

Implementation and Verification of the IDEAS Building Energy Simulation Library

F. Jorissen^{a,c*}, G. Reynders^{b,c}, R. Baetens^{b,d}, D. Picard^a, D. Saelens^{b,c}, L. Helsen^{a,c}

^aUniversity of Leuven, Department of Mechanical Engineering, Celestijnenlaan 300 - bus 2421, 3000 Leuven, Belgium; ^bUniversity of Leuven, Department of Civil Engineering, Kasteelpark Arenberg 40 - bus 2448, 3000 Leuven, Belgium; ^cEnergyVille, Thor Park 8310, 3600 Genk, Belgium; ^d3E nv, Kalkkaai 6, 1000 Brussels, Belgium;

(Received 00 Month 20XX; final version received 00 Month 20XX)

Keywords: Modelica; Building Energy Simulation; Library; IDEAS; Computation Time; Verification

Building and district energy systems become increasingly complex, requiring accurate simulation and optimisation of systems that combine building envelope, heating ventilation and air conditioning, electrical distribution grids and advanced controllers. Hence, it becomes more challenging for existing simulation tools to provide integrated solutions for these multi-physics problems. Moreover, common building simulation frameworks tightly integrate model equations and their solvers in the program code, which affects model transparency and hampers tool extensions. This is contrasted by equation-based tools such as Modelica, for which different solvers can be used. In this context, the Integrated District Energy Assessment by Simulation (IDEAS) library is developed. After a recent development shift towards more detailed, multi-zone models, this paper presents a comprehensive, well-documented, overview of the buildings part of IDEAS. This includes new computational aspects of the library, improved usability aspects, an updated intercomparison with BESTEST and a verification based on IEA EBC Annex 58.

Nomenclature

Indices

c	Cooling
con	Convective
db	Dry bulb
dif	Diffuse
dir	Direct
ext	Exterior
h	Heating
in	Inlet of a flow
inf	Infiltration
int	Interior
lat	Latent
lw	Longwave
nom	Nominal
rad	Radiative

*Corresponding author. Email: flip.jorissen@kuleuven.be

sen	Sensible
sw	Shortwave
w	Window
wat	Water
z	Zone

Symbols

α	Thermal diffusivity
ΔT	Temperature difference
\dot{m}	Mass flow rate
\dot{Q}	Heat flow rate
ϵ	Emissivity
λ	Latent heat of evaporation
ϕ	Inclination angle
ρ	Density
σ	Stefan-Boltzmann constant
clo	Clothing insulation level
fl	Floor/ceiling indicator
RH	Relative humidity
A	Surface area
a	Absorption coefficient
C	Heat capacity
D	Hydraulic diameter
d	Thickness
E	Thermal energy
F	View factor
f	Fraction
G	Thermal conductance
H	Solar irradiance
h	Convective heat transfer coefficient
h	Specific enthalpy
L	Length or distance
m	Mass
P	Power
p	Number of occupants
q	Heat flow rate per person
$S(\cdot, \cdot, \cdot, \cdot)$	Splice function
T	Dry bulb temperature
t	Transmission coefficient
v	Wind speed
X	Absolute humidity

1. Introduction

With the increasingly stringent energy performance requirements for buildings and the increasing complexity of building energy systems, the need for Building Energy Simulations (BES) tools is becoming stronger. Optimization of building design and control requires simulation tools that can model building dynamics, HVAC (Heating, Ventilation and Air Conditioning) systems, electrical systems and their control. Moreover, Baetens et al. (2012) demonstrated that for instance to correctly evaluate net zero energy buildings (nZEB) state of the art BES tools should also account for the interaction with the energy distribution network. BES tools thus require models from different physical domains, with vastly different time scales and allowing both continuous time modelling

for physical processes and discrete time and event modelling for advanced control strategies. Due to the wide scope and rapid development of BES, model libraries should be easily extendible with new functionalities.

This flexibility and rapid tool development may be hampered by the use of imperative programming languages as applied in common building simulation tools (Wetter et al. 2015b; Wetter, Bonvini, and Nouidui 2016) like EnergyPlus (Crawley et al. 2001), and TRNSYS (Klein 2010). Imperative programming languages assign variable values in sequence, defining the order in which the code should be executed. Also, numerical solution methods are often tightly intertwined with model equations, making it difficult to e.g. use a different solver (Wetter, Bonvini, and Nouidui 2016). The use of imperative languages further complicates the extraction of model equations, which is a requirement for the efficient implementation of optimal control methods such as gradient descent methods (Wetter, Bonvini, and Nouidui 2016). Wetter, Bonvini, and Nouidui (2016) demonstrated a constrained non-linear optimal control problem of a single room where a gradient descent method is 2200 times faster than a gradient-free algorithm. The gradient descent method could be used thanks to the availability of the model equations implemented in the equation-based modelling language Modelica. Similarly, Jorissen and Helsen (2016) and Picard et al. (2016) efficiently implemented a linear MPC controller by extracting a state space representation from a Modelica model (Picard, Jorissen, and Helsen 2015).

To overcome these limitations associated with imperative modelling languages, new generation computational tools (Wetter and van Treeck 2017) and modelling languages such as IDA/ICE (Björnsell et al. 1999) and Modelica libraries such as IDEAS are currently being developed using equation-based (acausal) languages instead of imperative languages. Alternatively, existing frameworks may adopt equation-based modelling. E.g. Spawn-of-EnergyPlus is a partially new implementation of EnergyPlus that uses open standards, including Modelica, to address these problems (Wetter et al. 2015b).

The goal and contribution of the IDEAS (Integrated District Energy Assessment by Simulation) library is to provide a general purpose library in the Modelica environment for performing building energy simulations such that IDEAS meets the requirements stated above. The Modelica language is chosen because it is an open-source, multi-disciplinary, equation-based modelling language, and it is easily adjustable and extendible thanks to its object-oriented structure and the access to all equations. Development of IDEAS started in 2010, whereby the library initially focussed on simultaneous transient simulation of thermal, electric and control systems at the *district level* (Baetens et al. 2012; Baetens 2015). IDEAS was further used to assess the potential of building structural thermal mass (Reynders, Nuytten, and Saelens 2013) and domestic hot water storage (De Coninck et al. 2014) for demand-side management in buildings, and as a benchmark for the identification of grey-box models (Reynders, Diriken, and Saelens 2014). OpenIDEAS (Baetens et al. 2015) extends IDEAS with STochastic Residential Occupant Behaviour (StROBe) models for modelling uncertainty in district energy simulations (Baetens and Saelens 2016) and with the **Fast Buildings** library, which was used together with the grey box toolbox (De Coninck et al. 2016) in a practical implementation of model predictive control for an office building in Brussels (De Coninck and Helsen 2016). Recently a validated air handling unit model was published in IDEAS (Jorissen, Boydens, and Helsen 2017) and IDEAS was used to assess heat pump and PV impact on residential low-voltage distribution grids (Protopapadaki and Saelens 2017).

In 2014 IDEAS was converted to use the IEA EBC Annex 60 library (Wetter et al. 2015a) *base classes*, which ensures compatibility and allows collaboration with other Modelica building libraries: the **Buildings** library from LBL (Wetter et al. 2014), the **AixLib** library from RWTH Aachen (Müller et al. 2016) and the **BuildingSystems** library from UdK Berlin (Nytisch-Geusen et al. 2013).

The focus of the developments in IDEAS was recently extended to more detailed simulations at the *building level*. This requires more detailed, multi-zone models. Recent developments therefore focussed on the scalability of such models in terms of computation time. Moreover, the principles

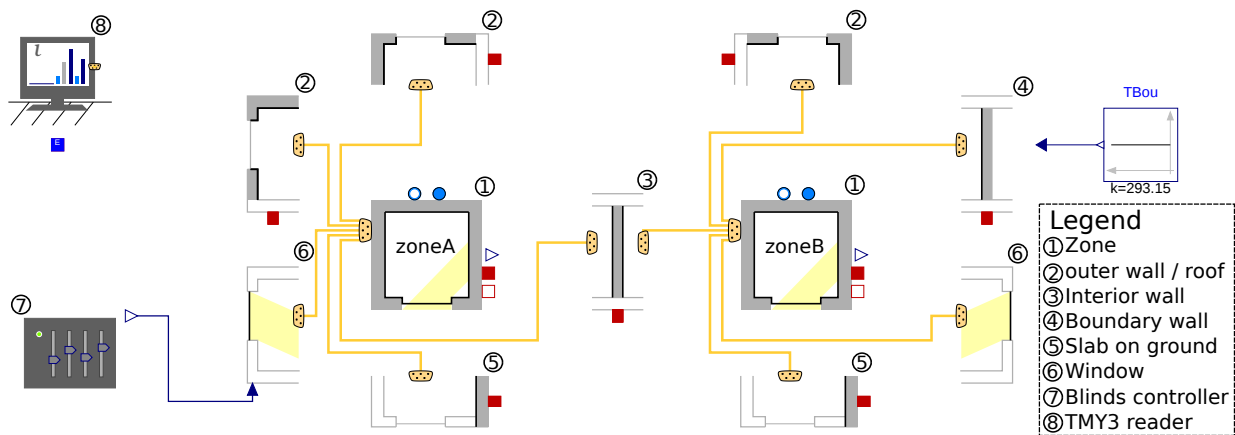


Figure 1.: Illustration of a 2-zone building model indicating the different IDEAS components.

to increase simulation speed in Modelica using explicit time integrators, as outlined by Jorissen, Wetter, and Helsen (2015), were applied to the library. This paper therefore presents the results of these combined efforts.

Future developments of on-going research include model development for detailed district heating network simulations (van der Heijde et al. 2017; van der Heijde, Aertgeerts, and Helsen 2017), Model Predictive Control using detailed controller models (Jorissen and Helsen 2016), modeling Building-Integrated PhotoVoltaics (BIPV), and integrating Building Information Modeling (BIM) and Graphical Information Systems (GIS).

This paper presents the resulting implementation of the building envelope models in version 1.0 of IDEAS (available at <https://github.com/open-ideas/IDEAS>) and verifies both numerical aspects and model accuracy. Although the IDEAS library also includes models for thermal and electrical systems, both on building and district level, the focus of this paper is on the building envelope models, since these are considered as the main difference with other Annex 60 libraries. Section 2 explains the model implementation starting from the implementation of heat transfer principles as a basis for the building component models. The main governing equations are included in Section 2 to support readability and interpretability when discussing the numerical aspects in Section 4, and since their implementation was changed compared to earlier descriptions of IDEAS. Section 3 demonstrates usability aspects as this was another central pillar when designing the library, next to numerical performance. Having a clear and flexible work-flow is important for users since it reduces their model development time. Furthermore it reduces the chance of introducing model errors.

In the second main part of the paper, the performance of the building envelope models in version 1.0 of IDEAS is evaluated. The IDEAS equations have been optimised to reach a good overall trade-off between simulation speed and accuracy. Section 4 discusses several design choices that were made to reduce model complexity and presents benchmarks that demonstrate how computation time scales for two main classes of integration methods. Finally, Section 5 presents several methods that are used to verify the IDEAS models. IDEAS simulation results were compared against IEA EBC BESTEST reference results and the IEA EBC Annex 58 Twin House experiments were used to further verify and compare the library with other simulation tools.

2. Model Description

Modelica, and therefore also IDEAS, uses an object-oriented approach for composing models. IDEAS building envelope models are composed of six main object types: a Zone and five types of building structure *components*: Window, OuterWall, InternalWall, BoundaryWall and SlabOnGround.

Figure 1 shows an example of a 2-zone model consisting of these main components. Each icon represents one of these components and contains equations and variables. **Ports** are used to connect the components to each other, which exposes the variables of one model to the other model and which allows them to interact.

This section presents the modelled physics of the IDEAS models. Section 2.1 elaborates on the implementation of main heat transfer equations. These heat transfer principles are used in the description of the component models in Section 2.2. Although these heat transfer principles may be considered as general knowledge, the equations are explicitly handled in this section to support the discussion in Section 4 that deals with the steps taken to improve the simulation speed. Boundary conditions are discussed in Section 2.3.

