Sub-zonal computational fluid dynamics in an object-oriented modelling framework

Marco Bonvini[1] (✉), **Mirza Popovac**[2], **Alberto Leva**[3]

1. Lawrence Berkeley National Laboratory, 1 Cyclotron Road, Berkeley, CA 94704, USA
2. Austrian Institute of Technology, Energy Department, Giefinggasse 2, 1210 Vienna, Austria
3. Dipartimento di Elettronica Informazione e Bioingegneria, Politecnico di Milano, Via Ponzio 34/5, 20133 Milano, Italy

Received: 7 October 2013
Revised: 2 January 2014
Accepted: 7 January 2014

Abstract

Airflow modelling is of fundamental importance for evaluating ventilation performance and energy consumption in buildings, and various approaches to the problem—starting from purely empirical up to the CFD ones—have been proposed and evaluated in the past years. Moreover, since the ultimate goal is *whole building* modelling, airflow simulation needs coupling with Energy Simulation (ES), in order to assess the *overall* energy performance. Due to the substantial differences between the software employed for airflow and ES, co-simulation is very often felt as the only way to handle such a problem. For example, in recent years a lot of effort has been spent in to couple ES and CFD tools. This paper proposes an alternative, in the form of an approach for solving the Navier–Stokes equations in a general multi-domain modelling framework. Since co-simulation is not involved, the correctness of the numerical solution relies on a single solver, thus being *really* transparent to the analyst. This is a first step towards a whole building simulation tool embedded in a *unique* framework capable of performing energy analysis, computing airflows, and representing control systems.

1 Introduction

Energy-related building simulation is a long lasting research field, and in the last years new and important challenges have been emerging. Among said challenges, three are the major ones: dynamic simulation capability, a transition from "subsystem-specific" to *efficient* "whole-building" tools, and the possibility of simulating a project at each of its steps, so as to perform "scalable-detail" and "system-level" studies. The work presented herein is part of a wider research aiming at fulfilling the demands just mentioned, and concentrates on a specific aspect that—if not treated properly—can significantly impair the necessary modularity, namely the modelling of airflows.

The mentioned problem comes to be of interest in cases where a fully mixed approach—i.e., considering the air properties as uniform—is not adequate. The zonal approach can serve this purpose. It is based on the idea to subdivide a space into zones connected through hydraulic resistances following the idea of "electric equivalents". For such models modularisation is not a difficult issue. The cases where fully mixed models are inadequate are those that most frequently oblige to use co-simulation, where air volumes are modelled along the CFD approach, thus difficult to integrate with models of other types—especially if modularity is an issue (Arias 2006; Zhai 2003; Beausoleil-Morrison 2000).

The main contributions of this manuscript can be summarised as follows. A particular spatial discretisation of the Navier–Stokes equations is proposed, taking care of reducing the number of empirical parameters to a minimum. It is also illustrated how the proposed approach, inflected in the object-oriented (OO) framework, allows to obtain *modular* models, with a scalable level of complexity (e.g., for turbulence descriptions) and a standardised interface. Examples show that the obtained results are in reasonable accordance (of course as long as the introduced simplifications are valid) with those of fine-scale CFD models,

E-mail: mbonvini@lbl.gov

while simulation performance improvements are observed. In addition, the standardised interface allows to couple the obtained models with others, also from different physical domains, in a very straightforward manner.

A key aspect of this work is the use of an OO equation-based modelling language, namely Modelica (Mattsson et al. 1998). Models are written in a totally transparent manner with respect to the way the system of DAE (Differential and Algebraic Equations) obtained by connecting them, is solved. Modelica (and OO at large) models are a-causal, i.e., it is not necessary to specify at modelling time which variables will be inputs and which outputs. This allows to write models independently of how they will be assembled, or equivalently, of which type of boundary condition will be presented to them in the context of a larger model containing them. The interested reader can refer to this well organised book (Cellier and Kofman 2006).

All these features make OO modelling—and in particular the Modelica language—very suited for multi-domain simulations. Thanks to the modular structure paradigm offered by connectors, it is in fact possible to link models that belong to different domains (e.g. mechanical, thermal, as well as electrical and control) and simulate them together in a unique framework. For such a reason Modelica is a promising tool in the context of building energy simulation. Model libraries are already available to represent part of the building structure as well as HVAC plants and the control system (Wetter 2009; Wetter et al. 2011). Complementing such a *scenario* with models capable of accurately representing the temperature distribution and the airflow within rooms, is a step ahead in the direction of an effective integrated *whole* building simulation tool.

