66	13.62	13.51	13.34	11.75	11.77	11.74	11.66	11.53	11.16	11.18	11.16	11.08	10.96
Run 138	2055	34.25	25.77	23.88	22.52	21.59	21.03	17.26	17.25	17.12	16.92	16.66	13.63	13.66	13.62	13.51	13.34	11.75	11.77	11.75	11.66	11.53	11.16	11.18	11.16	11.08	10.97
Run 139	2070	34.5	25.77	23.88	22.52	21.59	21.03	17.26	17.25	17.12	16.92	16.66	13.63	13.66	13.62	13.51	13.34	11.75	11.78	11.75	11.67	11.54	11.17	11.19	11.16	11.09	10.97
Run 140	2085	34.75	25.77	23.88	22.52	21.59	21.03	17.27	17.26	17.13	16.92	16.66	13.64	13.66	13.62	13.51	13.35	11.75	11.78	11.75	11.67	11.54	11.17	11.19	11.16	11.09	10.97
Run 141	2100	35	25.77	23.88	22.52	21.59	21.03	17.27	17.26	17.13	16.92	16.66	13.64	13.66	13.62	13.51	13.35	11.76	11.78	11.75	11.67	11.54	11.17	11.19	11.16	11.09	10.97
Run 142	2115	35.25	25.77	23.88	22.52	21.59	21.03	17.27	17.26	17.13	16.92	16.66	13.64	13.66	13.62	13.52	13.35	11.76	11.78	11.75	11.67	11.54	11.17	11.19	11.16	11.09	10.97
Run 143	2130	35.5	25.78	23.88	22.52	21.59	21.03	17.27	17.26	17.13	16.92	16.66	13.64	13.66	13.62	13.52	13.35	11.76	11.78	11.75	11.67	11.54	11.17	11.19	11.16	11.09	10.97
Run 144	2145	35.75	25.78	23.88	22.52	21.59	21.04	17.27	17.26	17.13	16.92	16.67	13.64	13.67	13.62	13.52	13.35	11.76	11.78	11.75	11.67	11.54	11.17	11.19	11.16	11.09	10.97
Run 145	2160	36	25.78	23.89	22.52	21.59	21.04	17.27	17.26	17.13	16.92	16.67	13.64	13.67	13.63	13.52	13.35	11.76	11.78	11.75	11.67	11.54	11.17	11.19	11.16	11.09	10.97
Run 146	2175	36.25	25.78	23.89	22.52	21.59	21.04	17.27	17.26	17.13	16.92	16.67	13.64	13.67	13.63	13.52	13.35	11.76	11.78	11.75	11.67	11.54	11.17	11.19	11.17	11.09	10.97
Run 147	2190	36.5	25.78	23.89	22.52	21.59	21.04	17.27	17.26	17.13	16.92	16.67	13.64	13.67	13.63	13.52	13.35	11.76	11.78	11.75	11.67	11.54	11.17	11.19	11.17	11.09	10.97

Figure 4.4.5 – Parametric Table for the time simulation of the car glass heating example.

	1	2	3	4	5
T _{i,1}	25.78	23.89	22.53	21.6	21.04
T _{i,2}	17.27	17.26	17.13	16.93	16.67
T _{i,3}	13.64	13.67	13.63	13.52	13.35
T _{i,4}	11.76	11.79	11.76	11.68	11.54
T _{i,5}	11.18	11.2	11.17	11.1	10.98

Figure 4.4.6 – Arrays Table with calculated temperatures after 25 minutes.

Figure 4.4.7 shows the time evolution of the temperatures of different volumes/nodes. Of course, the highest temperature occurs always in volume/node (1,1), as this is where the heating element is located. The temperatures at higher y (or i) values (farther from the wire) do not change significantly with x (or j).

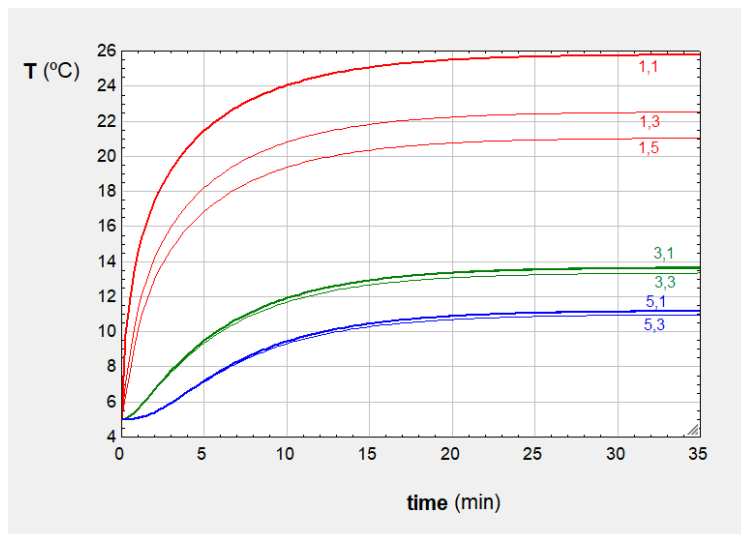


Figure 4.4.7 – Time evolution of temperatures in different nodes/volumes.

As Figure 4.4.7 shows, the major temperature changes occur in the initial 15 minutes. After about 25 minutes there is practically no further change in temperatures: steady-state is attained. The maximum temperature, at the wire location, is about 26°C. The minimum temperature, which is important to prevent condensation, is about 11°C.

EES allows obtaining a graphical distribution of the temperature in the 2 directions of space, at a given moment. For that, you need to choose in the menu: Plots → New Plot Window → X-Y-Z Plot; then, in the appearing window, choose Table – Arrays Table, 2-D table data, Isometric Lines, or Color Bands, and also define the scales – see Figure 4.4.8. With or without defining the coordinates of the different nodes ($x[j]$ and $y[i]$), it is also possible to obtain a 3D type graph, by choosing 3-D Surface. This will be shown in the example of section 4.5. That type of graph may be rotated to have access to different perspective views.

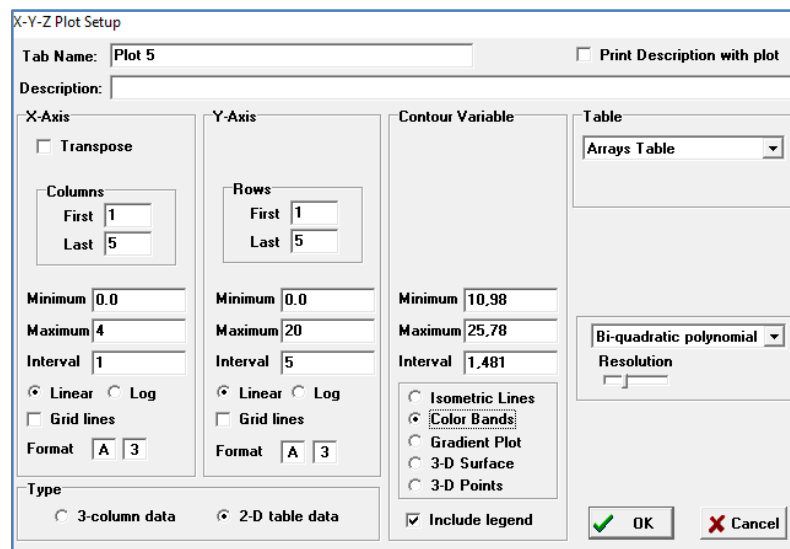


Figure 4.4.8 – Dialog window for X-Y-Z Color Bands Plot.

Figure 4.4.9 shows colour band graphs that may be obtained from the Arrays Table. You may produce different graphs for different instants of time (by using the respective Arrays Table). In the case of Figure 4.4.9, all the graphs are valid for a time of 25 minutes.

We should note that, because the colour bands and isothermal lines are obtained from the calculated node temperatures, there is the need perform an interpolation with the Arrays Table values; therefore, if the number of points in the Arrays Table is limited, a crude representation will be obtained. This may be noticed when comparing the graphs in Figure 4.4.9(a) and 4.4.9(b). Figure 4.4.9(a) was obtained with the Arrays Table from Figure 4.4.6, using the 25 previously calculated nodes, while Figure 4.4.9(b) was obtained with an extended number of nodes of $50 \times 50 = 2500$ nodes. The isothermal lines are significantly different, particularly near the electric wire region.

As requested, the influence of changing the internal temperature to 15°C, while maintaining the same external temperature and initial glass temperature at 5°C, was analysed. It is very simple to introduce this modification in the problem: it is simply needed to change T_{int} in the Equations Window to 15°C, maintaining all the other values. Figure 4.4.10 presents the time evolution of several temperatures in the glass. The evolution is similar to Figure 4.4.7, but, as the glass loses less heat, the temperatures are higher: the maximum temperature is now slightly above 28°C, and the minimum is above 13.5°C.

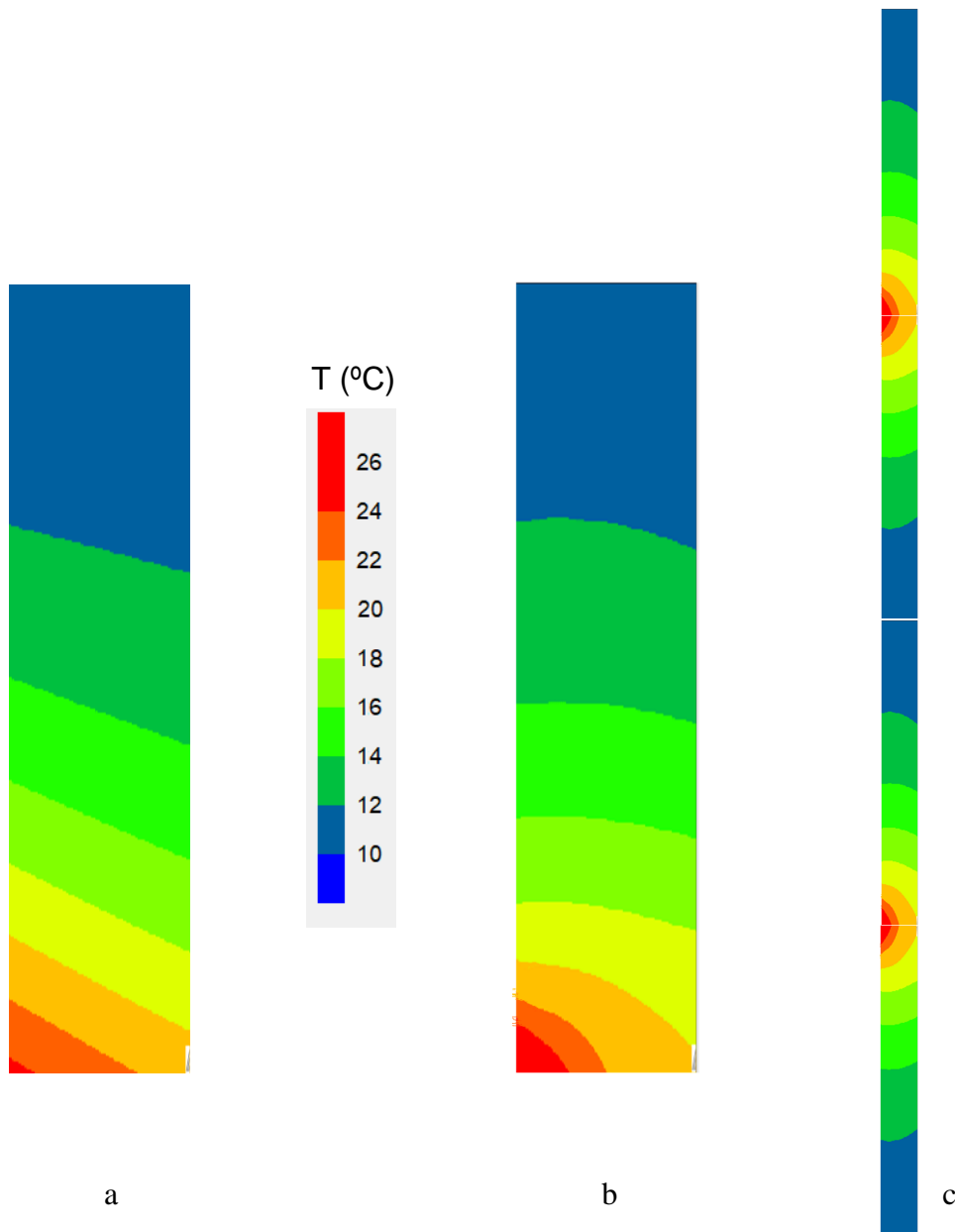


Figure 4.4.9 – Temperature colour bands for the car glass heating example: (a) using 25 nodes; (b) using 2500 nodes; (c) composition of 4 symmetrical regions in the glass.

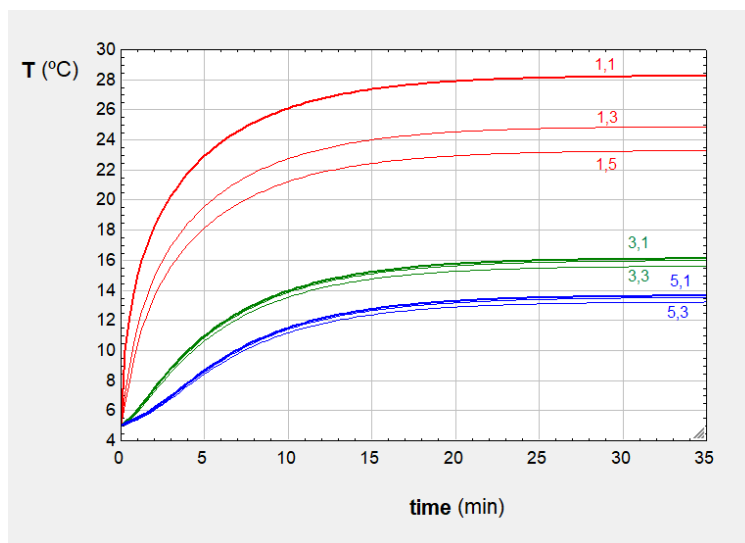


Figure 4.4.10 – Time evolution of temperatures in different nodes/volumes when $T_{int} = 15^{\circ}\text{C}$.

4.5 Dynamic cooling of a concrete beam

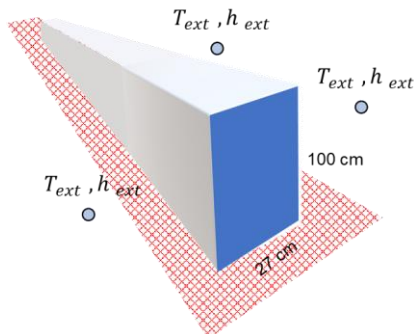


Figure 4.5.1 – Concrete beam cooling.

To increase the strength of concrete beams, they are subjected to a long heating cycle, up to a temperature of 80°C. Consider a long concrete beam, with a section of 27 × 100 cm, initially at a uniform temperature of 80°C. The beam is cooled to ambient air at $T_{ext}=20^{\circ}\text{C}$, with a heat transfer coefficient $h_{ext}=20\text{ W/m}^2\text{K}$ in all surfaces, except the base, which can be considered insulated. The concrete has the following properties: $k=1.2\text{ W/m}^{\circ}\text{C}$, $\rho=2300\text{ kg/m}^3$, $c_p=880\text{ J/kgK}$.

Obtain the temperature distribution in the beam section after 1 hour, 2 hours and 4 hours of cooling. Use a grid with $\Delta x=3\text{ cm}$ and $\Delta y=4\text{ cm}$. Calculate the evolution of the heat transfer rate to the outside.

The beam will be discretised in 2D, since its length is very large, and therefore the temperature is assumed to vary only in the section. Due to the symmetrical conditions, one half of the section could be considered, with 13.5 cm along the horizontal direction, and 100 cm along the vertical direction. However, we shall use the full width of 27 cm, with a grid with 10×26 nodes, with $\Delta x=3\text{ cm}$ and $\Delta y=4\text{ cm}$.

The discretised equations are very similar to those in section 4.4: one for the internal nodes, equal to equation (4.4.1) for 8×24 nodes, one for each of the boundaries excluding the corners – bottom (8 nodes), top (8 nodes), left (24 nodes) and right (24 nodes) surfaces – and one for each corner, with smaller volume sizes. The top, left and right surfaces have a heat transfer condition, while the bottom surface has no heat transfer (adiabatic).

The first index (i) is used for y (vertical) and the second index (j) is used for x (horizontal). Figure 4.5.2 presents the Formatted Equations Window and Figure 4.5.3 the Equations Window. Besides the temperatures, heat rate array variables – $\dot{Q}_{top}[j]$, $\dot{Q}_{left}[i]$, $\dot{Q}_{right}[i]$ – were defined to calculate the transfer rates to the outside in every element. Then they are summed to obtain the total surface transfer – $\dot{Q}_{top,total}$, $\dot{Q}_{left,total}$, $\dot{Q}_{right,total}$ – and the 3 are finally added to obtain the total heat transfer rate, \dot{Q}_{ext} .

The Parametric Table (“Table 1”) expresses the different time steps (one Run for each time step), starting with all temperatures equal to 80°C in the first row (initial condition), and contains the calculation results. A time step of 60 s was used in the simulations. As array variables were used for the temperature and heat rates, those results are also available in the Arrays Table, for the final simulation/Run. As the situation at 3 different moments is wanted (1 hour, 2 hours and 4 hours), the Parametric Table can be run 3 different times, stopping in each time at the required moment. Then, the Arrays Table will contain the wanted temperature and heat rate values, that will be used to obtain graphical representations of the temperatures in the 3 different moments.

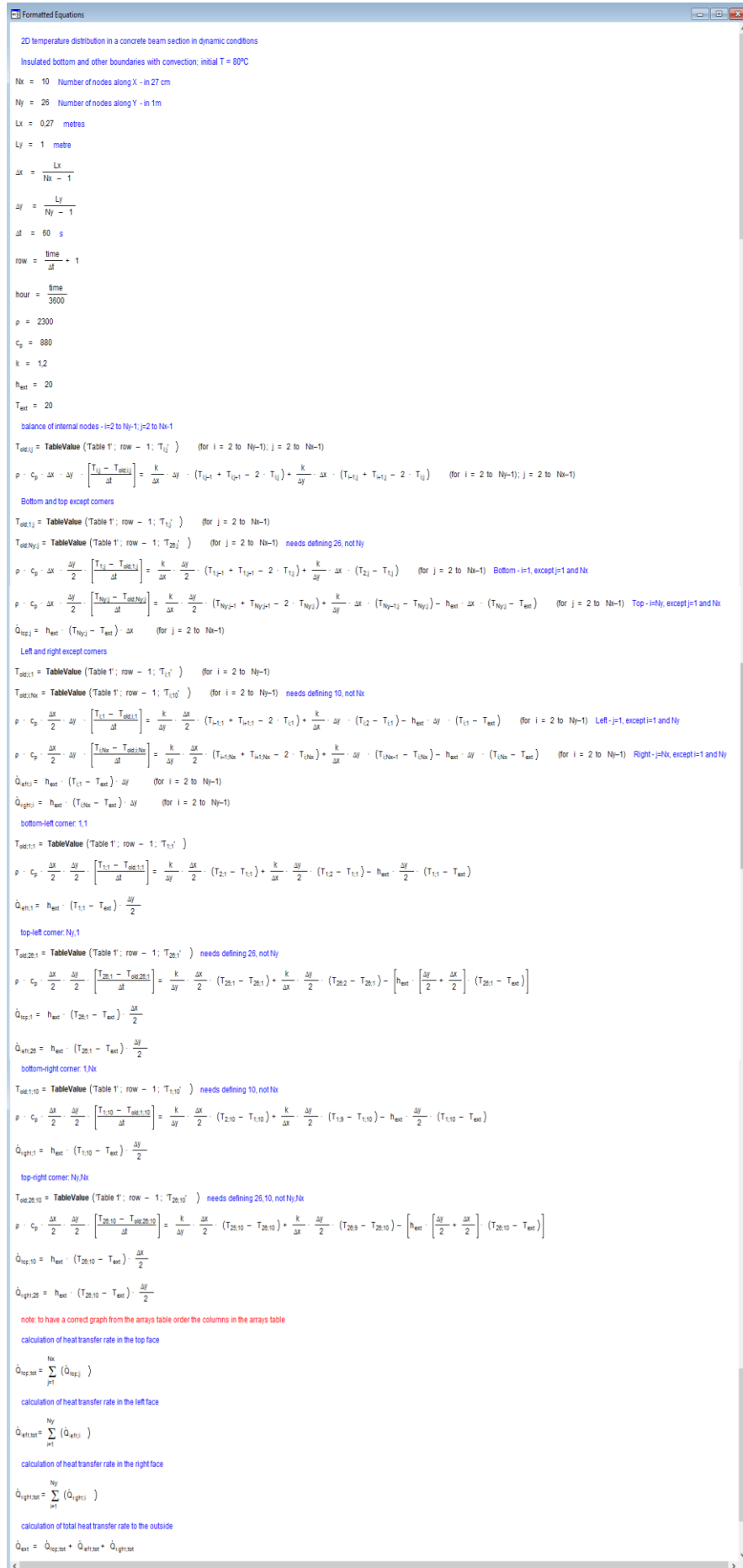


Figure 4.5.2 – Formatted Equations Window for the concrete beam cooling example.

```

Equations Window
"2D temperature distribution in a concrete beam section in dynamic conditions"
"Insulated bottom and other boundaries with convection; initial T = 80°C"

Nx=10 "Number of nodes along X - in 27 cm"
Ny=26 "Number of nodes along Y - in 1m"
Lx=0.27 "metres"
Ly=1 "metre"
DELTAx=Lx/(Nx-1)
DELTAy=Ly/(Ny-1)
DELTA=60 "s"
row=time/DELTA+1
hour=time/3600
rho=2300
c_p=880
k=1.2
h_ext=20
T_ext=20

"balance of internal nodes - i=2 to Ny-1; j=2 to Nx-1"
duplicate i=2:Ny-1
  duplicate j=2:Nx-1
    T_old[i,j]=TABLEVALUE (Table 1; row-1,#T[i,j])
    rho*c_p*DELTAx*DELTAy*(T[i,j]-T_old[i,j])/DELTA=k*DELTAx*DELTAy*(T[i,j-1]+T[i,j+1]-2*T[i,j])+k*DELTAy*DELTAx*(T[i-1,j]+T[i+1,j]-2*T[i,j])
  end
end

"Bottom and top except corners"
duplicate j=2:Nx-1
  T_old[1,j]=TABLEVALUE (Table 1; row-1,#T[1,j])
  T_old[Ny,j]=TABLEVALUE (Table 1; row-1,#T[26,j]) "needs defining 26, not Ny"
  rho*c_p*DELTAx*DELTAy/2*(T[1,j]-T_old[1,j])/DELTA=k*DELTAx*DELTAy/2*(T[1,j-1]+T[1,j+1]-2*T[1,j])+k*DELTAy*DELTAx*(T[2,j]-T[1,j]) "Bottom - i=1, except j=1 and Nx"
  rho*c_p*DELTAx*DELTAy/2*(T[Ny,j]-T_old[Ny,j])/DELTA=k*DELTAx*DELTAy/2*(T[Ny,j-1]+T[Ny,j+1]-2*T[Ny,j])+k*DELTAy*DELTAx*(T[Ny-1,j]-T[Ny,j]) "Top - i=Ny, except j=1 and Nx"
  Q_dot_top[j]=h_ext*(T[Ny,j]-T_ext)*DELTAx
end

"Left and right except corners"
duplicate i=2:Ny-1
  T_old[i,1]=TABLEVALUE (Table 1; row-1,#T[i,1])
  T_old[i,Nx]=TABLEVALUE (Table 1; row-1,#T[i,10]) "needs defining 10, not Nx"
  rho*c_p*DELTAx*DELTAy/2*(T[i,1]-T_old[i,1])/DELTA=k*DELTAx*DELTAy/2*(T[i,1]-T[i+1,1]-2*T[i,1])+k*DELTAy*DELTAx*(T[i,2]-T[i,1]) "Left - j=1, except i=1 and Ny"
  rho*c_p*DELTAx*DELTAy/2*(T[i,Nx]-T_old[i,Nx])/DELTA=k*DELTAx*DELTAy/2*(T[i,Nx]-T[i+1,Nx]-2*T[i,Nx])+k*DELTAy*DELTAx*(T[i,Nx-1]-T[i,Nx]) "Right - j=Nx, except i=1 and Ny"
  Q_dot_left[i]=h_ext*(T[i,1]-T_ext)*DELTAy
  Q_dot_right[i]=h_ext*(T[i,Nx]-T_ext)*DELTAy
end

"bottom-left corner: 1,1"
T_old[1,1]=TABLEVALUE (Table 1; row-1,#T[1,1])
rho*c_p*DELTAx*DELTAy/2*(T[1,1]-T_old[1,1])/DELTA=k*DELTAx*DELTAy/2*(T[2,1]-T[1,1])+k*DELTAy*DELTAx/2*(T[1,2]-T[1,1]) "h_ext*DELTAy/2*(T[1,1]-T_ext)"
Q_dot_left[1]=h_ext*(T[1,1]-T_ext)*DELTAy/2

"top-left corner: Ny,1"
T_old[Ny,1]=TABLEVALUE (Table 1; row-1,#T[26,1]) "needs defining 26, not Ny"
rho*c_p*DELTAx*DELTAy/2*(T[Ny,1]-T_old[Ny,1])/DELTA=k*DELTAx*DELTAy/2*(T[Ny,1]-T[Ny-1,1])+k*DELTAy*DELTAx/2*(T[Ny,2]-T[Ny,1]) "h_ext*(DELTAy/2+DELTAx/2)*(T[Ny,1]-T_ext)"
Q_dot_top[1]=h_ext*(T[Ny,1]-T_ext)*DELTAx/2
Q_dot_left[Ny]=h_ext*(T[Ny,1]-T_ext)*DELTAy/2

"bottom-right corner: 1,Nx"
T_old[1,Nx]=TABLEVALUE (Table 1; row-1,#T[1,10]) "needs defining 10, not Nx"
rho*c_p*DELTAx*DELTAy/2*(T[1,Nx]-T_old[1,Nx])/DELTA=k*DELTAx*DELTAy/2*(T[2,Nx]-T[1,Nx])+k*DELTAy*DELTAx/2*(T[1,Nx-1]-T[1,Nx]) "h_ext*DELTAy/2*(T[1,2Nx]-T_ext)"
Q_dot_right[1]=h_ext*(T[1,Nx]-T_ext)*DELTAy/2

"top-right corner: Ny,Nx"
T_old[Ny,Nx]=TABLEVALUE (Table 1; row-1,#T[26,10]) "needs defining 26,10, not Ny,Nx"
rho*c_p*DELTAx*DELTAy/2*(T[Ny,Nx]-T_old[Ny,Nx])/DELTA=k*DELTAx*DELTAy/2*(T[Ny,Nx]-T[Ny-1,Nx])+k*DELTAy*DELTAx/2*(T[Ny,Nx-1]-T[Ny,Nx]) "h_ext*(DELTAy/2+DELTAx/2)*(T[Ny,Nx]-T_ext)"
Q_dot_top[Nx]=h_ext*(T[Ny,Nx]-T_ext)*DELTAx/2
Q_dot_right[Ny]=h_ext*(T[Ny,Nx]-T_ext)*DELTAy/2

"Note: to have a correct graph from the arrays table order the columns in the arrays table"

"calculation of heat transfer rate in the top face"
Q_dot_top_tot=SUM(Q_dot_top[j],j=1:Nx)
"calculation of heat transfer rate in the left face"
Q_dot_left_tot=SUM(Q_dot_left[i],i=1:Ny)
"calculation of heat transfer rate in the right face"
Q_dot_right_tot=SUM(Q_dot_right[i],i=1:Ny)
"calculation of total heat transfer rate to the outside"
Q_dot_ext=Q_dot_top_tot+Q_dot_left_tot+Q_dot_right_tot

```

Figure 4.5.3 –Equations Window for the concrete beam cooling example.

