EMBEDDED PIPE MODELS

VALIDATION/VERIFICATION

Reference	**PRxxxxxx**
Author	**Roel De Coninck**
Date	**01/09/2011**
Version	**Draft**
Distribution scope	**3E, KULeuven**

1 PURPOSE

This document describes the verification and validation of embedded pipe models for the following applications:

- concrete core activation (tabs)
- capillary tube concrete core activation
- wet (massive) floor heating systems

The models are built for usage for dynamic simulations on building and district level. Therefore, the computation speed is important and a trade-off between speed and precision has to be made.

The most important variables for those simulations are the outlet water temperature and the total heat emitted by the tabs. Therefore, the verification of the models will be based on these two variables.

2 MODELS

2.1 OVERVIEW

An overview of the embedded pipe to be modelled is given in Figure 1. All models are based on (Koschenz and Lehmann 2000) and the derived norm (prEN 15377-1 2005). Details for capillary tubes and floor heating systems are based on (Transsolar 2007). However, as these models are designed for situations with mass flow rate > 0 in the pipe, they are slightly modified in order to account for the dynamic behaviour of the system under on/off flow rate conditions.

info@3E.eu
www.3E.eu

3E nv/sa
Rue du Canal 61
B-1000 Brussels

T +32 2 217 58 68
F +32 2 219 79 89

Fortis Bank 230-0028290-83
IBAN: BE14 2300 0282 9083
SWIFT/BIC: GEBABEBB

RPR Brussels
VAT BE 0465 755 594

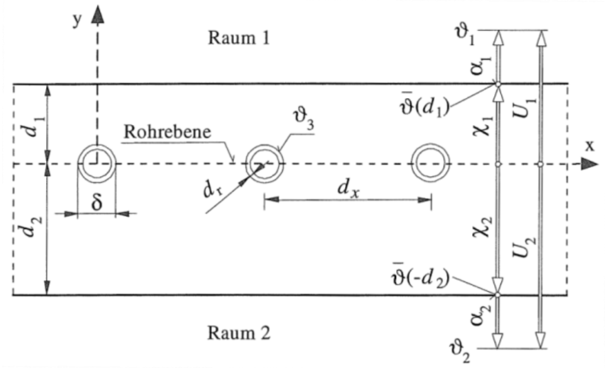

Figure 1 - Section of an embedded pipe (Koschenz and Lehmann 2000)

Koschenz and Lehmann developed a one-dimensional model to compute the mean core temperature θ_k in an activated slab, represented by a resistance network as shown in Figure 2. From this mean core temperature, the heat or cold storage and emission of the slab can be simulated based on an RC model of the slab.

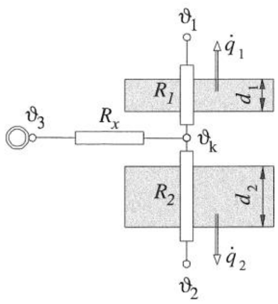

Figure 2 - Equivalent resistance network (Koschenz and Lehmann 2000)

The model is composed of four resistances in series, describing the relation between the entering water temperature and the mean core temperature. The first resistance, R_z, relates the entering water temperature with the mean water temperature in the pipe

$$R_z = \frac{1}{m_{flow,sp} \cdot c_p} \tag{1}$$

$$m_{flow,sp} = \frac{m_{flow}}{d_x \cdot l} \tag{2}$$

with l the total length of the embedded pipe. This resistance is only defined when a mass flow rate is present and does not take into account the inertia of the water in the pipe nor the time it takes for the water to run through the pipe.

Therefore, a different approach was chosen in which the water in the pipe is modelled explicitly to define the mean water temperature TMean. This approach replaces the calculation of R_z. The rest of the model, containing R_w, R_r and R_x is identical to the original one except for the computation of R_x, where variations were taken from (Transsolar 2007) as described in 2.3

2.2 DYNAMIC MODEL DEVELOPMENT

The dynamic model is developed in order to have a more realistic transient behaviour of the system and to enable the simulation of situations without mass flow rate. To this end, the energy balance of the water mass (m) in the embedded pipe is simulated (Figure 3).

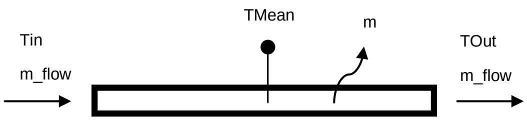

Figure 3 – Dynamic pipe model

The pipe model is not discretised, meaning that the water with mass m is supposed to be completely mixed (all capacity is lumped to a single temperature state). However, this model can be used to discretise the tube into n elements, where each of the elements has the same model. Results of both approaches are compared in the verification section of this document.

It is important to note that the heat flow from the water to the pipe and structure is based upon TMean.

2.2.1 Capacity lumped to TMean

A first model implementation lumps the capacity of the water to TMean directly. As we know that TMean should be the average of TIn and TOut, we can compute TOut based on the other two temperatures. This leads to the following set of equations:

$$m.c_p.\frac{d(T_{Mean})}{dt} + m_{flow}.c_p.(T_{In} - T_{Out}) + Q_{flow} = 0 \tag{3}$$

$$\mathrm{Q_{flow}} = \frac{1}{(\mathrm{R_w}+\mathrm{R_r}+\mathrm{R_x})}(\mathrm{T_{Mean}} - \theta_\mathrm{k}) \tag{4}$$

$$T_{Mean} = \frac{T_{In} + T_{Out}}{2} \tag{5}$$

The advantage of this model is the continuous TMean. The backside is a discontinuous TOut if TIn is subject to a step function. Moreover, TOut can be completely wrong in case of such a step function as will be showed in the validation section.

2.2.2 Capacity lumped to TOut

The most common solution is to lump the capacity to Tout (and specify TMean as the average of TOut and TIn). Therefore equation (3) changes into

$$m.c_p.\frac{d(T_{Out})}{dt} + m_{flow}.c_p.(T_{In} - T_{Out}) + Q_{flow} = 0 \tag{6}$$

Equations (4) and (5) remain, but we need an additional equation to change the value of TIn when there is no flow rate:

$$T_{In} = if\ m_{flow} > 0\ then\ T_{In}\ else\ T_{\mathrm{Out}} \tag{7}$$

This equation is required because when the flow rate drops to zero, we cannot maintain TIn at the outlet temperature of the upstream component. Suppose for instance that the upstream component is a thermal storage tank at 50°C: TIn would 50°C and TMean would be kept at a high temperature, leading to wrong (high) values of Qflow.

