

IMPROVED CONDUCTION TRANSFER FUNCTION COEFFICIENTS GENERATION IN TRNSYS MULTIZONE BUILDING MODEL

Benoit Delcroix¹, Michaël Kummert¹, Ahmed Daoud², and Marion Hiller³

¹École Polytechnique de Montréal, Dept of Mechanical Engineering, Montréal, QC, Canada

²Laboratoire des Technologies de l'Énergie, Hydro-Québec Research Institute, Shawinigan, QC, Canada

³TRANSSOLAR Energietechnik GmbH, Stuttgart, Germany

ABSTRACT

Many building energy performance simulation programs (including TRNSYS) use the Conduction Transfer Function (CTF) method to compute 1-D transient heat conduction through multi-layer slabs. Problems have been reported with the current CTF implementation in the TRNSYS multizone building model, especially during the CTF coefficients generation. These problems are related to heavy and highly insulated slabs and short time-step simulations (less than 15-minute time-step). This paper describes the implementation of a new CTF coefficients generation method in the TRNSYS building preprocessor (TRNBuild). The efficiency and the limitations of this method are also discussed.

INTRODUCTION

Transient conduction heat transfer through building slabs is a key aspect of cooling and heating loads calculation. These calculations can be performed by several Building Performance Simulation (BPS) programs such as EnergyPlus (EnergyPlus Documentation, 2012), ESP-r (Energy Systems Research Unit, 1998) and TRNSYS (TRANSSOLAR Energietechnik GmbH, 2011). The current interest for net-zero energy buildings and for better power demand management strategies requires accurate transient simulation of heavy and highly insulated slabs with short time-steps (lower than 15 minutes). It then represents a challenge for codes that were mainly developed for yearly energy load calculations with a time-step of 1 hour. It is the case of the TRNSYS building model (called Type 56) which is known to have limitations with heavy and highly insulated slabs and with short time-steps. These limitations come from the method used by TRNSYS for modeling conduction heat transfer through slabs which is known as the Conduction Transfer Function (CTF) method. In particular, problems have been identified in the generation of CTF coefficients, and a solution has been proposed (Delcroix, Kummert, Daoud, & Hiller, 2012). This paper reports on the implementation of a new method of CTF coefficients generation in the TRNSYS building preprocessor, known as TRNBuild. The method is explained, and a simple example is provided. Results obtained using the improved version of TRNBuild for different wall

types that cause problems are presented and discussed, including a full house model for a net-zero energy renovation project.

STATE OF THE ART

The CTF method has been implemented in many BPS programs (including TRNSYS and EnergyPlus) to model 1-D transient conduction heat transfer through building slabs. It was introduced by Mitalas and Stephenson (1971) and consists in time series which allow to compute the inside and outside surface heat flows (\dot{q}_{si} and \dot{q}_{so}) from current and past values of surface temperatures (T_{si} and T_{so}) and past values of heat flows themselves:

$$\dot{q}_{si,t} = \sum_{k=0}^{n_b} b_{t-k\Delta t_b} T_{so,t-k\Delta t_b} - \sum_{k=0}^{n_c} c_{t-k\Delta t_b} T_{si,t-k\Delta t_b} - \sum_{k=1}^{n_d} d_{t-k\Delta t_b} \dot{q}_{si,t-k\Delta t_b} \quad (1)$$

$$\dot{q}_{so,t} = \sum_{k=0}^{n_a} a_{t-k\Delta t_b} T_{so,t-k\Delta t_b} - \sum_{k=0}^{n_b} b_{t-k\Delta t_b} T_{si,t-k\Delta t_b} - \sum_{k=1}^{n_d} d_{t-k\Delta t_b} \dot{q}_{so,t-k\Delta t_b} \quad (2)$$

Where:

$$\frac{\sum_{k=0}^{n_a} a_{t-k\Delta t_b}}{\sum_{k=0}^{n_d} d_{t-k\Delta t_b}} = \frac{\sum_{k=0}^{n_b} b_{t-k\Delta t_b}}{\sum_{k=0}^{n_d} d_{t-k\Delta t_b}} = \frac{\sum_{k=0}^{n_c} c_{t-k\Delta t_b}}{\sum_{k=0}^{n_d} d_{t-k\Delta t_b}} = U \quad (3)$$

The coefficients a , b , c and d are known as the CTF coefficients. These coefficients allow characterizing the dynamic behavior of a slab. They are generated only once before the simulation for a certain timebase value Δt_b which is the CTF time-step. The timebase must be distinguished from the simulation time-step. TRNSYS simulations can run with a time-step that is shorter than the timebase, as long as the latter is an integer multiple of the former. The ideal case is to have equivalent values, but this cannot always be achieved for heavy and highly insulated

slabs, which require longer timebase values (sometimes several hours) to be simulated in TRNSYS. A difference between timebase and time-step creates a stair-step effect which becomes more pronounced as the difference increases.