In synthesis, the purpose of this paper is to illustrate how the adoption of OO modelling, if suitably exploited, offers CFD-like simulation capabilities, at a level certainly adequate for energy-related system studies in buildings, without the need to use and coordinate different tools. Models are thus described in a single language, and solution issues are made transparent to the analyst. Apart from efficiency, this allows for great improvements as for model creation, maintenance, and above all readability. Note that this particular exploitation of OO modelling is definitely innovative: analogous works such as (Ljubijankic et al. 2011), for example, are still tied to co-simulation.

2 State of the art

Airflow models accounting for the thermodynamic state of the air are very important in the context of building simulation. Such models can help create a thermally comfortable environment—e.g., with a desired temperature or humidity distribution. Simulating the model allows to predictively assess the building performance at design time—in fact the scalable-detail paradigm proposed here permitting this at earlier design stages than CAD-based ones do—and therefore to adapt its envelope, as well as HVAC components and controls, to satisfy the required comfort criteria.

Representing the entire building with a suitable tool (or a suite of them) addressing all of its subsystems, is another widespread and well assessed idea (Crawley et al. 1997). This was demonstrated in the past e.g. by the merging of BLAST and DOE-2, two historical tools developed by the US government (Pedersen et al. 1997). At that time, the need to simulate the air motion inside buildings (e.g. rooms or atriums) in a fast and reliable way, became a relevant issue. One of the first attempts to face the problem of air motion and ventilation performance prediction in a simplified and thus computationally light manner was proposed in (Allard et al. 1990), introducing the so called "subzonal" models. The basic idea was to split the spatial domain in a coarse mesh of volumes, and then apply simplified relationships to compute the airflow field. The idea was introduced also in some commercial tools like COMIS (Ren and Stewart 2003). Further improvements, detailed in (Mora et al. 2003), were proposed to improve the behaviour of the models in some cases (e.g. forced convection) where the shortcomings of the mentioned approach were too significant. Since the improvements were based on empirical assumptions, however, they could not avoid some loss of generality and applicability.

A different approach, that proved to yield good results and was thus extensively adopted in the following years, was proposed in (Chen and Xu 1998). In that paper they present a zero equation turbulence model particularly suited for the context of building energy simulation. Some simplification (e.g. in the turbulence model) and the usage of quite coarse meshes, together with the increasing computational power, enlarged the penetration of CFD codes in the context of building simulation (Chen 2009). Such an approach is still used by the main Energy Simulation (ES) software that typically use co-simulation to couple the whole building model together with a CFD code simulating the contained air.

In recent years, thanks to the increase of computer performances as well as to the introduction of new architectures, new approaches have been studied. A relevant work in this context is the Fast Fluid Dynamics (FFD) proposed in (Zuo and Chen 2010). FFD uses a semi-lagrangian approach, that is natively parallelisable, thus can be significantly accelerated if implemented on parallel architectures such as Graphic Processing Unit (GPU) ones. The same authors also presented an improved version of their models, in which undesired numerical diffusion was reduced through higher-order discretisation technique (Zuo et al. 2010).

Despite coupling between CFD end ES is at present a standard practice, it is in general a complex and resource-consuming process, that can sometimes prove error-prone unless correctly managed to prevent erroneous behaviours, see e.g. (Zhai and Chen 2004). An interesting work about co-simulation in the context of buildings and HVAC simulation tools is (Trčka et al. 2009, 2010). The authors evidence how to manage in a consistent and correct way the simulations performed with the involved tools and their step size. Of course, as the number of players (i.e., the individual simulation tools) increases, the time spent for synchronisation and management becomes relevant, diminishing the overall simulation performance.

The presented *scenario* is correctly summarised by the sentences taken from (Sahlin et al. 2004) and marked as fairly broad consensus issues: "The existing simulators were too rigid in their structure to be able to accommodate the improvements and flexibility that would be called for in the future. Each added feature required a larger implementation effort than the previous one." and "A promising technology for the future were the general simulation methods offered by equation-based methods using program-neutral model descriptions and domain-independent solution methods". The same author in (Sahlin 2000) stresses the idea of replacing the present tools with equation-based ones, leading to a true whole building simulation tool.