This solution has the advantage that TOut will be continuous at all conditions. The disadvantages are that TMean will not be continuous if a step function in TIn is applied or if mflow drops to zero.

2.2.3 Flow rate dependent capacity lumping

A last model is a variant on the previous model in order to ensure continuity in TMean when the flow rate drops to zero. To achieve this, we want the capacity of the water to be lumped on TOut during flow rate conditions, and to TMean at no flow rate conditions.

In Modelica, we can use events to switch equation systems when a condition is reached. In this case, we need to create two events: when the flow rate starts and when it stops.

An auxiliary variable TMeanDyn is created with exactly the same behaviour as TOut and equations (5) and (7) change. The total set of equations is given below :

$$m.c_p.\frac{d(T_{Out})}{dt} + m_{flow}.c_p.(T_{In} - T_{Out}) + Q_{flow} = 0 \tag{8}$$

$$m.c_p.\frac{d(T_{MeanDyn})}{dt} + m_{flow}.c_p.(T_{In} - T_{Out}) + Q_{flow} = 0 \tag{9}$$

$$\mathrm{Q_{flow}} = \frac{1}{(\mathrm{R_w+R_r+R_x})}\,(\mathrm{T_{Mean}} - \theta_\mathrm{k}) \tag{10}$$

$$if\ m_{flow} > m_{flowStart}\ then\ T_{Mean} = \frac{T_{In}+T_{Out}}{2}\ else\ T_{Mean} = T_{MeanDyn} \tag{11}$$

$$when\ m_{flow} < m_{flowStart}\ then\ (EVENT)\ reinit(T_{MeanDyn},\ pre(T_{Mean})) \tag{12}$$

$$when\ m_{flow} > m_{flowStart}\ then\ (EVENT)\ reinit(T_{Out},\ pre(T_{MeanDyn})) \tag{13}$$

Equations (12) and (13) generate an event, and the following reinit() statements are only executed at that event. At the first event, the dynamic state TMeanDyn is reinitialised at the temperature of TMean right before the event. This ensures the continuity of TMean and the correct dynamic behaviour of TMean at no flow conditions.

The second event is required for if the flow rate remains to zero for a 'long' time. As TOut is determined based on QFlow, and QFlow depends on TMean (which is higher than TOut), TOut can go to values below the ambient temperature. This is no problem during no-flow conditions, but at the following start up, TOut needs to have a correct value. Therefore, it is reinitialised to TMean.

Note that when the flow rate is interrupted for a short time, this model will show strange behaviour of TOut (discontinuities).

2.3 DEFINITION OF R_X

The resistance R_x is calculated according to :

$$R_x = \frac{d_x \left[ln\left(\frac{d_x}{\pi \delta}\right) + \sum_{s=1}^{\infty} \frac{g_1(s) + g_2(s)}{s} \right]}{2\pi \lambda_b}$$

$$g_i(s) = \frac{\dfrac{\frac{\alpha_i}{\lambda_b} d_x + 2\pi s}{\frac{\alpha_i}{\lambda_b} d_x - 2\pi s} \cdot e^{-\frac{4\pi s}{d_x} d_{3-i}} - e^{-\frac{4\pi s}{d_x}(d_1+d_2)}}{e^{-\frac{4\pi s}{d_x}(d_1+d_2)} - \dfrac{\frac{\alpha_1}{\lambda_b} d_x + 2\pi s}{\frac{\alpha_1}{\lambda_b} d_x - 2\pi s} \cdot \dfrac{\frac{\alpha_2}{\lambda_b} d_x + 2\pi s}{\frac{\alpha_2}{\lambda_b} d_x - 2\pi s}}$$

Equations 14 and 15: calculation of R_x (Koschenz and Lehmann 2000)

The summation term can be neglected if the following conditions hold:

$$\frac{\delta}{d_x} < 0.2 \qquad (16)$$

$$\frac{d_i}{d_x} > 0.3 \qquad (17)$$

In a floor heating however, d_2 is often smaller than $0.3d_x$. Therefore, the function $g_1(s)$ cannot be neglected and has to be taken into account. This leads to the different formulas for R_x to be used depending on the geometry of the embedded pipe, as shown in Table 1.

Table 1 – Calculation of R_x according to (Transsolar 2007). Note the error in the original document.

	Resistance in x-direction	Criteria
Radiant heating or cooling system (ceiling or wall)	$R_x \approx \frac{d_x \cdot \ln\left(\frac{d_x}{\pi \cdot \delta}\right)}{2 \cdot \pi \cdot \lambda_b}$	$d_x \geq 5.8 * \delta$
Capillary tube system	$R_x \approx \frac{d_x \cdot \frac{1}{3} \cdot \left(\frac{d_x}{\pi \cdot \delta}\right)}{2 \cdot \pi \cdot \lambda_b}$	$dx < 5.8 * \delta$
Floor heating systems	$R_x = \frac{d_x \cdot \left(\ln\left(\frac{d_x}{\pi \cdot \delta}\right) + \sum_{s=1}^{100} \frac{g_1(s)}{s}\right)}{2 \cdot \pi \cdot \lambda_b}$ $g_1(s) = -\frac{\frac{\alpha_2}{\lambda_b} \cdot d_x - 2 \cdot \pi \cdot s}{\frac{\alpha_2}{\lambda_b} \cdot d_x + 2 \cdot \pi \cdot s} \cdot e^{\frac{-4 \cdot \pi \cdot s}{d_x} d_2}$	$\alpha_2 = \frac{\lambda_{Insulation}}{d_{Insulation}} < 1.212 \frac{W}{m^2 K}$ $\frac{\delta}{2} \leq d_2$; $\frac{d_1}{d_x} \leq 0.3$

also equations (16) and (17)

also equations (16) and (17) (for d_1 only)

Wrong!

Has to be $d_1/d_x \geq 0.3$

At this moment, only the radiant heating/cooling system has been implemented and will be validated below.

3 VALIDATION

The original radiant heating/cooling model has been validated in comparison with different measurements and FEM simulations in (Koschenz and Lehmann 2000). The following section will compare the dynamic models with the validated static one (in which R_z is calculated according to equations (1) and (2)).

For convenience, the models will be named according to Table 2.

Table 2 – model naming for validation and verification

Name	Description
static	Static model according to (Koschenz and Lehmann 2000)
dynTMean	Dynamic model, water capacity lumped to TMean
dynTOut	Dynamic model, water capacity lumped to TOut
dynSwitch	Dynamic model, water capacity lumped depending on flowrate
xxx_n	Model xxx, n times discretized

> The most important variables for those simulations are the outlet water temperature and the total heat emitted by the tabs. Therefore, the verification of the models will be mainly based on these two variables.