Several methods exist for generating the CTF coefficients. Spitler and al. made a comparison between the different available methods (Li, Chen, Spitler, & Fisher, 2009). The two methods most often used in practice are the Direct-Root Finding (DRF) and the State-Space (SS) methods. The first one was developed by Mitalas and Arseneault (1972) and is used in TRNSYS. The SS method is for example described by Seem (1987) and is currently used in EnergyPlus. Several papers (including (Li, Chen, Spitler, & Fisher, 2009)) demonstrated that the SS method is more efficient because it allows generating the CTF coefficients for a lower timebase value.

With the stair-step effect, there is a second drawback with the CTF method which is the difficulty to take into account time-variant properties, since the coefficients are generated only once before the simulation (pre-processing). This is generally not an issue with conventional walls but it is an obstacle to modeling phase change materials (PCM) embedded in walls and slabs.

MATHEMATICAL DESCRIPTION

The method to generate CTF coefficients presented in this section is adapted from Seem (1987). The principle is to obtain a state-space (SS) representation of a slab and to convert that model into a transfer function representation.

Generating the CTF coefficients requires 5 steps:

- Selection of the number of nodes and their positioning.
- Construction of the SS model.
- Discretization of the SS model.
- Calculation of the CTF coefficients.
- Check of the generated CTF coefficients.

Selection of the number of nodes and their positioning

The first step consists in spatially discretizing each layer of the slab by nodes. The number of nodes is chosen according to layers characteristics, and more especially the Fourier number Fo which is computed as shown in Equation (4). This method is adapted from the one used in EnergyPlus (EnergyPlus Documentation, 2012).

$$Fo = \frac{\alpha \Delta t_b}{L^2} \quad (4)$$

Where the thermal diffusivity α is defined as:

$$\alpha = \frac{k}{\rho C_p} \quad (5)$$

The principle is now to define an optimal spacing between the nodes, named Δx (Equation (6)). A high Fourier number will give more accuracy (more nodes) but a higher computational time. A value of 200 was selected after initial testing.

$$\Delta x = \sqrt{\frac{\alpha \Delta t_b}{Fo}} \quad (\text{where } Fo = 200) \quad (6)$$

With the value of Δx , it is possible to define the number of nodes by dividing the layer thickness by Δx . Then the largest following integer is chosen.

A maximum number of nodes for a slab was also defined to avoid excessively long calculation times. The limit was fixed at 400 (calculation time per slab of approximately 4 seconds). In the current implementation, if the number of nodes reaches that value, an error message prompts the user to adapt (increase) the timebase.

Nodes are distributed equally, with half-nodes located at each interface between layers. The surface nodes do not take into account inside and outside convection coefficients, which are handled separately.

Construction of the SS model

An SS model of a linear time-invariant (LTI) system with n nodes, 2 inputs (T_{si} and T_{so}) and 2 outputs (\dot{q}_{si} and \dot{q}_{so}) can be expressed as following (Myers, 1971):

$$d \begin{bmatrix} T_1 \\ \dots \\ T_n \end{bmatrix} \frac{1}{dt} = [A] \begin{bmatrix} T_1 \\ \dots \\ T_n \end{bmatrix} + [B] \begin{bmatrix} T_{si} \\ T_{so} \end{bmatrix} \quad (7)$$

$$\begin{bmatrix} \dot{q}_{si} \\ \dot{q}_{so} \end{bmatrix} = [C] \begin{bmatrix} T_1 \\ \dots \\ T_n \end{bmatrix} + [D] \begin{bmatrix} T_{si} \\ T_{so} \end{bmatrix} \quad (8)$$

The variables $T_1 \dots T_n$ are the temperatures at each node and are known as the state variables. A , B , C and D are constant coefficients matrices with a size of respectively (n,n) , $(n,2)$, $(2,n)$ and $(2,2)$. These matrices characterize the system and can be determined if the nodes properties are known.

Equations (7) and (8) can be rewritten as Equations (9) and (10), where U_l and U_r are respectively the heat transfer coefficients to the left and the right of the node i .

$$C_i \frac{dT_i}{dt} = U_l(T_{i-1} - T_i) + U_r(T_{i+1} - T_i) \quad (9)$$

$$\dot{q}_s = U(T_s - T_i) \quad (10)$$

Discretization of the SS model

The third step is the discretization of the SS model. It means that we have to discretize the matrices A , B , C and D with relation to time (the discretization time-step is the timebase). The discretization method used is the First-Order Hold method which assumes a

linear interpolation between the discretized data, to be consistent with TRNSYS conventions to pass average values over the time-steps between components.