Figure 4.5.4 presents temperature profiles in the 3 moments. They were obtained by choosing to represent Color Bands (after Plots → New Plot Window → X-Y-Z Plot, choosing Table – Arrays Table, 2-D table data, Color Bands). At any moment, the maximum temperature in the beam occurs at the middle of the base, and the minimum temperatures in the top corners. After 1 hour of cooling the maximum temperature is still 78°C and the minimum is 35°C; after 2 hours the maximum temperature is 72°C and the minimum 30°C; after 4 hours the maximum temperature is 60°C and the minimum 25°C. By choosing “Gradient Plot” one can also represent the heat flux vectors, which obviously are normal to the isothermal lines – see Figure 4.5.5.

A 3D type graph can be obtained, by choosing “3-D Surface”, and different options of scales and resolutions are available – see Figure 4.5.6. The resulting graph is generated by EES and may afterwards be rotated, allowing different perspective views of the dependent variable (temperature, Z) as a function of X and Y – see Figures 4.5.7 and 4.5.8.

Figure 4.5.9 represents the time evolution of the total heat transfer rate from the beam to the outside, varying between 2600 W and 700 W after 4 hours. After 2 hours the transfer rate is already down to about 1000 W.

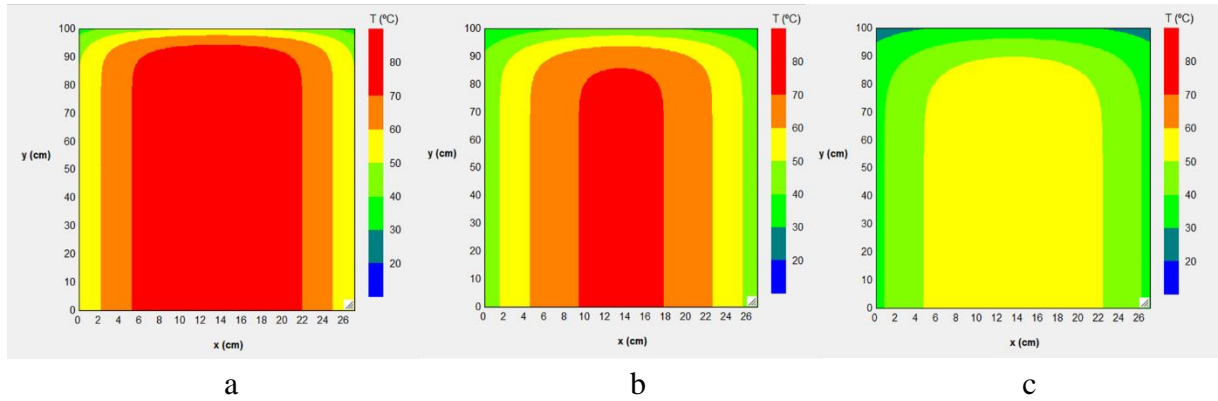


Figure 4.5.4 – Temperature profiles during cooling: (a) after 1 hour; (b) after 2 hours; (c) after 4 hours.

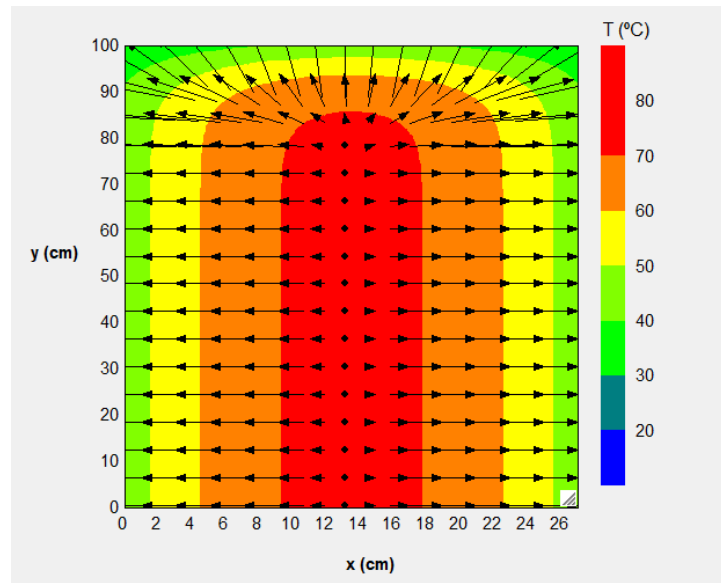


Figure 4.5.5 – Temperature colour bands and heat fluxes after 2 hours of cooling.

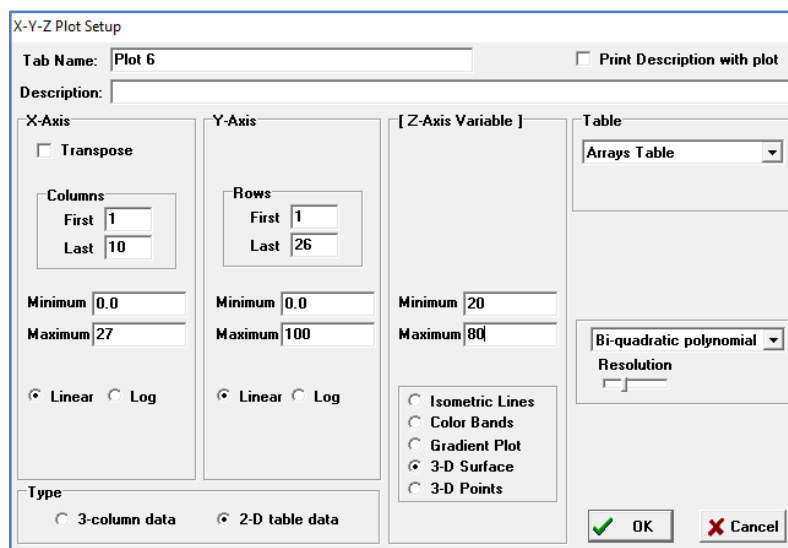


Figure 4.5.6 – Dialog window for X-Y-Z 3-D Surface Plot.

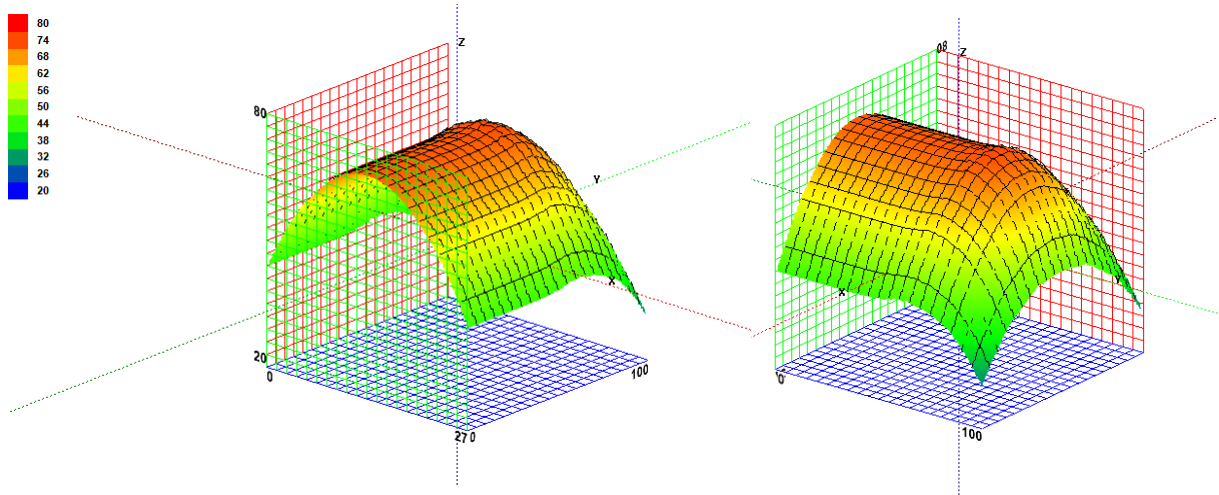


Figure 4.5.7 –Two views of a 3-D Surface Plot of temperature distribution after 2 hours.

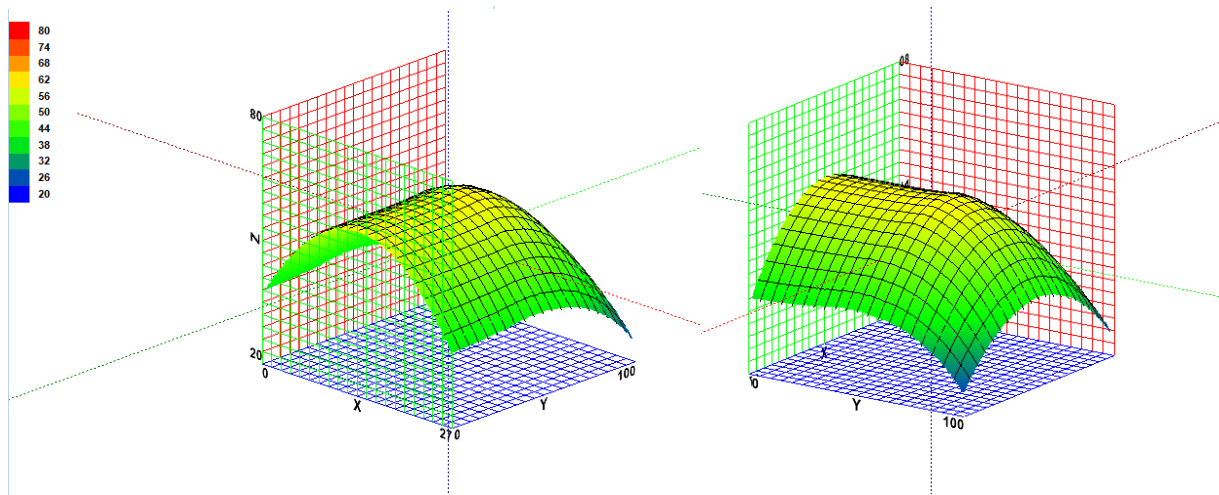


Figure 4.5.8 –Two views of a 3-D Surface Plot of temperature distribution after 4 hours.

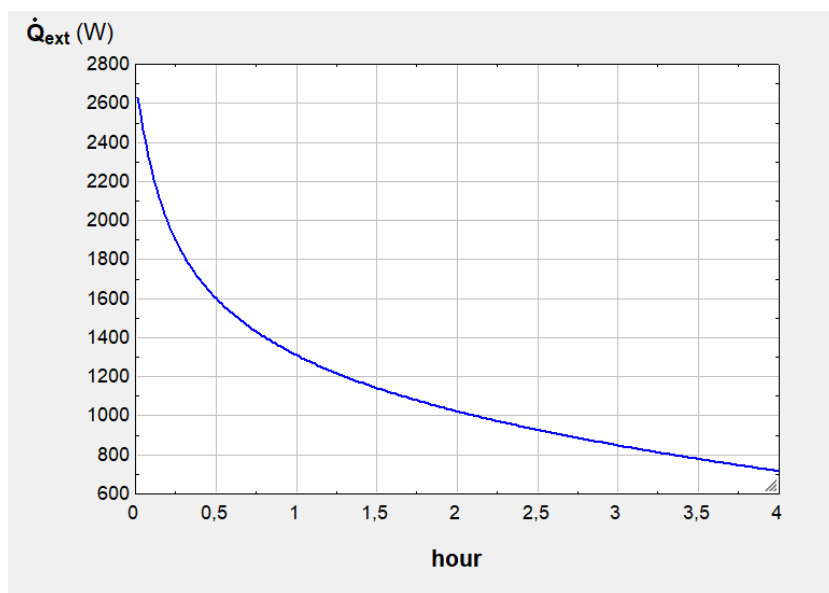


Figure 4.5.9 – Time evolution of the total heat transfer rate from the beam.

4.6 Dynamic heat transfer in a roof solar pond

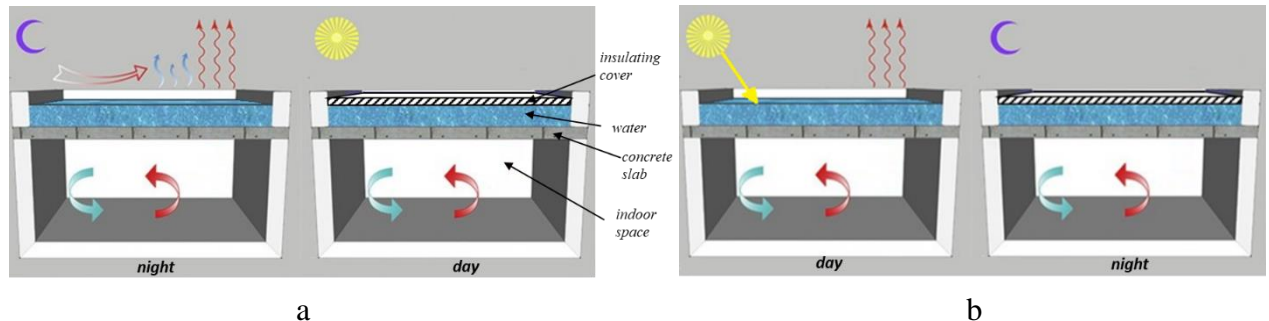


Figure 4.6.1 – Roof solar pond: (a) Summer operation; (b) Winter operation.

The above figure represents the operation of a roof pond, during the Summer and Winter seasons. The water is contained in a transparent plastic bag that does not affect heat transfer but prevents water evaporation.

In Summer – Figure 4.6.1(a) – at night the water loses heat to the outside, cooling the concrete slab (ceiling), which allows the removal of heat from the indoor space. During the daytime an insulating cover is placed over the water surface, limiting the transfer of heat with the outside – either the absorption of solar radiation, or conduction heat gains – so that the water is kept colder than the slab, allowing further indoor space cooling.

In Winter – Figure 4.6.1(b) – during the daytime the water receives solar radiation; in spite of the heat losses to the outside, the concrete slab (ceiling) is heated, which allows heating the indoor space. During the nighttime an insulating cover is placed over the water surface, limiting the transfer of heat with the outside, allowing further indoor space heating.

Consider a concrete slab and ceiling with 20 m² and a thickness of 15 cm ($\rho=2300$ kg/m³, $c_p=880$ J/kgK, $k=1.2$ W/mK) and a water pond with the same area and 20 cm thick ($\rho=1000$ kg/m³, $c_p=4190$ J/kgK). The convection coefficient to the outside air is equal to 20 W/m²K (constant) and the water emissivity is equal to 0.9 (equal to the absorption coefficient). The emissivity of the ceiling surface is equal to 0.9. Assume the following conditions for Summer and Winter operation:

- Summer: indoor space at a constant temperature of 24°C, with a variable heat transfer coefficient to the ceiling; the temperature of the water pond may be considered as uniform (due to convective currents), and the slab temperature varies along its thickness; in the slab upper surface the convective coefficient to the water also varies; during the night, with clear sky conditions, the effective sky temperature (for radiation exchanges) may be obtained with $T_{sky} = 0.0552 T_{amb}^{1.5}$, with temperatures in K;
- Winter: indoor space at a constant temperature of 20°C, with a variable heat transfer coefficient to the ceiling; consider the variation of the water pond temperature and the slab temperature along their thicknesses; the effective sky temperature (for radiation exchanges during the day) may be obtained with $T_{sky} = T_{amb} - 6$.

Use a numerical model and EES to obtain the variation of the slab temperature and water temperature, during one Summer day (24 hours) and one Winter day (24 hours), with the

climatic data in the following tables. Use a time step $\Delta t=300$ s, and consider that the cover is used between 6:00 and 20:00 in Summer, and between 17:00 and 8:00 in Winter. Analyse the cover schedule effect. Assume that, when used, the cover is a perfect thermal insulator. Calculate also the energy transferred with the indoor space during the two 24-hour periods.

<i>SUMMER - hour</i>	1	2	3	4	5	6	7	8	9	10	11	12
T_{amb} (°C)	17.6	17.2	17.2	16.6	16.0	15.4	15.2	17.2	19.6	22.2	23.6	25.4
I_{sol} (W/m ²)	0	0	0	0	0	39	206	408	594	747	861	897

<i>SUMMER - hour</i>	13	14	15	16	17	18	19	20	21	22	23	24/0
T_{amb} (°C)	27.0	29.0	31.6	32.0	33.2	33.0	30.8	27.0	24.8	24.0	22.0	18.0
I_{sol} (W/m ²)	889	831	725	569	367	169	150	58	0	0	0	0

<i>WINTER - hour</i>	1	2	3	4	5	6	7	8	9	10	11	12
T_{amb} (°C)	8.4	8.2	8.2	7.8	7.0	6.2	5.2	7.1	10.0	11.2	12.8	12.6
I_{sol} (W/m ²)	0	0	0	0	0	25	142	417	450	711	664	606

<i>WINTER - hour</i>	13	14	15	16	17	18	19	20	21	22	23	24/0
T_{amb} (°C)	14.0	14.8	15.6	16.4	17.2	16.2	12.6	11.6	11.0	11.0	10.6	8.4
I_{sol} (W/m ²)	522	214	356	286	58	0	0	0	0	0	0	0

Two different numerical models will be considered, due to the different climatic data, the different operating strategies, and the different thermal behaviour of the pond water in the 2 seasons (Summer and Winter).

In Summer, the slab will be warmer than the water, with the water receiving heat from the slab below it, which creates upward convective currents that tend to uniformize the water temperature. Therefore, a global model will be used for the water pond.

In Winter, the pond water will be warmer than the slab, and therefore no upward currents will occur; the water will be still, as the warmer elements will be on the top and the colder elements on the bottom. Thus, a distributed model, similar to the slab one, will be used in the water, taking into account its thermal storage capacity, and heat conduction, from the top to the bottom.

The 2 models, and respective results, will be presented separately, starting with the model for Summer operation.

4.6.1 Summer operation

The model will lead to the calculation of slab temperatures and water temperature, over time. Five volumes will be considered in the slab, together with a global water temperature. Figure 4.6.2 represents the temperatures and heat transfer rates.

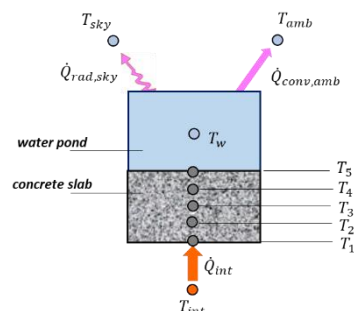


Figure 4.6.2 – Temperatures and heat transfer rates in the roof solar pond – Summer operation.

The discretised equation for the water global model, using the implicit formulation, is:

$$M_w c_{p,w} \frac{T_w^{t+\Delta t} - T_w^t}{\Delta t} = h_w^{t+\Delta t} (T_5^{t+\Delta t} - T_w^{t+\Delta t}) A - [(\dot{Q}_{conv,amb}^{t+\Delta t} + \dot{Q}_{rad,sky}^{t+\Delta t})]^{+} \quad (4.6.1)$$

where

$$\dot{Q}_{conv,amb}^{t+\Delta t} = h_{ext} (T_w^{t+\Delta t} - T_{amb}^{t+\Delta t}) A \quad (4.6.2)$$

$$\dot{Q}_{rad,sky}^{t+\Delta t} = \varepsilon_w \sigma (T_w^{t+\Delta t^4} - T_{sky}^{t+\Delta t^4}) A \quad (4.6.3)$$

To implement the model equations in EES, the cover condition ($[]^{+}$) will be defined with a FUNCTION or PROCEDURE, using a multiplying factor f_{cover} that is either 0 (cover on) or 1 (cover off).

Note that in equation (4.6.1) no water evaporation was considered, due to the plastic water container, which has no further thermal influence.

For the slab internal volumes ($i = 2$ to 4) the following discretised equation applies:

$$\rho_s c_{p,s} \Delta x \frac{(T_i^{t+\Delta t} - T_i^t)}{\Delta t} = \frac{k_s}{\Delta x_s} (T_{i-1}^{t+\Delta t} + T_{i+1}^{t+\Delta t} - 2T_i^{t+\Delta t}) \quad (4.6.4)$$

while for the volume in contact with water:

$$\rho_s c_{p,s} \frac{\Delta x}{2} \frac{(T_5^{t+\Delta t} - T_5^t)}{\Delta t} = \frac{k_s}{\Delta x_s} (T_4^{t+\Delta t} - T_5^{t+\Delta t}) - h_w^{t+\Delta t} (T_5^{t+\Delta t} - T_w^{t+\Delta t}) \quad (4.6.5)$$

The volume in contact with the indoor space will receive heat by convection (from the indoor air), but because the ceiling surface will have a significantly lower temperature than the other indoor surfaces, the thermal radiation effect will be considered. A detailed model would need to consider the temperatures of the different indoor surfaces, and respective view factors, but we will simply assume that all the surfaces except the ceiling will have the same temperature as the indoor air, in this case 24°C. Then, we may use a simple equation to include the thermal radiation effect, and have

$$\rho_s c_{p,s} \frac{\Delta x}{2} \frac{(T_1^{t+\Delta t} - T_1^t)}{\Delta t} = \frac{k_s}{\Delta x_s} (T_2^{t+\Delta t} - T_1^{t+\Delta t}) + h_{int}^{t+\Delta t} (T_{int} - T_1^{t+\Delta t}) + \varepsilon_s \sigma (T_{int}^4 - T_1^{t+\Delta t^4}) \quad (4.6.6)$$

Equations (4.6.1) to (4.6.6) allow calculating the 6 unknown temperatures in each time step. The convective heat transfer coefficients will be calculated in every step with the EES heat transfer database, as a function of the temperature differences (free convection). The heat rate removed from the space (\dot{Q}_{int}) is equal to the sum of the convective and radiative terms in equation (4.6.6), multiplied by the ceiling area.

Figure 4.6.3 shows the Equations Window for the Summer model. The values of the climatic variables were defined in a Lookup Table (“Lookup 1”) and interpolated to obtain the relevant values, step after step. The Parametric Table (“Table 1”) – see Figure 4.6.4 – starts with row/run number 1, with initial temperatures that were obtained after a few daily cycles, assuming the same day is repeated over and over. The f_{cover} factor was defined in Procedure COVER and the convective coefficients for horizontal surfaces were obtained with **fc_plate_horizontal1**,

from the EES heat transfer database, which is valid for an horizontal surface hotter than the above fluid (case of h_w), or an horizontal surface colder than the fluid below (h_{int}).

```

"Roof pond - Summer - July day"

Procedure cover (hour:f_cover) "factor=0 from 6 to 20"
If (hour<6) OR (hour>20) Then
  f_cover=1 "cover off"
Else
  f_cover=0 "cover on"
Endif
End

A_ceil =20 [m2]"roof pond, slab and ceiling area"
V_w=A_ceil*0.2 [m3] "water volume in m3"
L_slab=0,15 [m]
c_p_w=4190 [J/kgK]
rho_w=1000 [kg/m3]
epsilon_w=0,9
c_p_slab=880 [J/kgK]
rho_slab=2300 [kg/m3]
k_slab=1,2 [W/mK]
epsilon_slab=0,9
T_int=24 [°C]
h_ext=20 [W/m2K]

DELTA_t =60*5 [s] "5 minutes"
hour=time/3600
row=1+time/DELTA_t
T_amb=interpolate2(July;T_amb;hour;hour=hour)
T_sky=0,0552*(T_amb+273,15)^1,5-273,15 "in °C"

"water balance; implicit method"
Call COVER (hour:f_cover)
rho_w*V_w*c_p_w*(T_w-T_w_old)/DELTA_t=h_w*A_ceil*(T[5]-T_w)-f_cover*(h_ext*A_ceil*(T_w-T_amb)+epsilon_w*sigma#*A_ceil*((T_w+273,15)^4-(T_sky+273,15)^4))
T_w_old=tablevalue(Table 1; row-1;#T_w)
"slab - 5 volume elements; implicit method; 1 - interior; 5 - interface slab/water"
DELTA_x=L_slab/4
Duplicate i=2:4
  rho_slab*DELTA_x*c_p_slab*(T[i]-T_old[i])/DELTA_t=k_slab/DELTA_x*(T[i+1]+T[i-1]-2*T[i])
  T_old[i]=tablevalue(Table 1; row-1;#T[i])
End
rho_slab*DELTA_x/2*c_p_slab*(T[1]-T_old[1])/DELTA_t=h_int*(T_int-T[1])+epsilon_slab*sigma#*((T_int+273,15)^4-(T[1]+273,15)^4)+k_slab/DELTA_x*(T[2]-T[1])
Call fc_plate_horizontal1(Air; T[1];T_int;100; L_conv:h_int; Nusselt_int; Ra_int)
L_conv=A_ceil/sqrt(A_ceil)
T_old[1]=tablevalue(Table 1; row-1;#T[1])
rho_slab*DELTA_x/2*c_p_slab*(T[5]-T_old[5])/DELTA_t=k_slab/DELTA_x*(T[4]-T[5])-h_w*(T[5]-T_w)
T_old[5]=tablevalue(Table 1; row-1;#T[5])
Call fc_plate_horizontal1(Water; T[5]; T_w; 100; L_conv: h_w; Nusselt_w; Ra_ag)

"removal heat rate"
h_int_tot=h_int+epsilon_slab*sigma#*((T_int+273,15)^4-(T[1]+273,15)^4)/(T_int-T[1])
Q_dot_int=h_int_tot*(T_int-T[1])*A_ceil

```

Figure 4.6.3 – Equations Window for the roof solar pond example – Summer operation.

time [s]	hour	T ₁ [°C]	T ₂ [°C]	T ₃ [°C]	T ₄ [°C]	T ₅ [°C]	T _w [°C]	Q _{int} [W]	h _w [W/m ² K]	h _{int} [W/m ² K]	h _{int,tot} [W/m ² K]	T _{amb} [°C]	T _{sky} [°C]
0		21,35	20,75	20,22	19,73	19,23	19,05						
300	0,08333	21,35	20,76	20,22	19,73	19,2	19,02	399,9	111,7	2,272	7,557	17,97	1,032
600	0,1667	21,36	20,76	20,23	19,72	19,18	18,99	398,7	112,6	2,27	7,555	17,93	0,9854
900	0,25	21,37	20,77	20,23	19,72	19,15	18,96	397,7	113,4	2,268	7,554	17,9	0,9383
1200	0,3333	21,37	20,78	20,23	19,71	19,13	18,93	396,7	114,1	2,267	7,552	17,87	0,8912
1500	0,4167	21,38	20,78	20,23	19,7	19,1	18,9	395,7	114,8	2,265	7,55	17,83	0,8441
1800	0,5	21,38	20,79	20,24	19,69	19,07	18,88	394,8	115,4	2,263	7,549	17,8	0,797
2100	0,5833	21,39	20,79	20,24	19,68	19,05	18,85	394	116	2,262	7,548	17,77	0,75
2400	0,6667	21,39	20,8	20,24	19,67	19,02	18,82	393,3	116,5	2,261	7,546	17,73	0,7029
2700	0,75	21,4	20,8	20,23	19,66	19	18,79	392,6	117	2,259	7,545	17,7	0,6558
3000	0,8333	21,4	20,81	20,23	19,65	18,97	18,76	391,9	117,5	2,258	7,544	17,67	0,6088
3300	0,9167	21,41	20,81	20,23	19,64	18,95	18,73	391,4	117,9	2,257	7,543	17,63	0,5617
3600	1	21,41	20,81	20,23	19,62	18,92	18,7	390,9	118,4	2,256	7,543	17,6	0,5146
3900	1,083	21,41	20,81	20,22	19,61	18,89	18,67	390,5	118,8	2,256	7,542	17,57	0,4676
4200	1,167	21,41	20,81	20,22	19,6	18,87	18,64	390,1	119,1	2,255	7,541	17,53	0,4205
4500	1,25	21,42	20,81	20,22	19,58	18,84	18,62	389,8	119,5	2,254	7,541	17,5	0,3735
4800	1,333	21,42	20,81	20,21	19,57	18,82	18,59	389,6	119,8	2,254	7,541	17,47	0,3264
5100	1,417	21,42	20,81	20,21	19,55	18,79	18,56	389,4	120,2	2,254	7,54	17,43	0,2793
5400	1,5	21,42	20,81	20,2	19,54	18,76	18,53	389,3	120,5	2,254	7,54	17,4	0,2323
5700	1,583	21,42	20,81	20,19	19,53	18,74	18,5	389,3	120,8	2,253	7,54	17,37	0,1853
6000	1,667	21,42	20,81	20,19	19,51	18,71	18,47	389,3	121,1	2,253	7,54	17,33	0,1382

Figure 4.6.4 – Parametric Table initial rows for the roof solar pond example – Summer operation.