3.1 EQUATION CHECK

This first step is to check the implementation of the equations. The example of (Koschenz and Lehmann 2000), par 4.6 is used.

All models compute exactly the same values for R_z (only static), R_w, R_r and R_x.

Therefore, we can conclude that there are no typos in the formulas for these resistances.

3.2 STEP TO TIN

This validation is identical to the validation described in (Koschenz and Lehmann 2000), par 4.5.1. The boundary conditions are shown in Figure 4, the expected results in Figure 5.

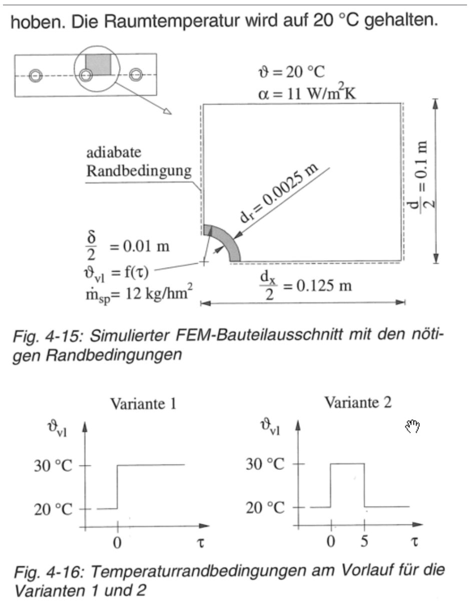
hoben. Die Raumtemperatur wird auf 20 °C gehalten.

Fig. 4-15: Simulierter FEM-Bauteilausschnitt mit den nötigen Randbedingungen

Fig. 4-16: Temperaturrandbedingungen am Vorlauf für die Varianten 1 und 2

Figure 4 – Boundary conditions for step and pulse validations

It's important to note that not all parameters are indicated, the following values are missing: the total floor surface, conduction coefficients of pipe and concrete, density and specific heat concrete. Also, it's not very clear (to me) what the adiabatic conditions mean: is there only heat transfer from the pipe into the upper-right quadrant?

3.2.1 Static model

First, the static model is verified.

I tried to obtain the same results by tuning the missing parameters, for three cases: heat emission only through the top surface, through both surfaces, and through both surfaces, but with d_x=0.125 m (instead 0.25 m). Computing an RMS was not done because the reference temperatures were visually copied from Figure 5, introducing some visible error. The best fits are shown in Figure 6 to Figure 8.

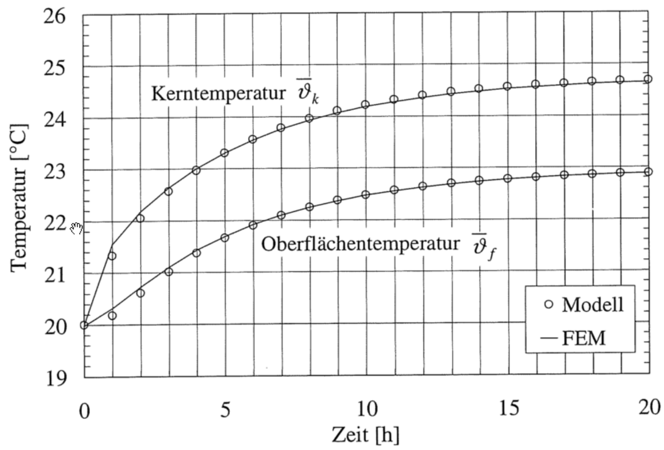

Figure 5 – Reference output for a step to TIn

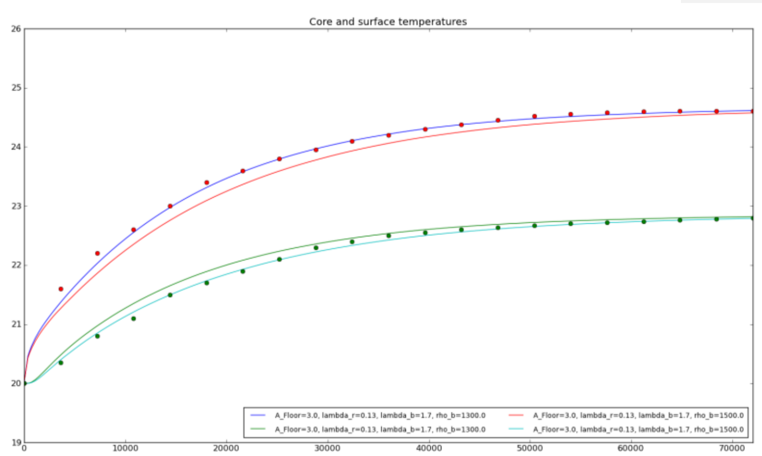

Figure 6 – Simulated output for step to TIn, single sided heat emission

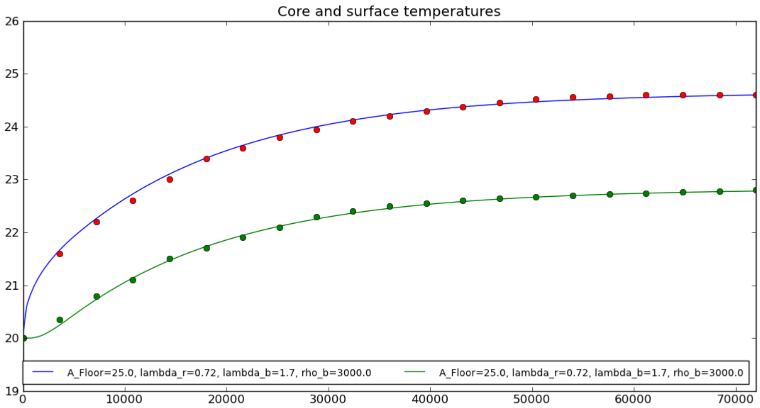

Figure 7 – Simulated output for step to TIn, double sided heat emission

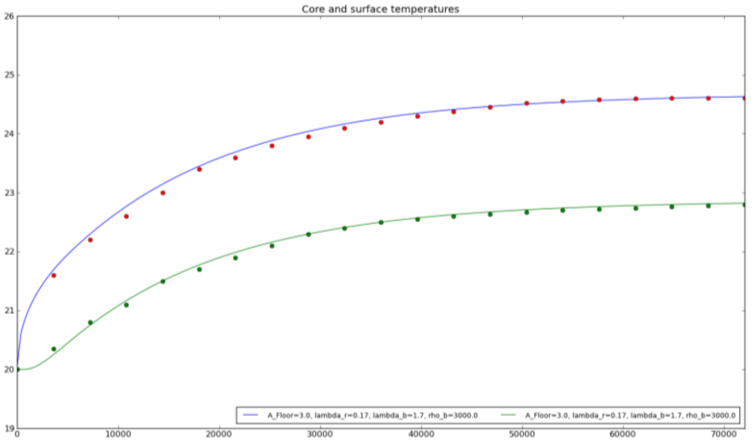

Figure 8 – Simulated output for step to TIn, double sided heat emission and half interdistance

It can be seen that the double sided fits give the best results. Regarding the thermal conductivity of the pipe wall, it has to be noted that the very low value of 0.17 W/mK for the last case is very unlikely.