First, a new matrix M is built, including matrices A and B . It also includes an identity matrix I with the size (2,2) which is divided by the timebase. The matrix M can be presented as follows:

$$M = \begin{bmatrix} A(nx, nx) & B(nx, nu) & \text{zeros}(nx, nu) \\ \text{zeros}(nu, nx) & \text{zeros}(nu, nu) & \frac{I(nu)}{\Delta t_b} \\ \text{zeros}(nu, nx) & \text{zeros}(nu, nu) & \text{zeros}(nu, nu) \end{bmatrix} \quad (11)$$

We can now compute the matrix exponential of M , which is named Φ :

$$\Phi = e^{M\Delta t_b} \quad (12)$$

Numerous methods exist to compute a matrix exponential (see e.g. Moler and Van Loan (2003) for a comparative review of 19 methods). The Padé approximation method implemented in “DGPADM” routine (Sidje, 1998) was selected.

Φ allows defining 2 intermediate matrices ($F1$ and $F2$) and the discretized matrix A_d :

$$F1 = \Phi(1: nx, nx + 1: nx + nu) \quad (13)$$

$$F2 = \Phi(1: nx, nx + nu + 1: nx + 2 nu) \quad (14)$$

$$A_d = \Phi(1: nx, 1: nx) \quad (15)$$

$F1$, $F2$ and A_d then give the discretized matrices B_d , C_d and D_d :

$$B_d = F1 + A_d * F2 - F2 \quad (16)$$

$$C_d = C \quad (17)$$

$$D_d = D + C_d * F2 \quad (18)$$

Calculation of the CTF coefficients

CTF coefficients are computed from the discretized matrices. This step consists in the conversion of the SS model in a transfer function representation. The computation process is for example documented in the Matlab documentation (The MathWorks Inc., 2010).

First, the d coefficients are calculated using the eigenvalues of the matrix A_d . In FORTRAN, the eigenvalues of a matrix are computed with a routine named “DGEEVX” which is included in the LAPACK package (2011).

The 3 other coefficients (a , b and c) are then computed from the d coefficients and several sets of eigenvalues calculated for different matrices involving the matrices A_d , B_d , C_d and D_d .

The d coefficients are dimensionless but a , b and c express the ratio between heat transfer rate and a temperature so care must be taken to comply with the non-standard TRNSYS units (kJ/h-K).

Check of the generated CTF coefficients

The validity of the computed CTF coefficients is checked using Equation (3): the calculated U-values must be within 0.001% of the “actual” value (i.e. computed from the layers description).

EXAMPLE

This section illustrates the CTF coefficients generation by a simple example of a slab with 3 nodes. The example (Figure 1) is a concrete slab with a thickness L of 0.3 m, a density ρ of 2200 kg/m³, a specific heat C_p of 0.84 kJ/kg-K and a thermal conductivity k of 1.7 W/m-K. The inside and outside film coefficients h_i and h_o have a value of respectively 8.3 and 34.5 W/m²-K (in TRNSYS, these coefficients would be handled in a different part of the program and would not be included in the CTF coefficients).

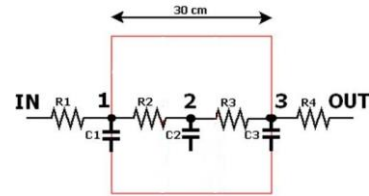


Figure 1 – Scheme of a three-node example

Once the nodes are defined, the A , B , C and D matrices of the SS model can be calculated. The thermal capacity and resistance (or U-value) for each node are presented in Equations (19) to (24):

$$C_w = \rho \times C_p \times L = 2200 \times 840 \times 0.3 = 554400 \frac{J}{m^2 K} \quad (19)$$

$$C1 = C3 = \frac{554400}{4} = 138600 \frac{J}{m^2 K} \quad (20)$$

$$C2 = \frac{554400}{2} = 277200 \frac{J}{m^2 K} \quad (21)$$

$$U1 = \frac{1}{R1} = h_i = 8.3 \frac{W}{m^2 K} \quad (22)$$

$$U2 = U3 = \frac{1}{R2} = \frac{k}{L} = \frac{1.7}{0.15} = 11.33 \frac{W}{m^2 K} \quad (23)$$

$$U4 = \frac{1}{R4} = h_o = 34.5 \frac{W}{m^2 K} \quad (24)$$

Differential equations similar to Equation (9) can then be written for each node:

$$\begin{aligned} \frac{dT_1}{dt} &= \frac{U1}{C1} \times (T_i - T_1) + \frac{U2}{C1} \times (T_2 - T_1) \\ &= \frac{8.3}{138600} \times (T_i - T_1) \\ &\quad + \frac{11.33}{138600} \times (T_2 - T_1) \end{aligned} \quad (25)$$