To compare present industry-standard tools with an equation-based one, in (Wetter and Haugstetter 2006) the authors performed a comparison between TRNSYS and the preliminary version of the Modelica Buildings library (Wetter and Haugstetter 2006). The work evidences that the time spent in the model development phase using Modelica is five to tenfold less with respect to TRNSYS. The main drawback of Modelica, evidenced at that time, was the limited set of models representing the various HVAC components and its computing time. The first drawback is becoming less relevant because thanks to an increasing interest in this research field the preliminary version of the Buildings library has been regularly updated and supported (Wetter et al. 2011). Thanks to the modularity of the language, adding a new model in Modelica requires less effort than any other traditional tool (e.g. TRNSYS, EnergyPlus, etc.), to the advantage of re-use and models' diffusion. Also, the computational aspects are made transparent by the equation-based approach, as stated e.g. in (Sahlin et al. 2004). On the other hand, new challenges involving parallel architectures in the context of equation-based approach are promising research fields (Ostlund et al. 2010).

At present the Modelica Buildings library (Wetter et al. 2011), as well as the Modelica Standard Library, contain fluid flow models, but do not aim at describing airflow inside "large" spaces, or more in general, at any case where CFD like models would be the choice of election if "accurate enough" results are needed (see e.g. (Ljubijankic et al. 2011) where Modelica is coupled together with a CFD code).

Based on the *panorama* just sketched, and on previous results such as (Bonvini et al. 2012), this work provides the improvements described in the introduction, and discussed more in detail in the following sections.

3 Description of the model

3.1 Governing equations

The flow of Newtonian fluid can be described with the mass, energy and momentum conservation equations. This set of equations, commonly referred to as the Navier–Stokes (NS) ones, can be written as

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \mathbf{v}) = 0 \qquad (\textit{mass}) \qquad (1a)$$

$$\frac{\partial (\rho e)}{\partial t} + \nabla \cdot (\rho \mathbf{v} h) = \nabla \cdot (k \nabla T) \qquad (\textit{energy}) \qquad (1b)$$

$$\frac{\partial (\rho \mathbf{v})}{\partial t} + \nabla \cdot (\rho \mathbf{v}\mathbf{v}^{\mathrm{T}}) + \nabla p = \nabla \cdot (\mu \nabla \mathbf{v}) + \mathbf{f} \qquad (\textit{momentum}) \qquad (1c)$$

where the fluid flow quantities p, T, e, h are respectively the absolute pressure, absolute temperature, specific energy and specific enthalpy of the fluid, together with the fluid properties ρ, k and μ representing the density, thermal-conductivity and dynamic viscosity respectively, while $\mathbf{v}$ and $\mathbf{f}$ represent the fluid velocity and the *external* forces (e.g. gravity). Vector quantities are here denoted by bold letters, in order to distinguish them from scalar ones.

For fluid density ρ, the ideal gas relationship can be approximated with a linearised model, since the discrepancy between the ideal gas and the linearised model is very limited in the typical operating temperature and pressure range. This leads to

$$\rho = \frac{p}{R^* T} \qquad (\textit{ideal gas relationship}) \qquad (2a)$$

$$\rho = \rho_o + \frac{1}{R^* T_o} P - \frac{p_o}{R^* {T_o}^2}(T - T_o) \quad (\textit{linearised model}) \qquad (2b)$$

with R^* being the specific ideal gas constant, and ρ_o, p_o and T_o representing respectively the fluid density, absolute pressure and absolute temperature at the linearisation point, and $P = p - p_o$ being the *relative* pressure, used in the rest of the paper instead of the absolute one, to better exploit the available numeric precision.

Finally, to close the set of equations, the specific energy and specific enthalpy are expressed as

$$e = c_\mathrm{v} T \qquad (\textit{specific energy}) \tag{3a}$$

$$h = e + \frac{p}{\rho} \qquad (\textit{specific enthalpy}) \tag{3b}$$

where c_v is the specific heat capacity at constant volume, and a linear relationship between specific energy and absolute temperature is assumed.