Figure 4.6.5 represents the evolution of temperatures and heat removal rate during the Summer day. Note that the slab is always warmer than the water, in particular $T_5 > T_w$, which means that the water pond removes indoor heat during the whole 24-hour period. The ceiling (T_1) has a very stable temperature, with a maximum of 21.4°C and a minimum of 20.8°C. The heat removal rate varies from 389 to 490 W, with a maximum value at about 10:30.

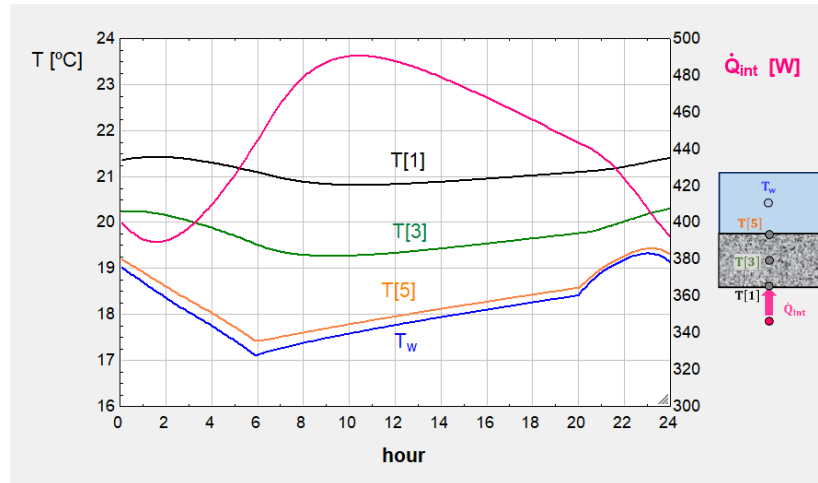


Figure 4.6.5 – Evolution of water and slab temperatures and removed heat – Summer operation.

Figure 4.6.6 shows the evolution of the variable heat transfer coefficients. The water convection coefficient has a larger variation (87 to 127 W/m²K), and the indoor coefficient is very stable; its value includes the convective and radiative contributions, with the radiative coefficient representing about twice the convective one.

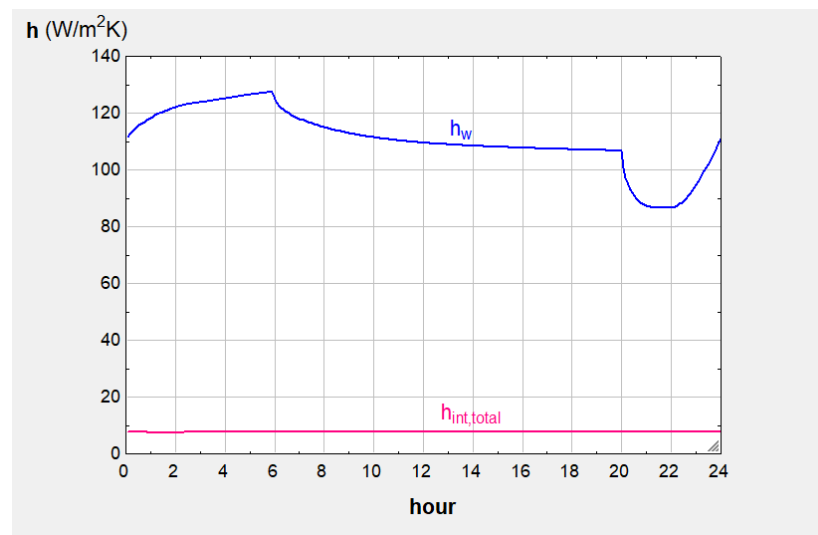


Figure 4.6.6 – Evolution of water and indoor heat transfer coefficients – Summer operation.

As can be seen in Figure 4.6.5, the water warms up during the daytime and is cooled at night, reducing its temperature by about 2.5°C between 23:00 and 6:00. This reduction is mostly due to the radiative cooling effect, which is a consequence of the low sky temperatures under clear sky conditions. Figure 4.6.7 compares the ambient air and sky temperatures, under those conditions.

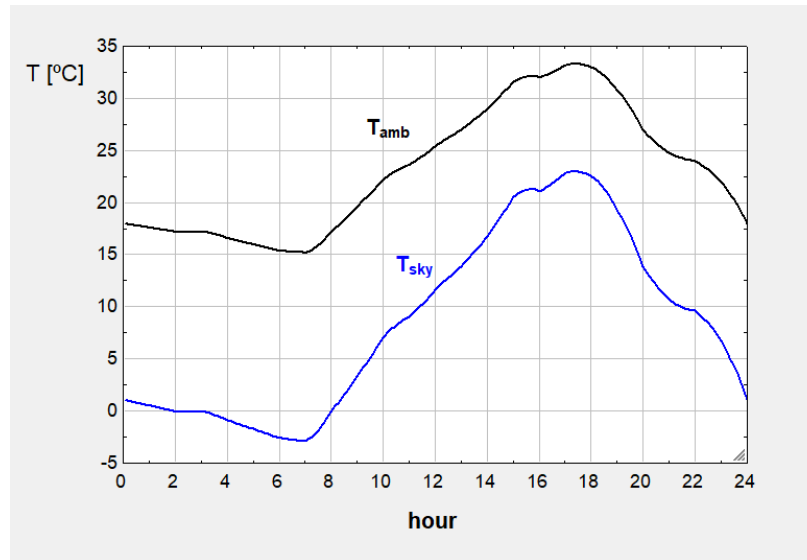


Figure 4.6.7 – Comparison of ambient air and sky temperatures with clear conditions – Summer operation.

Discontinuities in the water temperature evolution occur when the cover is removed and placed (at 6:00 and 20:00). It is noticeable from Figure 4.6.5 that after removing the cover at 20:00, the water still continues to heat up, due to the high outdoor air temperatures. During the period from 20:00 to 24:00 it would be better to keep the water covered, not increasing so much its temperature and improving the overall thermal performance (heat removal). This can be seen in Figure 4.6.8, which represents the evolutions when the cover is kept between 6:00 and 24:00. Those results were obtained by changing the cover schedule in the COVER Procedure of Figure 4.6.3. The minimum water temperature is now 16.8°C, and the maximum 18.4°C. The heat removal rate varies now from 440 to 522 W, which represents a significant increase by just adjusting the cover schedule. The cover could also be placed only after 6:00, as up to 7:00 the ambient and sky temperatures are still falling; but after 7:00 they will start to rise, and also solar radiation will start to affect the water temperature.

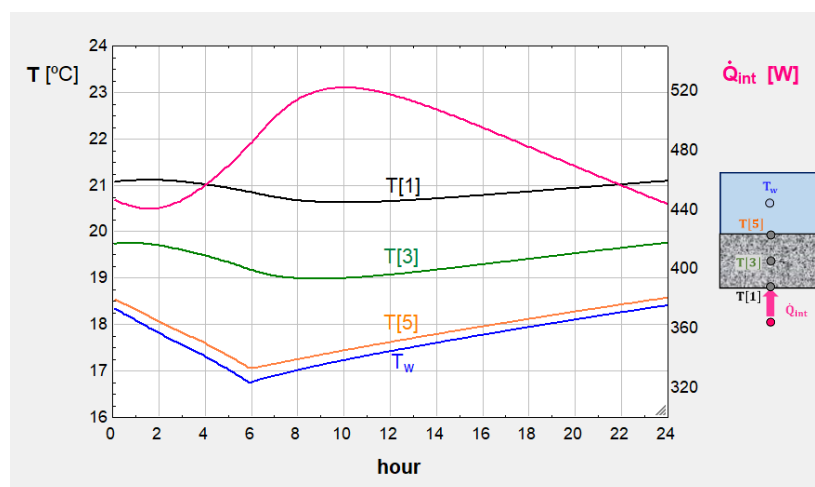


Figure 4.6.8 – Evolution of water and slab temperatures and removed heat with modified cover schedule – Summer operation.

4.6.2 Winter operation

The Winter operation model is based on the schematic discretisation of Figure 4.6.9. Besides the 5 nodes along the slab thickness ($\Delta x_s = 3.75$ cm), the water will also be divided with 5 equally spaced nodes ($\Delta x_w = 5$ cm). As heat is conducted in the downward direction, no convection currents will occur in the water; it will therefore be treated as a solid material. The interface between slab and water corresponds to slab node 5 and water node 1; Figure 4.6.9(b) shows the interface volume that includes 2 different materials.

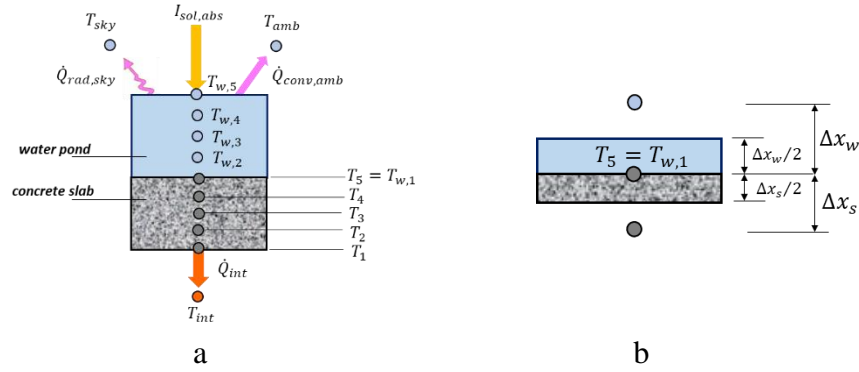


Figure 4.6.9 – Discretisation of the roof solar pond system – Winter operation: (a) temperatures and heat transfer rates; (b) interface (slab/water) volume.

The equations for slab nodes T_1 (4.6.6) and T_2 to T_4 (4.6.4) remain valid. For the interface volume ($T_5 = T_{w1}$) we may write

$$\left(\rho_s c_{p,s} \frac{\Delta x_s}{2} + \rho_w c_{p,w} \frac{\Delta x_w}{2} \right) \frac{(T_5^{t+\Delta t} - T_5^t)}{\Delta t} = \frac{k_s}{\Delta x_s} (T_4^{t+\Delta t} - T_5^{t+\Delta t}) + \frac{k_w}{\Delta x_w} (T_{w,2}^{t+\Delta t} - T_5^{t+\Delta t}) \quad (4.6.7)$$

For water internal nodes ($i = 2$ to 4):

$$\rho_w c_{p,w} \Delta x_w \frac{(T_{w,i}^{t+\Delta t} - T_{w,i}^t)}{\Delta t} = \frac{k_w}{\Delta x_w} (T_{w,i-1}^{t+\Delta t} + T_{w,i+1}^{t+\Delta t} - 2T_{w,i}^{t+\Delta t}) \quad (4.6.8)$$

and for the upper node ($T_{w,5}$):

$$\rho_w c_{p,w} \frac{\Delta x_w}{2} \frac{T_{w,5}^{t+\Delta t} - T_{w,5}^t}{\Delta t} = \frac{k_w}{\Delta x_w} (T_{w,4}^{t+\Delta t} - T_{w,5}^{t+\Delta t}) + f_{cover} \left[\alpha_w I_{sol} - h_{ext} (T_{w,5}^{t+\Delta t} - T_{amb}^{t+\Delta t}) - \varepsilon_w \sigma (T_{w,5}^{t+\Delta t^4} - T_{sky}^{t+\Delta t^4}) \right] \quad (4.6.9)$$

where f_{cover} will be equal to 0 when the cover is on (nighttime) and to 1 when the cover is off (daytime).

It is a set of 9 equations, plus those that define $h_{int}^{t+\Delta t}$ as a function of $(T_1^{t+\Delta t} - T_{int}^t)$, and the climatic variables ($I_{sol}, T_{amb}, T_{sky}$).

Figure 4.6.10 shows the Equations Window for the Winter model. The hourly values of the climatic variables for the Winter day were defined in the “Lookup 1” Table, to be interpolated and obtain the relevant values, step after step. As before, the Parametric Table (“Table 1”) starts with row/run number 1, with initial temperatures that were obtained after a few daily cycles, assuming the same day is repeated over and over. The f_{cover} factor was defined in the Procedure COVER, so that the pond is covered at night and receives solar radiation during the daytime. The convective coefficient for indoor air (h_{int}) was obtained with `fc_plate_horizontal2`, from

the EES heat transfer database, as now the horizontal surface (ceiling) is hotter than the fluid below.

```

Equations Window
"Roof pond - Winter day"
Procedure cover (hour:f_cover) "from 8 to 17 cover off"
If (hour<8) OR (hour>17) Then!
  f_cover=0 "cover on"
Else
  f_cover=1 "cover off"
Endif
End
End

A_ceil =20 [m2] "roof pond, slab and ceiling area"
L_slab=0.15 [m]
L_w=0.20 [m]
c_p_w=4190 [J/kgK]
rho_w=1000 [kg/m3]
k_w=0.6 [W/mK]
epsilon_w=0.9
c_p_slab=880 [J/kgK]
rho_slab=2300 [kg/m3]
k_slab=1.2 [W/mK]
epsilon_slab=0.9
T_int=20 [°C]
h_ext=20 [W/m2K]

DELTA=60*5 [s] "5 minutes"
hour=time/3600
row=1+time/DELTA
T_amb=interpolate2(Winter,T_amb,'hour',hour=hour)
I_sol=interpolate2(Winter,Rad Sol Hor,'hour',hour=hour)
T_sky=T_amb-6 "in °C"

"slab - 5 volume elements; implicit method; 1 - interior; 5 - interface slab/water"
DELTAx_slab=L_slab/4
Duplicate i=2,4
  rho_slab*DELTAx_slab*c_p_slab*(T[i]-T_old[i])/DELTA=k_slab*DELTAx_slab*(T[i+1]+T[i]-2*T[i])
  T_old[i]=tablevalue(Table 1; row-1,#T[i])
End
rho_slab*DELTAx_slab/2*c_p_slab*(T[1]-T_old[1])/DELTA=h_int*(T_int-T[1])+epsilon_slab*sigma*((T_int+273.15)^4-(T[1]+273.15)^4)+k_slab/DELTAx_slab*(T[2]-T[1])
Call fc_plate_horizontal2(Air; T[1];T_int;100; L_conv.h_int; Nusselt_int; Ra_int)
L_conv=A_ceil/sqrt(A_ceil)
T_old[1]=tablevalue(Table 1; row-1,#T[1])
"interface volume: T_w[1]=T[5]"
(rho_slab*DELTAx_slab/2*c_p_slab+rho_w*DELTAx_w/2*c_p_w)*(T[5]-T_old[5])/DELTA=k_slab*DELTAx_slab*(T[4]-T[5])+k_w*DELTAx_w*(T_w[2]-T[5])
T_old[5]=tablevalue(Table 1; row-1,#T[5])
"water - 5 volume elements; implicit method; 1 - water interface; 5 - external surface"
DELTAx_w=L_w/4
Duplicate i=2,4
  rho_w*DELTAx_w*c_p_w*(T_w[i]-T_w_old[i])/DELTA=k_w*DELTAx_w*(T_w[i+1]+T_w[i]-2*T_w[i])
  T_w_old[i]=tablevalue(Table 1; row-1,#T_w[i])
End
T_w[1]=T[5]
T_w_old[1]=tablevalue(Table 1; row-1,#T_w[1])
Call COVER (hour:f_cover)
rho_w*c_p_w*DELTAx_w*(T_w[5]-T_w_old[5])/DELTA=k_w*DELTAx_w*(T_w[4]-T_w[5])+f_cover*(epsilon_w*h_ext*(T_w[5]-T_amb)-epsilon_w*sigma*((T_w[5]+273.15)^4-(T_sky+273.15)^4))
T_w_old[5]=tablevalue(Table 1; row-1,#T_w[5])

"removal heat rate"
h_int_tot=h_int+epsilon_slab*sigma*((T[1]+273.15)^4-(T_int+273.15)^4)/(T[1]-T_int)
Q_dot_int=h_int_tot*(T[1]-T_int)*A_ceil

```

Figure 4.6.10 – Equations Window for the roof solar pond example – Winter operation.

Figure 4.6.11 represents the ambient air and sky temperatures, and incident horizontal solar radiation, during the Winter day.

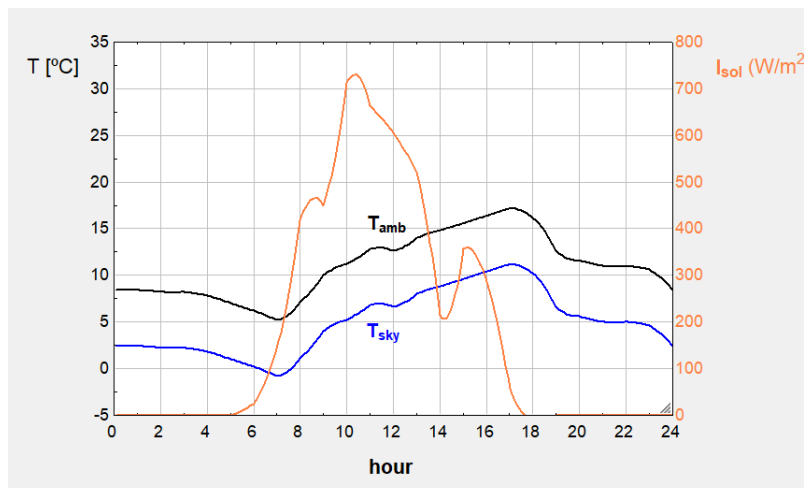


Figure 4.6.11 – Ambient air and sky temperatures, and incident solar radiation – Winter operation.

Figure 4.6.12 represents the evolution of temperatures and heat input rate during the Winter day. Note that the slab and lower water volumes have very stable temperatures, with the ceiling temperature (T_1) varying between 21.6 and 21.7°C. As a consequence, the heat input rate (\dot{Q}_{int}) varies only between 176 and 181 W. The indoor heat transfer coefficient is mostly due to radiation, as free convection is not significant with a warmer ceiling, with a total of about 5 W/m²K.

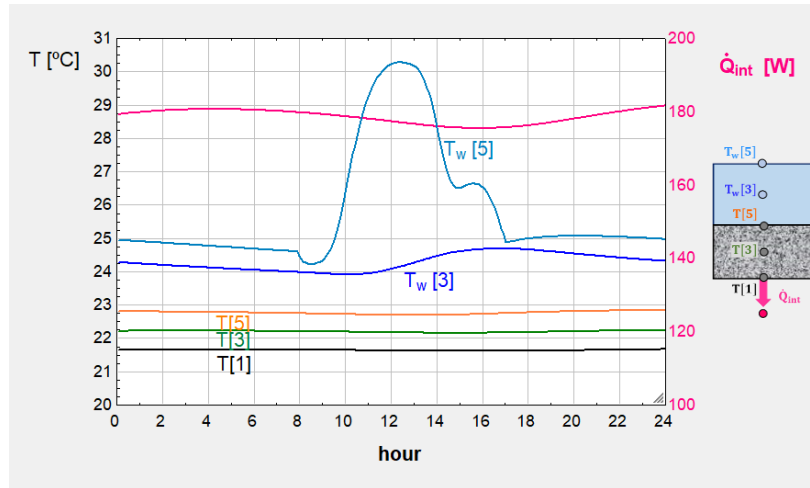


Figure 4.6.12 – Evolution of water and slab temperatures and input heat – Winter operation.

There is a large temperature swing in the top water volume during the daytime, due to the absorption of solar radiation. The minimum top water temperature is equal to 24.2°C, and the maximum (at 12:30) is 30.3°C. Because of the water being still and its low thermal conductivity, it is difficult to conduct heat to the other water layers and slab; this means that a large part of the solar gains are lost again to the outside. It would be better to increase the heat transfer to the lower depths of water, by agitating it. If a water mixing or stirring device was used, then the water temperature would (ideally) be uniform. To assess this effect, the model was adapted, becoming similar to the Summer one (global water temperature). The differences lay on the climatic data, the cover schedule, and the use of the function `fc_plate_horizontal2`, instead of `fc_plate_horizontal1`, for the water free convection coefficient between the water and the slab, due to the water being warmer than the slab; of course, the stirring would increase the coefficient, compared to pure free convection, so this is a conservative value. Figure 4.6.13 shows the new simulation results.

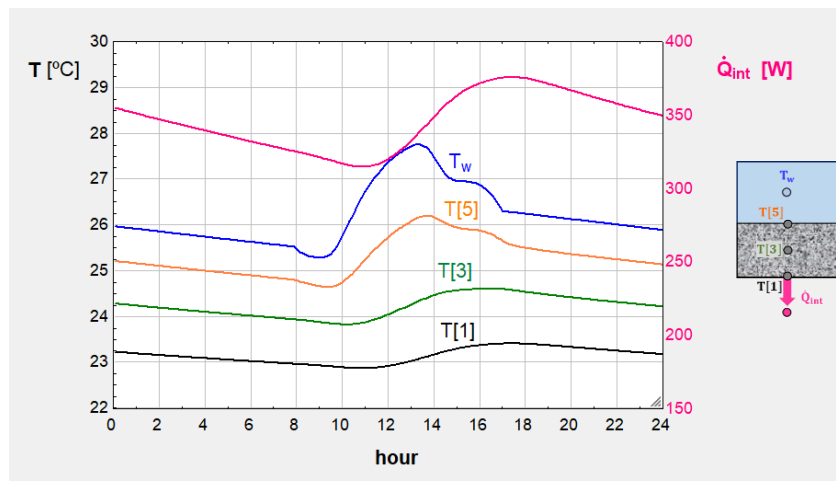


Figure 4.6.13 – Evolution of mixed water and slab temperatures and input heat – Winter operation.

As Figure 4.6.13 shows, more solar gains are now transmitted by conduction to the slab, and the different slab volumes have larger temperature swings, with higher temperatures. The indoor heat input would now be much larger, varying between 315 (at 11:00) and 376 W (at 17:00). Therefore, even a slight agitation of the water (if not ideal) will be beneficial to increase the heating performance of the roof solar pond.

Figure 4.6.14 represents the water free convection coefficient and the total (radiation and convection) indoor heat transfer coefficient. They are much lower than the Summer operation values, due to the different heat flow direction. The water coefficient is more stable than in Summer, around $20 \text{ W/m}^2\text{K}$, increasing during the daytime when the temperature difference is higher. The indoor coefficient is almost constant, at $5.5 \text{ W/m}^2\text{K}$, and most of it due to radiation.

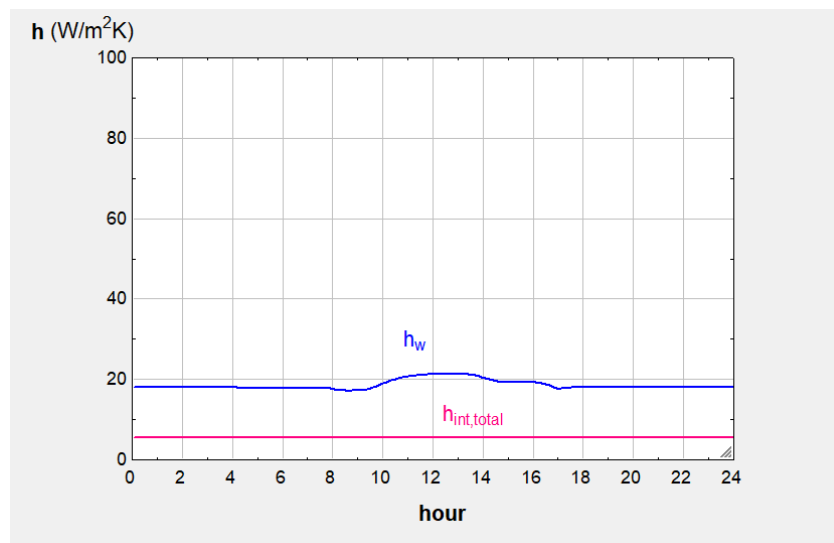


Figure 4.6.14 – Evolution of mixed water and indoor heat transfer coefficients – Winter operation.

4.7 Heat transfer in a laminar flow in a tube



Figure 4.7.1 – Laminar flow in a tube.

Consider a laminar flow of water in a 25 mm diameter tube, with a length of 10 m. The velocity profile at the tube inlet is developed, with a velocity at the axis of 6.5 cm/s. The inlet temperature is equal to 80°C (for all values of r).

The tube wall loses heat to the outside air at 20°C , with an outside heat transfer coefficient of $20 \text{ W/m}^2\text{K}$. Using EES with an appropriate grid, neglecting viscous dissipation and conduction in the wall (negligible thickness), obtain the flow temperature distribution and the evolution of the convection heat transfer coefficient along x (calculated after obtaining the fluid temperature $T(r, x)$).

Figure 4.7.2 resumes the problem conditions. The flow is laminar: as the average velocity is equal to $v_{max}/2$, the Reynolds number is $Re=2225$.

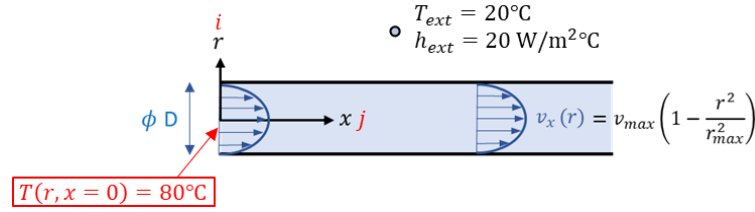


Figure 4.7.2 – Conditions for the laminar water flow in a tube, with fully developed velocity profile.