$$\begin{aligned} \frac{dT_2}{dt} &= \frac{U2}{C2} \times (T_1 - T_2) + \frac{U3}{C2} \times (T_3 - T_2) \\ &= \frac{11.33}{277200} \times (T_1 - T_2) \\ &\quad + \frac{11.33}{277200} \times (T_3 - T_2) \end{aligned} \quad (26)$$

$$\begin{aligned} \frac{dT_3}{dt} &= \frac{U3}{C3} \times (T_2 - T_3) + \frac{U4}{C3} \times (T_o - T_3) \\ &= \frac{11.33}{138600} \times (T_2 - T_3) \\ &\quad + \frac{34.5}{138600} \times (T_o - T_3) \end{aligned} \quad (27)$$

Equations (25) to (27) can be written in a matrix form, where the 3-by-3 matrix is *A* and the 3-by-2 matrix is *B* in the SS model:

$$\begin{aligned} &\frac{d}{dt} \begin{bmatrix} T_1 \\ T_2 \\ T_3 \end{bmatrix} \\ &= \begin{bmatrix} -\frac{19.63}{138600} & \frac{11.33}{138600} & 0 \\ \frac{11.33}{277200} & -\frac{11.33}{277200} & \frac{11.33}{277200} \\ 0 & \frac{11.33}{138600} & -\frac{45.83}{138600} \end{bmatrix} \begin{bmatrix} T_1 \\ T_2 \\ T_3 \end{bmatrix} \\ &+ \begin{bmatrix} \frac{8.3}{138600} & 0 \\ 0 & 0 \\ 0 & \frac{34.5}{138600} \end{bmatrix} \begin{bmatrix} T_i \\ T_o \end{bmatrix} \end{aligned} \quad (28)$$

Heat flows are computed from the first and last nodes and boundary conditions:

$$\dot{q}_{si} = h_i(T_i - T_1) = 8.3 (T_i - T_1) \quad (29)$$

$$\dot{q}_{so} = h_o(T_o - T_3) = 34.5 (T_o - T_3) \quad (30)$$

Or, in a matrix form:

$$\begin{aligned} \begin{bmatrix} \dot{q}_{si} \\ \dot{q}_{so} \end{bmatrix} &= \begin{bmatrix} -8.3 & 0 & 0 \\ 0 & 0 & -34.5 \end{bmatrix} \begin{bmatrix} T_1 \\ T_2 \\ T_3 \end{bmatrix} \\ &\quad + \begin{bmatrix} 8.3 & 0 \\ 0 & 34.5 \end{bmatrix} \begin{bmatrix} T_i \\ T_o \end{bmatrix} \end{aligned} \quad (31)$$

The 2-by-3 and 2-by-2 matrices in Equation (31) are respectively *C* and *D* in the SS model.

The four matrices are discretized for a time-step of 1 hour (3600 seconds):

$$A_d = \begin{bmatrix} 0.6147 & 0.1999 & 0.01142 \\ 0.09994 & 0.7725 & 0.07353 \\ 0.01142 & 0.1471 & 0.3133 \end{bmatrix} \quad (32)$$

$$B_d = \begin{bmatrix} 0.1355 & 0.01029 \\ 0.02049 & 0.06137 \\ 0.002475 & 0.3142 \end{bmatrix} \quad (33)$$

$$C_d = \begin{bmatrix} -8.3 & 0 & 0 \\ 0 & 0 & -34.5 \end{bmatrix} \quad (34)$$

$$D_a = \begin{bmatrix} 7.537 & -0.009171 \\ -0.009171 & 23.68 \end{bmatrix} \quad (35)$$

The CTF coefficients are calculated from the discretized matrices as explained previously. Table 1 shows the results:

Table 1 – Values of the CTF coefficients

	a	b	c	d
1	85.2408	0.033	27.1343	1.0000
2	-183.9657	0.2512	-50.1883	-1.7004
3	127.8714	0.169	28.1085	0.8785
4	-28.6833	0.01	-4.5913	-0.1361
Sum	0.4632	0.4632	0.4632	0.042
U-value [kJ/h-m ² -K]	11.0286	11.0286	11.0286	/

The actual U-value (including the film coefficients) of the concrete slab is 11.045 kJ/h-m²-K while the U-value calculated with the coefficients is 11.029 kJ/h-m²-K. The observed error is 0.15 % which is greater than the limit of 0.001 %. So the coefficients should be considered as non-reliable. This result is normal because the number of nodes (3) is not sufficient for that type of heavy wall. Equation (6) would require 74 nodes for this concrete slab.

IMPLEMENTATION IN TRNSYS

The developed SS method has been implemented in a development version of TRNSYS 17.