3.2 Generic form of the governing equations

In order to present the implementation of the fluid flow equations in Modelica, we shall recast the set of governing equations into the generic Convection–Diffusion (CD) form, i.e.,

$$\underbrace{\frac{\partial \rho \Phi}{\partial t}}_{unsteady} + \underbrace{\frac{\partial \rho v_i \Phi}{\partial x_j}}_{convective} = \underbrace{\frac{\partial}{\partial x_i}\left(\Gamma_\Phi \frac{\partial \Phi}{\partial x_i}\right)}_{diffusive} + \underbrace{S_\Phi}_{source} \tag{4}$$

where the transported quantity is represented by the generic scalar Φ, with its diffusivity coefficient Γ_Φ, and its generation rate per unit volume S_Φ.

The idea is to approach the set of governing equations in a unified manner (see (Patankar 1980; Versteeg and Malalasekera 2007)). This is summarised in Table 1, showing how Eqs. (1a), (1b) and (1c) are represented using the generic CD Eq. (4) with a suitable choice of variables. For brevity and to lighten the notation, the mentioned table (as well as the rest of this paper) presents the equations in a 2D Cartesian coordinate system (i.e., the velocity $\mathbf{v} = (v_x, v_y)$, and the coordinates $\mathbf{x} = (x, y)$), but everything can be extended to 3D without loss of generality and in a straightforward manner.

Table 1 The terms in the generic form of the Convective–Diffusive unsteady equation

Generic form

$$\frac{\partial \rho \Phi}{\partial t} + \frac{\partial \rho v_x \Phi}{\partial x} + \frac{\partial \rho v_y \Phi}{\partial y} = \frac{\partial}{\partial x}\left(\Gamma_\Phi \frac{\partial \Phi}{\partial x}\right) + \frac{\partial}{\partial y}\left(\Gamma_\Phi \frac{\partial \Phi}{\partial y}\right) + S_\Phi$$

	Φ	Γ_Φ	S_Φ
Mass	1	0	0
Energy	$c_\mathrm{p} T$	k_eff	Q_source
x-momentum	v_x	μ_eff	$\frac{\partial P}{\partial x}$
y-momentum	v_y	μ_eff	$\frac{\partial P}{\partial y} - \rho g$

3.3 Turbulence model

In the simulation of a turbulent flow (which is the most common flow type of engineering importance), one has to account for the small scale interactions between fluid flow structures. For engineering applications this is typically done by enhancing the diffusion (which resembles extensive mixing caused by these interactions) through increased diffusivity coefficient. In the momentum equation this implies the introduction of the *turbulent viscosity* μ_T, in addition to the molecular viscosity of the fluid μ. By analogy to the molecular viscosity, the turbulent viscosity is defined through "characteristic length" and "velocity scales" (although these are the characteristics of the flow, and not the characteristics of the fluid), as described by given *turbulence model.*

Different turbulence models yield the characteristic length and velocity scales by solving additional (algebraic or differential) model equations. To keep the model complexity as low as possible we use a zero-equation turbulence model: no additional equations are solved, but rather algebraic expressions are used for defining the characteristic scales. Following Prandtl's mixing length idea, it is assumed that the turbulent viscosity varies linearly with the minimal distance from the wall y_w, as

$$\mu_\mathrm{T}^+ = \kappa y^+ \tag{5}$$

where $\mu_\mathrm{T}^+ = \mu_\mathrm{T} / \mu$ is the normalised eddy viscosity, $y^+ = \rho u_\tau y_\mathrm{w} / \mu$ is the normalised minimal distance from the wall, and the normalisation is performed using the friction velocity u_τ, while κ=0.41 is the Von Karman constant.

In this model, the minimum distance from the wall y_w represents the characteristic length scale, while the friction velocity u_τ is taken as the corresponding velocity scale. Based on the momentum boundary layer theory, the friction velocity is defined here with the velocity magnitude U calculated in the immediate wall vicinity $u_\tau = c_U U$, with the proportionality constant $c_U = 0.0945$ proposed by Chen and Xu (Chen and Xu 1998), and adopted as a standard practice for the HVAC simulations. It has to be noted that this simple model is derived for the equilibrium flow conditions, hence for the non-equilibrium flows (e.g. with stagnation point or strong recirculation) it can introduce a modelling error.

For the calculations of temperature distribution in wall-bounded flows, the same idea of describing turbulence effects by introducing the turbulent thermal conductivity k_T is used. The analogy between the momentum and thermal boundary layer states that the Prandtl number Pr defines the ratio between the momentum and thermal diffusivity $Pr = c_\mathrm{p}\mu/k$ (assumed to be constant for a specific fluid). In the same way is the turbulent conductivity k_T

expressed through the turbulent viscosity μ_T, and turbulent Prandtl number Pr_T, hence the modelling of turbulence effects in the energy equation is directly coupled to the turbulence modelling in the momentum equation.