2.1. Heat Transfer Principles

The implementation of thermal conduction, convective heat transfer and radiative heat transfer is discussed in this section. Thereby, this section builds on the detailed description of the heat transfer equations implemented in version 0.3 of IDEAS as presented by Baetens (2015). The description in this section is hence limited to the main governing equations and the treatment of non-linear heat transfer equations - in order to make a clear link with the numerical aspects treated in Section 4.

2.1.1. Thermal Conduction through Solids

As a first fundamental assumption, IDEAS treats thermal conduction through building components as a 1 dimensional heat transfer problem, assuming the temperature is evenly distributed over the surface of the component. This is a common assumption in BES models (Crawley et al. 2008). Edge effects are neglected and the material is assumed to have homogeneous and constant thermal properties.

Walls typically consist of one or more material layers. Thermal conduction through these layers is modelled using a series connection of one-dimensional heat conduction models. Each layer is modelled using a series discretisation of lumped thermal conductors and heat capacitors (nodes).

The implementation of the model is elaborated below since it has a large impact on the computation speed of IDEAS, which will be discussed in Section 4.

Each wall layer i consists of n_i time-dependent temperature states $T_{i,j}(t)$, $j \in [1, n_i]$ with heat capacity $C_{i,j}$ and $n_i - 1$ thermal conductors with thermal conductance $G_{i,j}$. $G_{i,j}$ and $C_{i,j}$ are computed assuming homogeneous specific heat capacity c_i , thermal conductivity k_i , density ρ_i , layer thickness d_i and surface area A_i . Heat flow rates $\dot{Q}_{i,j}(t)$, $j \in [1, n_i - 1]$ are computed from Equation (1).

$$\dot{Q}_{i,j}(t) = G_{i,j} [T_{i,j}(t) - T_{i,j+1}(t)]. \quad (1)$$

Heat flow rates $\dot{Q}_{i,0}(t)$ and $\dot{Q}_{i,n_i}(t)$ are respectively the heat flow rates entering the material layer at the boundary with temperature $T_{i,1}(t)$ and leaving the material layer at the boundary with temperature $T_{i,n_i}(t)$.

The time derivative of $T_{i,j}(t)$ is computed from Equation (2).

$$\frac{dT_{i,j}(t)}{dt} = C_{i,j}^{-1} [\dot{Q}_{i,j-1}(t) - \dot{Q}_{i,j}(t)] \quad \forall j \in [1, n_i + 1] \quad (2)$$

Based on Wetter (2004), the number of nodes n_i of layer i is computed as

$$n_i = \max\left(2, 3\frac{\Pi_i}{\Pi_{ref}}\right)$$

$$\Pi = \frac{d}{\sqrt{\alpha}}$$

where α_i is the thermal diffusivity of layer i and Π_{ref} is computed for a concrete slab of 20 cm.

2.1.2. Convective Heat Transfer

Two types of convective heat transfer are distinguished: free interior convection due to buoyancy effects that occur inside the building and exterior forced convection due to the wind. Both are computed using heat transfer correlations that have the form of Equation (3)

$$\dot{Q}_{i,con}(t) = A_i h_i(t) \Delta T_i(t) \quad (3)$$

where A_i is the surface area of component i , $h_i(t)$ is the convective heat transfer coefficient and $\Delta T_i(t)$ is the difference between the zone air temperature and the interior surface temperature of the component, or between the ambient dry bulb temperature $T_{db}(t)$ and the exterior surface temperature of the component.

2.1.2.1. Interior convection. For interior convective heat transfer the correlations from Awbi and Hatton (1999) are used. For a heated floor or cooled ceiling $h_i(t)$ is defined by Equation (4), for a cooled floor or heated ceiling by Equation (5) and for a vertical component by Equation (6)

$$h_{i,1}(t) = \frac{2.175}{D_i^{0.076}} \Delta T_i(t)^{0.308} \quad (4)$$

$$h_{i,2}(t) = \frac{0.704}{D_i^{0.601}} \Delta T_i(t)^{0.133} \quad (5)$$

$$h_{i,3}(t) = \frac{1.823}{D_i^{0.121}} \Delta T_i(t)^{0.293} \quad (6)$$

where D_i is the hydraulic diameter of the component. For vertical components D_i is computed using a default zone height of 2.7 m. For ceilings and floors a square geometry is assumed such that $D_i = \sqrt{A_i}$. Optionally Equation (3) may be linearised by evaluating (4) - (6) using a user-defined nominal value for $\Delta T_i(t)$.

In order to avoid discontinuous switching between $h_{i,1}(t)$ and $h_{i,2}(t)$ when $\Delta T_i(t)$ changes sign, $h_i(t)$ is implemented using Equation (7) where fl is a variable that equals 1 for ceilings and -1 for floors and $S(a, b, c, d)$ is the `spliceFunction` from the IEA EBC Annex 60 library (Wetter et al. 2015a), that creates a smooth transition from variable a to b for rising values of c where $-d \leq c \leq d$.

$$h_i(t) = S(h_{i,2}(t), h_{i,1}(t), \text{fl}\Delta T_i(t), 0.1) \quad (7)$$

2.1.2.2. Exterior convection. The exterior convective heat transfer coefficient $h_{ext}(t)$ is computed using Equation (8) from Jürges (1924) where $v(t)$ is the free-stream air speed that is read from the

TMY3 reader.

$$h_{ext}(t) = S(5.6 + 4v(t), 7.1 \text{ abs}(v(t))^{0.78}, 5 - v(t), 0.5) \quad (8)$$

2.1.3. Radiative Heat Transfer

Longwave radiative heat exchange and shortwave solar irradiation are considered separately and are here discussed in more detail.

2.1.3.1. Longwave radiation. Internal and external radiative heat exchanges between components are modelled using the same Stefan-Boltzmann equation (Equation (9)). In the default setting, this equation is linearised for *internal* longwave heat exchange into Equation (10). $G_{rad,i}$ is an equivalent radiative thermal conductivity. $T_{1,nom,i}$ is the nominal temperature of the first heat exchange surface and $\Delta T_{nom,i}$ is a nominal temperature difference between the heat exchanging surfaces.

$$\dot{Q}_{rad,i}(t) = G_{rad,i} (T_1(t)^4 - T_2(t)^4) \quad (9)$$

$$\dot{Q}_{rad,i}(t) = G_{rad,i} (2T_{1,nom} + \Delta T_{nom}) (T_{1,nom}^2 + (T_{1,nom,i} + \Delta T_{nom,i})^2) (T_1(t) - T_2(t)) \quad (10)$$

External longwave radiative heat exchange of component i is computed from the radiative environment temperature $T_2(t) = T_{e,i}(t)$ (see Section 2.3), $T_1(t) = T_{i,1}$ where $T_{j,1}$ is the first node temperature of the outermost layer j of component i . $G_{rad,i}$ is computed from Equation 11 where $\sigma = 5.67 \times 10^{-8} \text{ W}/(\text{m}^2\text{K}^4)$ is the Stefan-Boltzmann constant, A_i is the surface area and $\epsilon_{lw,i}$ is the longwave emissivity of the outer layer of component i .

$$G_{rad,i} = \sigma \epsilon_{lw,i} A_i \quad (11)$$

Internal longwave radiative heat exchange in the zone is computed using the Mean Radiant Temperature Network model (Liesen and Pedersen 1997) as described by Carroll (1980). For a zone consisting of i components, i instances of Equation (9) are used where $T_1(t)$ equals the component surface temperature and $T_2(t)$ equals the radiant star temperature $T_{star}(t)$. Since energy must be conserved, the sum of these radiative heat flow rates must be zero, from which the value of $T_{star}(t)$ is computed. $G_{rad,i}$ is computed from Equations (12)-(13) (Carroll 1980) where A_i and ϵ_i are respectively the surface area and longwave emissivity of component i and $\sigma = 5.67 \times 10^{-8} \text{ W m}^{-2} \text{ K}^{-4}$.

$$F_i = \frac{1}{1 - \frac{A_i F_i}{\sum_i A_i F_i}} \quad (12)$$

$$G_{rad,i} = \frac{\sigma A_i}{\frac{1-\epsilon_i}{\epsilon_i} + \frac{1}{F_i}} \quad (13)$$

Equation (12) applies to each component i , composing an implicit system of equations that the Modelica tool needs to solve when initialising the model. When the solution to this problem cannot be found or does not exist, $G_{rad,i}$ is computed from a simplified, explicit formulation that uses emittance-area products to compute the view factors (Baetens 2015).

2.1.3.2. Shortwave radiation. Incident solar irradiation on component i consists of a diffuse component $H_{dif,i}(t)$ and a direct component $H_{dir,i}(t)$ (Baetens 2015). Objects or shading devices cause shade, requiring a correction of these solar irradiance components. The modified diffuse irradiance $H_{dif,i}(t)'$ and direct irradiance $H_{dir,i}(t)'$ are therefore defined depending on the chosen shading type and the declination, inclination and zenith angles of the sun.

Exterior shortwave radiation causes a heat flow rate $\dot{Q}_{sw,i,ext}(t)$ to be injected on the outside of wall i . This heat flow rate is computed using Equation (14)

$$\dot{Q}_{sw,i,ext}(t) = \epsilon_{sw,i} A_i (H_{dir,i}(t)' + H_{dif,i}(t)') \quad (14)$$

where $\epsilon_{sw,i}$ is the shortwave emissivity of the outer material of the wall.

Shortwave radiation enters the building through windows, causing heating on the inner surfaces of the zone components. This is discussed in Section 2.2.2.

2.2. Component Models

To compose a building model, IDEAS groups the fundamental heat transfers equations into component models. Two basic component types exist: zones and surfaces. IDEAS contains five *surface* models: one for modelling windows and four different opaque wall types. The following paragraphs summarize the main features of each *component model*.

2.2.1. Opaque walls

The four opaque wall models – as well as the window model – use the same code to model conductive heat transfer (see Section 2.1.1) through a series of material layers. Each wall model has different boundary conditions for the outer side of the material layers. Table 1 provides an overview of these boundary conditions. Side a of each component needs to be connected to a zone. Side b of the component is either connected internally to a submodel, or needs to be connected to a second zone. The structural and thermal properties of the material layers are defined using predefined or custom templates (**records**). Model parameters **A**, **inc** and **azi** allow to define the surface area, inclination angle and azimuth (tilt) angle of the component. Each model uses Equations (4)-(6) for computing convective heat transfer coefficients between the component and the zone air temperature, and Equation (9) for radiative heat exchange. Figure 2 graphically illustrates how the model equations are linked to each other for the outer wall model. Shortwave heat gains are not included in Figure 2 since the shortwave heat gains do not depend on the wall surface temperature.

The outer wall model should be used when modelling a wall between a zone and the surroundings. The internal wall model should be used when modelling a wall between two zones. The boundary wall model allows the user to define a custom boundary condition for the temperature or heat flow rate at the wall surface at side b . The slab on ground model may be used to model a floor element that is in direct contact with the ground. The model follows the calculation procedure defined in Annex D of ISO 13370. Quasi-stationary losses to the ground are computed as described by (Baetens 2015, Annex A.2.3).

2.2.2. Window

In addition to the surface area, inclination angle and tilt angle, the window model contains parameters that define the frame surface area fraction f , the frame type, the glazing type and the shading system type. Previously explained equations for interior and exterior convection and interior and exterior long wave radiative heat exchange are used to compute heat transfer with the frame and glazing surfaces.

Table 1.: Overview of the five available component models and their boundary conditions

Icon	Name	Boundary condition <i>a</i>	Boundary condition <i>b</i>
	InternalWall	zone <i>a</i>	zone <i>b</i>
	OuterWall	zone	weather model
	BoundaryWall	zone	ideal boundary
	Slab on Ground	zone	ground model
	Window	zone	weather model

Table 2.: Legend of symbols used in model structure illustrations.