CTF coefficients used in the multizone building model (Type 56) are generated by routines called by TRNBuild, the building preprocessor. They are implemented in a DLL called GenTRN, which is coded in FORTRAN. Several subroutines are included in this program. Figure 2 below presents a general scheme of the GenTRN new structure when the SS method is implemented as an alternative to the current method.

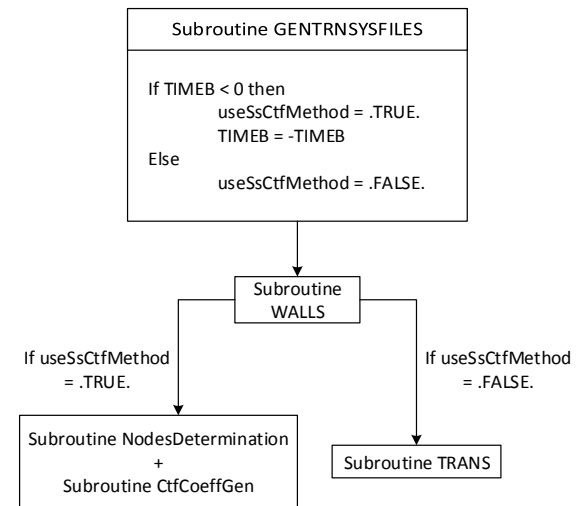


Figure 2 – Scheme of the implementation of the SS method in TRNSYS

The subroutine “GENTRNSYSFILES” is the main entry point into the DLL, which is called by TRNBuild and calls all the other subroutines in the DLL. The main routine receives all the building description data from TRNBuild (walls, layers,...). In the current test version, the user selects the new CTF

coefficients generation method by specifying a negative timebase value. This choice is caught by the main routine and passed to the WALLS subroutine which calls the appropriate CTF coefficients generation routine according to the user choice.

WALL TESTS IN TRNSYS

Extensive tests were performed to debug and quality-check the new CTF coefficients generation method. This paper reports on 2 wall types: an insulated concrete form (ICF) wall and a plain wooden wall. The characteristics of each slab are presented in the following table (from inside to outside).

Table 2 – Description of the presented walls

Slabs	L [m]	k [W/(m·K)]	ρ [kg/m ³]	C _p [kJ/(kg·K)]	R [m ² ·K/W]
Wall-only U-value: 0.188 W/(m ² ·K)					
G01 - Gypsum board	0.016	0.16	800	1.09	0.100
I03 - EPS (expanded polystyrene) board	0.076	0.03	43	1.21	2.533
M15 - Heavyweight concrete	0.203	1.95	2240	0.9	0.104
I03 - EPS board	0.076	0.03	43	1.21	2.533
F07 - Stucco	0.025	0.72	1856	0.84	0.035
Wall-only U-value: 0.260 W/(m ² ·K)					
Wood wall	0.500	0.130	600	1.88	3.846

The plain 50-cm wooden wall is an extreme case that illustrated the limits of the CTF method (as currently implemented and with the new coefficients generation method).

The scenario that we have tested is the one presented in the Figure 3. It represents a typical set-back scenario.

The outside temperature is kept constant at 0°C. No radiative heat flows are considered. The heating system power is initially set to maintain a steady-state indoor temperature of 20°C. The heating system is then stopped for 4 hours, and restarted with a heating power equal to 1.5 times the initial value.

The simulation time-step is one minute and the inside and outside convection coefficients have a value of 8.3 and 34.5 W/m²·K, respectively.

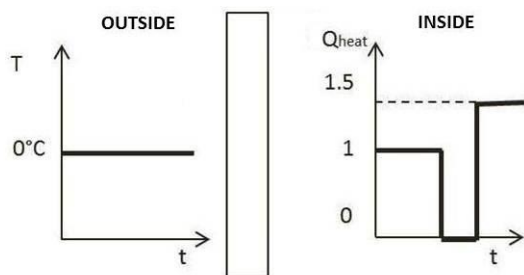


Figure 3 – Description of the presented scenario

In the following 4 graphs, 3 kind of results are presented: firstly, in dotted lines, the results coming from the current version of TRNSYS; secondly, in dashed lines, the results performed by using the SS method for generating the CTF coefficients; thirdly, in solid lines, a reference solution implemented in Matlab (The MathWorks Inc., 2010) by simulating a discretized wall model with a large number of nodes. All graphs present the evolution of the inside surface temperature according to time.

The first 2 graphs (Figures 4 and 5) concern the ICF wall (Figure 5 is a zoom into Figure 4 to highlight the differences between the curves). The results show that the original and modified versions of TRNSYS match the reference solution at the end of each timebase but the presence of a stair-step effect is influenced by that timebase. The minimum timebase value that can be reached with the new method is 12 minutes, while it was 1 hour with the DRF method.