Therefore, the second case (= double sided fit, normal interdistance) is considered as the best fit.

The high values of the thermal conductivity of the pipe wall and the density of the concrete are unusual (specific heat of concrete was equal to 840 J/kgK for all cases).

All fits were carried out with a discretization of the concrete slab in 50 steps, both above and below the pipe. The effect of this discretization is shown in Figure 9. Discretization has an important effect on the behaviour shortly after the step impulse, but values of n higher than 4 only slightly change the result.

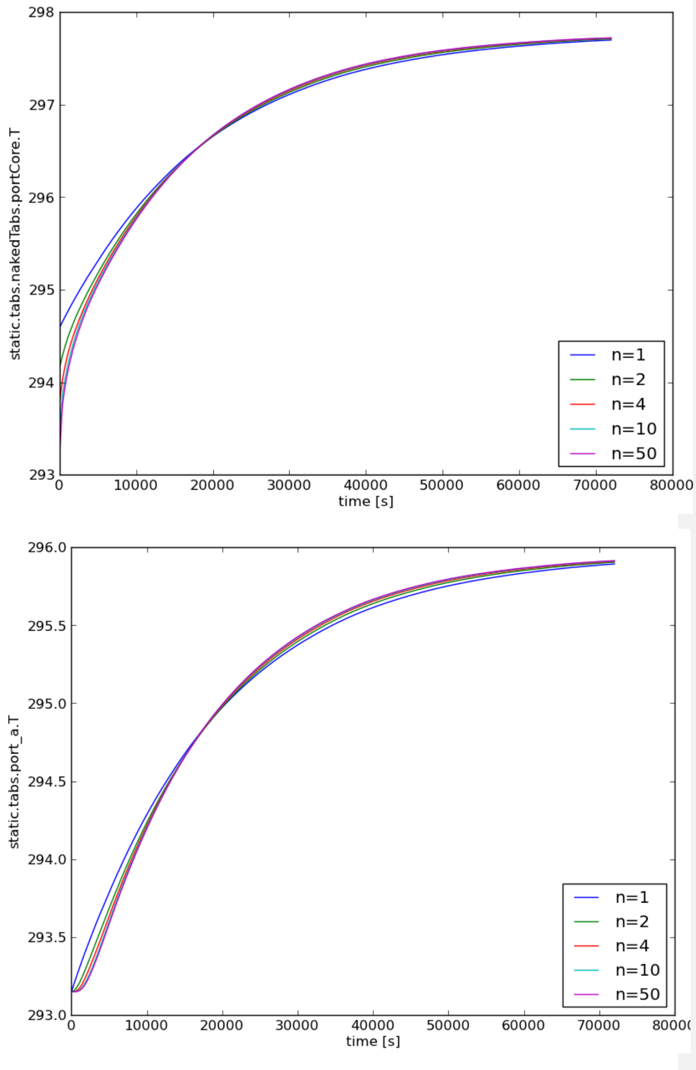

Figure 9 – Effect of discretization on core (upper) and surface temperature (lower)

From this validation exercise, unfortunately we cannot conclude that our model is validated because there are too much unknowns. However, put aside the high values we identified for the thermal conductivity of the pipe wall and the density of the concrete, our model behaves similarly as the validation case.

Another element we did not yet study in detail is the model of the heat conduction from the core to the surface of floor. As indicated, this model is discretized in n1 and n2 elements, above and below the core layer respectively. However, in the previous results, there was no capacity lumped to the core layer itself. For large n1 and n2, there is hardly any difference between both approaches. However, for rough discretizations, which will be interesting in case simulation speed and model size matter, a significant difference can be observed, as shown in Figure 10.

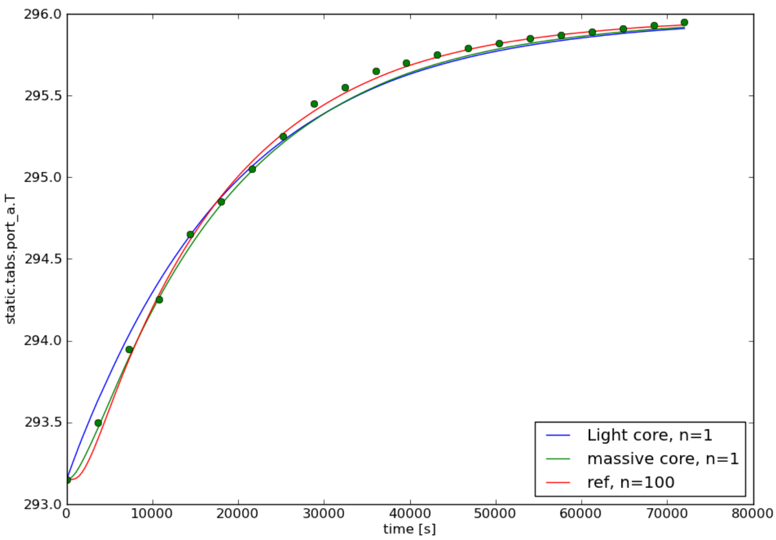

Figure 10 – Comparison of surface temperature for massive and light core to reference case

From this figure it can be seen that immediately after the step impulse, the massive core with n1=n2=1 reacts much closer to the reference results. For values around the time constant of the system, the light core approaches the reference better, but the deviations are smaller. Therefore, it is concluded that the massive core leads to better results for low discretizations.

Based on the previous results, a fit was made for a massive core with n1=n2=4 (see Figure 11) It can be seen that the fit is satisfying with lower (= more realistic) values for the density of the concrete. This result shows that also the discretization and the floor model have an impact on the fitting values.