To obtain the flow temperature distribution we need to discretise the domain. For that, we will assume a 2D temperature distribution; as the circumferential conditions (around the tube section perimeter) can be considered constant, the temperature will vary along the radius (r) and flow length (x). We will then consider several volume elements with a ring shape, with a radial dimension equal to Δr and a length equal to Δx . Figure 4.7.3 represents a general control volume element, as well as the energy exchanges related to it.

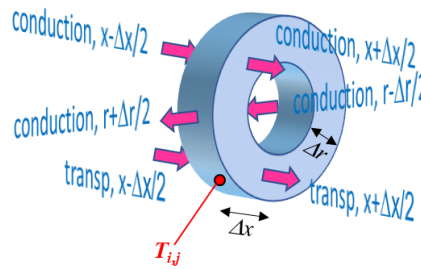


Figure 4.7.3 – General control volume element with radial and axial temperature variation and energy exchanges.

The flow section area for this elementary volume (identified with indices i, j) is:

$$\Delta S_i = \pi \left((r_i + \Delta r/2)^2 - (r_i - \Delta r/2)^2 \right) \quad (4.7.1)$$

and its energy balance, neglecting viscous dissipation, states that the change in energy transported by the flow leaving the volume is a result of the conductive heat exchanges with the surrounding volumes:

$$\rho c_p v_x(r_i) \Delta S_i (T_{i,j} - T_{i,j-1}) = \frac{k}{\Delta r} 2\pi \left(r_i - \frac{\Delta r}{2} \right) \Delta z (T_{i-1,j} - T_{i,j}) + \frac{k}{\Delta r} 2\pi \left(r_i + \frac{\Delta r}{2} \right) \Delta z (T_{i+1,j} - T_{i,j}) + \frac{k \Delta S_i}{\Delta z} (T_{i,j-1} + T_{i,j+1} - 2T_{i,j}) \quad (4.7.2)$$

Note that when expressing the transported energy in and out of the control volume (i, j), the flow temperatures at the borders are considered as the upstream values – see Figure 4.7.4.

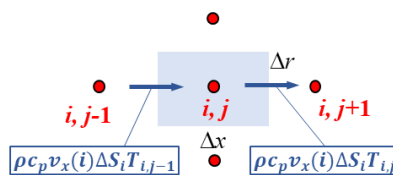


Figure 4.7.4 – General control volume (i, j) and upstream temperatures at flow inlet and outlet.

Assuming a total of N_r nodes/volumes along r and N_x along x , equation (4.7.2), together with the expression for the parabolic velocity profile, are valid for nodes $i = 2$ to $N_r - 1$ and $j = 2$ to $N_x - 1$. For nodes $(i, 1)$ – tube entrance – the temperatures are known (80°C).

The remaining nodes are those at the axis – $(1, j = 2$ to $N_x - 1)$ – those at the wall ($N_r, j = 2$ to $N_x - 1)$ – and those at the exit section ($j = N_x$). The volumes at the axis are small cylinders with a radius $\Delta r/2$ – see Figure 4.7.5(a). The volumes at the wall have a radial length of $\Delta r/2$ and, as the node is located at the wall, the flow velocity in and out of the volume is zero; the volumes also transfer heat to the outside (at 20°C) – see Figure 4.7.5(b). The exit volumes ($i = 1$ to N_r, N_x) – see Figure 4.7.5(c) – are the only ones with a length of $\Delta x/2$, as the $(i, 1)$ entrance volumes are not used, because of the fixed inlet temperature; we have to distinguish the $(i = 2$ to $N_r - 1, N_x)$ volumes and the 2 limits – $(1, N_x)$ and (N_r, N_x) .

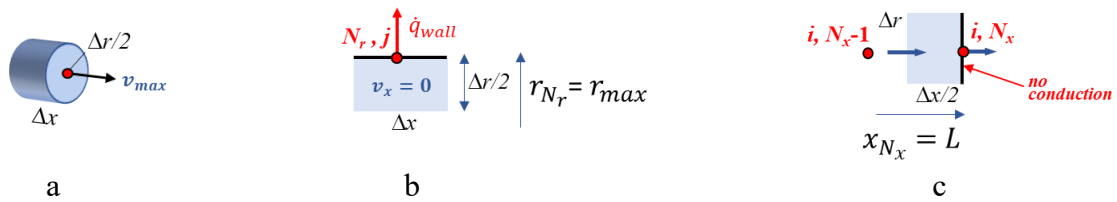


Figure 4.7.5 – Boundary volumes: (a) at the axis; (b) at the wall; (c) at the tube exit.

At the exit volumes – see Figure 4.7.5(c) – the transport or convective energy term will use the upstream temperature at the inlet ($T_{i, N_x - 1}$), and the node temperature (T_{i, N_x}) at the exit; and because there are no nodes after (i, N_x) , the conduction term at the exit section will be neglected – this has a negligible effect in the energy balance, as the axial conduction has much less importance (smaller gradients) than radial conduction.

The $N_r \times N_x$ equations (including the fixed inlet temperature) can be written in EES with double index/array variables, as seen before. The grid size is more decisive in the radial direction, because of the higher temperature gradients at the wall. Results will be presented for a 21×41 grid, with a constant $\Delta r = 0.625$ mm and $\Delta x = 250$ mm. This grid has similar results to a 41×81 , or even larger grids, as Δr is already considerably small. Results for smaller grids will later be compared.

Figure 4.7.6 presents the Formatted Equations Window, and Figure 4.7.7 the Equations Window. Other array variables are the $r[i]$ and $x[j]$ coordinates, the wall surface temperature ($T_s[j]$), the mixed mean or average temperature in the section ($T_{ave}[j]$) and the local convection coefficient ($h_{conv}[j]$). The convection coefficient is calculated after the wall and flow section average temperatures, as the heat flux at the wall divided by the temperature difference; and, as the heat flux at the wall is equal to the heat flux transferred to the outside:

$$h_{conv}[j] = \frac{\dot{q}_{wall}}{(T_{ave}[j] - T_s[j])} = \frac{h_{ext}(T_s[j] - T_{ext})}{(T_{ave}[j] - T_s[j])} \quad (4.7.3)$$

The section average temperature, or mixed mean temperature, is calculated from the available section temperatures:

$$T_{ave}[j] = \frac{\sum_{i=1}^{N_r} T_{i,j}(vA)_i}{\sum_{i=1}^{N_r} (vA)_i} \quad (4.7.4)$$

where A_i is the i element of section area – same as ΔS_i in equation (4.7.1).

At the end of the 2 windows, two EES procedures (**pipeflow** and **pipeflow_{local}**) from the heat transfer database are called, to compare the calculated convective coefficients with the theoretical available values.

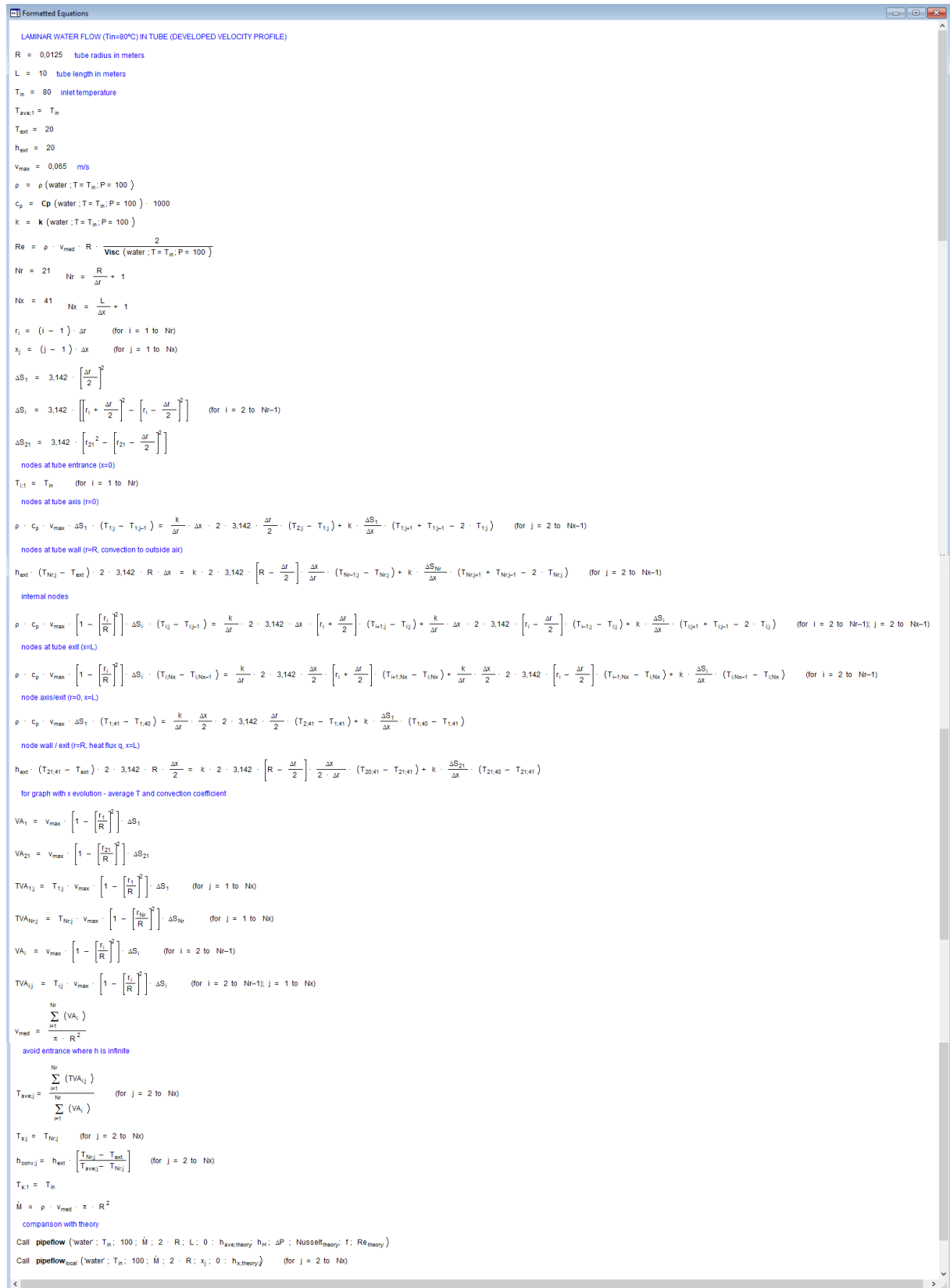


Figure 4.7.6 – Formatted Equations Window for the heat transfer in a laminar flow example.

```

Equations Window
"LAMINAR WATER FLOW (Tm=80°C) IN TUBE (DEVELOPED VELOCITY PROFILE)"
R=0,0125 "tube radius in meters"
L=10 "tube length in meters"
T_in=80 "inlet temperature"
T_ave[1]=T_in
T_ext=20
h_ext=20
v_max=0,065 "m/s"
rho=Density(Water,T=T_in,P=100)
c_p=Cp(Water,T=T_in,P=100)*1000
k=Conductivity(Water,T=T_in,P=100)
Re=rho*v_med*R^2/viscosity(Water,T=T_in,P=100)
Nr=21, Nr=R/DELTAx+1
Nx=41, Nx=L/DELTAx+1
duplicate i=1, Nr
  r[i]=(i-1)*DELTAx
end
duplicate j=1, Nx
  x[j]=(j-1)*DELTAx
end
DELTAx[1]=pi#*(DELTAx/2)^2
duplicate i=2, Nr-1
  DELTAx[i]=pi#*((r[i]+DELTAx/2)^2-(r[i]-DELTAx/2)^2)
jend
DELTAx[Nr]=pi#*(r[Nr]^2-(r[Nr]-DELTAx/2)^2)
"nodes at tube entrance (x=0)"
duplicate i=1, Nr
  T[i,1]=T_in
end
"nodes at tube axis (r=0)"
duplicate j=2, Nx-1
  rho*c_p*v_max*DELTAx[1]*(T[1,j]-T[1,j-1])=k*DELTAx*DELTAx^2*pi#*DELTAx/2*(T[2,j]-T[1,j])+k*DELTAx[1]*DELTAx*(T[1,j+1]+T[1,j]-2*T[1,j])
end
"nodes at tube wall (r=R, convection to outside air)"
duplicate j=2, Nx-1
  h_ext*(T[Nr,j]-T_ext)^2*pi#*R*DELTAx=k^2*pi#*(R-DELTAx/2)*DELTAx*DELTAx*(T[Nr-1,j]-T[Nr,j])+k*DELTAx[Nr]*DELTAx*(T[Nr,j+1]+T[Nr,j]-2*T[Nr,j])
end
"internal nodes"
duplicate i=2, Nr-1
duplicate j=2, Nx-1
  rho*c_p*v_max*(1-(r[i]/R)^2)*DELTAx[i]*(T[i,j]-T[i,j-1])=k*DELTAx^2*(r[i]+DELTAx/2)*(T[i+1,j]-T[i,j])+k*DELTAx*DELTAx^2*pi#*(r[i]-DELTAx/2)*(T[i-1,j]-T[i,j])+k*DELTAx[i]*DELTAx*(T[i,j+1]+T[i,j]-2*T[i,j])
end
"nodes at tube exit (x=L)"
duplicate i=2, Nr-1
  rho*c_p*v_max*(1-(r[i]/R)^2)*DELTAx[i]*(T[i,Nx]-T[i,Nx-1])=k*DELTAx^2*pi#*DELTAx/2*(r[i]+DELTAx/2)*(T[i+1,Nx]-T[i,Nx])+k*DELTAx*DELTAx^2*pi#*(r[i]-DELTAx/2)*(T[i-1,Nx]-T[i,Nx])+k*DELTAx[i]*DELTAx*(T[i,Nx+1]-T[i,Nx])
end
"node axis/exit (r=0, x=L)"
rho*c_p*v_max*DELTAx[1]*(T[1,Nx]-T[1,Nx-1])=k*DELTAx*DELTAx^2*pi#*DELTAx/2*(T[2,Nx]-T[1,Nx])+k*DELTAx[1]*DELTAx*(T[1,Nx+1]-T[1,Nx])
"node wall / exit (r=R, heat flux q_s)"
h_ext*(T[Nr,Nx]-T_ext)^2*pi#*R*DELTAx/2=k^2*pi#*(R-DELTAx/2)*DELTAx/2*DELTAx*(T[Nr-1,Nx]-T[Nr,Nx])+k*DELTAx[Nr]*DELTAx*(T[Nr,Nx+1]-T[Nr,Nx])

"for graph with x evolution - average T and convection coefficient"
VA[1]=v_max*(1-(r[1]/R)^2)*DELTAx[1]
VA[Nr]=v_max*(1-(r[Nr]/R)^2)*DELTAx[Nr]
duplicate j=1, Nx
  TVA[1,j]=T[1,j]*v_max*(1-(r[1]/R)^2)*DELTAx[1]
  TVA[Nr,j]=T[Nr,j]*v_max*(1-(r[Nr]/R)^2)*DELTAx[Nr]
end
duplicate i=2, Nr-1
  VA[i]=v_max*(1-(r[i]/R)^2)*DELTAx[i]
duplicate j=1, Nx
  TVA[i,j]=T[i,j]*v_max*(1-(r[i]/R)^2)*DELTAx[i]
end
v_med=sum(VA[i],i=1,Nr)/(pi*R^2)
duplicate j=2, Nx "avoid entrance where h is infinite"
  T_ave[j]=sum(TVA[i,j],i=1,Nr)/sum(VA[i],i=1,Nr)
  T_s[j]=T[Nr,j]
  h_conv[j]=h_ext*(T[Nr,j]-T_ext)/(T_ave[j]-T[Nr,j])
end
T_s[1]=T_in
M_dot=rho*v_med*pi*R^2
"comparison with theory"
call PipeFlow(Water,T_in,100,M_dot,2*R,L,0,h_ave_theory, h_H,DELTAx, Nusselt_theory, f, Re_theory)
duplicate j=2, Nx
  call PipeFlow_local(Water,T_in,100,M_dot,2*R,x[j],0,h_x_theory[j], h_x_H_theory[j], dPdx[j])
end

```

Figure 4.7.7 – Equations Window for the heat transfer in a laminar flow example.

After performing the calculations (Calculate → Solve) all the main results, namely temperatures and convection coefficients, will be available in the Arrays Table. With 21×41 nodes, the table will contain 21 rows and 41 columns just for the fluid temperatures; there will be 41 rows for variables such as $T_s[j]$ and $h_{conv}[j]$. Figure 4.7.8 shows a partial view of the Arrays Table.

From this table several graphical outputs may be obtained. Figure 4.7.9 shows the temperature distribution in 3 different flow sections (x coordinates). As can be seen, the water reduces its temperature along the flow direction, as it loses heat to the outside, and a large part of the sections has a small temperature variation; the variation is larger in the region near the wall. Figure 4.7.10 shows a Color Bands graph representing the 2D variation of $T(x, r)$.

Figure 4.7.11 represents the change of the section average temperature, of the wall surface temperature, and of the convection coefficient along the flow length. Figure 4.7.12 shows the evolution of the calculated convection coefficient compared to existing theoretical models obtained with the EES procedure: constant wall temperature or constant wall heat flux. Our example has variable T_s and variable \dot{q}_s , with h_{conv} values standing between those 2 cases.

Sort	ΔS_i	$h_{conv,i}$	r_i	$T_{t,1}$	$T_{t,2}$	$T_{t,3}$	$T_{t,4}$	$T_{t,5}$	$T_{t,6}$	$T_{t,7}$	$T_{t,8}$	$T_{t,9}$
[1]	3.068E-07		0	80	80	80	80	79.99	79.98	79.96	79.93	79.89
[2]	0.000002454	292	0.000625	80	80	80	80	79.99	79.98	79.96	79.93	79.88
[3]	0.000004909	222,6	0.00125	80	80	80	80	79.99	79.97	79.95	79.91	79.86
[4]	0.000007363	193,1	0.001875	80	80	80	79.99	79.98	79.96	79.93	79.89	79.83
[5]	0.000009817	176	0.0025	80	80	80	79.99	79.97	79.94	79.9	79.85	79.77
[6]	0.00001227	164,4	0.003125	80	80	79.99	79.98	79.96	79.92	79.86	79.78	79.69
[7]	0.00001473	156	0.00375	80	80	79.99	79.97	79.93	79.87	79.79	79.69	79.57
[8]	0.00001718	149,6	0.004375	80	80	79.98	79.95	79.89	79.8	79.69	79.56	79.41
[9]	0.00001963	144,5	0.005	80	79.99	79.96	79.91	79.82	79.7	79.55	79.38	79.19
[10]	0.00002209	140,3	0.005625	80	79.98	79.94	79.85	79.72	79.55	79.36	79.14	78.91
[11]	0.00002454	136,9	0.00625	80	79.97	79.89	79.75	79.57	79.34	79.09	78.82	78.54
[12]	0.000027	134	0.006875	80	79.95	79.81	79.61	79.35	79.05	78.74	78.41	78.08
[13]	0.00002945	131,6	0.0075	80	79.9	79.68	79.39	79.04	78.67	78.29	77.91	77.53
[14]	0.00003191	129,5	0.008125	80	79.82	79.49	79.07	78.62	78.17	77.73	77.29	76.88
[15]	0.00003436	127,6	0.00875	80	79.68	79.19	78.63	78.09	77.56	77.05	76.58	76.12
[16]	0.00003682	126	0.009375	80	79.46	78.76	78.06	77.42	76.82	76.27	75.77	75.29
[17]	0.00003927	124,6	0.01	80	79.11	78.17	77.34	76.62	75.98	75.4	74.87	74.38
[18]	0.00004172	123,4	0.01063	80	78.58	77.41	76.48	75.72	75.05	74.46	73.92	73.42
[19]	0.00004418	122,3	0.01125	80	77.85	76.5	75.52	74.73	74.06	73.46	72.93	72.43
[20]	0.00004663	121,4	0.01188	80	76.91	75.49	74.5	73.72	73.05	72.46	71.93	71.44
[21]	0.00004909	120,5	0.0125	80	75.84	74.44	73.47	72.7	72.05	71.47	70.95	70.47
[22]		119,7										
[23]		119,1										
[24]		118,5										
[25]		117,9										
[26]		117,4										
[27]		117										
[28]		116,6										
[29]		116,2										
[30]		115,9										
[31]		115,6										
[32]		115,4										
[33]		115,2										
[34]		114,9										

Figure 4.7.8 – Arrays Table for the heat transfer in a laminar flow example.

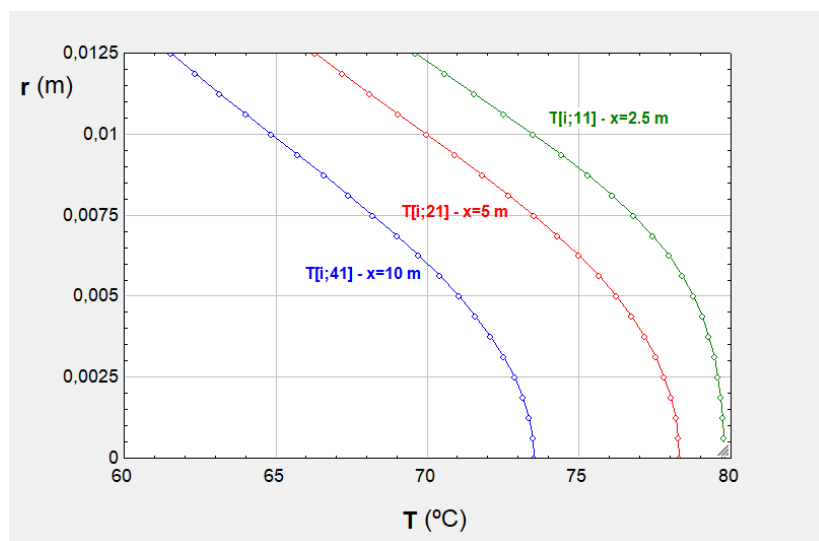


Figure 4.7.9 – Temperature distribution in 3 flow sections for the heat transfer in a laminar flow example.

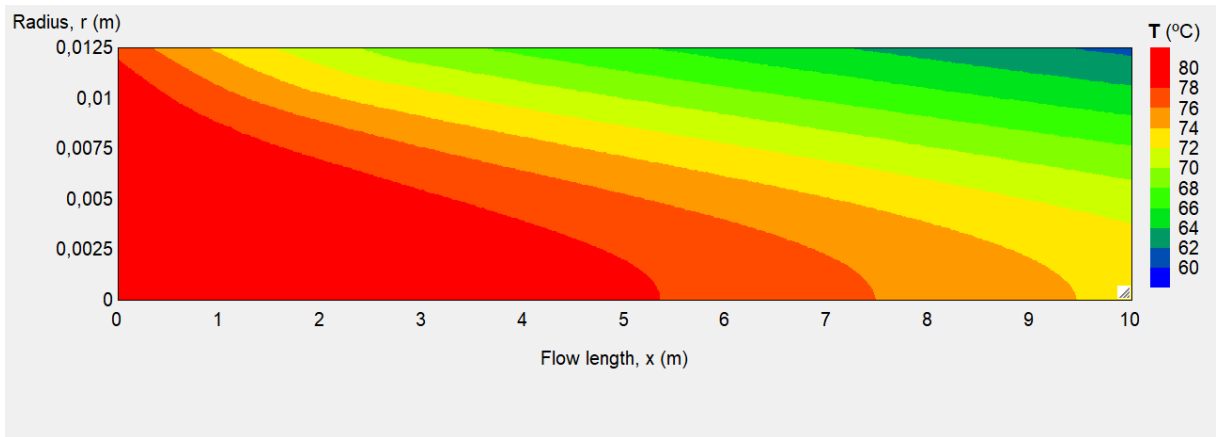


Figure 4.7.10 – Temperature distribution in the heat transfer laminar flow example – colour bands.

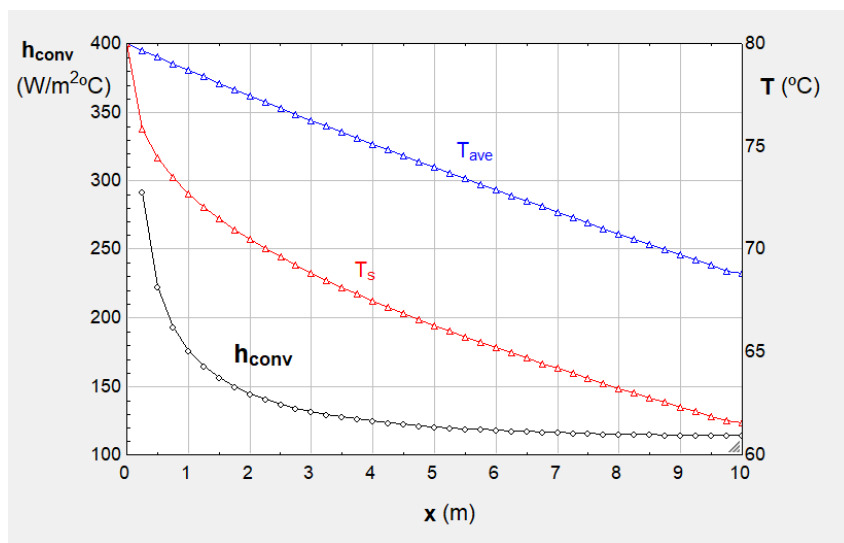


Figure 4.7.11 – Evolution of average temperature, wall surface temperature and convection coefficient along the flow length.

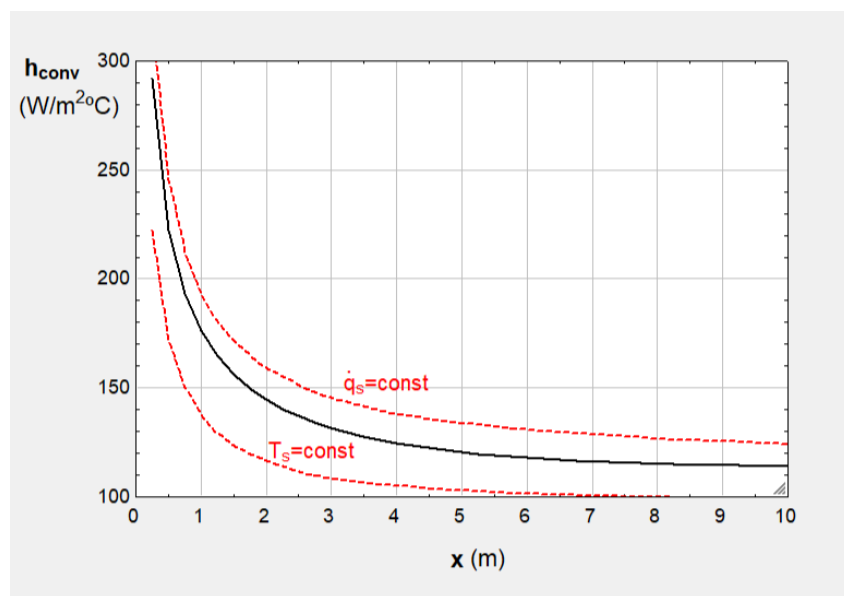


Figure 4.7.12 – Comparison of the calculated convection coefficient with available theoretical solutions.

As said before, the previous results were obtained with a uniform 21×41 grid, with $\Delta r = 0.625$ mm and $\Delta x = 250$ mm. A larger number of nodes leads to negligible differences. A smaller number of nodes was also tested. The number of radial nodes is more important, and Figure 4.7.13 shows additional results for a 11×41 grid and a 6×41 grid. There is a more noticeable difference only for the 6 radial nodes grid (with $\Delta r = 2.5$ mm).