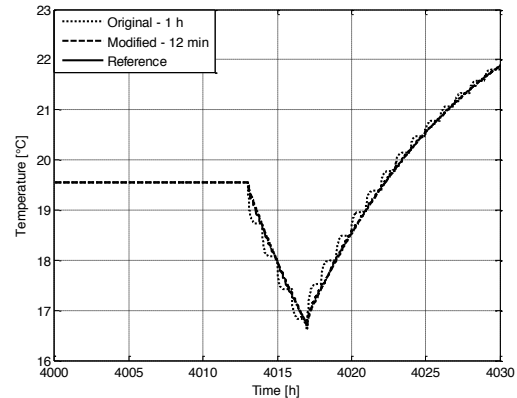


Figure 4 – Evolution of the inside surface temperature for an ICF wall

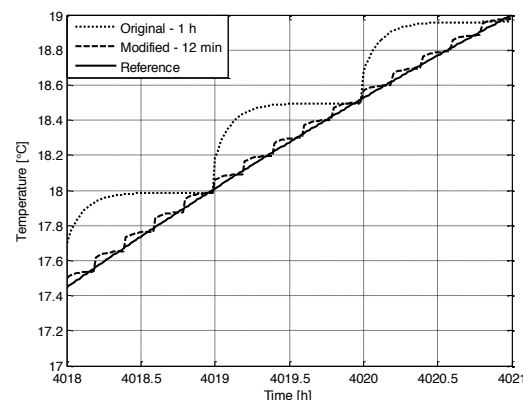


Figure 5 – Evolution of the inside surface temperature for an ICF wall (zoom)

The next two graphs are about the plain wooden wall with a thickness of 0.5 m, the second one being a zoom where the minimum temperature is reached. This wall is not typical from walls encountered in real buildings but it allows highlighting problems with the CTF method. These problems appear clearly in the graphs. The results generated by the current and modified versions of TRNSYS are affected by a stair-step effect. That effect is more pronounced for the current TRNSYS version (minimum timebase value = 3 hours, vs. 1 hour for the new method). During the temperature drop, the modified version behaves better while it is the opposite during the temperature increase. It is not clear at this stage why the new CTF coefficient generation shows an offset in addition to the stair-step effect. This will be investigated in the future.

Figure 7 shows that both curves present a negative peak that is not present in the reference solution.

Again, future work will aim at clarifying why the response of the wooden wall is incorrect.

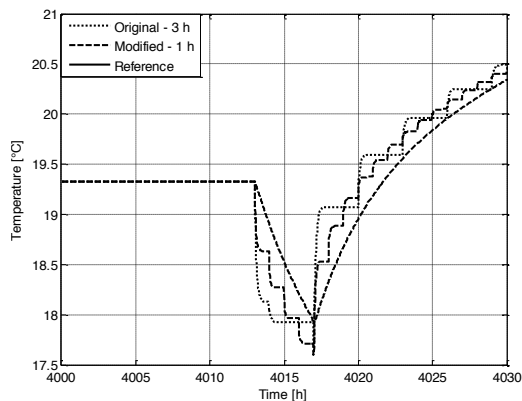


Figure 6 – Evolution of the inside surface temperature for a wooden wall

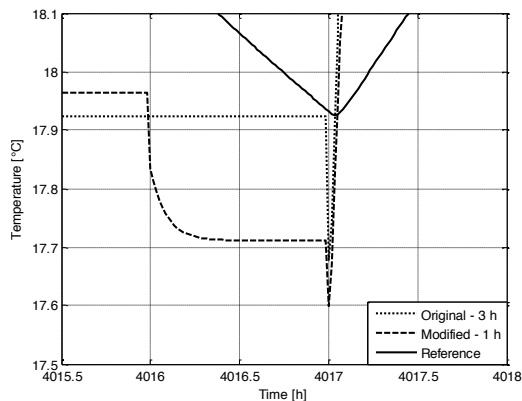


Figure 7 – Evolution of the inside surface temperature for a wooden wall (zoom)

CTF coefficients are generated once per simulation, in the building pre-processing stage. Computational time is imperceptible with the current method. The SS method takes much longer, and the computational time depends on the number of nodes for each wall and the timebase. Table 3 sums up the results obtained for various tests on 6 wall types. The ICF and wooden slabs were described above. The 4 others are a double stud wall with brick veneer (DST), a structural insulated panel (SIPS), a concrete slab (0.5 m thick) and an insulation slab (0.5 m thick).