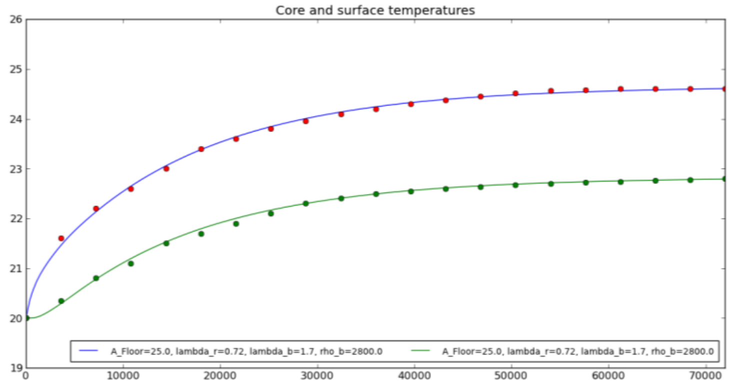

Figure 11 – Best fit for massive core with n1=n2=4

3.2.2 Dynamic models

We compare the result of the dynamic models with the 'verified' static model from Figure 11. As shown in Figure 12, all models show exactly the same heat transfer. This is an important conclusion, meaning that Rz can be replaced by a computation of TMean between TIn and TOut of the embedded pipe.

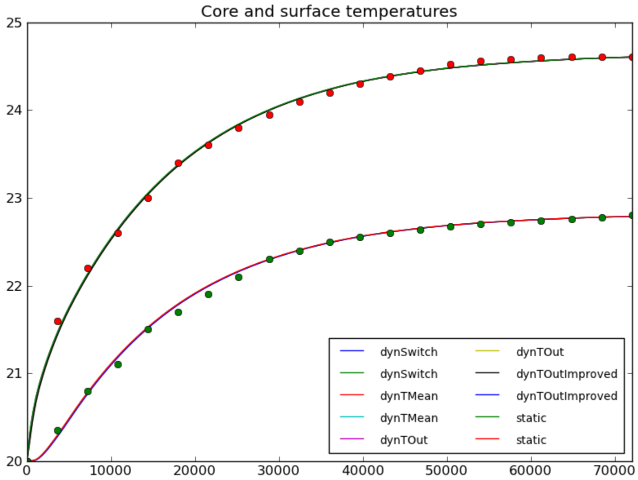

Figure 12 – Comparison of the dynamic and static models for response to a step in TIn

3.3 PULSE TO TIN

Next, the case from (Koschenz and Lehmann 2000), par 4.5.1 “Variante 2” is simulated.

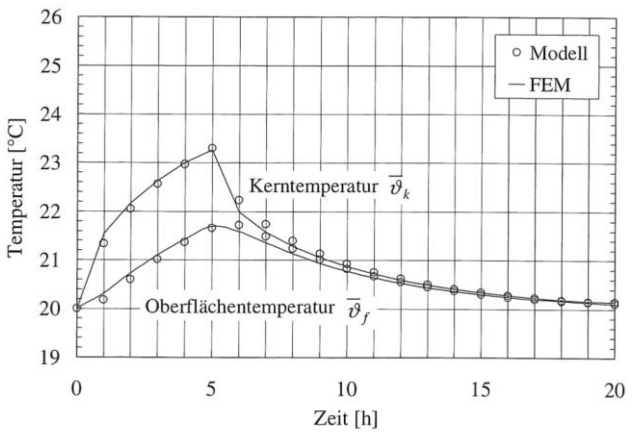

Figure 13 – Expected result for pulse to TIn

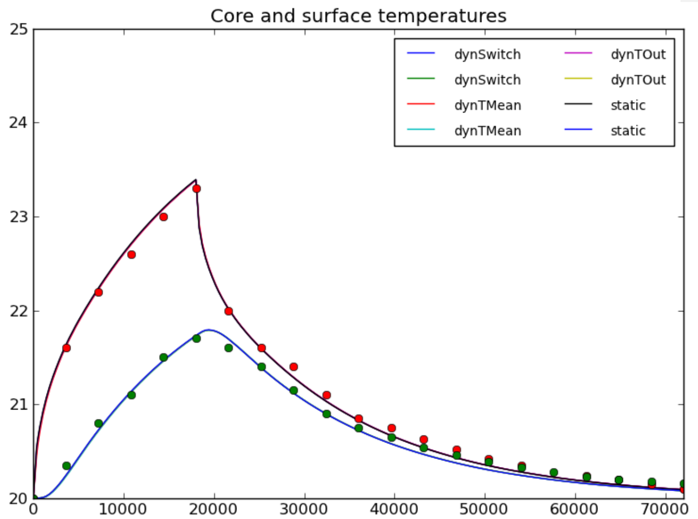

Figure 14 – Achieved result for pulst to TIn, models are as in Figure 7

It can be seen that the models more or less show the same behaviour as the validation case. However, there is more deviation than for the step simulations, probably due to the uncertainty on certain model parameters.

3.4 ENERGY BALANCE

The same boundary conditions as variant 2 of the previous section are taken. Now the energy balance, being the difference between the thermal energy going in and out of the embedded pipe is checked.

The energy balance is made by comparing the heat emitted by the upper and lower surface of the tabs (QTabsOut) to the heat entering the tabs through the fluid flow (QTabsIn). The cumulative values of these variables are shown in Figure 15 for the different models. The resulting energy balance can be found in Table 3.

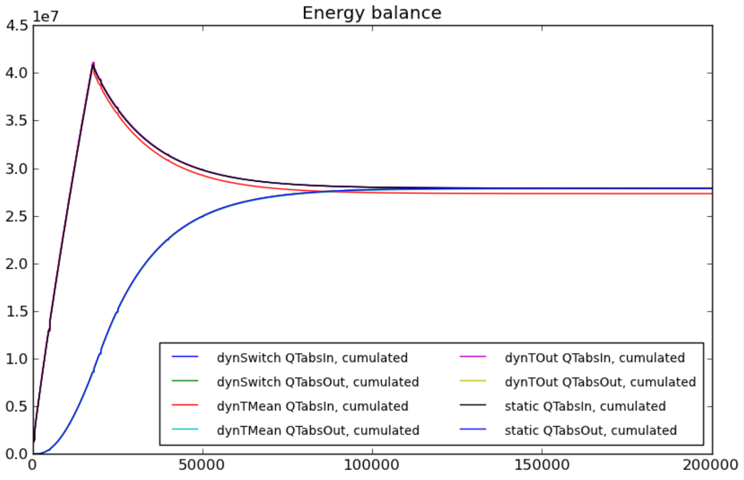

Figure 15 – Cumulated energy flows in and out of the tabs

From these results it can be seen that the model dynTMean has a wrong energy balance, the other models perform well.