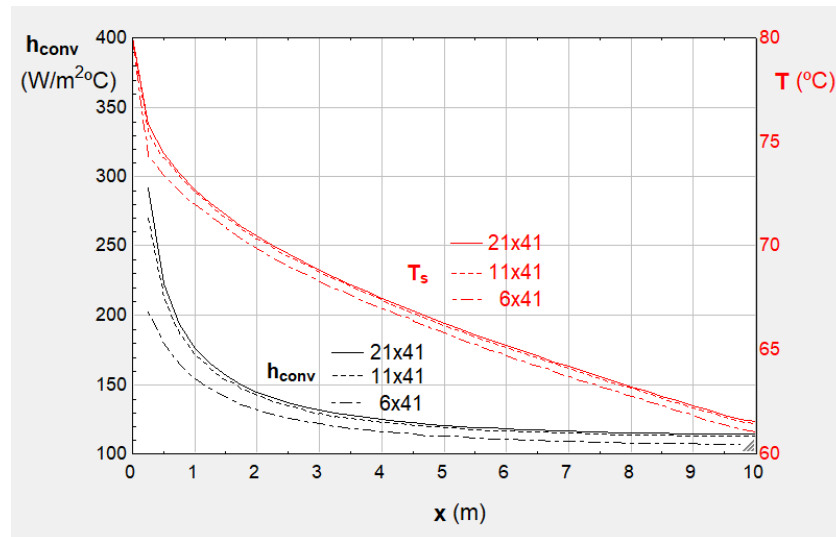


Figure 4.7.13 – Comparison of the calculated wall surface temperatures and local convection coefficients for different grid sizes.

4.8 Heat transfer in a laminar flow over a flat plate

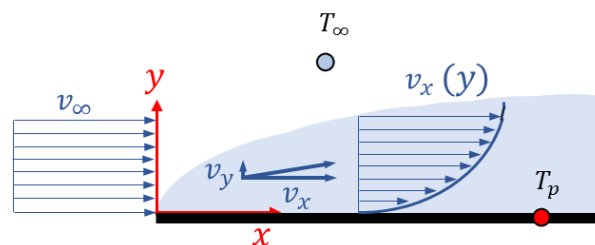


Figure 4.8.1 – Laminar flow over a flat plate.

Consider a flow of air over a flat plate, with a velocity $v_\infty = 5$ m/s (parallel to the surface). The air is at 30°C (T_∞) and the plate at a constant temperature of 60°C (T_p). The plate length is equal to 1.5 m.

With EES, using a grid of 31×31 nodes/elements, with a maximum distance to the plate (along y) of 3 cm, calculate the flow velocity components (v_x and v_y) and the flow temperature distribution ($T(x, y)$). After the temperature field calculate the variation of the convection coefficient along x .

While in the previous example the flow was dynamically developed, and the velocity profile was known, in this case the dynamic and thermal boundary layers are under development.

We will assume a 2D geometry, with the velocities (v_x and v_y) and temperature varying along the flow direction, x , and y . The index i will be associated with y , varying between 1 and N_y , and the index j will be associated with x , varying between 1 and N_x .

For the entrance nodes ($i, 1$) the velocities and temperatures are known: $v_x(i, 1) = v_\infty$, $v_y(i, 1) = 0$, $T(i, 1) = 30^\circ\text{C}$. For the other volumes, with dimensions Δx and Δy , we need 3 equations to calculate the 3 unknown properties (the 2 velocity components and temperature). We may use the well-known continuity, momentum and energy differential equations, and apply a finite differences discretisation, or perform mass, momentum and energy balances for the chosen control volumes (finite volumes approach).

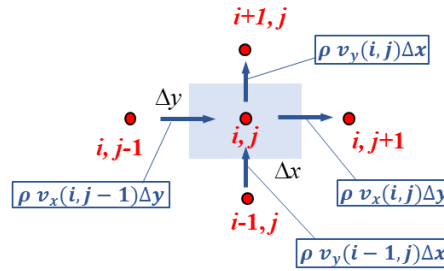


Figure 4.8.2 – General control volume (i, j) and mass balance with upstream velocities at volume borders (inlets and outlets).

Using Figure 4.8.2 as a reference, for the steady-state incompressible flow, the mass balance is

$$[v_x(i, j) - v_x(i, j - 1)] \Delta y + [v_y(i, j) - v_y(i - 1, j)] \Delta x = 0 \quad (4.8.1)$$

which could also be obtained by discretising the continuity differential equation with upwind differences:

$$\frac{\partial v_x}{\partial x} + \frac{\partial v_y}{\partial y} = 0 \Rightarrow \frac{[v_x(i, j) - v_x(i, j - 1)]}{\Delta x} + \frac{[v_y(i, j) - v_y(i - 1, j)]}{\Delta y} = 0 \quad (4.8.2)$$

Regarding momentum in the x direction, the change in momentum is due to the only force acting in that direction: viscous stress. The differential equation is:

$$v_x \frac{\partial v_x}{\partial x} + v_y \frac{\partial v_x}{\partial y} = \nu \frac{\partial^2 v_x}{\partial y^2} \quad (4.8.3)$$

which after discretisation becomes

$$v_x(i, j) \frac{[v_x(i, j) - v_x(i, j - 1)]}{\Delta x} + v_y(i, j) \frac{[v_x(i, j) - v_x(i - 1, j)]}{\Delta y} = \nu \frac{[v_x(i + 1, j) + v_x(i - 1, j) - 2v_x(i, j)]}{\Delta y^2} \quad (4.8.4)$$

Regarding the conservation of energy, the change in energy transported by the flow is due to the exchanges of the volume with neighbour volumes, by conduction. We may write the discretised equation (neglecting energy dissipation due to low velocity):

$$\begin{aligned} \rho c_p \left[v_x(i, j) \frac{[T(i, j) - T(i, j - 1)]}{\Delta x} + v_y(i, j) \frac{[T(i, j) - T(i - 1, j)]}{\Delta y} \right] = \\ = k \left[\frac{[T(i, j + 1) + T(i, j - 1) - 2T(i, j)]}{\Delta x^2} + \frac{[T(i + 1, j) + T(i - 1, j) - 2T(i, j)]}{\Delta y^2} \right] \end{aligned} \quad (4.8.5)$$

Then, equations (4.8.2), (4.8.4) and (4.8.5) allow the calculation of v_x , v_y and T .

Besides the fixed velocity and temperature conditions at the entrance section – nodes ($i = 1$ to $N_y, 1$) – the conditions at the plate are also known: for nodes ($i = 1, j = 2$ to N_x) both velocity components are equal to 0, and the temperature is the plate temperature. Far from the plate, outside the boundary layer – nodes ($i = N_y, j = 2$ to N_x) –, the velocity is equal to v_∞ and the temperature to T_∞ . As the boundary layer is smaller than the maximum distance to the plate used (3 cm), those conditions are valid for $i = N_y$. The conduction term at the exit section will be neglected by imposing the same temperature at ($i, N_x + 1$) and (i, N_x).

The local convection coefficient is equal to the heat flux at the wall divided by the temperature difference between plate and T_∞ . The heat flux at the wall is evaluated as the heat conduction rate in the fluid, between the 2 nodes nearer to the wall (neglecting longitudinal conduction):

$$h_{conv}[j] = \frac{\dot{q}[j]}{(T_p - T_\infty)} = \frac{\frac{k}{\Delta y}(T_p - T[2,j])}{(T_p - T_\infty)} \quad (4.8.6)$$

Figure 4.8.3 presents the Equations Window, and Figure 4.8.4 the Formatted Equations.

```

"LAMINAR AIR FLOW PARALLEL TO A PLATE AT CONSTANT T"
H=0.03 "max distance from plate to infinit flow = 0.03 m"
L=1.5 "plate / flow length in meters"
v_infinity=5 "velocity far from plate m/s"
T_in=30
T_p=60

"Air properties"
rho=1.2
c_p=1005
k=0.025
nu=1.5e-5
Re=v_infinity*L/nu
Pr=nu*rho*c_p/k

Ny=31: Ny=H/DELTAy+1
Nx=31: Nx=L/DELTAx+1
duplicate i=1: Ny
  y[i]=(i-1)*DELTAy
end
duplicate j=1: Nx
  x[j]=(j-1)*DELTAx
end

"nodes at entrance (x=0, j=1)"
duplicate i=1: Ny
  v_x[i;1]=v_infinity
  v_y[i;1]=0
  T[i;1]=T_in
end
"nodes at plate (y=0, i=1 - at T_p) and infinite (y=H, i=Ny - at T_in)"
duplicate j=2: Nx
  v_x[1;j]=0
  v_y[1;j]=0
  T[1;j]=T_p
  v_x[Ny;j]=v_infinity
  v_y[Ny;j]=v_y[Ny-1;j]
  T[Ny;j]=T_in
end
"internal nodes"
duplicate i=2: Ny-1
duplicate j=2: Nx
  (v_x[i;j]-v_x[i;j-1])/DELTAx+(v_y[i;j]-v_y[i-1;j])/DELTAy=0 "continuity equation - mass balance"
  v_x[i;j]*(v_x[i;j]-v_x[i;j-1])/DELTAx+nu*(v_x[i-1;j]+v_x[i+1;j]-2*v_x[i;j])/DELTAx^2 "momentum equation - v_y term removed for convergence"
  v_x[i;j]*(T[i;j]-T[i;j-1])/DELTAx+v_y[i;j]*(T[i;j]-T[i-1;j])/DELTAy=k/(rho*c_p)*((T[i;j-1]+T[i;j+1]-2*T[i;j])/DELTAx^2+(T[i-1;j]+T[i+1;j]-2*T[i;j])/DELTAy^2) "energy equation"
end
end
"nodes exit x - Nx+1"
duplicate i=2:Ny
  T[i;Nx+1]=T[i;Nx]
end

"for graph with x evolution - convection coefficient"
duplicate j=2:Nx
  q_dot[j]=k*DELTAy*(T_p-T[2,j]) "neglecting longitudinal conduction"
  h_conv[j]=q_dot[j]/(T_p-T_in)
end

"comparison with theory"
duplicate j=2: Nx
  Re[j]=v_infinity*x[j]/nu
  Nusselt[j]=0.332*Re[j]^0.5*Pr^(1/3)
  h_theory[j]=Nusselt[j]*k/x[j]
end

```

Figure 4.8.3 – Equations Window for the laminar flow over a flat plate example.

Formatted Equations

LAMINAR AIR FLOW PARALLEL TO A PLATE AT CONSTANT T

H = 0,03 max distance from plate to infinit flow = 0.03 m

L = 1,5 plate / flow length in meters

$v_{\infty} = 5$ velocity far from plate m/s

$T_{in} = 30$

$T_p = 60$

Air properties

$\rho = 1,2$

$c_p = 1005$

$k = 0,025$

$\nu = 0,000015$

$Re = v_{\infty} \cdot \frac{L}{\nu}$

$Pr = \nu \cdot \rho \cdot \frac{c_p}{k}$

$N_y = 31 \quad N_y = \frac{H}{\Delta y} + 1$

$N_x = 31 \quad N_x = \frac{L}{\Delta x} + 1$

$y_i = (i - 1) \cdot \Delta y \quad (\text{for } i = 1 \text{ to } N_y)$

$x_j = (j - 1) \cdot \Delta x \quad (\text{for } j = 1 \text{ to } N_x)$

nodes at entrance (x=0, j=1)

$v_{x,i,1} = v_{\infty} \quad (\text{for } i = 1 \text{ to } N_y)$

$v_{y,i,1} = 0 \quad (\text{for } i = 1 \text{ to } N_y)$

$T_{i,1} = T_{in} \quad (\text{for } i = 1 \text{ to } N_y)$

nodes at plate (y=0, i=1 - at T_p) and infinite (y=H, i=Ny - at T_{in})

$v_{x,1,j} = 0 \quad (\text{for } j = 2 \text{ to } N_x)$

$v_{y,1,j} = 0 \quad (\text{for } j = 2 \text{ to } N_x)$

$T_{1,j} = T_p \quad (\text{for } j = 2 \text{ to } N_x)$

$v_{x,Ny,j} = v_{\infty} \quad (\text{for } j = 2 \text{ to } N_x)$

$v_{y,Ny,j} = v_{y,Ny-1,j} \quad (\text{for } j = 2 \text{ to } N_x)$

$T_{Ny,j} = T_{in} \quad (\text{for } j = 2 \text{ to } N_x)$

internal nodes

$\frac{v_{x,i,j} - v_{x,i,j-1}}{\Delta x} + \frac{v_{y,i,j} - v_{y,i-1,j}}{\Delta y} = 0 \quad (\text{for } i = 2 \text{ to } N_y-1; j = 2 \text{ to } N_x) \quad \text{continuity equation - mass balance}$

$v_{x,i,j} \cdot \left[\frac{v_{x,i,j} - v_{x,i,j-1}}{\Delta x} \right] = \nu \cdot \left[\frac{v_{x,i-1,j} + v_{x,i+1,j} - 2 \cdot v_{x,i,j}}{\Delta y^2} \right] \quad (\text{for } i = 2 \text{ to } N_y-1; j = 2 \text{ to } N_x) \quad \text{momentum equation - } v_y \text{ term removed for convergence}$

$v_{x,i,j} \cdot \left[\frac{T_{i,j} - T_{i,j-1}}{\Delta x} \right] + v_{y,i,j} \cdot \left[\frac{T_{i,j} - T_{i-1,j}}{\Delta y} \right] = \frac{k}{\rho \cdot c_p} \cdot \left[\frac{T_{i,j-1} + T_{i,j+1} - 2 \cdot T_{i,j}}{\Delta x^2} + \frac{T_{i-1,j} + T_{i+1,j} - 2 \cdot T_{i,j}}{\Delta y^2} \right] \quad (\text{for } i = 2 \text{ to } N_y-1; j = 2 \text{ to } N_x) \quad \text{energy equation}$

nodes exit x - Nx+1

$T_{i,Nx+1} = T_{i,Nx} \quad (\text{for } i = 2 \text{ to } N_y)$

for graph with x evolution - convection coefficient

$\dot{q}_j = \frac{k}{\Delta y} \cdot (T_p - T_{2,j}) \quad (\text{for } j = 2 \text{ to } N_x) \quad \text{neglecting longitudinal conduction}$

$h_{conv,j} = \frac{\dot{q}_j}{T_p - T_{in}} \quad (\text{for } j = 2 \text{ to } N_x)$

comparison with theory

$Re_j = v_{\infty} \cdot \frac{x_j}{\nu} \quad (\text{for } j = 2 \text{ to } N_x)$

$Nusselt_j = 0,332 \cdot Re_j^{0,5} \cdot Pr^{(1/3)} \quad (\text{for } j = 2 \text{ to } N_x)$

$h_{theory,j} = Nusselt_j \cdot \frac{k}{x_j} \quad (\text{for } j = 2 \text{ to } N_x)$

Figure 4.8.4 – Formatted Equations Window for the laminar flow over a flat plate example.

Results were obtained for a 31×31 grid, which, with a flow length of 1.5 m and a height of 3 cm, means $\Delta x = 5$ cm and $\Delta y = 1$ mm.

When running the program a few convergence problems occurred. The initial guess values were changed for the temperatures – choosing 50°C , as a value between 30 and 60°C – and also for the vertical velocities, $v_y(i, j)$ – a value of 0.01 m/s was used. Besides, it was found that the term related to $v_y(i, j)$ in the momentum equation (4.8.4) was causing no convergence. Therefore, this term was eliminated, which should not produce bad results, as the velocity vertical component (v_y) is very small compared to the horizontal component (v_x). The error will be assessed by comparing the convective coefficient results with the theoretical solution for a laminar flow over a flat surface, which was added at the end of the Equations Window.

After performing the calculations (Calculate → Solve) all the main results, namely velocities, temperatures and convection coefficients, are available in the Arrays Table. From this table several graphical outputs may be obtained.

Figure 4.8.5 shows the horizontal velocity (v_x) distribution, for 4 different x coordinates. The velocity gradient decreases along y , as expected, and the dynamic boundary layer thickness when the air leaves the plate is equal to about 1 cm. Figure 4.8.6 shows the same distribution for all values of x and y , using colour bands. Figure 4.8.7 represents the vertical velocity (v_y) distribution, for 4 different x coordinates. The velocity values are much lower than the vertical ones, reducing in the main flow direction. The maximum value at $x=0.25$ m is about 0.011 m/s.

Figures 4.8.8 and 4.8.9 show the temperature distribution in the flow. Temperature gradients are high near the plate wall, as expected. The thermal boundary layer is slightly thicker than the dynamic boundary layer: at the exit ($x=1.5$ m) the thermal boundary layer is about 1.25 cm thick. This is expected from theory, as the Prandtl number for air is lower than 1 (with $Pr=1$ the two boundary layers would have the same size). In this air flow the Reynolds number is in the laminar limit of 5×10^5 , and the theoretical calculation indicates a dynamic boundary thickness of 0.0106 m at the plate exit. Since $Pr=0.724$, the theoretical thermal boundary layer thickness should be equal to 0.0118 m. The values obtained with the numerical model are very close to the theoretical predictions.

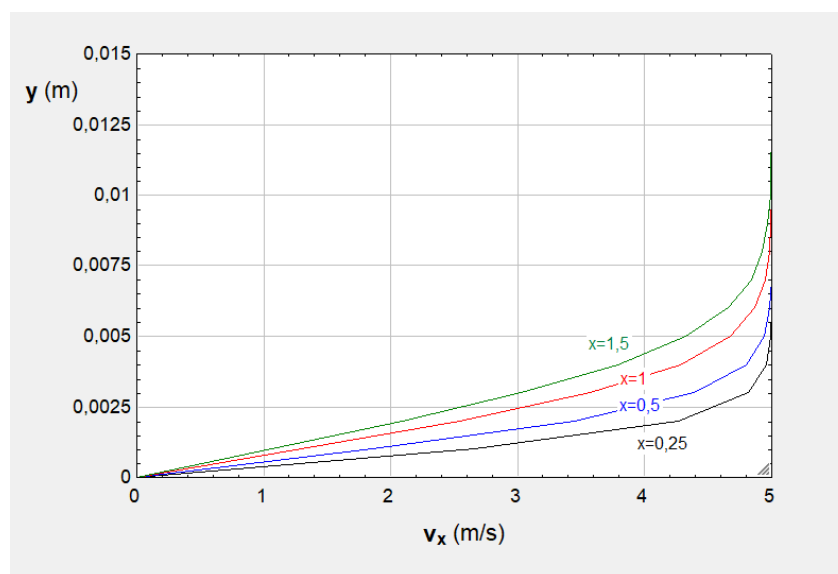


Figure 4.8.5 – Horizontal velocity distribution in the laminar flow over a flat plate, at four x locations.

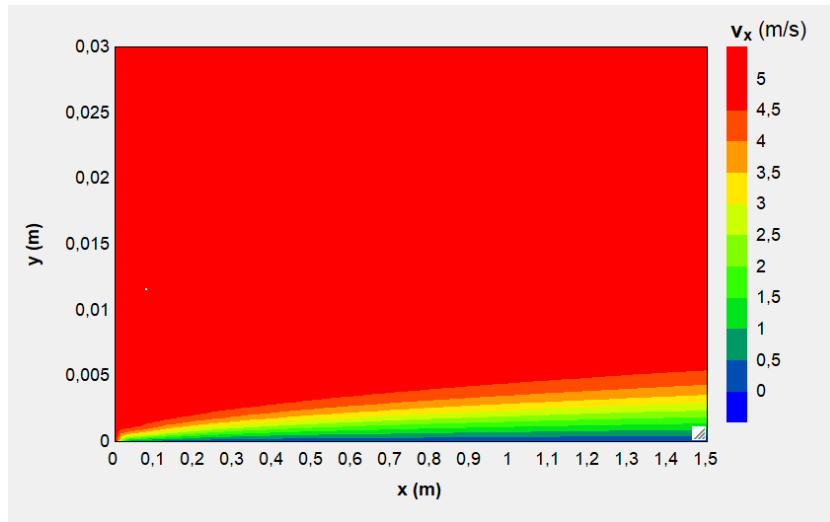


Figure 4.8.6 – Horizontal velocity distribution in the laminar flow over a flat plate – colour bands.

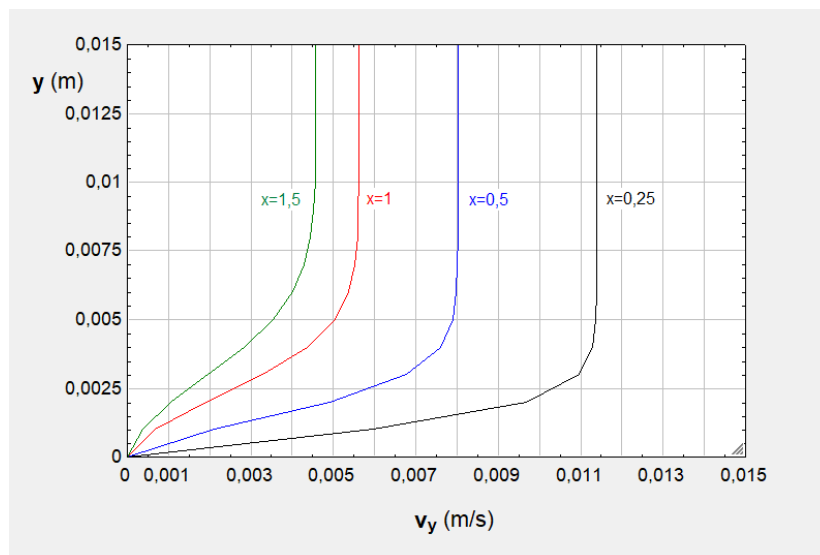


Figure 4.8.7 – Vertical velocity distribution in the laminar flow over a flat plate, at four x locations.

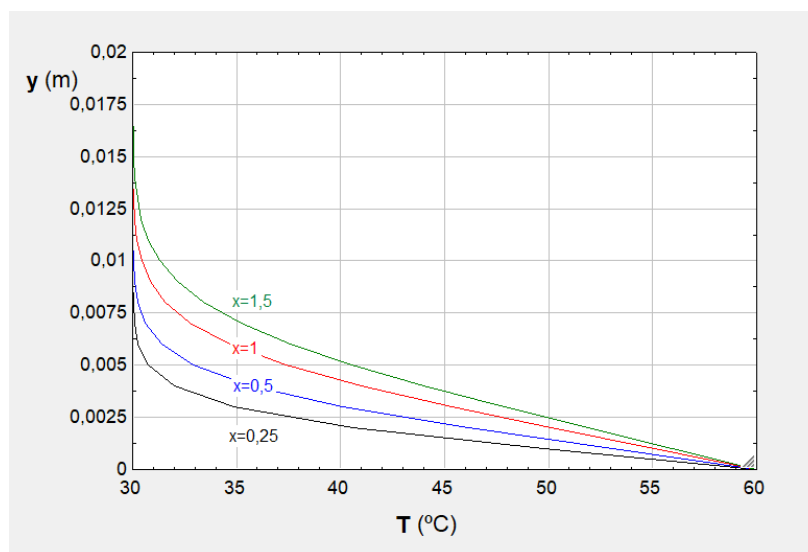


Figure 4.8.8 – Temperature distribution in the laminar flow over a flat plate, at four x locations.

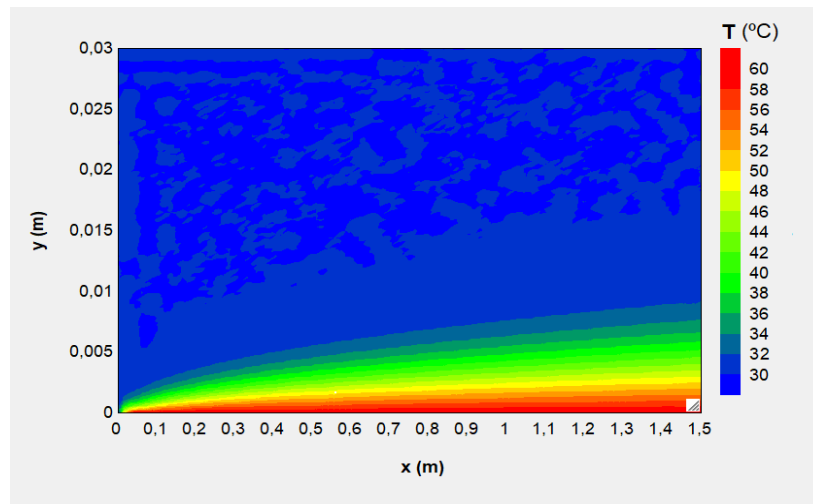


Figure 4.8.9 – Temperature distribution in the laminar flow over a flat plate – colour bands.

Following equation (4.8.6), the model was used to calculate the evolution of the convective heat transfer coefficient ($h_{conv}[x]$). The results are presented in Figure 4.8.10, where a comparison is made with the theoretical values from the expression for laminar flow over a flat plate. The results of the numerical model are fairly close, with a slight underprediction for low values of x (a slight overprediction in nodes $j=3$ and $j=4$). The differences are generally lower than 5%. This means that the approximations made, namely regarding the momentum equation, are fairly valid.

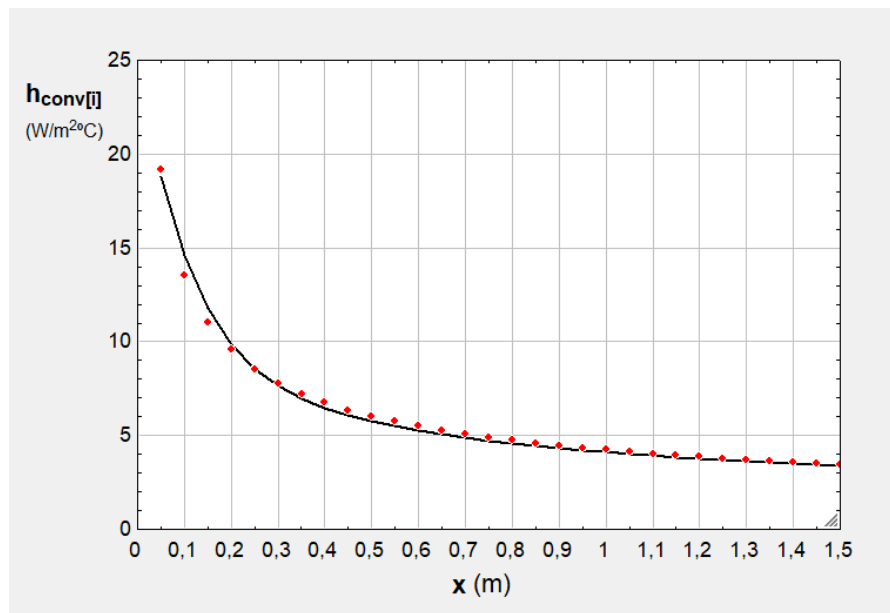


Figure 4.8.10 – Comparison of the calculated convection coefficient (black colour curve) with the laminar flow theoretical solution (red colour dots).