Table 3 – Comparison of the minimum timebase value and the calculation time between the DRF and SS methods

	Original timebase (DRF method)	New timebase (SS method)	Number of nodes (SS method)	Calculation time (SS method)
	[h]	[h]		[s]
ICF	1	0.2	255	2
DST	0.25	0.1	323	3.4
SIPS	0.15	0.05	350	3.7
Concrete	0.25	0.2	268	2.2
Insulation	0.5	0.1	333	3.4
Wood	3	1	347	3.5

The table shows that the SS method allows decreasing the timebase values (three times in average). This improvement decreases the inaccuracies related to the stair-step effect. The maximum cost of this improvement in terms of

calculation time per slab is 4 seconds (once per simulation).

FULL BUILDING TEST IN TRNSYS

In this section, results coming from the original and modified versions of TRNSYS on a complete house model are presented.

The test was realized in the context of a renovation project, called Zero Energy House Renovation and located in Oud-Heverlee, Belgium (Peeters & Mols, 2012). The building (Figure 8) is an old house built in 1931 and composed of 3 floors (basement, ground floor and 1st floor). The project consists in renovating this house by trying to make it a smart nearly zero energy building. This example is typical of cases encountered by TRNSYS users where the limitations in the current CTF method become apparent, and sometimes make it difficult to obtain meaningful results for short-term analyses (e.g. regarding demand-side management strategies or transients in heating / cooling system controls).



Figure 8 – Picture of the house in project ZEHR (Zero Energy House Renovation) (Peeters & Mols, 2012)

The presented results focus on the evolution of the operative temperature in the kitchen, as shown in Figure 9.

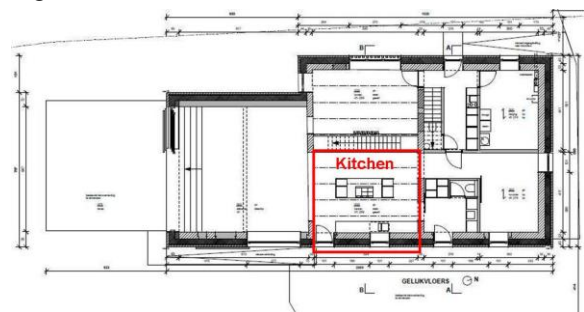


Figure 9 – Plan of the house's ground floor (Peeters & Mols, 2012)

Table 4 below describes the external wall of the kitchen, which is a highly insulated ($U = 0.115$ W/m²-K) and heavy slab.

Table 4 – Description of the external wall in the kitchen (Peeters & Mols, 2012)

	L [m]	k [W/m-K]	ρ [kg/m ³]	Cp [kJ/kg-K]	R [m ² -K/W]
Cavity	/	/	/	/	0,140
Foamglass	0,300	0,038	130	0,840	7,895
Brick	0,330	0,500	1500	0,840	0,660
Plaster	0,010	0,700	1400	0,840	0,014

The setback scenario presented above is extended to this case. The outside temperature is fixed at 0°C.

There are no solar gains. The only room to be heated is the kitchen (convective heater). The heating system stops during 4 hours before a restart with more power (1.5 times the initial power).

The simulation has been carried out twice: firstly, with the original version of TRNSYS (DRF method – Timebase = 5 hours, the minimum achievable) and secondly, with the modified version of TRNSYS (SS method – Timebase = 1.5 hours, the minimum achievable).

Figure 10 below shows the evolution of the operative temperature in the kitchen according to time. The results present a clear difference between the 2 methods (DRF and SS) for generating the CTF coefficients. The SS method allows reducing the stair-step effect, as shown in the single-wall examples. The observed maximum and mean temperature differences between the 2 methods are respectively 0.49 and 0.16°C.

Another important parameter to take into account is the impact of the method on the computational time during the simulation (and not during the CTF coefficients generation). For a 168-day simulation with a time-step of 1 minute, the computational times for the original and modified versions of TRNSYS are respectively 21 and 23 minutes.

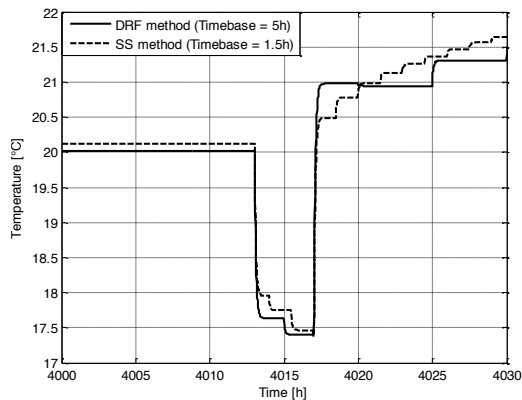


Figure 10 – Evolution of the operative temperature in the kitchen

DISCUSSION AND CONCLUSIONS

This paper describes the implementation of a new CTF coefficients generation method in TRNSYS. The method was successfully implemented in a development version of the TRNSYS building pre-processor (TRNBuild). This work is intended to respond to problems faced by TRNSYS users who simulate heavy, highly insulated walls and want to perform short term analyses to study demand-side management strategies or transients in heating and cooling system controls.