Table 3 – Energy balance

Model	Energy balance (TIn-TOut)/TIn
dynSwitch	-7.33265e-05
dynTMean	-0.0188974
dynTOut	-7.33265e-05
static	-1.12527e-05

3.5 T_{OUT}

An important check is the output temperature of the tabs. Even if the emitted heat from the tabs surfaces is the same, there can be differences in the output temperature, and these differences can have an important impact on the HVAC system, more specifically on heat pump performance, stratification in a storage tank, controls.

Figure 16 and Figure 17 show the response of TOut after a positive respectively a negatiive step in TIn. First, it can be observed that the model dynTMean shows totally unacceptable and physically wrong behaviour. Looking at the model structure, this behaviour is mathematically correct. Therefore, the model dynTMean will be excluded from further analyses.

The static model is not correct either. It is not possible that a step to TIn immediately causes a step to TOut. In the previous sections it was shown that the dynTOut and dynSwitch models behave exactly the same as the static model, therefore the rest of the analysis will only be done with these two models.

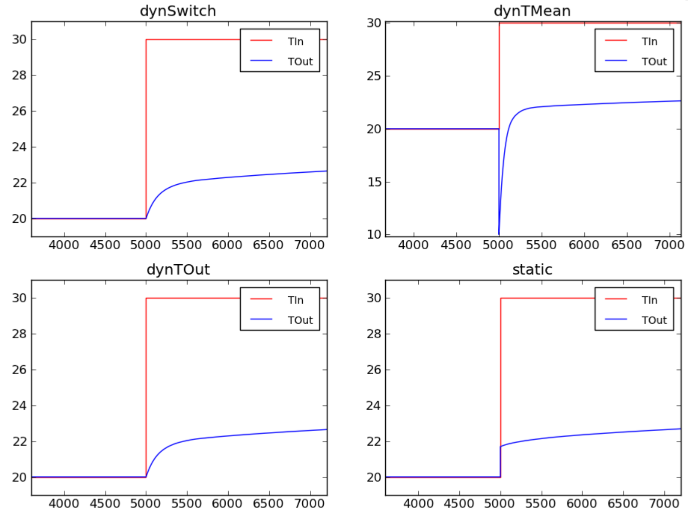

Figure 16 – Response of TOut for an upward step in TIn

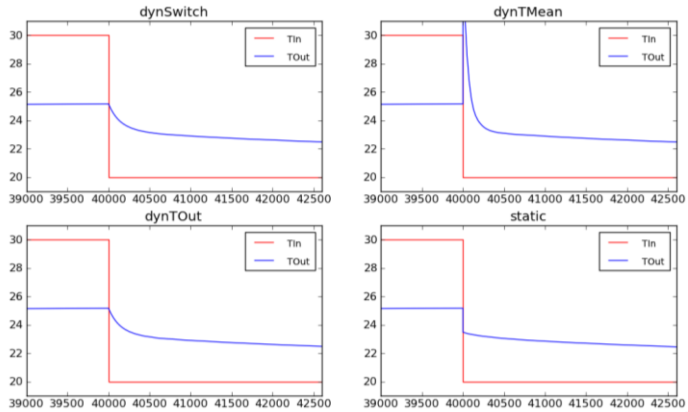

Figure 17 – Response of TOut for a downward step in TIn

There is however a physical effect that is not represented by the two remaining models: the time lag. As it takes a finite amount of time TLag for the water to run through the embedded pipe, it is not possible that TOut changes within TLag of a change of temperature of TIn. Only around T0 + TLag we expect to see a change in TOut. Without resorting to algebraic means and discrete modelling, we expect to solve this issue by a discretization of the length of the embedded pipe. This is developed in the next section.

4 DISCRETIZATION IN THE FLOW DIRECTION

To discretize in the flow direction, the tabs model is split up in identical smaller tabs models, for which only the floor surface is changed. Initially, all surface temperatures are considered identical. . The results for a pulse to TIn are shown in Figure 18 and Figure 19. The results for the dynSwitch are identical; they are omitted from the graph.

Comment [RDC1]: I thought this could be wrong, so I made a second model in which each discrete tabs element has a different surface temperature. I should still check the difference between both models in detail, to be sure in the first model there is no heat flowing from the first to eg. the last surface element

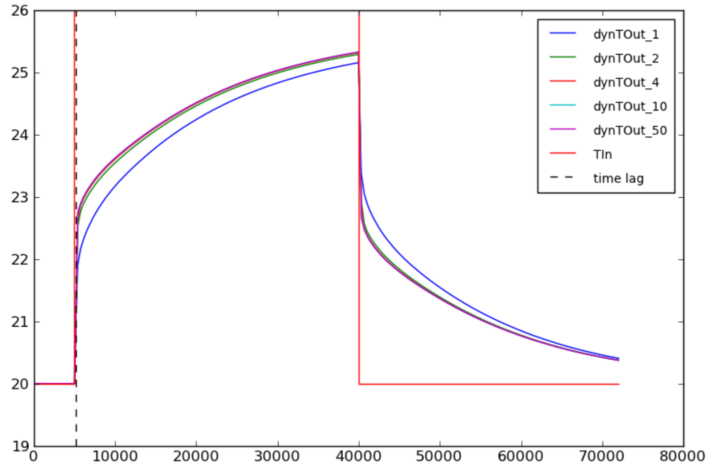

Figure 18 – Response in function of discretization (1, 2, 4, 10, 50 steps)

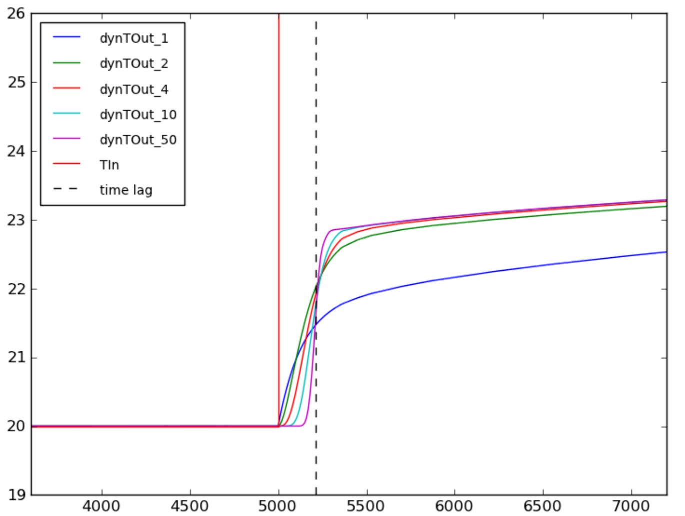

Figure 19 – Response in function of discretization (1, 2, 4, 10, 50 steps), detail at the step

The figures show that the discretization has a big impact on the response immediately after a sudden change in the input temperature. The expected behaviour, being the rise of the outlet temperature only after a time lag equal to the flow through time, is only observed for large discretization numbers. Even with 50 discrete elements, the output temperature starts rising before the time lag has passed.