Finally, as shown in Table 1, the turbulence model implementation is reduced to introducing into the generic CD Eq. (4) the *effective* viscosity μ_{eff} instead of the dynamic viscosity μ regarding the momentum Eq. (1c), and introducing the *effective* thermal conductivity k_{eff} instead of the thermal conductivity k regarding Eq. (1b), i.e.

$$\mu_{\text{eff}} = \mu + \mu_{\text{T}} = \mu + \rho\kappa u_\tau y_{\text{w}} \tag{6a}$$

$$k_{\text{eff}} = k + k_{\text{T}} = k + \frac{\rho\kappa c_{\text{p}} u_\tau y_{\text{w}}}{Pr_{\text{T}}} \tag{6b}$$

3.4 Near-wall treatment

In wall-bounded turbulent flows, the steepest variations in the distribution of flow variables (e.g. velocity, temperature etc.) occur in the immediate vicinity of walls, and the accuracy of the numerical predictions (the wall fluxes in particular) strongly depends on the accurate representation of this variation. A straightforward way to accurately represent this steep variation is to increase the spatial resolution in the wall-normal direction; however, this can be computationally very demanding. Since however a fully rigorous—thus very detailed—representation of the mentioned local steep variation is not relevant to catch energy-related facts at a system level, one can use a pre-defined logarithmic profile (log-law) which was proven to be valid for an attached plane flow with high Reynolds number. This yields

$$U^+ = \frac{1}{\kappa}\ln(Ey^+) \tag{7}$$

where $U^+ = U / u_\tau$ is the normalised velocity, and E =8.9 is the log-law constant.

The idea behind this near-wall treatment is to satisfy the log-law behaviour even in the situation when the wall-normal resolution is not sufficient to fully resolve the spatial variation of the flow characteristics in the near-wall region. To this purpose, the effective viscosity—derived using Eq. (7)—is imposed at the walls (whence the name *wall viscosity* μ_{w}) in order to provide for a suitable boundary condition in the momentum equation regardless of the spatial resolution in the near-wall region, obtaining

$$\mu_{\text{w}} = \begin{cases} \mu & \text{if } y^+ < 5 \\ \dfrac{\rho\kappa u_\tau y_{\text{w}}}{\ln(Ey^+)} & \text{if } y^+ > 30 \end{cases} \tag{8}$$

with the linear interpolation used in the y^+ range between 5 and 30. Clearly, using a boundary layer analogy, the same approach can be applied for the thermal equation as well, thus introducing the *wall conductivity*

$$k_{\text{w}} = \begin{cases} k & \text{if } y^+ < 5 \\ \dfrac{c_{\text{p}}\mu_{\text{w}}}{Pr_{\text{T}}} & \text{if } y^+ > 30 \end{cases} \tag{9}$$

4 Implementation in Modelica

4.1 The grid

The governing Eqs. (1a), (1b), (1c) need to be discretised over a spatial domain. The grid adopted here for that purpose is a structured and non-uniform one, as shown in Fig. 1 (left).

The result of such a representation (in the 2D case) is a grid of rectangular shaped boxes or Control Volumes (CV); extension to 3D is straightforward. The spatial domain is thus described by its dimensions (respectively base and height), while the grid is described by the number of CVs along the x and y axes (I and K) and the percentage of space occupied by the volumes contained into the grid. A grid composed by 5 × 5 volumes uniformly spaced is described by two vectors **X**=[0.2, 0.2, 0.2, 0.2, 0.2] and **Y**=[0.2, 0.2, 0.2, 0.2, 0.2]. If non-uniform these vectors became e.g. **X**=[0.1, 0.2, 0.4, 0.2, 0.1] and **Y**=[0.1, 0.1, 0.6, 0.1, 0.1]. The complexity in the implementation when using a non-uniform grid is practically the same as in the uniform case. The notion of "staggered grid" recalls that the CVs where the energy and mass preservation are discretised (usually called Pressure or P-cells) differ from the ones where x and y momentum are, as shown in Fig. 1 (right). For such a reason three different coordinate systems for univocally numbering the CVs have been adopted. The three systems are respectively for identifying scalar quantities inside P-cells, and the V_x, V_y air velocities, as shown in Fig. 2.