Symbol	Type	Description
●	State variable	Variable that is computed from differential equation.
□	Algebraic variable	Variable that is computed from algebraic equation
▷	Input variable	Variable whose value is known
■	Connector variable	Variable depending on equations outside of the considered model.
—	Identity	Connected variables are equal.
—□—	Linear equation	Equation that is a linear function of the connected variables.
—◯—	Non-linear equation	Equation that is a non-linear function of the connected variables.

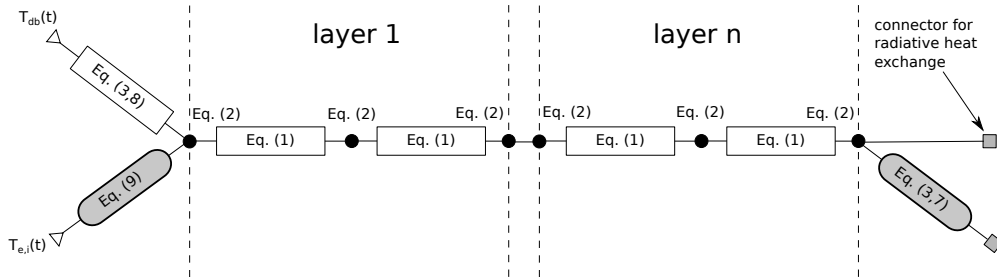


Figure 2.: Equation structure of the OuterWall model. See Table 2 for a legend of the used symbols.

Thermal losses through the frame are computed using its surface area $A_f = fA$ and an equivalent U-value that is defined in the frame type. Thermal losses through the glazing system are computed using static models for the conductive heat transfer through glazing. Thermal mass of the glazing is lumped as a single state to the inside of the window. Convective heat transfer through gas layers is modelled using correlations from Wright (1996) and Hollands, Raithby, and Konicek (1975). Glazing type properties such as the longwave emissivity are used to compute the radiative heat transfer between the glazing sheets.

Direct and diffuse shortwave radiation causes heat gains on the outside of the frame (see Equation (14)), in the zone and in the glazing sheets. As an input for these models, the modified diffuse and direct solar irradiance $H'_{dif,i}(t)$ and $H'_{dir,i}(t)$ are used (see Section 2.1.3.2). Solar heat gains $\dot{Q}_{w,i,k}(t)$ in each of the glazing sheets k of window i are computed using Equation (15), where solar absorption coefficients $a_{dif,i,k}$ and $a_{dir,i,k}(\cdot)$ are respectively constants and a piecewise linear function of the inclination angle $\phi(t)$, which is the angle between the component surface outward normal and the direction of the sun rays. Heat flow rates $\dot{Q}_{w,i,k}(t)$ are injected between two thermal conductors

that represent glazing sheet k .

$$\dot{Q}_{w,i,k}(t) = A_i(1 - f_i)(a_{dif,i,k}H'_{dif,i}(t) + a_{dir,i,k}(\phi(t))H'_{dir,i}(t)) \quad (15)$$

Diffuse solar gains $\dot{Q}_{w,dif,i}(t)$ and direct solar gains $\dot{Q}_{w,dir,i}(t)$ from window i are transmitted into the zone. Their values are computed from Equations (16) - (17) where $t_{dif,i}$ and $t_{dir,i}(\cdot)$ are solar transmission coefficients similar to the absorption coefficients. These and other window glazing properties are obtained from the LBL windows and daylighting software (Finlayson et al. 1993).

$$\dot{Q}_{w,dif,i}(t) = A_i(1 - f_i) H'_{dif,i}(t) t_{dif,i} \quad (16)$$

$$\dot{Q}_{w,dir,i}(t) = A_i(1 - f_i) H'_{dir,i}(t) t_{dir,i}(\phi(t)) \quad (17)$$

2.2.3. Building or Window Shade

Outer walls or windows may be shaded, which reduces the direct and diffuse solar radiation intensity on the surface. Shading models are therefore implemented that compute $H'_{dif,i}(t)$ and $H'_{dir,i}(t)$ (see Section 2.1.3.2). Outer walls or windows can both be shaded by exterior objects such as buildings. The `BuildingShade` model computes solar shading from an object with a distance L from the building and height difference h with respect to the top of the wall or window.

For windows following additional shading options are supported (Baetens 2015).

- Screen: Controllable screen represented by a shading factor.
- Overhang: Opaque window overhang with finite width and length.
- Side fins: Opaque window side fins with finite width and length.
- Box: Combination of overhang and side fins.

Two or more shading options can be combined using a template.

2.2.4. Building Zone

Each component must be connected to one or two (see Table 1) zone models. These connections expose the component variables, such as the radiative temperature, emissivity and inclination angle to the following zone sub-models:

- a zone air model,
- the longwave radiative heat exchange model of Section 2.1.3.1,
- a radiative heat gain distribution model,
- an occupant internal gain and thermal comfort model.

These models are here discussed in more detail.

2.2.4.1. Zone air model. The default zone air model assumes that the zone air is well mixed with a uniform temperature T_{air} and absolute humidity X_{air} , which has units of kg water per kg of moist air. CO₂ concentrations may be computed depending on the chosen `Medium`, which is a Modelica package that contains medium properties (e.g. specific heat capacity) and equations (e.g. the ideal gas law). External ventilation components can be connected to the zone air model through two fluid ports (see blue and blue-white circular ports in Figure 3). Convective heat flow rates for all connected components are computed using Equations (3)-(6). The air thermodynamic properties are governed by Equations (18)-(20)

$$T_{air}(t) = T(h_{air}(t), X_{air}(t)) \quad (18)$$

$$m_{air} \frac{d h_{air}(t)}{d t} = \sum_i \dot{Q}_{con,i}(t) + \dot{Q}_{con,ext}(t) + \dot{Q}_{con,int}(t) + [h_{air,in}(t) - h_{air}(t)] \dot{m}_{air}(t) + [h_{air,ext}(t) - h_{air}(t)] \dot{m}_{inf}(t) \quad (19)$$

$$m_{air} \frac{d X_{air}(t)}{d t} = \dot{m}_{lat} + [X_{air,in}(t) - X_{air}(t)] \dot{m}_{air}(t) + [X_{air,ext}(t) - X_{air}(t)] \dot{m}_{inf}(t) \quad (20)$$

where $T(\cdot, \cdot)$ is a function that depends on the chosen **Medium** (see Wetter et al. (2015a)), m_{air} is the zone air thermal mass, $h_{air}(t)$ is the specific enthalpy of the air, $\dot{m}_{air}(t)$ and $\dot{m}_{inf}(t)$ are the air mass flow rate entering the zone through the fluid ports and the infiltration mass flow rate, $h_{air,in}(t)$ and $X_{air,in}(t)$ are respectively the enthalpy and moisture content of the air entering the zone through the zone's fluid ports, $h_{air,ext}(t)$ and $X_{air,ext}(t)$ are respectively the enthalpy and moisture content of the air that infiltrates from the surroundings, \dot{m}_{lat} [kg s^{-1}] is the mass flow rate related to the latent heat load (see Section 2.2.4.3), $\dot{Q}_{con,ext}(t)$ are optional convective heat gains from external models (e.g. radiators) that are connected to the zone and $\dot{Q}_{con,int}(t)$ are internal convective heat gains that are computed from the occupant model. The infiltration air flow rate is computed assuming a constant n50-value, which is a parameter of the zone model. Interzone air flow must be modelled explicitly using the zone fluid ports since such models are not yet integrated into the component models.

2.2.4.2. Radiative heat gain distribution. For each zone three sources of radiative heat gains exist: diffuse solar gains $\dot{Q}_{z,dif}(t)$, direct solar gains $\dot{Q}_{z,dir}(t)$ and other radiative heat sources $\dot{Q}_{rad}(t)$ such as occupants, appliances and radiators. They are computed from Equations (21)-(23)

$$\dot{Q}_{z,dif}(t) = \sum_i \dot{Q}_{w,dif,i}(t) \quad (21)$$

$$\dot{Q}_{z,dir}(t) = \sum_i \dot{Q}_{w,dir,i}(t) \quad (22)$$

$$\dot{Q}_{rad}(t) = \dot{Q}_{rad,ext}(t) + \dot{Q}_{rad,int}(t) \quad (23)$$

where $\dot{Q}_{w,dif,i}(t)$ and $\dot{Q}_{w,dir,i}(t)$ are the diffuse and direct solar gains (see Equations (16)-(17)) from each window that is connected to the zone, $\dot{Q}_{rad,ext}(t)$ are radiative heat gains from models that are connected to the zone externally (e.g. radiators) and $\dot{Q}_{rad,int}(t)$ are internal radiative heat gains from the occupant model.

$\dot{Q}_{z,dif}(t)$ and $\dot{Q}_{rad}(t)$ are distributed over the connected components using weights that are proportional to the area-emissivity product of each component. $\dot{Q}_{z,dir}(t)$ is distributed over the connected components assuming that a fixed fraction $(1 - \epsilon_{sw,floor})\dot{Q}_{z,dir}(t)$ is absorbed by the floor(s). The remaining heat is diffusely spread over the other components using weights that are proportional to the area-emissivity product of each component.

2.2.4.3. Occupancy models. The zone model contains optional models for computing occupant internal gains and thermal comfort. Both models require the user to choose an occupancy type and act as an alternative for the stochastic residential occupant behavior modelling approach provided through the `StROBe` package. This `record` defines the sensible and latent heat loads per person \dot{q}_{sen} , \dot{q}_{lat} , the radiative fraction f_{rad} and the clothing insulation level `clo`. Various occupancy types are available using data from (ASHRAE 1997).

When using the occupant internal gain model, the number of occupants $p(t)$ needs to be connected as an input to the model. The internal gains are then computed from Equations (24)-(26) where $\lambda_{wat} = 2418$ kJ/kg is the latent heat of evaporation of water at 35 °C. CO₂ gains from occupants are only computed when the chosen `Medium` declares the CO₂ concentration as a variable. The model then assumes a constant CO₂ generation rate per occupant depending on the occupant metabolic rate, which is estimated from $\dot{q}_{sen} + \dot{q}_{lat}$. By default internal gains from occupants are not computed such that $\dot{Q}_{rad,int} = \dot{Q}_{con,int} = \dot{m}_{lat} = 0$ and $p(t)$ need not be defined.

$$\dot{Q}_{rad,int}(t) = f_{rad} \dot{q}_{sen} p(t) \quad (24)$$

$$\dot{Q}_{con,int}(t) = (1 - f_{rad}) \dot{q}_{sen} p(t) \quad (25)$$

$$\dot{m}_{lat}(t) = \frac{\dot{q}_{lat}}{\lambda_{wat}} p(t) \quad (26)$$

The zone model further includes an optional model for evaluating occupant thermal comfort. Fanger’s model (Fanger 1970) as described in (ASHRAE 1997) and implemented by Wetter et al. (2014) uses the values of `clo`, \dot{q}_{sen} and \dot{q}_{lat} from the occupancy type `record` and the zone air relative humidity, radiative temperature and air temperature to compute the Percentage of People Dissatisfied (PPD) and the Predicted Mean Vote (PMV), which are outputs of the model.

2.3. Climatic Boundary Conditions

Boundary conditions are those variables (or equations) that affect the ‘state’ of the building, but whose value is not affected by it. For instance the ambient temperature affects the state (temperature) of the building, but we do not consider the effect of the building on the ambient temperature.

The outdoor dry bulb temperature $T_{db}(t)$, the beam solar irradiation on a horizontal surface $H_{dir,hor}(t)$, the diffuse solar irradiation on a horizontal surface $H_{dif,hor}(t)$, the wind speed $v(t)$, relative humidity $RH(t)$ and cloud coverage are read from an EnergyPlus based TMY3 file. They are used to compute other boundary conditions, e.g. the radiative environment temperature $T_{e,i}(t)$ and the external shortwave radiation. For some of these computations, code is now used from the jointly developed IEA EBC Annex 60 library (Wetter et al. 2015a). Other parts of the implementation were presented by Baetens (2015).