It is remarkable to observe a much lower output temperature for the case without discretization compared to all discretized cases. The explanation must be searched in the non-linear temperature profile over the length of the pipe. As shown before, the embedded pipe model supposes a linear temperature profile, whereas in reality, a non-linear profile is present, as shown in Figure 20. Therefore, TMean is overestimated, leading to a higher heat exchange between the pipe and the concrete, and thus a lower output temperature. By discretizing, the overestimation of TMean is reduced, and therefore the output temperature is higher.

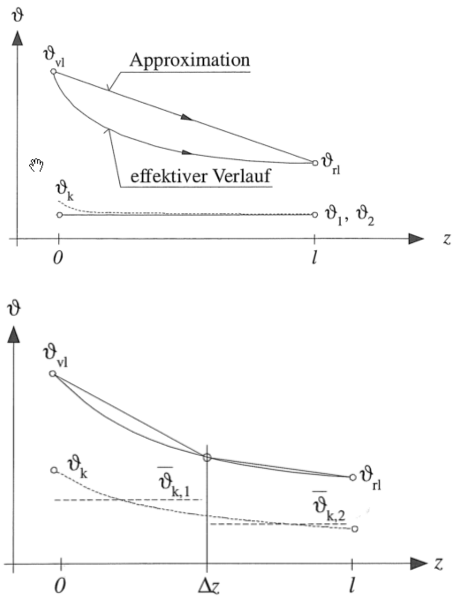

Figure 20 – Real and supposed temperature profile in the embedded pipe. Above: no discretization, below, discretization with n=2

To test this hypothesis, two more simulations are made, with different flow rates. Higher flow rates should lead to more linear temperature profiles, and therefore a smaller difference between the cases without and with discretization. Figure 21 shows the mean water temperatures (TMean) in the embedded pipes for different cases.

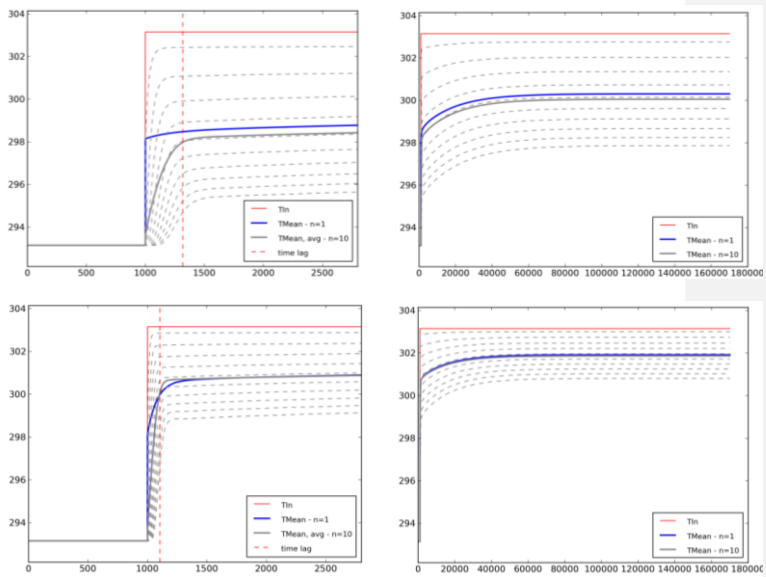

Figure 21 – Impact of flow rate on difference between discretization and no discretization. The upper row has a flowrate of 8 kg/hm², the lower row has 24 kg/hm².

The results confirm that the discretization has much less impact with higher flow rates.

Please note that the linearity of the water temperature profile is depending on the ratio between the flow rate and the total thermal conductivity between the embedded pipe and the tabs core. Therefore, in systems where the flowrate is low and the heat conductivity between the embedded pipe and the tabs core is high, discretization is required. If the flowrate is high compared to this conductivity, the temperature profile will be close to linear and discretization is not required (if speed matters).

The physical boundary for this relation is given by equation 4.72 from (Koschenz and Lehmann 2000) as shown below (*n* = number of discrete elements):

$$\dot{m}_{sp} * n * c_p * (R_w + R_r + R_x) > 0.5 \quad (18)$$

For the cases shown in Figure 21, the left side of equation (18) evaluates to 1.48 in case of 24 kg/hm² and to 0.57 in case of 8 kg/hm². From the results it can be seen that discretization is already desirable before reaching the physical boundary.

5 NO FLOW CONDITIONS

Of course, the flow rate in the model should be able to go to zero, eg. in case the circulation pump is not running. The behaviour of the model in these conditions, and at switching from flow to no flow conditions, is checked.

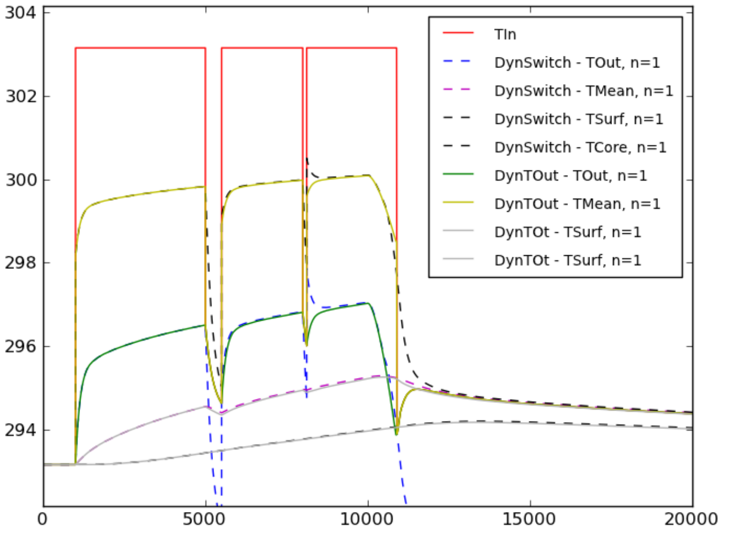

Figure 22 – Overview of a simulation with on/off mass flow conditions

The results of these simulations are shown in Figure 22 and Figure 23. From the detail, it can be seen that TMean is discontinuous for the dynTOut model, but continuous for the dynSwitch model. The fact that TOut is discontinuous in dynSwitch models is not necessariliy an issue because at moments of zero flow rate, this temperature has no function in the model, it could take any value. However, when the flow is restored, the dynSwitch model reinitialises TOut to TMean. It can be understood that this is a dangerous practice if the flow rate would only be interrupted for a brief period. TOut would then be initiated to values higher than it's value when the flow rate disappeared. TMean is the average between TIn and TOut during flow rate conditions, so also TMean can jump to higher values.