The proposed method alleviates the stair-step effect that occurs in these circumstances, allowing to reduce the minimum timebase by a factor of 3 in average. It provides very satisfactory results in some cases for time-steps down to the order of one minute.

Some walls present more difficulties such as the extreme 50-cm thick plain wooden slab selected for our test. In these cases, the new method delivers some improvements regarding the stair-step effects but does not compare favorably with a reference solution. Further work will aim at better understanding the limitations of the CTF method while the new method is released to more TRNSYS users.

The impact on computational time to generate the CTF coefficients is significant, but this operation only takes place once per simulation and the cost amounts to a few seconds per wall type. The impact on simulation time in a realistic case is acceptable (10% increase).

NOMENCLATURE

A, B, C and D = state-space matrices
 a, b, c and d = CTF coefficients
 C = thermal capacity [J/m²-K]
 C_p = specific heat [J/kg-K]
 Fo = Fourier number
 h = convection coefficient [W/m²-K]
 k = thermal conductivity [W/m-K]
 L = thickness [m]
 n = number of coefficients or nodes
 nx = nodes number
 nu = inputs number (= 2)
 \dot{q} = heat flow [W or kJ/h]
 R = thermal resistance [m²-K/W]
 T = temperature [°C or K]
 t = time [s]
 U = heat transfer coefficient [W/m²-K]
 x = position [m]
 Δt_b = timebase [s]

Greek symbols

α = thermal diffusivity [m²/s]
 Δ = difference
 ρ = density [kg/m³]

Subscript

d = discretized
 i = inside or i-node
 k = number of CTF coefficient
 l = left
 o = outside
 r = right
 s = surface
 w = wall

ACKNOWLEDGEMENT

The research work presented in this paper is financially supported by a grant of Hydro-Québec, FQRNT (Fonds Québécois de Recherche en Nature et Technologies) and NSERC (Natural Sciences and Engineering Research Council of Canada). Moreover the presentation of the full building test in this paper has been possible thanks to the cooperation of Leen Peeters who works on the “Zero Energy House Renovation” project.

REFERENCES

- Delcroix, B., Kummert, M., Daoud, A., & Hiller, M. (2012). Conduction transfer functions in TRNSYS multizone building model: current implementation, limitations and possible improvements. *SimBuild* (p. 8). Madison, Wisconsin.
- Energy Systems Research Unit. (1998). *ESP-r User Guide: The ESP-r System for Building Energy Simulation*. Glasgow.
- EnergyPlus Documentation. (2012). *Engineering Reference* (p. 1247).
- LAPACK. (2011). DGEEVX: computation of the eigenvalues of a general matrix. Retrieved from <http://www.netlib.org/lapack/individualroutines.html>
- Li, X. Q., Chen, Y., Spitler, J. D., & Fisher, D. (2009). Applicability of calculation methods for conduction transfer function of building constructions. *International Journal of Thermal Sciences*, 48(7), 1441–1451. doi:10.1016/j.ijthermalsci.2008.11.006
- Mitalas, G. P., & Arseneault, J. G. (1972). Fortran IV program to calculate z-transfer functions for the calculation of transient heat transfer through walls and roofs. Division of Building research, National Research Council Canada. Retrieved from <http://nparc.cisti-icist.nrc-cnrc.gc.ca/npsi/ctrl?action=shwart&index=an&req=5752842>
- Moler, C., & Van Loan, C. (2003). Nineteen Dubious Ways to Compute the Exponential of a Matrix, Twenty-Five Years Later. *SIAM Review*, 45(1), 3–49.
- Myers, G. E. (1971). *Analytical methods in conduction heat transfer* (McGraw-Hil.). New-York.
- Peeters, L., & Mols, R. (2012). Zero Energy House Renovation. Retrieved February 8, 2013, from <http://www.zehr.be>
- Seem, J. E. (1987). *Modeling of heat transfer in buildings*. University of Wisconsin, Madison.
- Sidje, R. B. (1998). Expokit: a software package for computing matrix exponentials. *ACM Transactions on Mathematical Software*, 24(1), 130–156.
- Stephenson, D. G., & Mitalas, G. P. (1971). Calculation of heat conduction transfer functions for multi-layer slabs. *ASHRAE Transactions*, 77(2), 117–126. Retrieved from <http://nparc.cisti-icist.nrc-cnrc.gc.ca/npsi/ctrl?action=shwart&index=an&req=5757086&lang=en>
- The MathWorks Inc. (2010). *Matlab help*. Natick, MA, USA.
- TRANSSOLAR Energietechnik GmbH. (2011). *TRNSYS 17 - Multizone building modeling with Type 56 and TRNBuild* (p. 258). Stuttgart.