3. Usability

This section discusses the usability and practical aspects of the library.

Interface

Composing a building envelope model in `IDEAS` typically consists of four main steps, for which `IDEAS` uses the graphical user interface of Modelica tools. Firstly, a single instance of the TMY3 reader model needs to be added from the component library. This model contains the TMY3 reader and centralises weather calculations. Secondly, individual component models (zones, windows, walls)

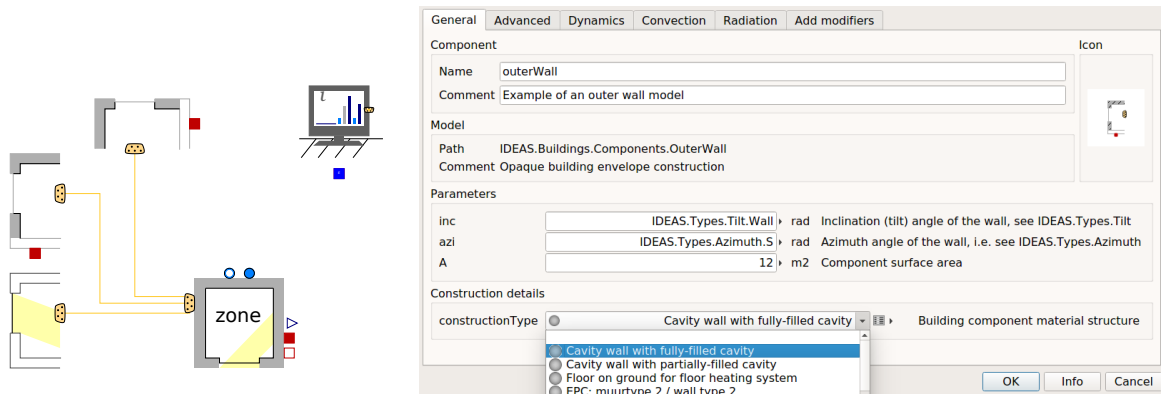


Figure 3.: Left: Graphical illustration of a simple one-zone TMY3 building model consisting of a window, roof and a wall represented using two `OuterWalls`. The TMY3 reader is shown in the top right. Right: Parameter view of the `OuterWall` model in Dymola.

may be dragged and dropped into the model. Thirdly, the TMY3 reader and component parameters need to be defined. Finally, each building component needs to be connected to a zone by connecting the respective connectors of the component and the zone. Figure 3 shows the result of these steps for a zone consisting of a wall, a roof and a window. It also shows how the parameter list of the outer wall model looks like in the commercial Modelica tool Dymola.

Records

The construction properties of a wall or glazing type of a window need to be defined by selecting the appropriate `record` using a drop down menu as shown in Figure 3. A number of predefined constructions exists. Defining constructions is intuitive, as illustrated using the example of a filled cavity in Listing 1.

Listing 1: Cavity wall record definition

```
record CavityWall "Cavity wall with fully-filled cavity"
  extends IDEAS.Buildings.Data.Interfaces.Construction(
    mats={IDEAS.Buildings.Data.Materials.Brick(d=0.08),
          IDEAS.Buildings.Data.Materials.Rockwool(d=0.1),
          IDEAS.Buildings.Data.Materials.Brick(d=0.14),
          IDEAS.Buildings.Data.Materials.Gypsum(d=0.015)});
end CavityWall;
```

Each material listed in the array `mats` points at a `record` that contains predefined material properties such as density, specific heat capacity and emissivities. The definition of each of these `records` is extended by supplying the material layer thickness `d` [m]. Glazing types are defined in a similar way.

Building Zone Template

Since most zones are rectangular, the work flow of composing a building model can be simplified. For such zones a template model has been created that consists of a zone, six components (a ceiling, a floor and four walls) and an optional window and external connection for each of the walls and the ceiling. Using this template, the wall properties and other parameters can be set more easily and connections are made internally, which allows a 1 zone model to be created in two minutes.

Replaceable Components

The shading model, comfort model, internal gains model and zone air model are *replaceable*, meaning that the model can be *redeclared* (replaced) by a different model as long as it contains the

same interface, or generally speaking, the same connectors. This makes the use and development of alternatives for these models easy, since templates exist that already contain the required output connectors. It also simplifies code re-usability and allows entire sub-models to be redeclared, which for instance facilitates the comparison of multiple heating or ventilation systems.

Documentation

Modelica models contain a documentation section. Most of the IDEAS models have been documented following Annex 60 documentation guidelines (Wetter et al. 2015a).

Limitations

IDEAS has some limitations. On the one hand, as with any Building Performance Simulation (BPS) tool, the applicability of IDEAS for a specific use case depends on the availability of the models required for this use case. E.g. Phase Change Materials are currently not supported by the library. However, since the library is open-source and equation based, existing models can easily be adjusted to be added to IDEAS. In this particular example, it satisfies to make parameter C in differential equation (2) temperature-dependent, and to expose this parameter at the user-level. Similarly, building component models that integrate both the thermal and electrical aspects of building integrated photovoltaics (BIPV) are currently being implemented. IDEAS usually does not accurately model fast transients of individual HVAC components. Therefore, these models are mostly suited for building energy use calculations, and less suited for e.g. 1) testing and development of control sequences that have dynamics with similar time scales as these HVAC dynamics, and 2) the computation of electrical power use transients.

On the other hand, the level of detail in IDEAS is comparable to that of other BPS tools. E.g. edge effects of walls are neglected and zone air temperatures are lumped. However, Modelica models need to be translated to C code and then compiled. This translation takes some time, before the actual simulation can start, but the resulting code is faster in general. Furthermore, generic Modelica tools such as Dymola do not have a graphical user interface that is tailored to BPS. This may limit the usability of the tool as a general engineering software. To reduce model development time, as well as the learning curve associated with Modelica, *template* models are included in IDEAS. These *template* models, such as rectangular zones and typical housing archetypes, exploit preconfiguration of groups of building components by propagating model parameters. At the same time, Python tool-chains are being developed to generate IDEAS building models directly from Building Information Models (BIM) (Reynders et al. 2017) or Graphical Information Systems (GIS) (De Jaeger, Reynders, and Saelens 2017).

4. Computational Aspects

When translating a model into an executable, a Modelica tool collects all differential-algebraic equations (DAEs) of the model and then determines how to efficiently solve these using computer algebra. The Modelica user only has to choose a time integration algorithm (e.g. Dassl, LSodar, Dopri45, Radau, explicit Euler) and tolerance or fixed time step size. This simplifies development work, since no algebraic loop solvers or time integrators need to be integrated into the code, but it also hides complexity from developer and user (Jorissen, Wetter, and Helsen 2015). This may lead to slow models and models that do not scale well with problem size.

This section first provides a qualitative theoretical motivation for the model equations and elaborates on some design choices in IDEAS that ensure fast and robust models. Secondly computational benchmarks demonstrate numerical impact on some benchmark models.

We do not compare Modelica with traditional BPS tools, as making a fair comparison could become quite involved. Wetter and Haugstetter (2006) report a case where Modelica is three to

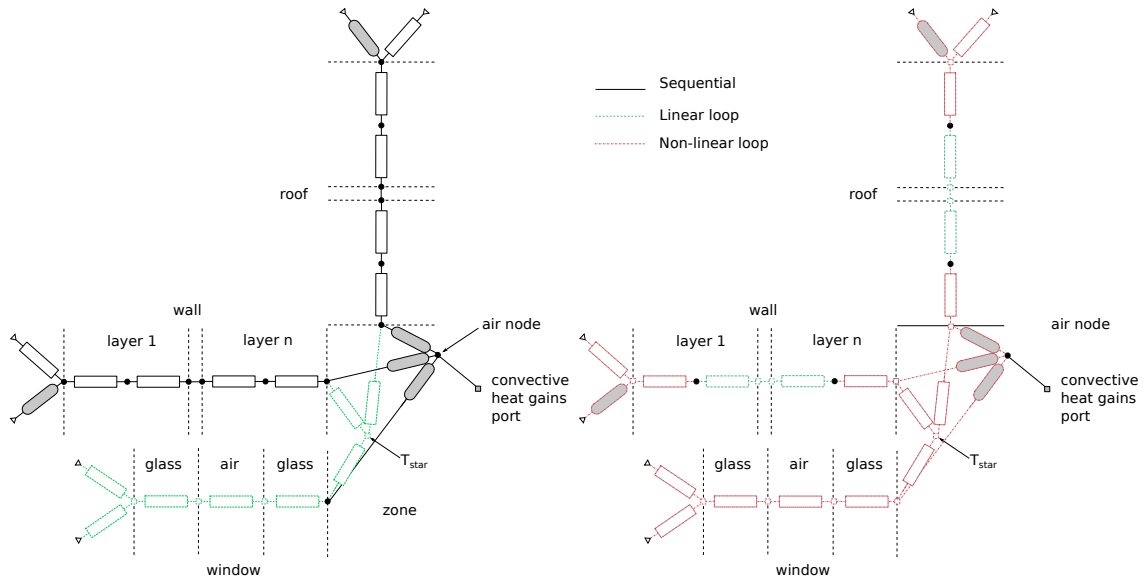


Figure 4.: Equation structure of the model in Figure 3 using symbols from Table 2. Equations and variables that are part of algebraic loops are indicated using specific line types and colours. In the left figure states are located on the surface of solid material layers and the window model contains a state representing the window dynamics. This design choice leads to fewer algebraic loops than in the right case where the material surface temperature is an algebraic variable and the window is considered quasi-stationary.

four times slower than TRNSYS. Their Modelica model however included time-consuming non-linear algebraic loops, which are most likely eliminated in our code, as will now be explained.

4.1. Motivation of Design Choices

Three fundamental design choices were made in the development of the IDEAS library to improve simulation time and scalability. These choices focus on algebraic loops, duplication of calculations and parameter divisions.

Jorissen, Wetter, and Helsen (2015) identified algebraic loops as important sources for increasing computation times. Algebraic loops may be generated while solving for algebraic variables that have mutual dependencies, e.g. a system of two equations $a = f_1(b)$ and $b = f_2(a)$. The extent of an algebraic loop (i.e. which equations it consists of and which variables it solves for) is bounded by state variables (see Table 2), or by what we in this discussion call *inputs* of the algebraic loop. These *inputs* are variables that do not depend (directly or indirectly) on the variables that are solved in the algebraic loop. When one or more of the algebraic loop equations is non-linear, then the entire algebraic loop is typically solved using a non-linear solver. This incurs a computational penalty since non-linear algebraic loops need to be solved iteratively. Moreover, when all elements of A in the linear algebraic loop $Ax = b$ are constant, then the model can be simplified off line, for instance by inverting A and by computing $x = A^{-1}b$. Since algebraic loops are bounded by states, algebraic loops can be avoided by introducing state variables, which typically have small time constants. This however deteriorates the integrator computation time (Jorissen, Wetter, and Helsen 2015).

IDEAS avoids both algebraic loops and small time constants by three main design choices. Firstly, we discretise heat conduction such that a state exists on the wall surface or differently formulated, the wall surface temperature is a state variable, defined by a differential equation (Equation (2)), and not an algebraic variable. This results in faster dynamics compared to reality, however using

a resistor instead of a capacitor at the wall surface would result in slower dynamics compared to reality. So both options incur a deviation, which can be limited by applying a fine discretisation. Secondly, by default the interior longwave heat exchange and window exterior longwave heat exchange are linearised, which is justified in small temperature ranges. Finally, by default the thermal mass of window glazing is lumped into a single state (in contrast to the conventional approach where window glazing is treated stationary).

These choices have a profound impact on the model complexity, as illustrated in Figure 4, which shows the equation structure that is formed when connecting the components of the example in Figure 3. The left part of the figure shows the equation structure with default options. The right part shows the equation structure if less favourable design choices would have been taken, i.e. the wall surface temperature is an algebraic variable, the window contains no thermal mass and longwave radiative heat exchange is not linearised.