4.9 Solar dryer heat and mass transfer

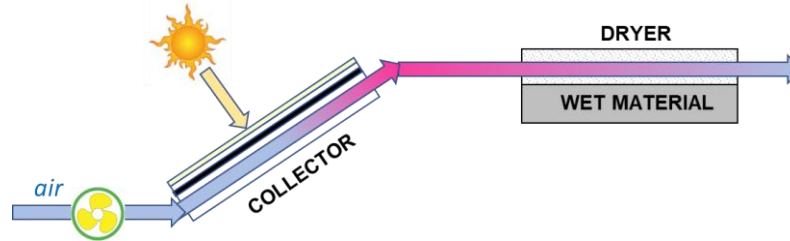


Figure 4.9.1 – Schematic representation of a solar drying system.

A small solar drying system is composed of a solar collector and a drying chamber. Ambient air is introduced in the collector, with an area of 2 m^2 , and a flowrate of 15 litres/s/m^2 ; under those conditions its efficiency is equal to 0.5. Climatic data during the drying period are given in the table below: air temperature (T_{amb}), humidity (ϕ_{amb}) and incident solar radiation on the collector surface ($I_{sol,col}$).

The drying chamber contains a wet material (fully saturated surface) with an air channel with a height of 5 cm. It is 2 m long and 1 m wide. The flow may be assumed as parallel to the material surface. Heat transfer with the outside of the chamber (top, bottom and sides) is negligible.

Assume quasi-steady conditions in the collector and drier. Considering the mixed-mean air properties in each flow section, dividing the dryer into volume elements, and using convection coefficient correlations, obtain with EES the evolution of air temperature and humidity along the channel, and the drying rate, during the drying period.

hour	9	10	11	12	13	14	15	16	17
T_{amb} (°C)	16.6	17.4	19.6	22.2	22.8	24.2	23.2	23.0	22.0
ϕ_{amb} (%)	55	51	46	42	43	41	39	48	51
$I_{sol,col}$ (W/m ²)	601	770	897	961	962	887	748	557	368

The model will start by calculating the solar collector outlet conditions (temperature and humidity) for the existing inlet (ambient air) conditions, taking into account the collector efficiency. Under quasi-steady conditions, we may write:

$$\dot{Q}_{col} = \eta_{col} I_{sol,col} A_{col} = \dot{M}_{col} c_p (T_{col,out} - T_{amb}) \quad (4.9.1)$$

As the air is heated in the collector, its relative humidity (ϕ) is reduced, although maintaining the same absolute humidity (ω). The ambient air absolute humidity (at the inlet) may be calculated after the ambient temperature and relative humidity. In EES, using the absolute humidity property function for AirH2O (“humrat”, from humidity ratio):

$$\omega_{in} = \text{humrat}(\text{AirH2O}; T = T_{amb}; R = \phi_{amb}; P = 100) \quad (4.9.2)$$

for air at atmospheric pressure (in kPa). This will also be the absolute humidity at the entrance of the dryer.

From equations (4.9.1) and (4.9.2) we know, for each climatic situation, the air temperature and humidity at the dryer inlet. The collector volumetric flow rate is always the same (imposed by the fan performance), but there is a slight variation of the total mass flow rate, which includes dry air flow rate plus water vapour flow rate. The relationship between them is given by:

$$\dot{M}_{col} = \dot{M}_{dry}(1 + \omega_{in}) \quad (4.9.3)$$

The dry air flow rate will be the same along the dryer length, but the total flow rate will increase, due to water evaporation.

The dryer will be divided into N nodes and $N-1$ finite control volumes, according to Figure 4.9.2. We shall use control volumes of the same size (Δx), because the inlet conditions (node 1) are already known: the inlet air properties are the absolute humidity ($\omega[1]$), the temperature ($T_{air}[1]$), and the air enthalpy per mass of dry air ($h_{ent}[1]$); the subscript *ent* was used for enthalpy, to distinguish from the h symbol for the convective coefficient. Equations (4.9.1) and (4.9.2) define $\omega[1]$ and $T_{air}[1]$, and the enthalpy is easily calculated with the EES function:

$$h_{ent}[1] = \text{enthalpy}(\text{AirH}_2\text{O}; T = T_{air}[1]; R = \phi_{amb}; P = 100) \quad (4.9.4)$$

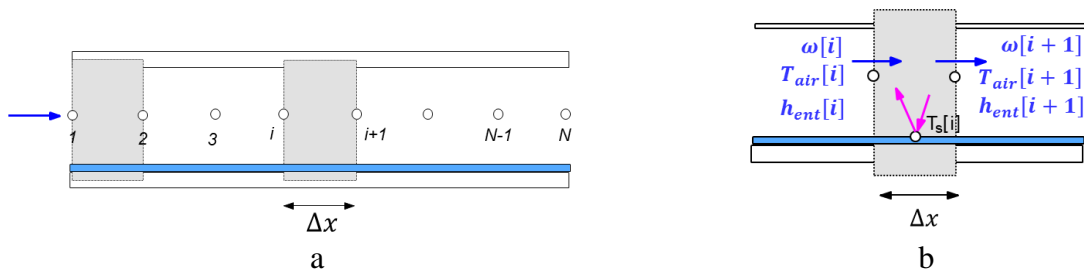


Figure 4.9.2 – Finite control volumes in the dryer channel: (a) $N-1$ volumes, N nodes; (b) generic volume.

For each volume (Figure 4.9.2(b)) the outlet properties ($i + 1$) will be calculated assuming that the average air properties in the volume are the inlet ones (i , upwind approach). They result from mass and energy balances, neglecting the longitudinal heat conduction; this means that only heat exchanges with the wet surface (at $T_s[i]$) are considered.

The mass balance states that the change in air mass is due to evaporation from the wet surface:

$$\begin{aligned} \dot{M}_{dry}(\omega[i + 1] - \omega[i]) &= \dot{M}_{evap}[i] = \\ &= h_m[i](\Delta x \cdot width)[\rho_{v,sat}(T_s[i]) - \phi_{air}[i] \rho_{v,sat}(T_{air}[i])] \end{aligned} \quad (4.9.5)$$

where the vapour concentration of saturated air can be calculated with the EES property database as a function of temperature ($T_s[i]$ or $T_{air}[i]$). The mass transfer coefficient may be related to the heat convection coefficient with the Lewis relationship, and this coefficient may be obtained with the EES heat transfer procedure “Ductflow local”, which provides the local convection coefficients for each location ($x[i] + \Delta x/2$ was used).

The energy balance at the wet surface states that the heat needed for water evaporation comes from air convection (longitudinal heat conduction is neglected):

$$\dot{M}_{evap}[i] \Delta h_{lv} = h[i](\Delta x \cdot width)(T_{air}[i] - T_s[i]) \quad (4.9.6)$$

and the air energy balance is:

$$\dot{M}_{dry}(h_{ent}[i + 1] - h_{ent}[i]) = \dot{M}_{evap}[i] \Delta h_{lv} - h[i](\Delta x \cdot width)(T_{air}[i] - T_s[i]) \quad (4.9.7)$$

which, according to (4.9.6) implies that the air enthalpy remains constant; this wouldn't happen if the dryer had heat losses to the outside.

The enthalpy is a function of ω and T_{air} , and therefore equations (4.9.5), (4.9.6) and (4.9.7) allow the calculation of $\omega[i + 1]$, $T_s[i]$ and $T_{air}[i + 1]$. The calculation is extended to all volumes, until the outlet values are obtained ($i = N$).

Figure 4.9.3 presents the Formatted Equations window, and Figure 4.9.4 the Equations Window. The total drying rate was calculated by summing the evaporation rate in all volumes. A drying system indicator (η_{dry}) was also calculated, by dividing the heat rate needed to evaporate the water by the heat rate input in the solar collector. A total of 10 volumes (11 nodes) were used, and its effect will be assessed later.

```

Formatted Equations
SOLAR DRYER HEAT & MASS TRANSFER

A_col = 2
eta_col = 0,5
M_dot_col = rho * 0,015 * A_col * 15 !5 l/s/m2 of solar collector
rho = rho (AIRH2O ; T = T_amb ; R = phi_amb ; P = 100 )
Q_dot_col = M_dot_col * Cp (Air ; T = T_amb ) * 1000 * (T_colout - T_amb )
Q_dot_col = eta_col * I_solcol * A_col
omega_in = omega (AIRH2O ; T = T_amb ; R = phi_amb ; P = 100 )
M_dot_col = M_dot_dry * (1 + omega_in )
T_air,1 = T_colout defines T_air,1 - dryer inlet
omega_1 = omega_in
omega_1 = omega (AIRH2O ; T = T_air,1 ; R = phi_air,1 ; P = 100 ) dryer inlet conditions ATTENTION TO phi_air,1 GUESS VALUES
enthalpy_1 = h (AIRH2O ; T = T_air,1 ; R = phi_air,1 ; P = 100 ) * 1000 dryer inlet conditions
L = 2 drying channel length
Width = 1 drying channel width
H = 0,05 drying channel height 5 cm
D_v,air = 0,000026
k_air = k (Air ; T = T_air,1)
N = 10 volumes, (N+1 nodes)
Delta_x = L / N
x_1 = 0
x_i,1 = x_i + Delta_x (for i = 1 to N)
M_dot_dry * (omega_i,1 - omega_i) = M_dot_evap,i (for i = 1 to N) water mass balance
omega_i,1 = omega (AIRH2O ; T = T_air,i ; R = phi_air,i ; P = 100 ) (for i = 1 to N)
M_dot_evap,i = h_m,i * Delta_x * Width * (rho_vs,surf,i - phi_air,i * rho_vs,air,i) (for i = 1 to N)
rho_vs,surf,i = omega (AIRH2O ; T = T_surf,i ; R = 1 ; P = 100 ) / v (AIRH2O ; T = T_surf,i ; R = 1 ; P = 100 ) (for i = 1 to N)
rho_vs,air,i = omega (AIRH2O ; T = T_air,i ; R = 1 ; P = 100 ) / v (AIRH2O ; T = T_air,i ; R = 1 ; P = 100 ) (for i = 1 to N)
h_m,i = h_i * D_v,air / k_air (for i = 1 to N)
Call ductflow_local [Air ; T_air,i ; 100 ; M_dot_dry ; H ; Width ; x_i + Delta_x / 2 ; 0 ; h_i] (for i = 1 to N)
M_dot_evap,i * Enthalpy_vaporization (water ; T = T_surf,i) * 1000 = h_i * Delta_x * Width * (T_air,i - T_surf,i) (for i = 1 to N) surface energy balance
M_dot_dry * (enthalpy_i,1 - enthalpy_i) = M_dot_evap,i * Enthalpy_vaporization (water ; T = T_surf,i) * 1000 - h_i * Delta_x * Width * (T_air,i - T_surf,i) (for i = 1 to N) air energy balance
enthalpy_i,1 = h (AIRH2O ; T = T_air,i ; R = phi_air,i ; P = 100 ) * 1000 (for i = 1 to N) alternative: enthalpy_i,1 = enthalpy_i
wb = WB (AIRH2O ; T = T_air,i ; R = phi_air,i ; P = 100 )
M_dot_evap,total = sum_{i=1}^N (M_dot_evap,i) * 1000 in grams/s
Q_dot_dry = M_dot_evap,total * Enthalpy_vaporization (water ; T = T_surf,1) in W
eta_dry = Q_dot_dry / Q_dot_col
count = hour
  
```

Figure 4.9.3 – Formatted Equations window for the solar air dryer example.

```

Equations Window
"SOLAR DRYER HEAT & MASS TRANSFER"
A_col=2
eta_col=0,5
M_dot_col=rho*0,015*A_col "15 l/s/m2 of solar collector"
rho=density(AirH2O,T=T_amb,R=phi_amb,P=100)
Q_dot_col=M_dot_col*cp(Air,T=T_amb)*1000*(T_col_out-T_amb)
Q_dot_col=eta_col*I_sol_col*A_col
omega_in=humrat(AirH2O,T=T_amb,R=phi_amb,P=100)
M_dot_col=M_dot_dry*(1+omega_in)

T_air[1]=T_col_out "defines T_air[1] - dryer inlet"
omega[1]=omega_in
omega[1]=humrat(AirH2O,T=T_air[1],R=phi_air[1],P=100) "dryer inlet conditions" "ATTENTION TO phi_air[i] GUESS VALUES"
enthalpy[1]=enthalpy(AirH2O,T=T_air[1],R=phi_air[1],P=100)*1000 "dryer inlet conditions"

L=2 "drying channel length"
Width=1 "drying channel width"
H=0,05 "drying channel height 5 cm"
D_v_air=2,6e-5
k_air=conductivity(Air,T=T_air[1])
N=10 "volumes, (N+1 nodes)"
DELTAx=L/N
x[1]=0
Duplicate i=1:N
x[i+1]=x[i]+DELTAx
M_dot_dry*(omega[i+1]-omega[i])=M_dot_evap[i] "water mass balance"
omega[i+1]=humrat(AirH2O,T=T_air[i+1],R=phi_air[i+1],P=100)
M_dot_evap[i]=h_m[i]*DELTAx*Width*(rho_vs_surf[i]-phi_air[i]*rho_vs_air[i])
rho_vs_surf[i]=humrat(AirH2O,T=T_surf[i],R=1,P=100)/volume(AirH2O,T=T_surf[i],R=1,P=100)
rho_vs_air[i]=humrat(AirH2O,T=T_air[i],R=1,P=100)/volume(AirH2O,T=T_air[i],R=1,P=100)
h_m[i]=h[i]*D_v_air/k_air
Call ductflow_local(air,T_air[i],100,M_dot_dry,H,Width,x[i]+DELTAx/2,0:h[i],h_H[i],dPdx[i])
M_dot_evap[i]*enthalpy_vaporization(Water,T=T_surf[i])*1000=h[i]*DELTAx*Width*(T_air[i]-T_surf[i]) "surface energy balance"
M_dot_dry*(enthalpy[i+1]-enthalpy[i])=M_dot_evap[i]*enthalpy_vaporization(Water,T=T_surf[i])*1000*h[i]*DELTAx*Width*(T_air[i]-T_surf[i]) "air energy balance" "alternative: enthalpy[i+1]=enthalpy[i]"
enthalpy[i+1]=enthalpy(AirH2O,T=T_air[i+1],R=phi_air[i+1],P=100)*1000
End
wb=wetbulb(AirH2O,T=T_air[1],R=phi_air[1],P=100)
M_dot_evap_total=sum(M_dot_evap[i],i=1:N)*1000 "in grams/s"
Q_dot_dry=M_dot_evap_total*enthalpy_vaporization(Water,T=T_surf[1]) "in W"
eta_dry=Q_dot_dry/Q_dot_col
count=hour
    
```

Figure 4.9.4 – Equations Window for the solar air dryer example.

A Parametric Table (“Table 1”) was also created, containing the time (“hour”) and respective climatic conditions, as well as selected calculation results – see Figure 4.9.5. To obtain those results, modifications had to be made in the guess values of $T_{air}[i]$ and $\phi_{air}[i]$, which by default are equal to 1; temperatures and relative humidities closer to the collector outlet conditions were used. Those guess values are only used in Run 1, as the guess/initial values for the other EES runs use the previous run results.

Table 1	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1.9	hour	T_{amb} [°C]	ϕ_{amb}	$I_{sol,col}$ [W/m ²]	$T_{air,1}$ [°C]	$T_{air,11}$ [°C]	$T_{surf,1}$ [°C]	$\phi_{air,1}$	$\phi_{air,11}$	$\dot{M}_{evap,total}$ [g/s]	\dot{M}_{col} [kg/s]	\dot{M}_{dry} [kg/s]	\dot{Q}_{col} [W]	\dot{Q}_{dry} [W]	η_{dry}
Run 1	9	16,6	0,55	601	33,26	30,21	16,62	0,2035	0,2862	0,0433	0,03593	0,0357	601	106,6	0,1774
Run 2	10	17,4	0,51	770	38,8	35	18,24	0,1464	0,2221	0,05353	0,03583	0,03561	770	131,6	0,1708
Run 3	11	19,6	0,46	897	44,71	40,17	20,23	0,111	0,1787	0,06344	0,03556	0,03533	897	155,6	0,1735
Run 4	12	22,2	0,42	961	49,35	44,24	21,92	0,09404	0,1561	0,07051	0,03524	0,03499	961	172,7	0,1797
Run 5	13	22,8	0,43	962	50,04	44,89	22,42	0,09649	0,1584	0,07084	0,03516	0,03489	962	173,4	0,1803
Run 6	14	24,2	0,41	887	49,44	44,42	22,44	0,1031	0,1664	0,06885	0,03498	0,03471	887	168,5	0,19
Run 7	15	23,2	0,39	748	44,4	39,96	20,42	0,1193	0,188	0,0613	0,03512	0,03488	748	150,3	0,201
Run 8	16	23	0,48	557	38,79	35,32	19,93	0,1949	0,2731	0,04803	0,03511	0,03481	557	117,9	0,2116
Run 9	17	22	0,51	368	32,4	29,78	18,06	0,2771	0,3607	0,03653	0,03523	0,03493	368	89,8	0,244

Figure 4.9.5 – Parametric Table for the solar air dryer example.

As an $x[i]$ coordinate was defined along the dryer length, every time EES is run the air properties for the different $x[i]$ values will be available in an Arrays Table. The evolutions may be graphically represented. Figure 4.9.6 shows the Arrays Table for the last Run in Table 1 (at 17:00), and Figure 4.9.7 represents the evolution of air and wet surface temperatures and relative humidity along the dryer at 17:00. This figure shows that the wet surface temperature is almost constant, and equal to the air wet bulb temperature. The same Figure shows that there is virtually no difference in the air temperature results if 40 volumes are used instead of 10.

Sort	1	2	3	4	5	6	7	8	9	10	11	12	13
	x_i	$\phi_{air,i}$	$T_{air,i}$	ρ_i	$P_{vs,surf,i}$	$P_{vs,air,i}$	$\dot{M}_{evap,i}$	$dPdx_i$	h_i	$h_{H_2O,i}$	$h_{m,i}$	$T_{surf,i}$	enthalpy _i
[1]	0	0.2771	32.4	0.008504	0.0154	0.03448	0.00000476	-0.0001202	4.08	4.08	0.004071	18.06	54317
[2]	0.2	0.287	32.06	0.00864	0.01542	0.03386	0.000004081	-0.0001055	3.588	3.588	0.00358	18.08	54317
[3]	0.4	0.2957	31.76	0.008757	0.01543	0.03333	0.000003849	-0.0001016	3.46	3.46	0.003452	18.09	54317
[4]	0.6	0.3041	31.49	0.008867	0.01544	0.03284	0.0000037	-0.0000997	3.398	3.398	0.00339	18.11	54317
[5]	0.8	0.3123	31.22	0.008973	0.01546	0.03238	0.000003582	-0.00009848	3.36	3.36	0.003352	18.12	54317
[6]	1	0.3205	30.97	0.009076	0.01547	0.03194	0.000003481	-0.00009762	3.334	3.334	0.003326	18.13	54317
[7]	1.2	0.3286	30.72	0.009176	0.01548	0.03151	0.00000339	-0.00009697	3.315	3.315	0.003307	18.15	54317
[8]	1.4	0.3367	30.47	0.009273	0.01549	0.0311	0.000003307	-0.00009645	3.3	3.3	0.003293	18.16	54317
[9]	1.6	0.3447	30.24	0.009367	0.01551	0.03071	0.000003228	-0.00009602	3.289	3.289	0.003281	18.17	54317
[10]	1.8	0.3527	30.01	0.00946	0.01552	0.03033	0.000003154	-0.00009566	3.279	3.279	0.003272	18.18	54317
[11]	2	0.3607	29.78	0.00955									54317

Figure 4.9.6 – Arrays Table for Run 9 of Table 1 (17:00) in the solar air dryer example.

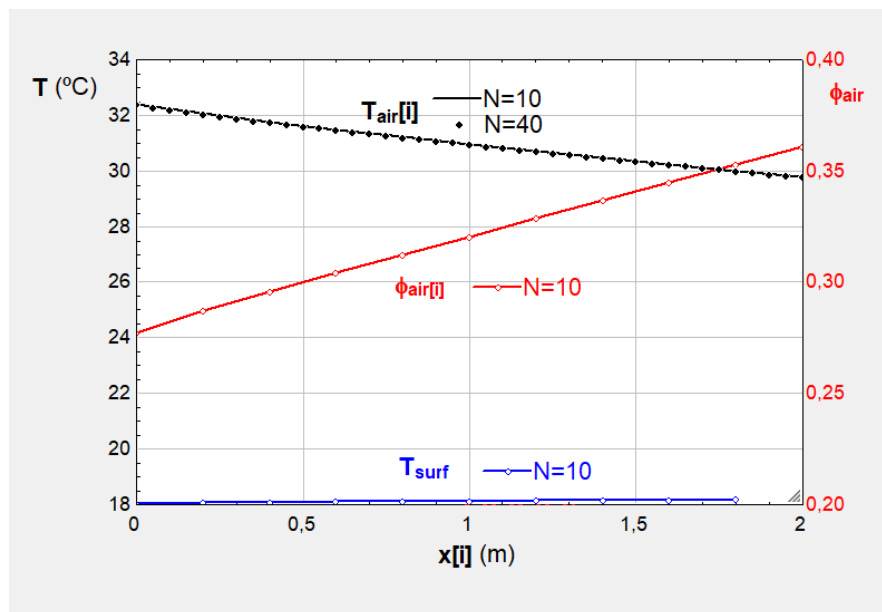


Figure 4.9.7 – Evolution of different temperatures and relative humidity along the dryer at 17:00.

From the Parametric Table in Figure 4.9.5 different graphs were obtained. Figure 4.9.8 shows the variation in dryer inlet and outlet conditions (air temperature and relative humidity) for the different climatic conditions considered. The collector is responsible for a significant increase in air temperature (from 22.8 up to 50°C at 13:00), with a corresponding decrease in relative humidity. The air reduces its temperature along the dryer, but still leaves with a high temperature (45°C at 13:00) and a low humidity; this indicates that a larger (longer) dryer could be used for increased system performance.

Figure 4.9.9 shows the evaporation rate and the drying system indicator (η_{dry}) for the different climatic conditions considered. The drying rate is higher at 13:00, due to the more favourable conditions, especially the higher solar collector radiation. The system indicator increases to about 0.24 at 17:00, due to a high drying heat rate combined with a lower collector heat rate – see Figure 4.9.10. Note that the value of this indicator is not only due to the solar collector input, as part of the drying rate is due to the ambient air temperature and humidity. Figure 4.9.10 also represents the change in collector air flow rate, change that is due to the varying humidity, and air density.

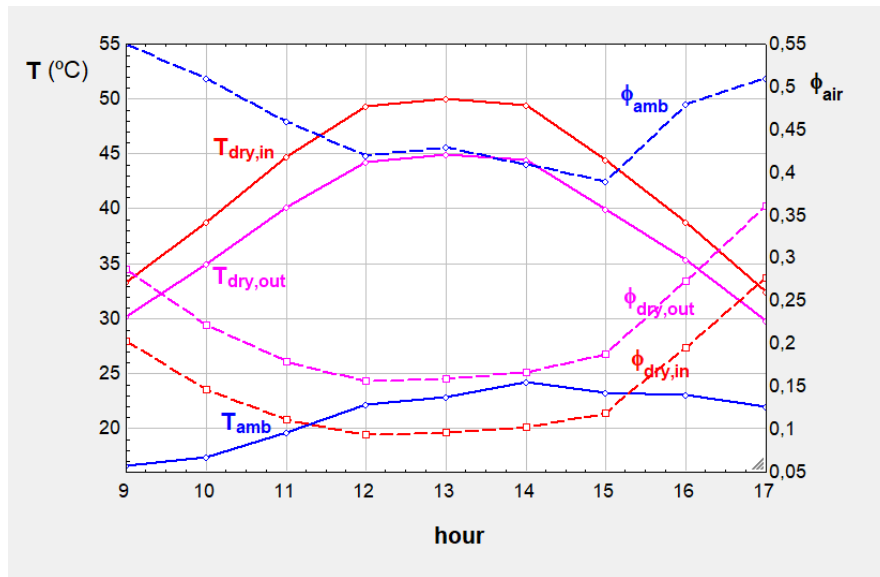


Figure 4.9.8 –Dryer inlet and outlet air conditions for the different ambient conditions.

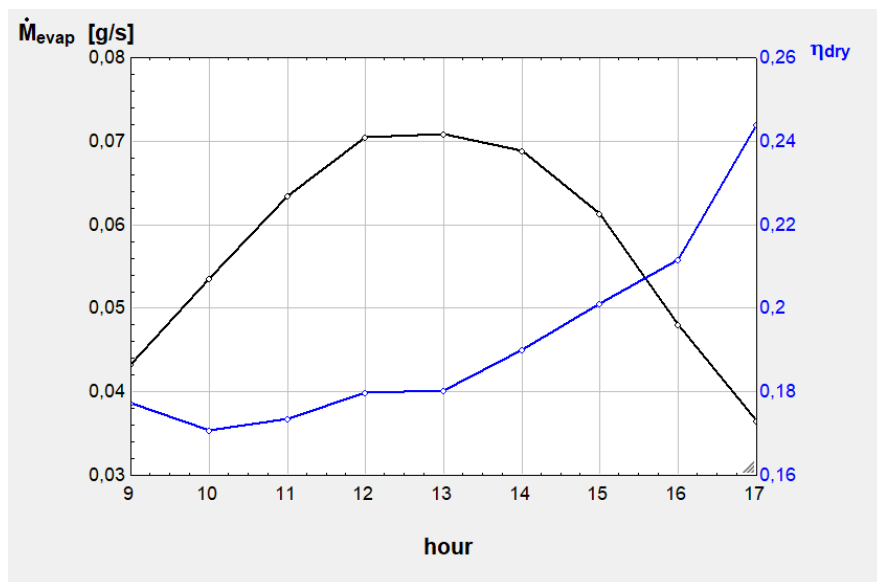


Figure 4.9.9 – Evaporation rate and drying system indicator for the different ambient conditions.

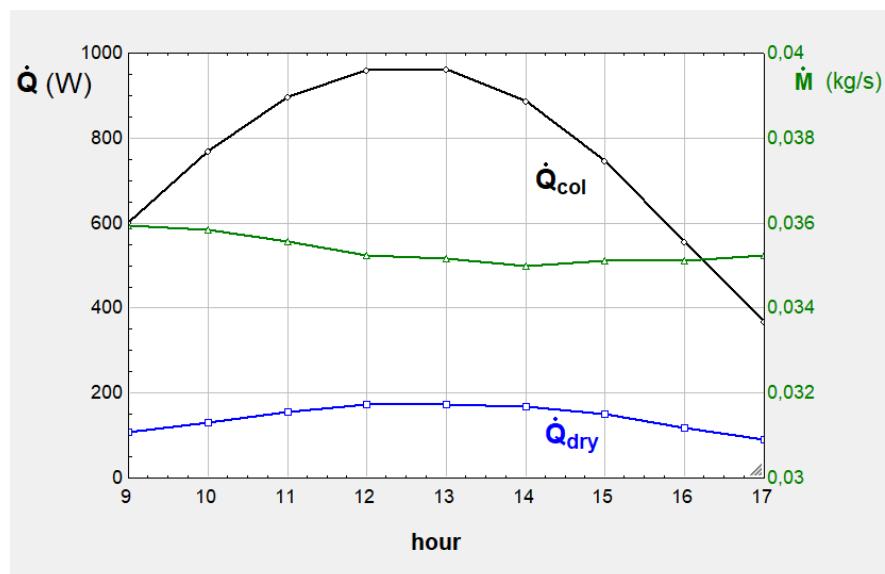


Figure 4.9.10 – Solar collector input and flow rate, and drying input for the different ambient conditions.