The dynTOut model sets TMean to TOut when the flow rate disappears. It is evident that at these moments, there is less heat transfer into the tabs. This can be seen in the results: the core temperature and also the surface temperature are lower for this model. The difference is however small, and will be small in realistic HVAC simulations.

It's also important to notice that discretization will reduce the differences between the models, and reduce the energy imbalance for all models. An overview of the energy balances for the simulated case (simulated until 300 000 s) is given in The results show that the TOut model has a much better energy balance, even without discretization at all. As a matter of fact, the energy balance of the switching models is unacceptable, because short on/off cycling of pumps is perfectly possible in HVAC simulations, and it is too risky to use a model that can introduce energy balance errors.

Table 4.

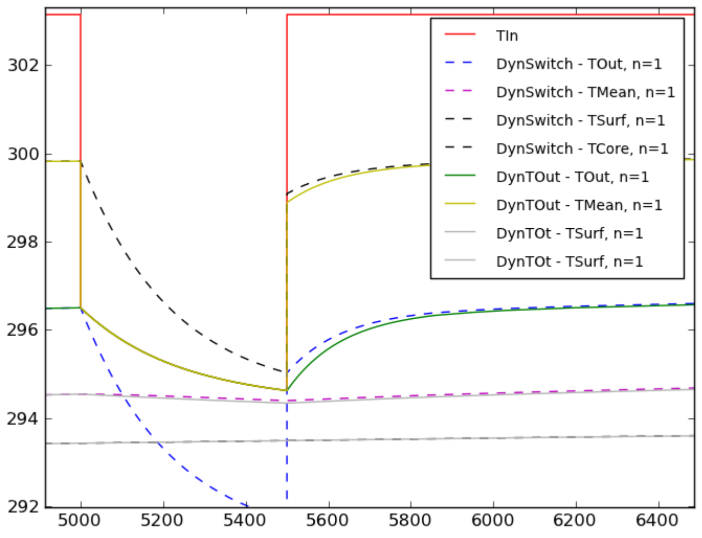

Figure 23 – Overview of a simulation with on/off mass flow conditions

The results show that the TOut model has a much better energy balance, even without discretization at all. As a matter of fact, the energy balance of the switching models is unacceptable, because short on/off cycling of pumps is perfectly possible in HVAC simulations, and it is too risky to use a model that can introduce energy balance errors.

Table 4 – Energy balance: (In - Out) / In for different models and discretizations

Model, discretization	Energy balance
Switch, n=1	-0.04012
Switch, n=2	-0.02196
Switch, n=4	-0.01091
Switch, n=10	-0.00433
TOut, n=1	0.00231
TOut, n=2	0.00006
TOut, n=4	0.00003

TOut, n=10	0.00002

6 CONCLUSION

A detailed verification of the embedded pipe / tabs model is made, leading to the following conclusions:

- The main equations determining Rz, Rw, Rr and Rx are correct
- However, it was NOT possible to fit the model to the case depicted in 4.5.1 and 4.5.2 from (Koschenz and Lehmann 2000) with standard values for some of the (unknown) parameters like pipe material conductivity, and concrete density.
- Discretization of the tabs capacity (y-direction) has a clear impact on the core temperature immediately after a sudden change in water temperature, but it does not seem necessary to go beyond 3-4 elements on each side of the core layer.
- Attributing a part of the tabs capacity to the core layer seems to give slightly better results, specifically for rough discretizations of the slab's capacity.
- The dynamic models, replacing the static calculation of Rz by simulation of the water temperature in the duct in detail, give exactly the same results (for cases where the flow rate does not drop to zero)
- Lumping the water mass to it's average temperature gives bad results for the energy balance and completely wrong results for TOut (in case of sudden changes on TIn).
- Also the static model gives very bad results on TOut because it does not reflect the inertia of the water nor the time lag caused by the circulation.
- Discretizing the embedded pipe (and tabs around it) in the flow direction revealed that the assumption of a linear water profile between in- and outlet in the pipe is not always satisfied. Therefore, it is important to discretize at least in 2 elements, where each element has it's own independent core and surface temperatures. In cases where the specific flow rate is low and the heat conductivity between the embedded pipe and the tabs core is high, discretization in more elements is required.
- Discretization in the flow direction is however required if we want to take into account the time lag between changes in the input and their reflection on the output. High discretization numbers (>50) are required to simulate this.
- Under on/off flow rate conditions, from the two remaining models, the one that permanently lumps the water capacity to the outlet temperature shows the most robust behaviour and the best energy balance.

In general, it can be concluded that the dynTOut model, that lumps the water capacity to the outlet temperature gives the best results, given at least two such elements in series and about 3-4 discrete elements in the concrete layers above and below the core. If the time lag has to be modelled in detail, much higher discretizations in the flow direction are required.

REFERENCES

Koschenz, Markus, and Beat Lehmann. 2000. *Thermoaktive Bauteilsysteme - tabs*. Dübendorf: EMPA Dübendorf.

Transsolar. 2007. Trnsys 16 - Multizone Building modeling with Type56 and TRNBuild. In *TRNSYS 16 Manual*. Vol. 6.

prEN 15377-1. 2005. *Heating systems in buildings - Design of embedded water based surface heating and cooling systems - Part 1 : Determination of the design heating and cooling capacity*.

7 REMAINING QUESTIONS

? Are there any more questions?

Of course, we can investigate everything. But for the record, these are things that were not checked in detail:

- Is it really important to keep the surface temperatures separated together with the discretization of the embedded pipe in flow direction?
- Would it be possible to discretize only the embedded pipe, and NOT the concrete?
- what about plug models for the pipe? Does it exist (these models specifically take into account the time lag
- We did not check other tab compositions (thicker slabs, different interdistances between the pipes, different pipe diameters, ...)
- we still have to implement the detailed models for floor heating etc. according to Table 1.
-