Using the default options, the convective heat exchange can be computed directly since it depends on two surface temperatures, both of which are *state* variables. Therefore their value is known from the time integrator. On the right side of Figure 4 the surface temperatures are *algebraic* variables, which depend on the radiative and the convective heat exchange, requiring all these equations to be solved simultaneously. Moreover, since the window does not contain a state variable the thermal dynamics of the window glazing and the exterior window heat exchange equations are coupled into the same algebraic loop. Since the convective heat exchange equations are non-linear a non-linear algebraic loop is formed. Smaller non-linear algebraic loops are also generated at the outer surface of the walls, since these surface temperatures are also algebraic variables. With default options an algebraic loop is also formed for the computation of T_{star} , but this is a linear algebraic loop since all longwave radiative heat exchange equations are linear. A second algebraic loop is formed in the window glazing, which is also linear since the window radiative heat exchange is linearised by default.

A design choice unrelated to algebraic loops is to avoid duplicate evaluation of variables such as $H_{dif,i}(t)$ for components i that have the same orientation. Typically only five distinct orientations are required, i.e. south, west, north, east and horizontal. This default list of orientations is computed centrally in the TMY3 reader. The individual component models only compute $H_{dif,i}(t)$ when the orientation of the component is not contained within the default list. The list can be changed or extended by the user.

A final design choice is to avoid parameter divisions. When a continuous time expression contains a division by a parameter, the compiled C-code may check at *each* time step whether that parameter equals zero to avoid division by zero. Equations (1) and (2) are therefore formulated as a product instead of as a division by using parameters G and C^{-1} , which are the inverse of more commonly used variables R and C . Jorissen, Wetter, and Helsen (2015) have shown that such reformulations can significantly reduce the computation time of these expressions. Such parameter inversions are implemented throughout the library.

4.2. Numerical Benchmarks

The impact of these design choices has been quantified in two ways. Firstly, the default IDEAS implementation is compared with an implementation with other design choices. Secondly, we quantify in three benchmarks how the computation time of the default IDEAS implementation scales with problem size. A benchmark using IDEAS models that includes the model accuracy when comparing several time integrators, tolerances and time steps to each other was described by Jorissen, Wetter, and Helsen (2015).

All simulations were performed using Dymola 2017 and Dassel with the default solver tolerance of 10^{-4} on a 2011 Macbook Pro with an i7 processor of 2.2 GHz running Ubuntu 14.04 in a virtual machine on OSX 10.11.6. Dymola options `Evaluate=true` and `Advanced.EfficientMinorEvents=true` were used and the C compiler flag `-O2` was used.

4.2.1. Comparison with Other Design Choices

To illustrate the impact of our design choices, the default IDEAS implementation is compared to an old implementation that has no window dynamics, that uses parameter divisions and where the wall surface temperature is an algebraic variable.

Both implementations were used to simulate IEA EBC BESTEST case 900 FF (see section 5.2) for one year. A free float model was used since these models only contain building envelope components. The default implementation requires 3.1 seconds compared to 47 seconds when using the old implementation. Both models generate one non-linear algebraic loop for computing the absolute humidity from the relative humidity. The old implementation further generates 5 non-linear algebraic loops of which the largest one has 19 iteration variables. The default implementation only generates linear algebraic loops.

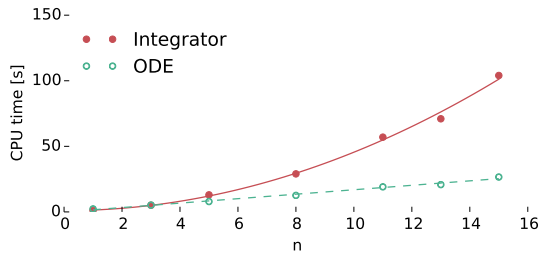
The same analysis was performed for a 9-zone free-floating model of a terraced house. For this model the simulation time was 60 seconds using the default implementation and 193 seconds using the old implementation. The relative speed increase is thus smaller. This is however caused by the integrator overhead, which increases strongly for large models (see Section 4.2.2). When only considering the CPU time required for evaluating the algebraic and differential equations, the computation time is 15.2 seconds instead of 139 seconds in the old implementation.

4.2.2. Scaling Benchmarks

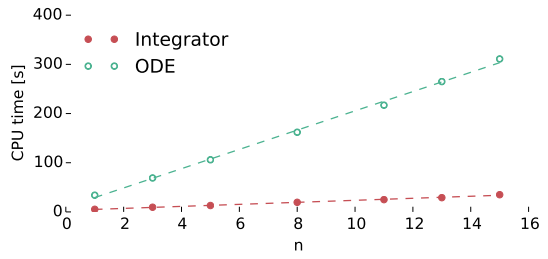
Since IDEAS is used to perform district and detailed building simulations, the computation time should not become prohibitively large for models with many zones and/or many components. Ideally the computation time scales linearly with the problem size. Three benchmarks illustrate how IDEAS scales. The first two benchmarks analyse how computation time scales with the number of zones. The third benchmark investigates how computation time scales with the number of components that are connected to one zone.

The first benchmark is a model that consists of n identical instances of IEA EBC BESTEST case 900 FF. Since the models are identical, their excitations are identical and the integrator step sizes therefore should not change a lot. This benchmark therefore assesses the impact of the *number* of equations on the computation time, without considering different dynamic excitations of these equations. The results of this benchmark are presented in Figure 5a and Figure 5b for respectively Dassl integration and Euler integration with a fixed step size of 30 s. In both cases the overhead required for evaluating the model (ODE) equations scales linearly with the number of zones n . However, the integration time of Euler is negligible and scales linearly whereas the integration time of Dassl scales more or less quadratically and dominates the computation time. Euler integration will therefore become faster for large models. When this transition occurs cannot be generalised. It depends a lot on the model excitation and the time constants of the model.

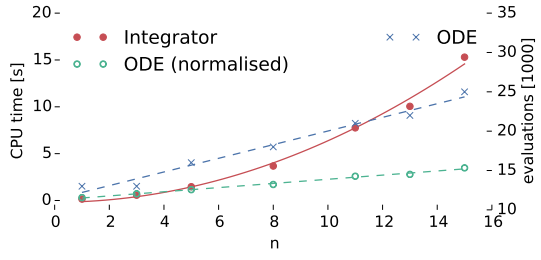
The second benchmark does include different excitations by using IEA EBC BESTEST case 900, i.e. with heating system. Moreover building $i \in [0, n - 1]$ is rotated by $360^\circ i/n$ relative to the default orientation. This means each model is excited differently, which may cause the number of integrator steps, and therefore also the number of ODE evaluations, to rise in a variable time step solver. Results are presented in Figure 5c. The right vertical axis shows the number of ODE evaluations, which rises linearly with the number of zones, although there is a large offset. In the first benchmark the number of evaluations stayed more or less constant. Figure 5c also presents the total integrator time and *normalised* ODE computation times \bar{t}_i , which equal $t_i e_0 / e_i$ where e_i (see right y-axis) is the number of ODE evaluations required for simulating the model with size n_i . Normalised ODE times are plotted since we are interested in the evaluation cost of a single ODE evaluation, regardless of what integrator is used. Figure 5c shows that in this case the integrator time scales more than quadratically. In addition to that the number of ODE evaluations also rises. The computation time per ODE evaluation however scales linearly with the problem size.



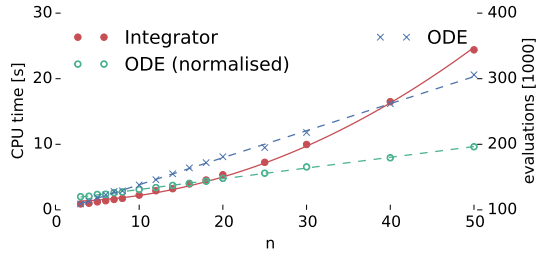
(a) Integrator and ODE computation times for n free floating zones using Dassl ($\text{tol} = 10^{-4}$)



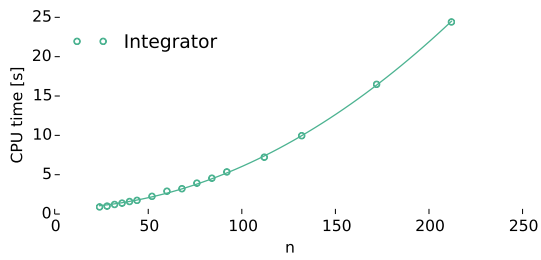
(b) Integrator and ODE computation times for n free floating zones using Euler ($\Delta t = 30$ s)



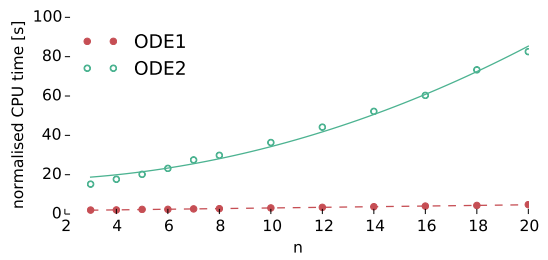
(c) Normalised integrator and ODE computation times and number of ODE evaluations for n conditioned zones



(d) Normalised integrator and ODE computation times and number of ODE evaluations for n walls in 1 zone



(e) Integrator time as a function of n state variables, instead of n walls in Figure 5d



(f) Normalised ODE evaluation time using default (ODE1) implementation and using the old design choices (ODE2) for n walls in 1 zone

Figure 5.: Benchmark computation times for models with various size indicators n . Dashed and full lines respectively indicate affine ($a + bn$) or quadratic ($a + bn^2$) trend lines. All computation times were obtained using Dassl unless specified otherwise.

The third benchmark assesses how computation time scales with the number of components that are connected to a single zone, since detailed zone models may contain a large number of components. A polygon-shaped zone with n sides is therefore modelled. The zone consists of n walls, a floor and a ceiling that contains a window. This model was simulated for increasing values of n . Figure 5d shows that normalised ODE evaluation time and the number of ODE evaluations again rise linearly with the problem size. The integrator time seems to rise less than linearly with the number of walls. Figure 5e however illustrates that this is caused by the fact that a relatively large number of state variables originate in the floor and ceiling, which offsets the curve. There does exist a quadratic relation between the integrator time and the number of states. Figure 5f uses benchmark three to compare the normalised ODE evaluation time of the default IDEAS implementation (ODE1) with an implementation that uses the design choices from Section 4.2.1 (ODE2). The old design choices would have caused an ODE evaluation time that scales almost quadratically with the number of components.

These results indicate that computation time for evaluating the ODE equations scales linearly with the model size. Dassl integration time scales more than quadratically for models with different excitations, while Euler integration time scales linearly with the problem size. Non-optimised design choices lead to an ODE evaluation time that scales quadratically instead of linearly with the number of zone components.

We therefore conclude that the variable time step solver Dassl can be used for relatively small and simple problems. For larger problems Euler integration can be faster, depending on the model time constants, which allows large IDEAS models to be solved with a computation time that scales linearly with problem size. These conclusions are in line with (Jorissen, Wetter, and Helsen 2015) where Euler integration was orders of magnitude faster than Dassl when simulating a complex building model using IDEAS.

5. Verification

The IDEAS building library is verified using several approaches to ensure consistency and correctness of the library implementation and results. Firstly, we verify the code by checking conservation of energy during simulation and verify consistency of the results with earlier code versions through the use of unit tests during development. These checks help to detect implementation errors. Secondly, IDEAS simulation results are compared with other tools using IEA EBC BESTEST. This check helps identifying implementation errors or model inaccuracies. Finally, the IEA EBC Annex 58 twin house experiments are used to compare IDEAS results against measurement data, which provides an estimate of the absolute accuracy of the library.

5.1. Individual Model Verifications

IDEAS allows verifying conservation of energy. When this option is enabled, the energy flows across the outer component boundaries are integrated and added to the internal energy of all components and zones. This sum should be a constant value when all energy flows are accounted for. In a 9-zone house model this check leads to an accumulated error in the order of 10^{-4} J for a simulation of one week. The total energy content of the building is in the order of 10^{10} J. Based on these two figures the precision can be estimated to be in the order of 10^{-14} , which is reasonably close to the machine precision of 10^{-16} . Therefore this error is assumed to originate from the numerical integrator or other numerical computations.