4.10 Dew point air cooler

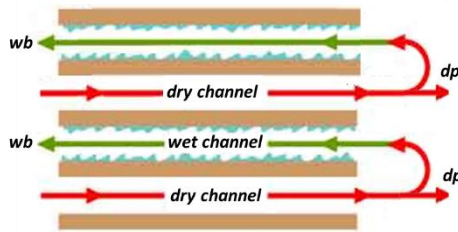


Figure 4.10.1 – Dew point air cooler.

A dew point air cooler achieves an air temperature close to the dew point temperature, lower than the wet bulb temperature of incoming air. To achieve that, the incoming air circulates in a dry channel, transferring heat to a stream of humid air in a wet channel; the humid air is the same air leaving the dry channel, that is partially diverted to the wet channel, and humidified by contact with a water saturated material.

Simulate the evolution of air temperature and humidity in a dry air channel with a height of 1 cm, a width and a length of 1 m, with an average air velocity of 10 cm/s. The incoming air is at 30°C, with a relative humidity of 40%. There is transfer of heat to one wet channel with the same dimensions. 50% of the air leaving the dry channel is diverted to the wet channel. The thickness of the wet material is 1 mm, and its effective conductivity is equal to 0.6 W/m°C; the thickness of the separating wall may be neglected.

This is another example with simultaneous heat and mass transfer. As in 4.9, we shall assume quasi-steady conditions in the two streams, and consider the mixed-mean air properties in each flow section. The two channels will be divided into volume elements, and convection coefficient correlations will be used for heat and mass transfer.

The inlet air absolute humidity may be calculated after the ambient temperature and relative humidity. In EES, using the absolute humidity property function for AirH2O (“humrat”, from humidity ratio):

$$\omega_{in} = \text{humrat}(\text{AirH2O}; T = T_{in}; R = \phi_{in}; P = 101.3) \quad (4.10.1)$$

for air at atmospheric pressure (101.3 kPa). This will also be the absolute humidity at the end of the dry channel, and at the flow start in the wet channel. The wet bulb and dew point temperatures corresponding to the inlet temperature and humidity can also be calculated with EES functions (WeTbulb and DewPoint AirH2O functions).

The numerical model will consider N nodes and $N-1$ volumes in each channel, with general volumes represented in Figure 4.10.2. The two streams flow in opposite directions, and to keep the same reference (index i in the array variables), while stream a flows with increasing i , flow b flows with decreasing i . In each volume the entrance properties are known; at the entrance of channel b (wet channel) $T_b[N] = T_a[N]$ and $\omega[N] = \omega_{in}$. In each wet channel section, only two properties are needed to define the humid air conditions: temperature and absolute humidity or temperature and relative humidity, as all the others (enthalpy as well) can be obtained from those 2 properties. Therefore, from an overall volume including the 2 channels, we need equations to obtain 4 properties: the dry channel exit air temperature ($T_a[i+1]$), the wet channel exit air humidity ($\omega[i]$), the wet channel exit air temperature ($T_b[i]$), and the water saturated surface temperature ($T_s[i]$).

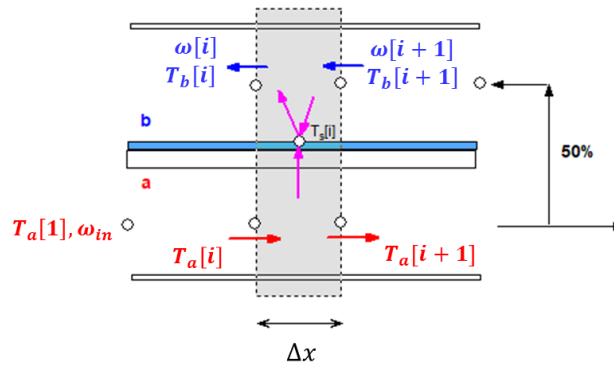


Figure 4.10.2 – Finite control volumes in the two channels: a – dry channel; b – wet channel.

The 4 equations for each volume are: the mass balance for the wet channel stream, the energy balance of the dry channel stream (exchange *a-s*), the energy balance of the wet channel stream (exchange *b-s*), and an overall energy balance (*a-b*). For each volume (Figure 4.10.2) the outlet properties (*i + 1*) will be calculated assuming that the average air properties in the volume are the inlet ones (upwind approach). Longitudinal heat conduction will be neglected.

The mass balance for the wet channel stream states that the change in air mass is due to evaporation from the saturated wet surface:

$$\begin{aligned} \dot{M}_{b,dry}(\omega[i] - \omega[i + 1]) &= \dot{M}_{evap}[i] = \\ &= h_m(\Delta x \cdot width)[\rho_{v,sat}(T_s[i]) - \phi[i] \rho_{v,sat}(T_b[i + 1])] \end{aligned} \quad (4.10.2)$$

where the vapour concentration of saturated air can be calculated with the EES property database as a function of temperature ($T_s[i]$ or $T_b[i + 1]$). The mass transfer coefficient was related to the heat convection coefficient, and only the average value along the flow length was considered; although there is a variation, this is not very significant for the results (small hydraulic diameter and low Reynolds number). The average convection coefficient was obtained with the EES heat transfer procedure “DuctFlow”.

The energy balance of the dry channel stream (*a*) expresses that the change in energy carried by the flow is due to the heat transfer across the separating wall and wet material. Other heat losses to the outside of the channel are neglected. We may write:

$$\dot{M}_a c_{p,a}(T_a[i] - T_a[i + 1]) = U(\Delta x \cdot width)(T_a[i] - T_s[i]) \quad (4.10.3)$$

where U is the overall heat transfer coefficient from the air up to the wet surface in contact with stream *b*; it includes the convection in stream *a*, and the conduction through the wall and wet material. The average convection coefficient for stream *a* was also calculated with “DuctFlow”. Note that although the *a* and *b* flow rates are different, the convection coefficients are very similar, because both flow regimes are laminar.

The energy balance of the wet channel stream, also assuming no losses to the outside, includes convection from the humid air (*b*) to the saturated surface (*s*), and the transfer of latent heat from the surface to the air; the difference between both is due to the heat coming across the separating wall. The corresponding equation is:

$$\dot{M}_{b,dry}(h_{ent}[i] - h_{ent}[i + 1]) = \dot{M}_{evap}[i] \Delta h_{lv} - h_b(\Delta x \cdot width)(T_a[i + 1] - T_s[i]) \quad (4.10.4)$$

An overall energy balance will state that the total energy received by the wet stream (*b*) comes from the dry stream (*a*). That is:

$$\dot{M}_{b,dry}(h_{ent}[i] - h_{ent}[i + 1]) = \dot{M}_a c_{p,a}(T_a[i] - T_a[i + 1]) \quad (4.10.5)$$

Equations (4.10.2) to (4.10.5) allow the calculation of the 4 unknown variables. However, after implementation in EES, it was found that convergence was almost impossible. Because the energy balance at the saturated surface – equation (4.10.4) – involves very small numbers, associated with the evaporation rate, and at the same time very large numbers, associated with the enthalpies, this leads to numerical instabilities and convergence problems. To avoid this, a different form of the energy balance equation was used, using the combined heat and mass methodology followed by ASHRAE, [8]. With this approach, equation (4.10.4) can be rewritten as

$$\dot{M}_{b,dry}(h_{ent}[i] - h_{ent}[i + 1]) = k_m(\Delta x \cdot width)(h_{ent,s}[i] - h_{ent}[i + 1]) \quad (4.10.6)$$

where k_m is a mass transfer coefficient, related to h_m through

$$k_m = h_m \rho_{dry\ air} = \frac{h}{c_{p,b}} \quad (4.10.7)$$

and $h_{ent,s}$ is the enthalpy of vapour at the wet surface:

$$h_{ent,s}[i] = \text{enthalpy}(\text{AirH2O}; T = T_s[i]; R = 1; P = 101.3) \quad (4.10.8)$$

Using the same methodology, equation (4.10.2) could have been replaced by

$$\dot{M}_{b,dry}(\omega[i] - \omega[i + 1]) = k_m(\Delta x \cdot width)(\omega_s[i] - \omega[i + 1]) \quad (4.10.9)$$

With equation (4.10.6) no convergence problems occurred, and it was not even necessary to change the default guess values for any variables, starting with all unknown variables equal to 1.

Figure 4.10.3 shows the Formatted Equations window, and Figure 4.10.4 the Equations Window. Note that the default energy units (kJ) were changed to J, to avoid the constant multiplication of enthalpies by 1000.

Two performance indicators were also calculated: the wet bulb efficiency (η_{wb}), quantifying the approach to the inlet air wet bulb temperature, and the dew point efficiency (η_{dp}), quantifying the approach to the inlet dew point temperature; they are defined as

$$\eta_{wb} = \frac{T_a[1] - T_a[N]}{T_a[1] - T_{wb}[1]} \quad (4.10.10)$$

and

$$\eta_{dp} = \frac{T_a[1] - T_a[N]}{T_a[1] - T_{dp}[1]} \quad (4.10.11)$$

The exit temperature ($T_a[N]$) will be lower than the wet bulb temperature, approaching the dew point temperature, and therefore the wet bulb efficiency will be higher than 1.

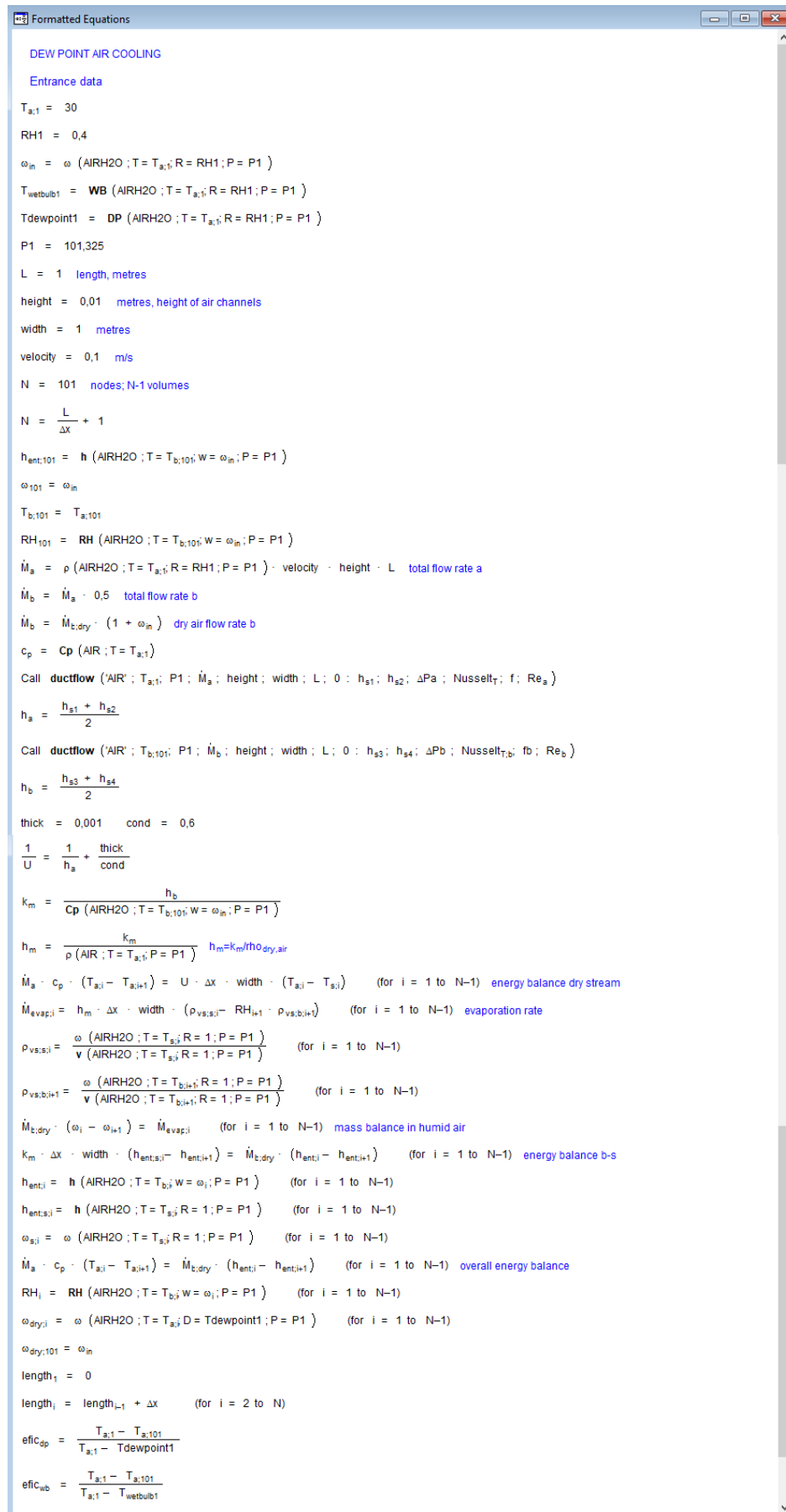


Figure 4.10.3 – Formatted Equations window for the dew point cooler example.


```

"DEW POINT AIR COOLING"
"Entrance data"
T_a[1]=30
RH1=0,4
omega_in=HUMRAT(AirH2O;T=T_a[1];R=RH1;P=P1)
T_wetbulb1=WeTbulb(AirH2O;T=T_a[1];r=RH1;P=P1)
Tdewpoint1=DewPoint(AirH2O;T=T_a[1];r=RH1;P=P1)
P1=101,325
L=1 "length, metres"
height=0,01 "metres, height of air channels"
width=1 "metres"
velocity=0,1 "m/s"
N=101 "nodes; N-1 volumes"
N=L/DELTAx+1
h_ent[N]=ENTHALPY(AirH2O;T=T_b[N];w=omega_in;P=P1)
omega[N]=omega_in
T_b[N]=T_a[N]
RH[N]=RelHum(AirH2O;T=T_b[N];w=omega_in;P=P1)

M_dot_a=Density(AirH2O;T=T_a[1];r=RH1;P=P1)*velocity*height*L "total flow rate a"
M_dot_b=M_dot_a*0,5 "total flow rate b"
M_dot_b_dry=M_dot_b*(1+omega_in) "dry air flow rate b"
c_p=Cp(Air;T=T_a[1])
call DuctFlow(air;T_a[1];P1;M_dot_a;height;width;L;0:h_s1;h_s2 ;DELTApa;Nusselt_T,f;Re_a)
h_a=(h_s1+h_s2)/2
call DuctFlow(air;T_b[N];P1;M_dot_b;height;width;L;0:h_s3;h_s4 ;DELTApb;Nusselt_T,b;fb;Re_b)
h_b=(h_s3+h_s4)/2
thick=0,001: cond=0,6
1/U=1/h_a+thick/cond
k_m=h_b/Cp(AirH2O;T=T_b[N];w=omega_in;P=P1)
h_m=k_m/Density(Air;T=T_a[1];P=P1) "h_m=k_m/rho_dry_air"

duplicate i=1; N-1
  M_dot_a*c_p*(T_a[i]-T_a[i+1])=U*DELTAx*width*(T_a[i]-T_s[i]) "energy balance dry stream"

  M_dot_evap[i]=h_m*DELTAx*Width*(rho_vs_s[i]-RH[i+1]*rho_vs_b[i+1]) "evaporation rate"
  rho_vs_s[i]=humrat(AirH2O;T=T_s[i];R=1;P=P1)/volume(AirH2O;T=T_s[i];R=1;P=P1)
  rho_vs_b[i+1]=humrat(AirH2O;T=T_b[i+1];R=1;P=P1)/volume(AirH2O;T=T_b[i+1];R=1;P=P1)
  M_dot_b_dry*(omega[i]-omega[i+1])=M_dot_evap[i] "mass balance in humid air"

  k_m*DELTAx*width*(h_ent_s[i]-h_ent[i+1])=M_dot_b_dry*(h_ent[i]-h_ent[i+1]) "energy balance b-s"
  h_ent[i]=Enthalpy(AirH2O;T=T_b[i];w=omega[i];P=P1)
  h_ent_s[i]=Enthalpy(AirH2O;T=T_s[i];r=1;P=P1)
  omega_s[i]=HumRat(AirH2O;T=T_s[i];r=1;P=P1)

  M_dot_a*c_p*(T_a[i]-T_a[i+1])=M_dot_b_dry*(h_ent[i]-h_ent[i+1]) "overall energy balance"
  RH[i]=RelHum(AirH2O;T=T_b[i];w=omega[i];P=P1)
  omega_dry[i]=HumRat(AirH2O;T=T_a[i];D=Tdewpoint1;P=P1)
end
omega_dry[N]=omega_in
length[1]=0
duplicate i=2; N
  length[i]=length[i-1]+DELTAx
end
efic_dp=(T_a[1]-T_a[N])/(T_a[1]-Tdewpoint1)
efic_wb=(T_a[1]-T_a[N])/(T_a[1]-T_wetbulb1)

```

Figure 4.10.4 –Equations Window for the dew point cooler example.

Figure 4.10.5 shows the results for 101 nodes (100 volumes) in each channel. Air in the dry channel reaches a temperature well below wet bulb. The dew point efficiency is close to 1. The wet channel air is almost saturated, during the flow length. Figure 4.10.6 shows the effect of reducing the number of nodes to 41. In this example, it is more advisable to use the higher number of nodes; a further increase produces similar results to 101.

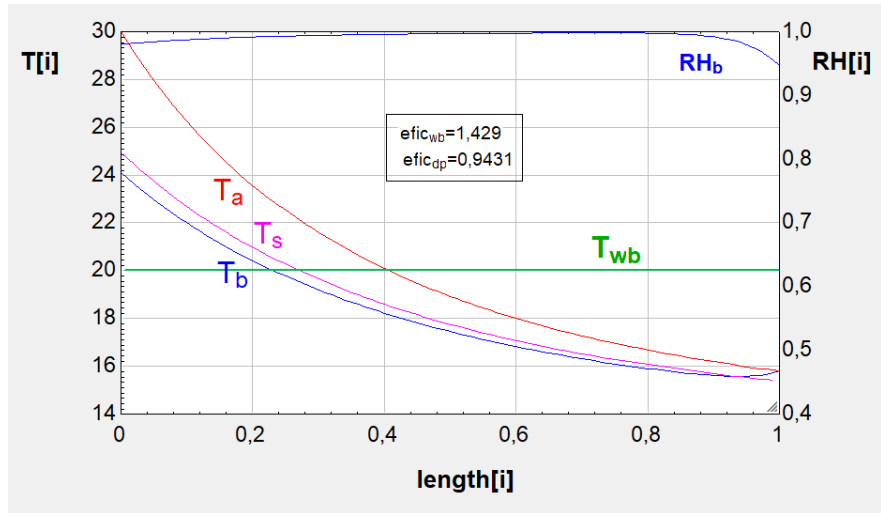


Figure 4.10.5 – Evolution of stream (*a* and *b*) temperatures, wet surface temperature and relative humidity (wet channel). Comparison with inlet air wet bulb temperature.

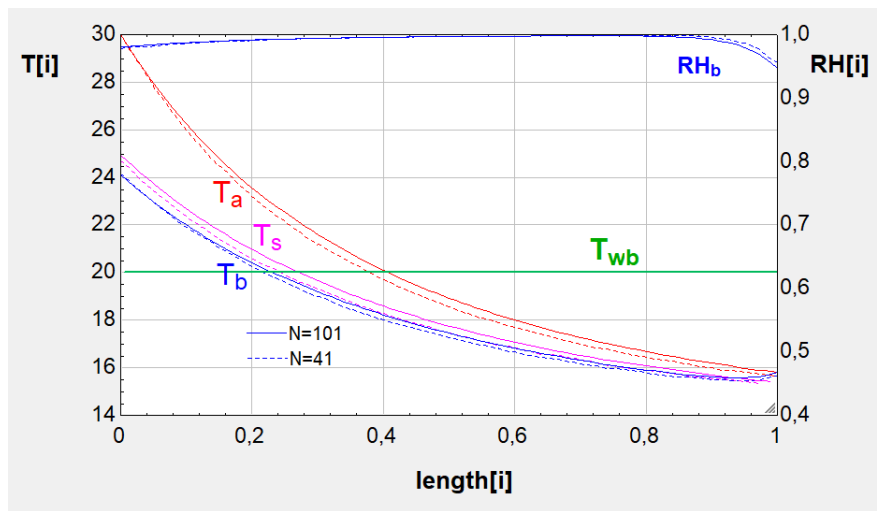


Figure 4.10.6 – Evolution of stream temperatures, wet surface temperature and relative humidity (wet channel) for two different node numbers (41 and 101).

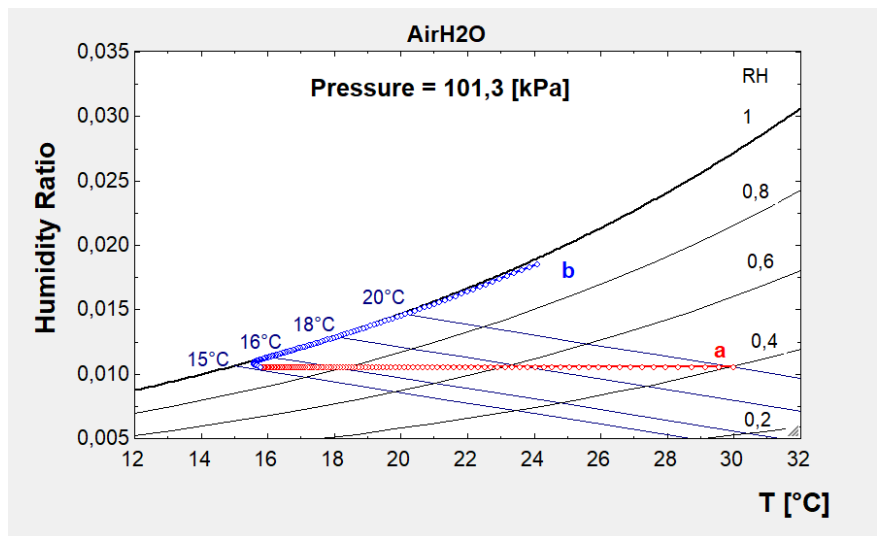


Figure 4.10.7 – Psychrometric chart representation of air stream evolutions in the 2 channels: *a* – dry channel; *b* – wet channel.

Figure 4.10.7 shows another type of graph that may be obtained with EES: a psychrometric graph, representing the air temperature and absolute humidity (ω) of the air, along the 2 channels. The dry stream (*a*) is cooled at constant absolute humidity (no mass transfer), from 30°C down to 15.8°C, and the wet stream (*b*) evolves always close to the saturation line, leaving at 24.1°C.

Note that the use of a higher inlet air flow rate may compromise the objective of reaching a lower than wet bulb temperature. To assess that, further results were obtained with an inlet velocity of 0.4 m/s (4 times higher). The results are shown in Figures 4.10.8 and 4.10.9. With the higher flow rates (maintaining 50% in the wet channel), the outlet temperature of the dry channel is now 20.7°C, higher than the wet bulb temperature. The wet bulb efficiency is now only equal to 0.93.

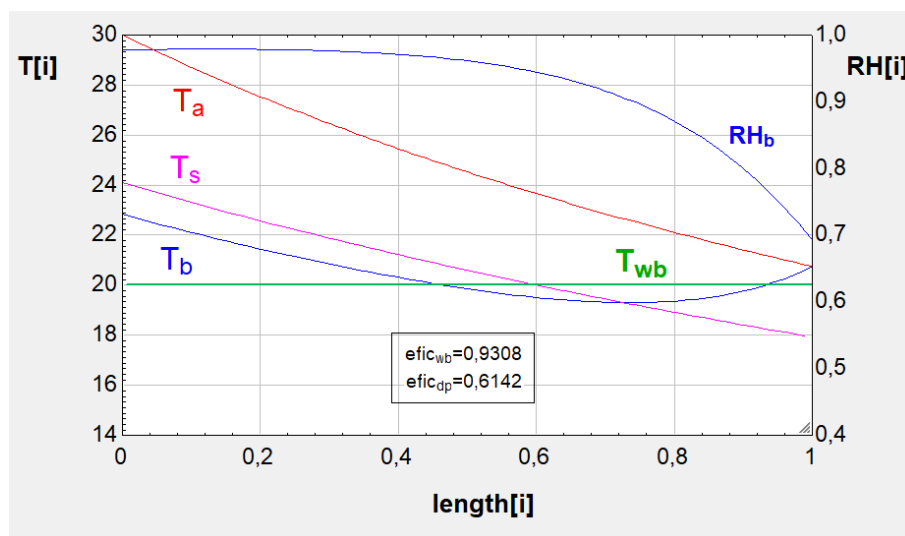


Figure 4.10.8 – Model results for an inlet velocity of 0.4 m/s (dry channel).

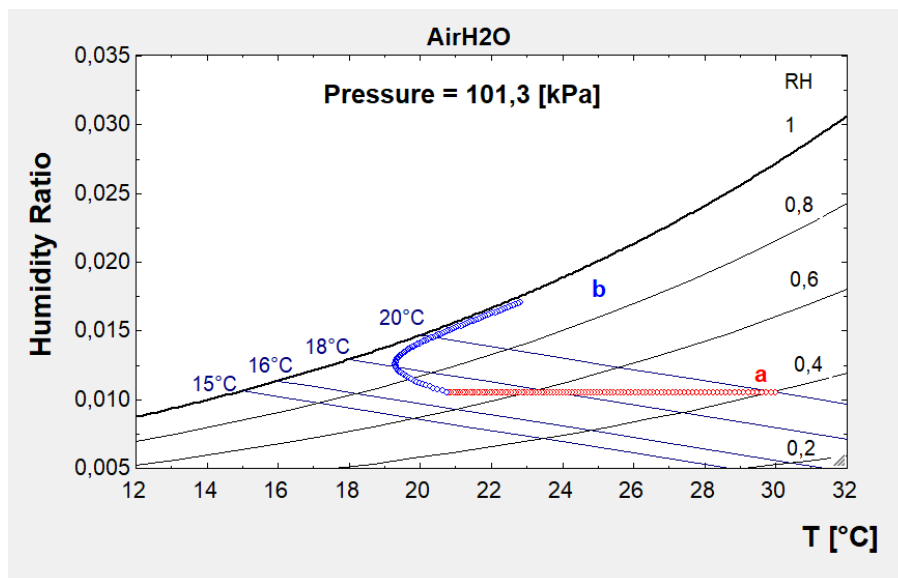


Figure 4.10.9 – Model results for an inlet velocity of 0.4 m/s (dry channel).

4.11 Domestic hot water solar system with thermal storage stratification

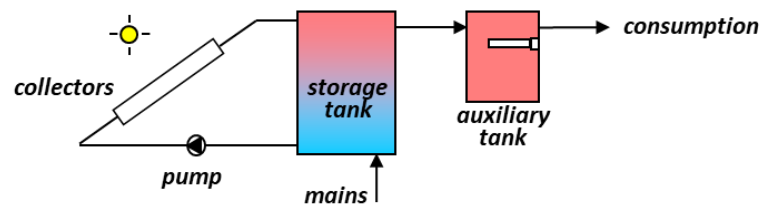


Figure 4.11.1 – Schematic representation of a DHW solar system.

Consider the domestic hot water solar system of example 3.5, in its last configuration, with a separate auxiliary tank, electrically heated with proportional control. We want to assess the effect of thermal stratification in the storage tank, using different volumes in the vertical direction. The auxiliary tank can still be considered at uniform temperature.

Additional data are:

- storage tank diameter: 0.5 m;
- storage tank height: 1.528 m;
- storage tank heat loss coefficient: $U=0.645 \text{ W/m}^2\text{C}$ ($UA=1.8 \text{ W/}^\circ\text{C}$).

The model of the stratified storage tank is based on its division into finite volumes, and 10 volumes of equal size will be considered. With a total tank height of 1.528 m, this means that each volume will have a height (ΔH) of 15.28 cm. The temperature is considered uniform in each volume and the water flows between elements when the pumps are activated (collector circuit or consumption circuit). Figure 4.11.2 shows the water circulation inside the tank, assuming vertical velocity only, under 2 different situations: collector flow rate is higher, or consumption flow rate is higher.

Under the specified consumption flow rates, when the collector pump operates the collector flow rate is always higher (0.08 kg/s) and there is a downward flow. When the collector pump is switched-off, if there is water consumption there will be an upward flow, and if no consumption occurs there will be no flow inside the tank.

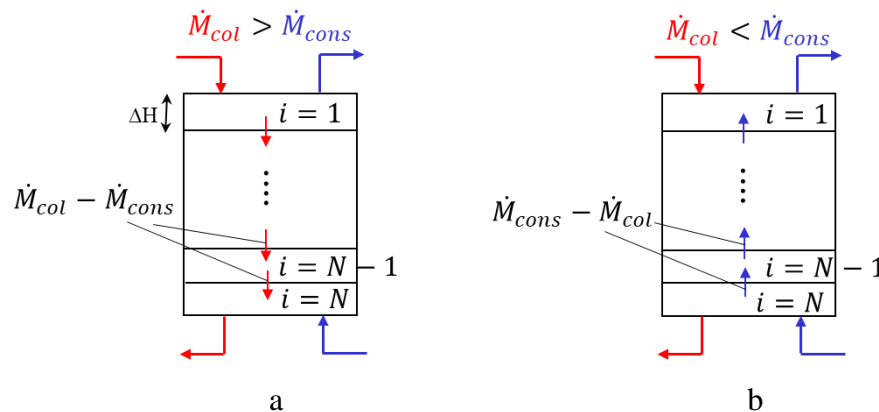


Figure 4.11.2 – Storage tank finite volumes and flow rates: (a) collector flow rate higher than consumption flow rate; (b) consumption flow rate higher than collector flow rate.

Figure 4.11.3 shows the mass and energy flows in different tank volumes, assuming that all entrances/exits are located in the extreme volumes (top - 1 - or bottom - N). Besides the energy transported by the flows, each volume will exchange heat by conduction with the neighbour volumes, and will loose heat through the tank wall to the outside (*ext*).

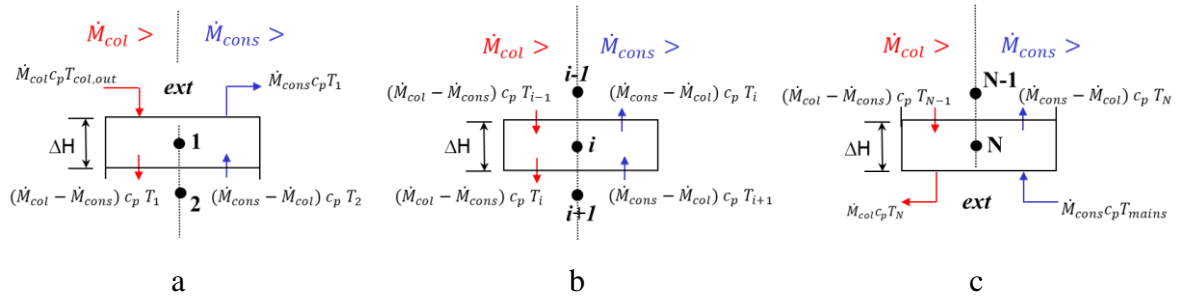


Figure 4.11.3 – Transported energy in finite volumes: (a) top volume ($i = 1$); (b) internal volume (i); (c) bottom volume ($i = N$). Two different situations are represented: higher collector flow rate, or higher consumption rate.

The two different situations represented in Figure 4.11.3 – higher collector flow rate (with 0 or positive water consumption) and higher consumption rate (with no collector flow in this case) may be written, for each of the 3 volume types, with the help of a pump factor, as already defined in example 3.5 (f_{pump}). This factor is equal to 0 when the pump is switched off, and to 1 when the pump is switched on (when the collector outlet temperature is higher than the inlet one). With the help of this factor the discretised equation for the top volume, using the implicit method, is:

$$\begin{aligned} \rho \Delta H A_s c_p \frac{T_1^{t+\Delta t} - T_1^t}{\Delta t} &= \dot{M}_{col}^{t+\Delta t} c_p T_{col,out}^{t+\Delta t} - \dot{M}_{cons}^{t+\Delta t} c_p T_1^{t+\Delta t} + \\ &+ (\dot{M}_{cons}^{t+\Delta t} - f_{pump} \dot{M}_{col}^{t+\Delta t}) c_p [f_{pump} T_1^{t+\Delta t} + (1 - f_{pump}) T_2^{t+\Delta t}] + \\ &+ \frac{k_w}{\Delta H} A_s (T_2^{t+\Delta t} - T_1^{t+\Delta t}) - U(\pi D \Delta H + A_{top}) (T_1^{t+\Delta t} - T_{ext}) \quad (4.11.1) \end{aligned}$$

For an internal volume ($i = 2$ to $N - 1$) it is:

$$\begin{aligned} \rho \Delta H A_s c_p \frac{T_i^{t+\Delta t} - T_i^t}{\Delta t} &= \\ &= (\dot{M}_{cons}^{t+\Delta t} - f_{pump} \dot{M}_{col}^{t+\Delta t}) c_p [f_{pump} (T_i^{t+\Delta t} - T_{i-1}^{t+\Delta t}) + (1 - f_{pump}) (T_{i+1}^{t+\Delta t} - T_i^{t+\Delta t})] - \\ &- \frac{k_w}{\Delta H} A_s (T_{i-1}^{t+\Delta t} + T_{i+1}^{t+\Delta t} - 2T_i^{t+\Delta t}) - U \pi D \Delta H (T_i^{t+\Delta t} - T_{ext}) \quad (4.11.2) \end{aligned}$$

and for the bottom volume:

$$\begin{aligned} \rho \Delta H A_s c_p \frac{T_N^{t+\Delta t} - T_N^t}{\Delta t} &= \dot{M}_{cons}^{t+\Delta t} c_p T_{mains} - \dot{M}_{col}^{t+\Delta t} c_p T_N^{t+\Delta t} - \\ &- (\dot{M}_{cons}^{t+\Delta t} - f_{pump} \dot{M}_{col}^{t+\Delta t}) c_p [f_{pump} T_{N-1}^{t+\Delta t} + (1 - f_{pump}) T_N^{t+\Delta t}] + \\ &+ \frac{k_w}{\Delta H} A_s (T_{N-1}^{t+\Delta t} - T_N^{t+\Delta t}) - U(\pi D \Delta H + A_{bot}) (T_N^{t+\Delta t} - T_{ext}) \quad (4.11.3) \end{aligned}$$

The additional equation needed is the collector energy balance:

$$0.8 I_{sol}^{t+\Delta t} A_{col} - 5(T_N^{t+\Delta t} - T_{amb}^{t+\Delta t}) A_{col} = \dot{M}_{col}^{t+\Delta t} c_p (T_{col,out}^{t+\Delta t} - T_N^{t+\Delta t}) \quad (4.11.4)$$

Equations (4.11.1) to (4.11.4) allow the calculation of the N storage tank temperatures (10 in our case) and the collector outlet temperature.

The auxiliary tank is translated in the model in the same manner as in example 3.5, taking into account that the inlet water temperature is equal to the top storage tank temperature (T_1):

$$\frac{T_{aux}^{t+\Delta t} - T_{aux}^t}{\Delta t} = \frac{1}{M_{aux} c_p} \left[P_{resist}^{t+\Delta t} + \dot{M}_{cons}^{t+\Delta t} c_p (T_1^{t+\Delta t} - T_{aux}^{t+\Delta t}) - (UA)_{tank} (T_{aux}^{t+\Delta t} - T_{ext}) \right] \quad (4.11.5)$$

with the resistance input (P_{resist}) depending on the control function.

Figure 4.11.4 shows the EES Equations Window containing the model equations. The PUMP and RESIST functions are the same as those used in example 3.5. There was, however, the need to obtain the f_{pump} factor using the inlet and outlet collector temperatures at the previous time (t) and not $t + \Delta t$; this was necessary to achieve convergence in the f_{pump} calculation. This represents an explicit method influence in the otherwise implicit method of discretisation. However, no significant errors occur, as the time steps are small (60 s).

Figure 4.11.5 shows the corresponding formatted equations window.

```

"solar DHW system with stratified storage and separate tank with aux heater with proportional control"
FUNCTION PUMP(T_col_in,T_col_out)
  IF T_col_out>T_col_in THEN
    PUMP=1
  ELSE
    PUMP=0
  ENDIF
END
FUNCTION RESIST(T)
  P_max=1500
  T_max=55
  T_min=50
  IF T>T_max THEN
    RESIST=0
  ELSE
    IF T<T_min THEN RESIST=P_max ELSE RESIST=P_max*(T_max-T)/(T_max-T_min)
  ENDIF
END
V=300/1000 "m3, storage tank"
DELTA=60*1 "s"
rho=Density(Water, T=50, p=100)
c_p=Cp(Water, T=50, p=100)*1000 "in J/kgK"
k_w=Conductivity(Water, T=50, p=100)
T_ext=20
T_mains=15
U=0.645
N=10
H=1.528
D=0.5
A_s=pi*D^2/4 "section area"
DELTAH=H/N
V_aux=50/1000 "m3, auxiliary tank"
UA_aux=0.6
eta_0_col=0.8
FU_col=5
A_col=4
M_dot_col=0.080 "kg/s"
hour=time/3600
line=1+time/DELTA
T_amb=Interpolate1(Lookup 1, T_amb, hour, hour=time/3600)
Rad_sol=Interpolate1(Lookup 1, Rad_sol, hour, hour=time/3600)
M_dot_cons=Interpolate1(Lookup 1, consumption, hour, hour=trunc(time/3600))/3600
Rad_sol*A_col*eta_0_col*FU_col*(T[N]-T_amb)=M_dot_col*c_p*(T_col_out-T[N])
"storage tank top volume, i=1"
T_old[1]=TableValue(Table 1, line-1, #T[1])
rho*A_s*DELTAH*c_p*(T[1]-T_old[1])/DELTA=U*(pi*D*DELTAH+A_s)*(T[1]-T_ext)+M_dot_cons*c_p*(T[1]+M_dot_col*f_pump*c_p*T_col_out+(M_dot_cons-M_dot_col*f_pump)*c_p*(f_pump*(T[1]+(1-f_pump)*T[2])+k_w*A_s/DELTAH*(T[2]-T[1]))
"storage tank internal volumes: 2 to N-1"
duplicate i=2, N-1
T_old[i]=TableValue(Table 1, line-1, #T[i])
rho*A_s*DELTAH*c_p*(T[i]-T_old[i])/DELTA=U*(pi*D*DELTAH*(T[i]-T_ext)+(M_dot_cons-M_dot_col*f_pump)*c_p*(f_pump*(T[i]-T[i-1])+(1-f_pump)*(T[i]+T[i-1]))+k_w*A_s/DELTAH*(T[i-1]+T[i-1]-T[i]))
end
"storage tank bottom volume, i=N"
T_old[N]=TableValue(Table 1, line-1, #T[10]) "Note: when changing N value, change also 10 here"
rho*A_s*DELTAH*c_p*(T[N]-T_old[N])/DELTA=U*(pi*D*DELTAH+A_s)*(T[N]-T_ext)+M_dot_cons*c_p*(T_mains-M_dot_col*f_pump*c_p*(T[N]-M_dot_cons-M_dot_col*f_pump)*c_p*(f_pump*(T[N-1]+(1-f_pump)*T[N])+k_w*A_s/DELTAH*(T[N-1]-T[N]))
f_pump=PUMP(T_old[N], T_col_out, old) "old values (previous time) for convergence"
T_col_out_old=TableValue(Table 1, line-1, #T_col_out)
P_resist=RESIST(T_aux)
T_aux_old=TableValue(Table 1, line-1, #T_aux) "recovers previous T_aux"
rho*V_aux*c_p*(T_aux-T_aux_old)/DELTA=M_dot_cons*c_p*(T[1]-T_aux)+P_resist*UA_aux*(T_aux-T_ext)
Energy_cons=Energy_cons_ant+P_resist*DELTA/1000/3600 "em kWh"
Energy_cons_ant=TableValue(Table 1, line-1, #Energy_cons)
Energy_sol=Energy_sol_ant+M_dot_col*f_pump*c_p*(T_col_out-T[N])/DELTA/1000/3600 "em kWh"
Energy_sol_ant=TableValue(Table 1, line-1, #Energy_sol)

```

Figure 4.11.4 – EES Equations Window for the DHW solar system example.

```

Formatted Equations

solar DHW system with stratified storage and separate tank with aux heater with proportional control

Function PUMP (T_coolin, T_coolout)
If (T_coolout > T_coolin) Then
    PUMP := 1
Else
    PUMP := 0
Endif
End PUMP

Function RESIST (T)
P_max := 1500
T_max := 55
T_min := 50
If (T >= T_max) Then
    RESIST := 0
Else
    If (T <= T_min) Then RESIST := P_max Else RESIST := P_max * (T_max - T) / (T_max - T_min)
Endif
End RESIST

V = 300 / 1000 m3, storage tank
dt = 60 * 1 s
rho = rho (water, T = 50, P = 100)
Cp = Cp (water, T = 50, P = 100) * 1000 in J/kgK
kw = k (water, T = 50, P = 100)
T_ext = 20
T_mains = 15
U = 0.645
N = 10
H = 1.528
D = 0.5
As = pi * D^2 / 4 section area
dH = H / N
V_aux = 50 / 1000 m3, auxiliary tank
UA_aux = 0.6
eta_cool = 0.8
FU_cool = 5
A_cool = 4
M_cool = 0.08 kg/s
hour = time / 3600
line = 1 + time / dt

T_amb = Interpolate1 [Lookup 1; 'hour'; 'T_amb'; 'hour' = time / 3600]
Rad_sol = Interpolate1 [Lookup 1; 'hour'; 'Rad_sol'; 'hour' = time / 3600]
M_ccs = Interpolate1 [Lookup 1; 'hour'; 'consumption'; 'hour' = Trunc (time / 3600)]
Rad_sol * A_cool * eta_cool - FU_cool * A_cool * (T_10 - T_amb) = M_cool * Cp * (T_coolout - T_10)

storage tank top volume, i=1
T_cool1 = TableValue (Table 1; line - 1; 'T_1')
rho * As * dH * Cp * (T_1 - T_cool1) / dt = -U * (pi * D * dH * As) * (T_1 - T_ext) - M_ccs * Cp * T_1 + M_cool * f_pump * Cp * T_coolout + (M_ccs - M_cool * f_pump) * Cp * (f_pump * T_1 + (1 - f_pump) * T_2) + kw * As / dH * (T_2 - T_1)

storage tank internal volumes, 2 to N-1
T_cooli = TableValue (Table 1; line - 1; 'T_i') (for i = 2 to N-1)
rho * As * dH * Cp * (T_i - T_cooli) / dt = -U * (pi * D * dH * As) * (T_i - T_ext) + (M_ccs - M_cool * f_pump) * Cp * (f_pump * (T_i - T_{i-1}) + (1 - f_pump) * (T_{i+1} - T_i)) + kw * As / dH * (T_{i+1} + T_{i-1} - 2 * T_i) (for i = 2 to N-1)

storage tank bottom volume, i=N
T_cool10 = TableValue (Table 1; line - 1; 'T_10') note: when changing N value, change also 10 here
rho * As * dH * Cp * (T_10 - T_cool10) / dt = -U * (pi * D * dH * As) * (T_10 - T_ext) + M_ccs * Cp * T_mains - M_cool * f_pump * Cp * T_10 - ((M_ccs - M_cool * f_pump) * Cp * (f_pump * T_9 + (1 - f_pump) * T_10)) + kw * As / dH * (T_9 - T_10)

f_pump = PUMP (T_cool10, T_coolout) old values (previous time) for convergence
T_cooloutold = TableValue (Table 1; line - 1; 'T_coolout')
P_resist = RESIST (T_coolout)
T_coolout = TableValue (Table 1; line - 1; 'T_coolout') recovers previous T_coolout
rho * V_aux * Cp * (T_coolout - T_cooloutold) / dt = M_ccs * Cp * (T_1 - T_coolout) + P_resist * UA_aux * (T_coolout - T_ext)

Energy_ccs = Energy_ccs_start + P_resist * dt / 1000 * 3600 em kWh
Energy_ccs_start = TableValue (Table 1; line - 1; 'Energy_ccs_start')
Energy_cool = Energy_cool_start + M_cool * f_pump * Cp * (T_coolout - T_10) * dt / 1000 * 3600 em kWh
Energy_cool_start = TableValue (Table 1; line - 1; 'Energy_cool_start')

```

Figure 4.11.5 – EES Formatted Equations window for the DHW solar system example.

The resistance electrical consumption and the solar collector contribution are also calculated, as done in example 3.5. For the energy gained in the solar collectors:

$$\text{Energy}_{\text{sol}} = \text{Energy}_{\text{sol,old}} + f_{\text{pump}} * \dot{M}_{\text{col}} * c_p * (T_{\text{col,out}} - T_{10}) * \Delta t \quad (4.11.6)$$

Figure 4.11.6 shows the storage tank temperature evolution in different volumes, as well as the auxiliary tank temperature. Temperatures in the storage tank have a large spatial variation, which exceeds 20°C during the nighttime. During sunshine hours there is circulation in the collectors and in the storage tank, with much closer temperature values (due to water mixing). Figure 4.11.7 compares the collector inlet (same as storage tank bottom) and outlet temperature evolution. The temperature at the top of the storage tank is always very close to the collector outlet temperature.

The consumption temperature (auxiliary tank temperature) is very stable, with values between 53 and 55°C; thermal stratification leads to a higher temperature at the storage top, increasing the auxiliary tank inlet temperature and decreasing its energy consumption.

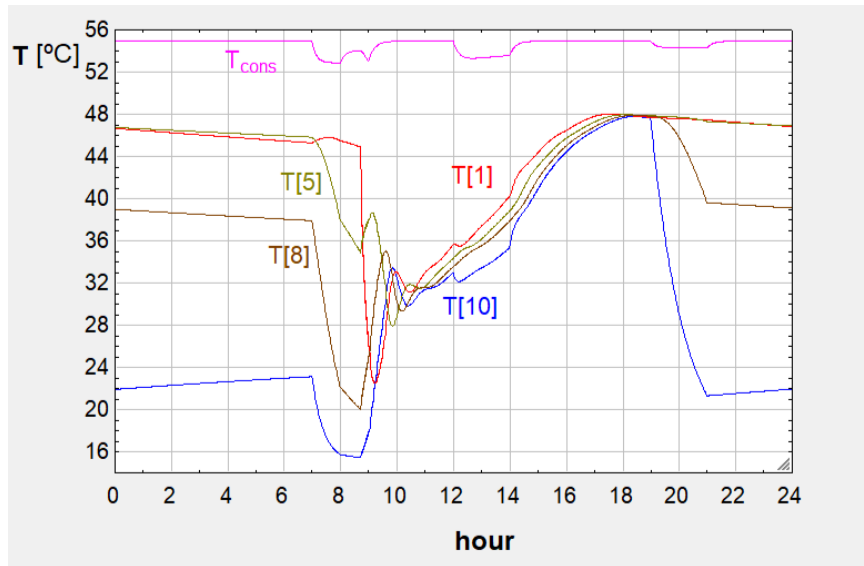


Figure 4.11.6 – Temperatures in the storage tank and auxiliary tank for the DHW solar system example.

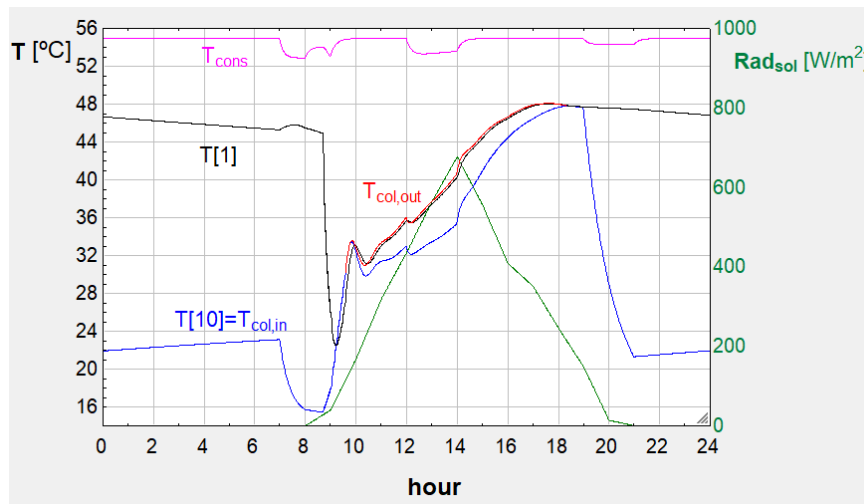


Figure 4.11.7 – Collector inlet and outlet temperatures for the DHW solar system example.

Besides the collector inlet and outlet temperature evolution, Figure 4.11.8 represents the pump factor. It is equal to 1 (there is collector circulation) during most of the sunshine hours. In the early morning period, between 8:45 and 9:30, the pump switches on and off several times, due to low solar radiation and very close inlet and outlet temperatures. This could be avoided by imposing circulation (pump switching on) with a larger temperature difference (outlet-inlet).

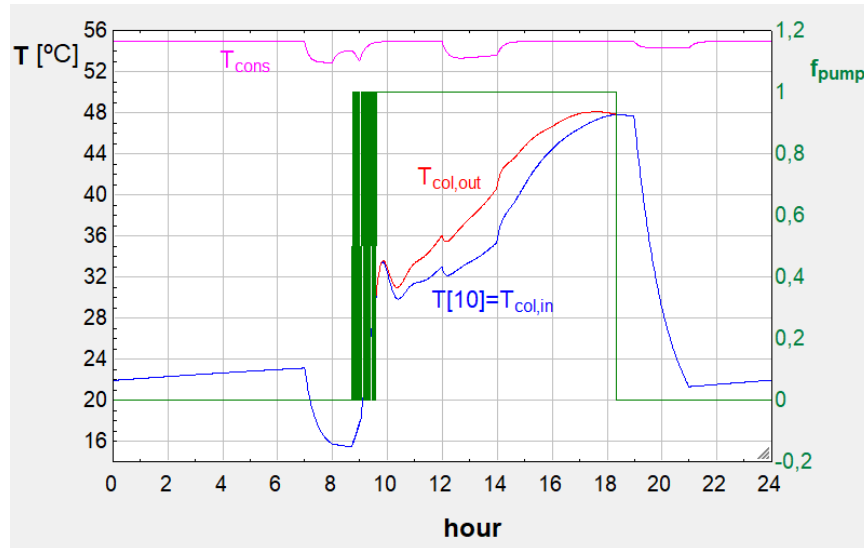


Figure 4.11.8 – Collector inlet and outlet temperatures and pump factor for the DHW solar system example.

Figure 4.11.9 shows the electrical resistance consumption. Due to the higher auxiliary tank inlet temperature the electrical consumption is smaller, and due to the lower inlet collector temperature, collector efficiency is higher, as well as solar contribution to water heating. Without thermal stratification the total water heat input was equal to 10.307 kWh and the solar contribution was equal to 67%; with thermal stratification the total water heat input is equal to 10.365 kWh and the solar contribution is equal to 73%.

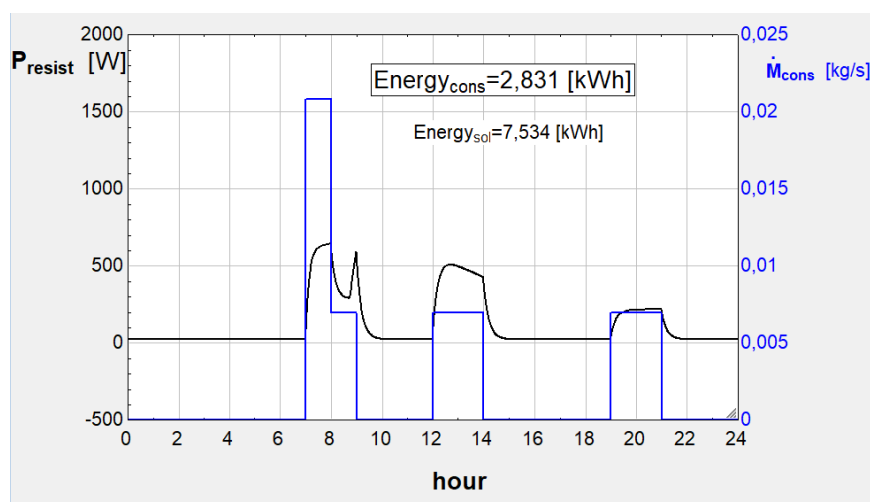


Figure 4.11.9 – Energy and water consumption for the DHW solar system example.

References

- [1] Suhas Patankar (1980). Numerical Heat Transfer and Fluid Flow. Ed. CRC Press.
- [2] Jean-Michel Bergheau, Roland Fortunier (2008). Finite Element Simulation of Heat Transfer. Ed. Wiley.
- [3] F-chart Software (2017). Engineering Equation Solver manual.
- [4] G. F. Nellis, S. A. Klein (2020). Introduction to Engineering Heat Transfer. Cambridge University Press.
- [5] Theodore Bergman, Adrienne Lavine, Frank Incropera, David Dewitt (2011). Fundamentals of Heat and Mass Transfer. 7th edition. Ed. John Wiley & Sons.
- [6] J. A. Orosa, J. Santos, A. C. Oliveira (2010). Introduction to Psychrometrics with EES. ISBN: 9783838365855. Lambert Academic Publishing.
- [7] C. J. Kobus, G. L. Wedekind (1995). An Experimental Investigation into Forced, Natural and Combined Forced and Natural Convective Heat Transfer from Stationary Isothermal Circular Disks. *Int J Heat and Mass Transfer*, vol 38, n 18, pp 3329-3339. Elsevier Science Ltd.
- [8] A.S.H.R.A.E. (2021). Handbook of Fundamentals, Chapter 6. American Society of Heating, Refrigerating and Air Conditioning Engineers.

HEAT TRANSFER: NUMERICAL MODELLING WITH EES APPLICATIONS

ARMANDO C. F. COELHO OLIVEIRA

Professor of Energy Systems at the University of Porto

This book briefly revisits numerical solutions that are applicable to the solution of heat transfer problems and thermal system simulations.

A wide range of numerical modelling examples are presented, including the solution with EES software and detailed analyses of the results.

ISBN 978-972-752-319-1



9 789727 523191 >