To guarantee consistency during model development, the unit test methodology from the IEA EBC Annex 60 library is used (Wetter et al. 2015a). The IDEAS library contains example models that test the building components. These examples include 1) tests of individual components, such as the zone or shading models, 2) tests that demonstrate the difference in results between two model options or 3) tests of groups of models (e.g. the BESTEST models). A unit test that stores the numerical results is added for each example. Whenever changes are made to the library the unit tests are run and the numerical results are compared with the old results. If the results change or when the model produces errors, the developer has to inspect and fix or accept the differences before the changes are added to the library.

5.2. BESTEST

The thermal building component models are verified using the BESTEST (short for ‘building energy simulation test and diagnostic method’). (Judkoff and Neymark 1995; ASHRAE 2011) The method implies an inter-model comparison between simulation tools for a set of test cases focusing on peak heating and cooling loads, annual heating and cooling demands and thermal comfort assessment.

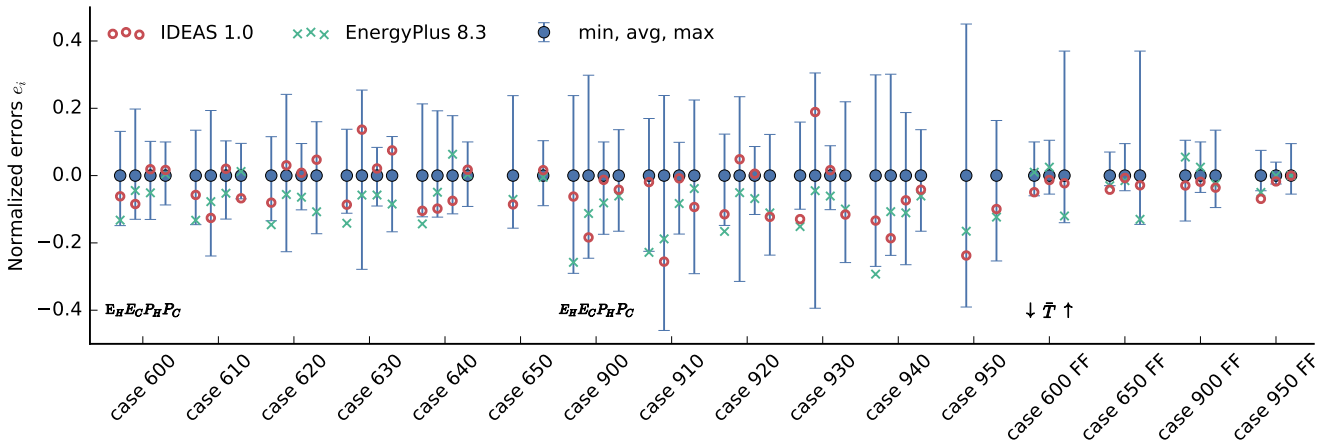


Figure 6.: Comparison of IDEAS (o) BESTEST results with EnergyPlus (x) and the minimum and maximum (error bars) and mean (•) results of other building energy simulation tools. For series 600 and 900: the normalized annual heating loads E_H , the normalized annual cooling loads E_C , the normalized peak hourly averaged heating loads P_H and normalized peak hourly averaged cooling loads P_C are presented. For four free float (‘FF’) cases the normalized minimum (\downarrow), average (\bar{T}) and maximum (\uparrow) temperatures are reported.

In this work only the *basic* test cases as described in (ASHRAE 2011) are discussed. Here, the total annual required energy for heating E_h and cooling E_c and peak heating P_h and cooling P_c loads are compared for a series of low mass one-zone buildings (600 series) and a series of heavy mass buildings (900 series) where strict temperature boundaries are tracked by an ideal heating and cooling system. Moreover free float minimum (\downarrow), average (\bar{T}) and peak (\uparrow) temperatures [°C] are reported for a subset of these cases indicated with ‘FF’.

Figure 6 presents a comparison of these results z_i between IDEAS 1.0, EnergyPlus 8.3.0-b45b06b780 and the minimum, mean (\bar{z}_i) and maximum of the results of ASHRAE Standard 140 (ASHRAE 2011) extended with BLAST 3.0-334 and DOE2.1E as reported by Henninger and Witte (2015). EnergyPlus results are reported since it is a well established tool and since recent results are available. The results are presented as relative errors e_i , which are defined as $e_i = (z_i - \bar{z}_i) / \bar{z}_i$ where $\bar{z}_i = \bar{z}_i$ for series 600 and 900 and $\bar{z}_i = 20K$ for the free float cases.

Overall, the results of IDEAS are within the range of reference results obtained by EnergyPlus and the other building energy simulation tools. The results fall outside the error bars three times (cases 930, 650 FF and 950 FF). For these cases a similar or stronger excursion of the error bars is observed for EnergyPlus. The peak heating and cooling loads are often below the reference result average. However, the results of EnergyPlus are on average even lower such that we consider these results acceptable. Only for case 630 and for case 930 IDEAS computes a total cooling load that is significantly larger than the mean and EnergyPlus results. Since these results are still well within the error bars, they are considered acceptable.

5.3. Twin Houses

Given the limitations of an inter-model comparison and the theoretical nature of the above 600- and 900-series of BESTEST, the thermal building models are additionally verified with the Twin House experiment conducted in IEA EBC Annex 58. The method implies a comparison of simulated temperature profiles and heating loads obtained by the multi-zone model against the measurements of the Twin House experiment. Firstly the experiment is described, then the model implementation is discussed and finally selected results are presented. The model implementation is available in package IDEAS.Examples.TwinHouses.

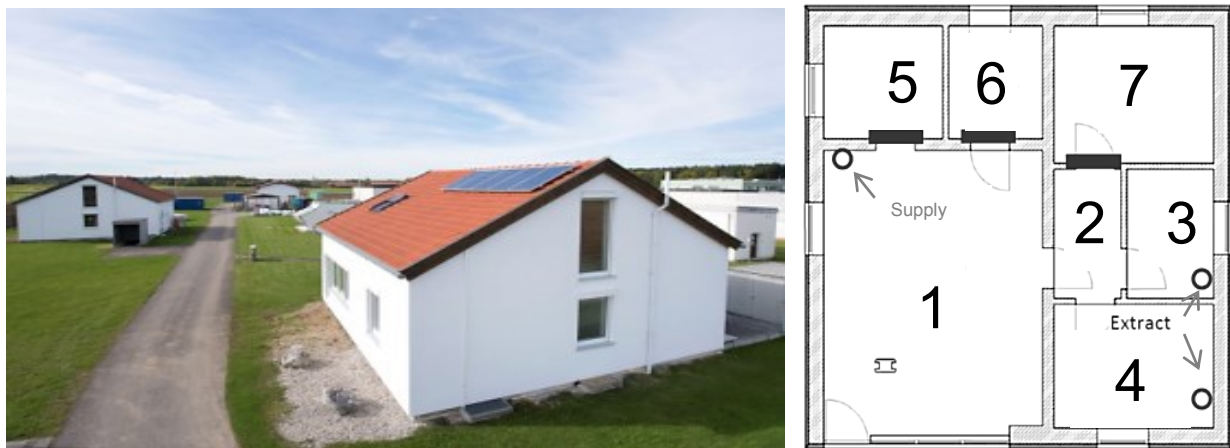


Figure 7.: Image (left) and floor plan (right) of the Twin Houses (Reynders 2015)

5.3.1. Experiment Description

This section gives a brief overview of the experiment description. A full description together with the available data files can be downloaded from <http://dx.doi.org/10.15129/94559779-e781-4318-8842-80a2b1201668>. The considered experiments are conducted in the N2 building at the Fraunhofer institute in Holzkirchen (Figure 7).

For the experiment only the ground floor of the building was used, while the attic and cellar are heated to obtain a constant temperature. The ground floor consists of 7 zones (Figure 7) and is divided into 2 parts, i.e. the controlled zones and the boundary zones. The controlled zones consist of the 4 south-oriented rooms: the living room (1), the corridor (2), the bathroom (3) and the bedroom (4). The North-oriented rooms, i.e. the kitchen (5), the doorway (6) and the parent bedroom (7) are referred to as the boundary zones since they are heated to a constant set-point temperature equal to that of the attic and the cellar. The doors between the controlled and boundary zones have been sealed. In contrast, the doors between the controlled zones are open to allow for a strong mixing of the indoor air.

The building has a ground floor area of 100 m² and is constructed and insulated in line with the German energy code (EnEV 2009). The outer walls are exterior insulated brick walls with U-values between 0.20 W/(m²K) and 0.27 W/(m²K). The windows have double-pane thermally insulating glazing with a U-value of 1.2 W/(m²K) and g-value of 0.57.

A balanced ventilation system is used for the controlled zones. A constant air flow rate of 60 m³/h is supplied to the living room and extracted from the bathroom (30 m³/h) and the bedroom (30 m³/h).

The air-tightness of the dwelling was measured using a blower-door test, resulting in an n₅₀-value of 1.67 ACH. A more detailed description of the building and its thermal properties is presented by Strachan et al. (2016).

Experiment 2 for building N2 is considered in this work. For this experiment, the full test sequence consists of a combination of a constant temperature initialisation period, free-floating temperature periods, constant temperature periods, and randomly ordered logarithmically distributed binary sequences.

The used temperature sensors have accuracies between ±0.04 K and ±0.14 K and their measurements are assumed to be correct in the following discussion.

5.3.2. Model Implementation

The building is implemented using a 7-zones model. The thermal properties and building geometry are implemented according to the modelling specifications provided in the common exercise of Subtask 4 of the IEA EBC Annex 58 (Strachan et al. 2016). Some physical phenomena, such

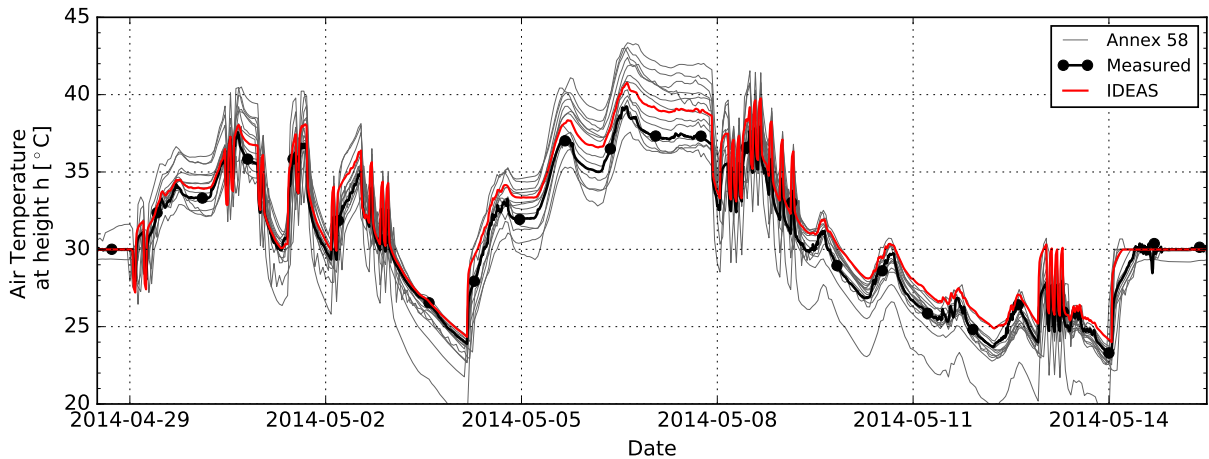


Figure 8.: Comparison of measured living-room air temperature at a height of 110 cm (‘Measured’) and the simulated air temperature obtained for IDEAS (‘IDEAS’) and in the IEA EBC Annex 58 common exercise (‘Annex 58’).

as thermal bridges and temperature driven inter-zone airflow through open doors, are not yet integrated into IDEAS v1.0 and are therefore not simulated in the model. The modelled airflow between the rooms of the south zones only includes the flow rates induced by mechanical ventilation.

The heating system consists of electric resistance heaters of which the electrical power use $P(t)$ was measured during the experiments. The electric heaters are modelled using Equations (27)-(28)

$$\dot{Q}_{con,ext}(t) = (1 - f_{rad})P(t) \quad (27)$$

$$\dot{Q}_{rad,ext}(t) = f_{rad}P(t) \quad (28)$$

where the radiant fraction f_{rad} equals 0.3 as prescribed in the experiment description. The heating system control for the controlled zones is split into 4 periods in which sequentially the measured indoor temperature and the heat input are tracked. During constant temperature periods, the measured air temperature is tracked using a proportional controller. Since the measured temperatures are tracked, the measured and computed heating powers can be compared during these periods. During other periods, $P(t)$ equals the measured heating power and the measured and computed air temperatures are compared. The north zones (zones 5-7) use a proportional controller with a set-point temperature of 22 °C.

As prescribed, an additional convective heat loss is introduced in zone 5 (kitchen), which results from an uninsulated ventilation duct that passes through the room. The heat flow rate is computed from the measured flow rate and inlet and outlet temperatures and averages -37 W.

5.3.3. Verification Results

Figure 8 and Figure 9 show verification results of the second experiment of building N2. These figures are only shown for the living room since it contains the most monitoring equipment. Rooms 2, 3 and 4 display mean absolute temperature errors of 1.48 K, 0.79 K and 0.74 K compared to 0.49 K for the living room. These larger errors may be explained by the fact that IDEAS currently only simulates forced convection between zones and no natural convection through open doors. This hypothesis is mainly supported by the time lag in the simulated temperature variations of zone 2 compared to the temperature variations in zone 1. This time lag is not visible in the measured data, suggesting a higher thermal coupling in reality.

Figure 8 compares the living room air temperature to the measured indoor air temperature at

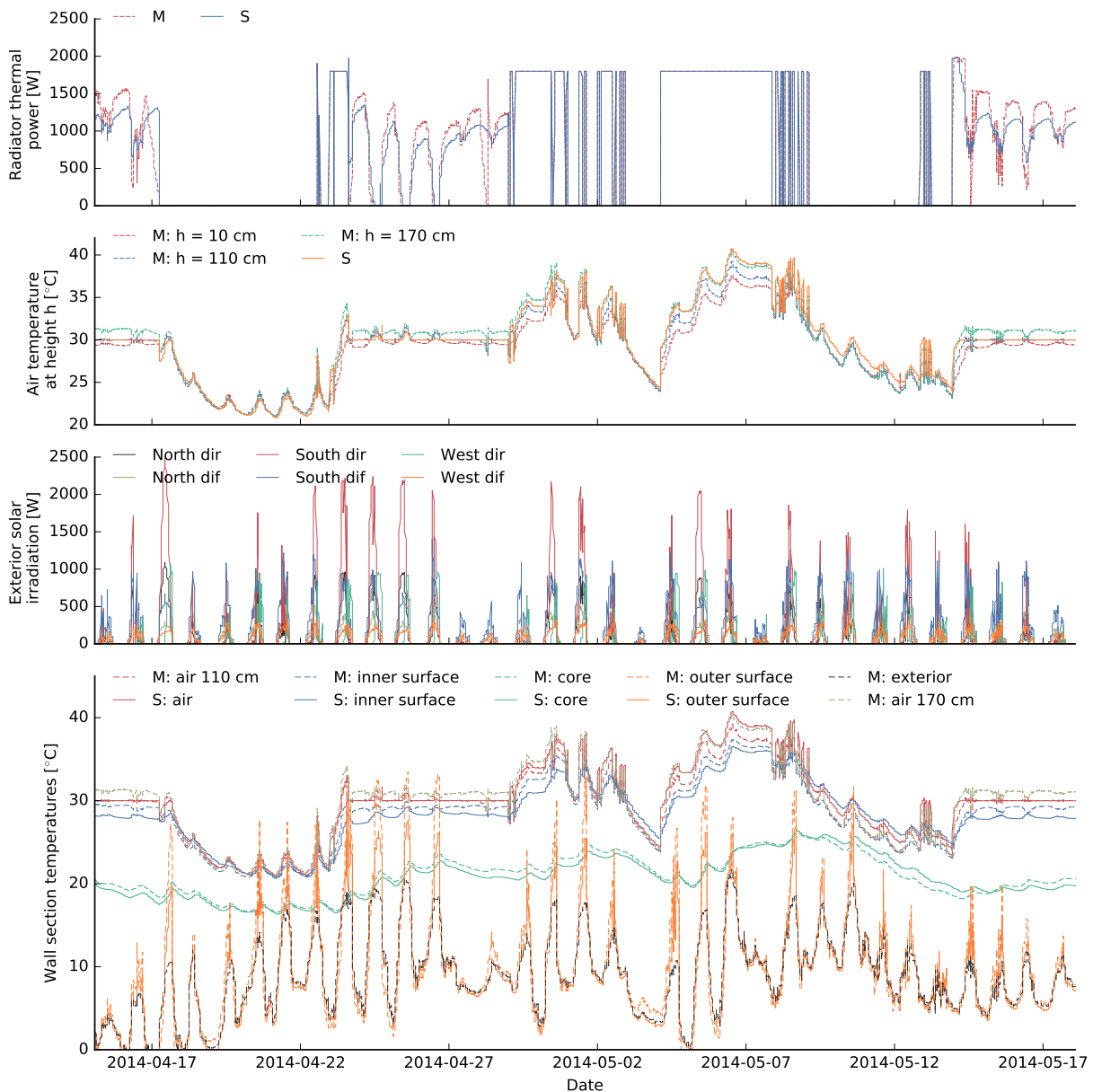


Figure 9.: Verification results of Twin House building N2 experiment 2 for the living room. Results denoted using ‘M’ and ‘S’ respectively correspond to measured or simulated results. The top subplot displays the measured and simulated radiator thermal power. The second subplot shows the well mixed simulated air temperature and the measured air temperature at three heights. Direct (dir) and diffuse (dif) solar irradiations of the door and south or west oriented windows are indicated in the third subplot. The bottom subplot contains detailed measurement and simulation data for the west oriented outer wall of the living room.

a height of 110 cm above the floor and to the results obtained by other simulation tools in the context of Subtask 4 of the IEA EBC Annex 58. Results are shown between April 28th and May 14th since in Figure 9 the largest deviations were obtained during this period. With a root mean squared error (RMSE) of 1.21°C compared to the measured data, the IDEAS building model is well within the range of the other simulation tools (Strachan et al. 2016). For the other tools RMSE values are between 0.62°C and 2.62°C with an average of 1.74°C. Unfortunately, a comparison to specific modelling tools is not possible as the Annex 58 results are treated anonymously.

Figure 9 allows a more detailed analysis of the experiment results. The first and second subplots respectively show heating powers and zone air temperatures. Note that the heating power was not logged for a few hours on the 17th of April and therefore the first and last measurement points are linearly interpolated on the figure during this period. The third subplot shows the computed diffuse and direct solar irradiation, which is an important input to the model. The last subplot shows a detailed comparison of measurements and simulations along the cross-section of the west oriented outer wall of the living room.

During the free float period (April, 17-22), the house thermal dynamics are well predicted with errors of about 0.5 K. The first subplot shows that the heating power requirements are underestimated (11-16 %) when tracking the set point temperature. Conversely, when tracking the heating power, the zone temperature exceeds the measured temperature with maximum temperature deviations of 2.6 °C. In the second free float interval (May, 9-12) the temperature dynamics are also correctly predicted, but a fixed temperature difference exists since the initial temperature of this period is overestimated. A systematic temperature overestimation is also seen during the constant heating period starting on May 4th. All these effects may be caused by the simplified well-mixed air model that is used in the simulations. This model may not be sufficiently accurate since the measurements show strong stratification with high temperatures at the top of the room. This concentrates heat dissipation at the top of the room and causes relatively low temperatures in the lower parts of the room, also after the heating period has passed. Moreover, the temperature at a height of 110 cm may not longer represent the average air temperature in the room, as assumed in the model comparison.

The outer wall cross section temperatures are well predicted although relatively large outside wall surface temperature deviations are observed during periods of high solar irradiation, which suggests that the outside heat transfer models could be improved. The simulated inside wall surface temperature is lower than the measurements during constant temperature periods. This may be caused by the location of the sensor, which is higher than the temperature sensor used for controlling the air temperature.

Although this verification shows that IDEAS is able to correctly predict most of the thermal dynamics and is well within the range of other BES tools, further detailed measurements are needed to clarify the origin of the deviations with more certainty. More detailed zone air models and inter-zonal air exchange could further improve the results. Nonetheless, it is found difficult to pinpoint specific model deficiencies based on measurements on building level.

6. Conclusion

This paper presents the building models of IDEAS v1.0, an open source Modelica library. The main physical modelling principles are explained and corresponding equations are presented. Usability aspects are discussed, which demonstrate practical usability of IDEAS. The contribution of IDEAS is that it provides transparent and flexible building models by exploiting the Modelica language. This equation-based approach facilitates efficient implementations of optimal control and other optimisation problems, as was demonstrated in earlier work. Furthermore, since Modelica is a generic simulation language, other systems such as HVAC, advanced controllers, district heating networks or electrical grids can be coupled in a single simulation tool.

In the second part of the paper, specific design choices of the library are motivated and checked using benchmarks showing that the computation time of IDEAS scales linearly to quadratically with the model size depending on the used solver. Especially the steps taken to avoid algebraic loops proved to significantly reduce the computation time. Implicit variable time-step solvers such as Dassl can be used for small problems, while explicit integrators such as Euler are suggested for

large problems.

Finally, the library is verified using several approaches. Both an inter-model comparison using the IEA EBC BESTEST and a verification against measured data from the Twin House experiments show that model accuracy falls within the range of other BES models. In the basic IEA EBC BESTEST test cases IDEAS falls beyond the bounds only three times. However, similar results are observed for EnergyPlus 8.3. For the IEA EBC Annex 58 twin house experiment 2 of building N2, IDEAS compares well to other building simulation tools. Some deviations are observed that may be explained by the use of a well-mixed zone air model, which is unable to model stratification caused by the heating system. To conclude, it is difficult to pinpoint model deficiencies based on whole-building measurements. More detailed measurements and further analysis may pinpoint which refinements are useful, which is highly influenced by the types of energy systems used, and thus the application considered. Low-exergy systems are less prone to stratification and should therefore have less deviations.

Acknowledgements

The IDEAS library incorporates models from the IEA EBC Annex 60 project, an international project conducted under the umbrella of the International Energy Agency (IEA) within the Energy in Buildings and Communities (EBC) Programme. Annex 60 develops and demonstrates new generation computational tools for building and community energy systems based on Modelica, Functional Mockup Interface and BIM standards. The authors acknowledge the contributions of Roel De Coninck as part of his joint PhD between KU Leuven and 3E.

Funding

This work was supported by the Agency for Innovation by Science and Technology in Flanders (IWT); under Grant 131012; and IWT and the WTCB in the frame of the IWT-VIS Traject SMART GEOTHERM. This project receives the support of the European Union, the European Regional Development Fund ERDF, Flanders Innovation & Entrepreneurship and the Province of Limburg.

References

- ASHRAE. 1997. *ASHRAE handbook: Fundamentals (SI edition)*. Atlanta: American Society of Heating, Refrigerating and Air Conditioning Engineers.
- ASHRAE. 2011. *ANSI/ASHRAE Standard 140-2011 - Standard Method of Test for the Evaluation of Building Energy Analysis Computer Programs*. RFC.
- Awbi, H.B., and a. Hatton. 1999. "Natural convection from heated room surfaces." *Energy & Buildings* 30 (3): 233–244, doi:10.1016/S0378-7788(99)00004-3.
- Baetens, R., R. De Coninck, J. Van Roy, B. Verbruggen, J. Driesen, L. Helsen, and D. Saelens. 2012. "Assessing electrical bottlenecks at feeder level for residential net zero-energy buildings by integrated system simulation." *Applied Energy* 96: 74–83, doi:10.1016/j.apenergy.2011.12.098.
- Baetens, Ruben. 2015. "On externalities of heat pump-based low-energy dwellings at the low-voltage distribution grid." PhD diss., KU Leuven.
- Baetens, Ruben, Roel De Coninck, Filip Jorissen, Damien Picard, Lieve Helsen, and Dirk Saelens. 2015. "OPENIDEAS - An Open Framework for Integrated District Energy Simulations." In *14th Conference of International Building Performance Simulation Association*, edited by Jyotirmay Mathur and Vishal Garg, Hyderabad, 347–354. International Building Performance Simulation Association.
- Baetens, Ruben, and Dirk Saelens. 2016. "Modelling uncertainty in district energy simulations by stochas-

- tic residential occupant behaviour.” *Journal of Building Performance Simulation* 9 (4): 431–447, doi:10.1080/19401493.2015.1070203.
- Björnell, Niclas, Axel Bring, Lars Eriksson, Pavel Grozman, Magnus Lindgren, Per Sahlin, Alexander Shapovalov, Bris Data AB, and Mika Vuolle. 1999. “IDA Indoor Climate and Energy.” In *Proceedings of the 6th IBPSA Conference*, Kyoto, Japan, 1035–1042.
- Carroll, Joseph A. 1980. “An “MRT Method” of Computing Radiant Energy Exchange in Rooms.” In *Systems Simulation and Economic Analysis*, San Diego, 343–360.
- Crawley, Drury B., Jon W. Hand, Michaël Kummert, and Brent T. Griffith. 2008. “Contrasting the capabilities of building energy performance simulation programs.” *Building and Environment* 43 (4): 661–673, doi:10.1016/j.buildenv.2006.10.027.
- Crawley, Drury B., Linda K. Lawrie, Frederick C. Winkelmann, W. F. Buhl, Y. Joe Huang, Curtis O. Pedersen, Richard K. Strand, et al. 2001. “EnergyPlus: creating a new-generation building energy simulation program.” *Energy & Buildings* 33: 319–331, doi:10.1016/S0378-7788(00)00114-6.
- De Coninck, R., R. Baetens, D/ Saelens, A. Woyte, and L. Helsen. 2014. “Rule-based demand-side management of domestic hot water production with heat pumps in zero energy neighbourhoods.” *Journal of Building Performance Simulation* 7 (4): 271–288, doi:10.1080/19401493.2013.801518.
- De Coninck, Roel, and Lieve Helsen. 2016. “Practical implementation and evaluation of model predictive control for an office building in Brussels.” *Energy & Buildings* 111: 290–298, doi:10.1016/j.enbuild.2015.11.014.
- De Coninck, Roel, Fredrik Magnusson, Johan Åkesson, and Lieve Helsen. 2016. “Toolbox for development and validation of grey-box building models for forecasting and control.” *Journal of Building Performance Simulation* 9: 288–303, doi:10.1080/19401493.2015.1046933.
- De Jaeger, Ina, Glenn Reynders, and Dirk Saelens. 2017. “Impact of spatial accuracy on district energy simulations.” *Energy Procedia* 132 (Supplement C): 561 – 566, doi:https://doi.org/10.1016/j.egypro.2017.09.741. 11th Nordic Symposium on Building Physics, NSB2017, 11-14 June 2017, Trondheim, Norway.
- Fanger, P.O. 1970. *Thermal comfort: Analysis and applications in environmental engineering*. Copenhagen: Danish Technical Press.
- Finlayson, Elizabeth U., Dariush K. Arasteh, Charlie Huizenga, Michael D. Rubin, and M. Susan Reilly. 1993. *Window 4.0: Documentation of Calculation Procedures*. Berkeley: Lawrence Berkeley Laboratory.
- Henninger, Robert H., and Michael J. Witte. 2015. *EnergyPlus 8.3.0-b45b06b780 Testing with Building Thermal Envelope and Fabric Load Tests from ANSI/ASHRAE Standard 140-2011*. Technical report. U.S. Department of Energy Energy. [Online; visited 30-March-2017], https://energyplus.net/sites/all/modules/custom/nrel_custom/eplus_files/current_testing_reports/ASHRAE140-Envelope-8.3.0-b45b06b780.pdf.
- Hollands, K. G. T., G. D. Raithby, and L. Konicek. 1975. “Correlation Equations for Free Convection Heat Transfer in Horizontal Layers of Air and Water.” *International Journal of Heat and Mass Transfer* 18: 879–884, doi:http://dx.doi.org/10.1016/0017-9310(75)90179-9.
- Jorissen, F., W. Boydens, and L. Helsen. 2017. “Validated air handling unit model using indirect evaporative cooling.” *Journal of Building Performance Simulation* doi:10.1080/19401493.2016.1273391. Advance online publication.
- Jorissen, Filip, and Lieve Helsen. 2016. “Towards an Automated ToolChain for MPC in Multi-zone Buildings.” In *International High Performance Buildings Conference*, West-Lafayette, Indiana, Paper 202.
- Jorissen, Filip, Michael Wetter, and Lieve Helsen. 2015. “Simulation Speed Analysis and Improvements of Modelica Models for Building Energy Simulation.” In *11th International Modelica Conference*, edited by Peter Fritzson and Hilding Elmqvist, Paris, 59–69. Modelica Association.
- Judkoff, R., and J. Neymark. 1995. *International Energy Agency building energy simulation test (BESTEST) and diagnostic method*. Technical report. Golden, CO: National Renewable Energy Laboratory (NREL).
- Jürges, W. 1924. “Der Wärmeübergang an einer ebenen Wand.” *Beihefte zum Gesundheits-Ingenieur* 19 (1).
- Klein, Sanford A. 2010. *TRNSYS 17, A Transient System Simulation Program*. Madison: Solar Energy Laboratory, University of Wisconsin.
- Liesen, Richard J., and Curtis O. Pedersen. 1997. “An Evaluation of Inside Surface Heat Balance Models for Cooling Load Calculations.” *ASHRAE Transactions* 3 (103): 485–502.
- Müller, D., M. Lauster, A. Constantin, M. Fuchs, and P. Remmen. 2016. “AIXLIB – An Open-Source Modelica Library Within the IEA-EBC Annex 60 Framework.” In *BauSIM*, edited by John Grunewald,

- Clemens Felsmann, Andreas Nicolai, and Joachim Seifert, Dresden, 3–9.
- Nytsch-Geusen, Christoph, Jörg Huber, Manuel Ljubijankic, and Jörg Rädler. 2013. “Modelica BuildingSystems- eine Modellbibliothek zur Simulation komplexer energietechnischer Gebäudesysteme.” *Bauphysik* 35 (1): 21–29.
- Picard, Damien, Filip Jorissen, and Lieve Helsen. 2015. “Methodology for Obtaining Linear State Space Building Energy Simulation Models.” In *11th International Modelica Conference*, edited by Peter Fritzson and Hilding Elmqvist, Paris, 51–58. Modelica Association and Linköping University Electronic Press.
- Picard, Damien, Maarten Sourbron, Filip Jorissen, Zdeněk Váňa, Jiří Cigler, Lukáš Ferkl, and Lieve Helsen. 2016. “Comparison of Model Predictive Control Performance Using Grey-Box and White-Box Controller Models of a Multi-zone Office Building.” In *International High Performance Buildings Conference*, West-Lafayette, Indiana, Paper 203.
- Protopapadaki, Christina, and Dirk Saelens. 2017. “Heat pump and PV impact on residential low-voltage distribution grids as a function of building and district properties.” *Applied Energy* 192: 268–281, doi:10.1016/j.apenergy.2016.11.103.
- Reynders, G., J. Diriken, and D. Saelens. 2014. “Quality of grey-box models and identified parameters as function of the accuracy of input and observation signals.” *Energy & Buildings* 82: 263–274, doi:10.1016/j.enbuild.2014.07.025.
- Reynders, G., T. Nuytten, and D. Saelens. 2013. “Potential of structural thermal mass for demand-side management in dwellings.” *Building and Environment* 64: 187–199, doi:10.1016/j.buildenv.2013.03.010.
- Reynders, Glenn. 2015. “Quantifying the impact of building design on the potential of structural storage for active demand response in residential buildings.” PhD diss., KU Leuven.
- Reynders, Glenn, Razakaniaina Ando Ludovic Andriamamonjy, Ralf Klein, and Dirk Saelens. 2017. “Towards an IFC-Modelica Tool Facilitating Model Complexity Selection for Building Energy Simulation.” In *15th International conference of the international building physics association: Building Simulation 2017*, .
- Strachan, Paul, Katalin Svehla, Matthias Kersken, and Ingo Heusler. 2016. *IEA EBC Annex 58 - Report of Subtask 4a: Empirical validation of common building energy simulation models based on in situ dynamic data*. Leuven: KU Leuven.
- van der Heijde, B., A. Aertgeerts, and L. Helsen. 2017. “Modelling steady-state thermal behaviour of double thermal network pipes.” *International Journal of Thermal Sciences* 117 (Supplement C): 316 – 327, doi:https://doi.org/10.1016/j.ijthermalsci.2017.03.026.
- van der Heijde, B., M. Fuchs, C. Ribas Tugores, G. Schweiger, K. Sartor, D. Basciotti, D. Müller, C. Nytsch-Geusen, M. Wetter, and L. Helsen. 2017. “Dynamic equation-based thermo-hydraulic pipe model for district heating and cooling systems.” *Energy Conversion and Management* 151 (Supplement C): 158 – 169, doi:https://doi.org/10.1016/j.enconman.2017.08.072.
- Wetter, Michael. 2004. “Simulation-Based Building Energy Optimization.” PhD diss., University of California.
- Wetter, Michael, Marco Bonvini, and Thierry S. Noudui. 2016. “Equation-based languages – A new paradigm for building energy modeling, simulation and optimization.” *Energy & Buildings* 117: 290–300, doi:10.1016/j.enbuild.2015.10.017.
- Wetter, Michael, Marcus Fuchs, Pavel Grozman, Lieve Helsen, Filip Jorissen, Moritz Lauster, Müller Dirk, et al. 2015a. “IEA EBC Annex 60 Modelica Library - An International Collaboration to Develop a Free Open-Source Model Library for Buildings and Community Energy Systems.” In *14th Conference of International Building Performance Simulation Association*, edited by Jyotirmay Mathur and Vishal Garg, Hyderabad, 395–402. International Building Performance Simulation Association.
- Wetter, Michael, Thierry S. Noudui, David Lorenzetti, Edward A. Lee, and Amir Roth. 2015b. “Prototyping the Next Generation EnergyPlus Simulation Engine.” In *14th Conference of International Building Performance Simulation Association*, edited by Jyotirmay Mathur and Vishal Garg, Hyderabad, 403–410. International Building Performance Simulation Association.
- Wetter, Michael, and Christoph van Treeck. 2017. *IEA EBC Annex 60: New Generation Computing Tools for Building and Community Energy Systems*. The Regents of the University of California and RWTH Aachen University.
- Wetter, Michael, Wangda Zuo, Thierry S. Noudui, and Xiufeng Pang. 2014. “Modelica Buildings library.” *Journal of Building Performance Simulation* 7 (4): 253–270, doi:10.1080/19401493.2013.765506.
- Wetter, Micheal, and Christophe Haugstetter. 2006. “Modelica Versus Trnsys – A Comparison Between an Equation-Based and a Procedural Modeling Language for Building Energy Simulation.” In *Proc. of the 2nd SimBuild Conference*, Cambridge, MA, USA, 262–269.

Wright, John L. 1996. "A Correlation to Quantify Convective Heat Transfer Between Vertical Window Glazings." *ASHRAE Transactions* 102 (1): 940–946, doi:10.1017/CBO9781107415324.004.