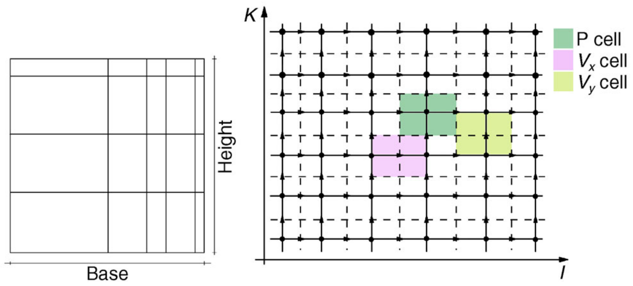


Fig. 1 (Left) non-uniform structured grid; (right) staggered grid for the discretisation of the NS equations

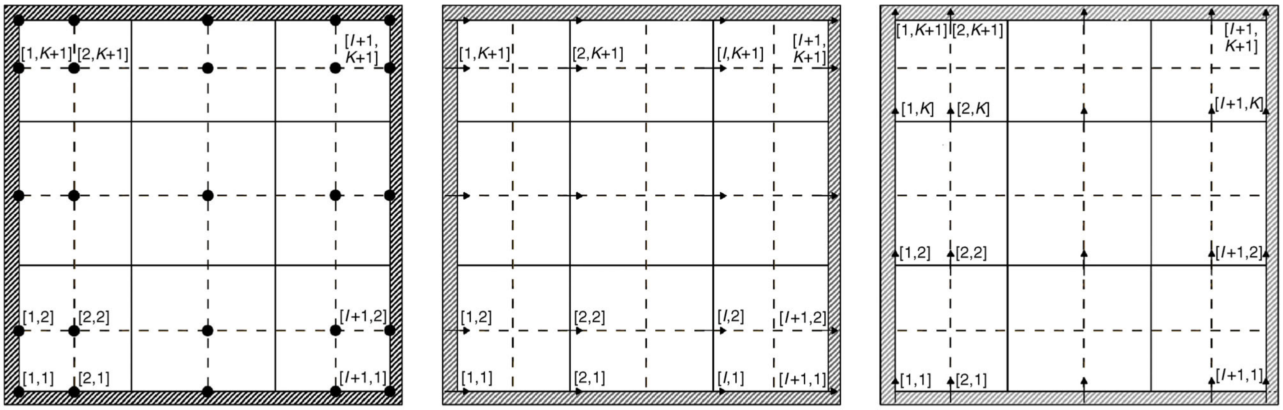


Fig. 2 Grid for the discretised version of the equations in Modelica: energy and mass (left), *x*-momentum (center) and *y*-momentum (right)

4.2 Domain boundaries

Boundary conditions, needed to solve the governing equations, are provided in terms of temperatures and air velocities on the domain boundaries (e.g., a wall surrounding the room). Such values can be assigned to a subset of variables located on the domain border, as shown in Fig. 2. Table 2 contains the subsets of variables that represent the boundaries.

Usually, the velocities on the boundaries are defined as null, except when describing inlets or outlets (e.g., for HVAC). Concerning the temperatures, in order to describe a room with a wall kept at a given temperature, it suffices to assign the desired temperature to each node in the boundary set (as defined in Table 3). On the contrary, an adiabatic wall should be described by imposing a null temperature gradient between adjacent cells (see Table 3).

Table 2 Set of variables on the domain boundaries

	Left	Right	Bottom	Top
T	$T[1,2,...K+1]$	$T[I+2,2,...K+1]$	$T[2,...I+1,1]$	$T[2,...I+1,K+2]$
V_x	$V_x[1,1,...K+2]$	$V_x[I+1,1,...K+2]$	$V_x[1,...I+1,1]$	$V_x[1,...I+1,K+2]$
V_y	$V_y[1,1,...K+1]$	$V_y[I+2,1,...K+1]$	$V_y[1,...I+2,1]$	$V_y[1,...I+2,K+1]$

Table 3 Boundary conditions for temperatures

	Left wall boundary
Fixed temperature	$T[1,2,...K+1] = T_{\mathrm{wall}}$
Adiabatic wall	$T[1,2,...K+1] = T[2,2,...K+1]$

4.3 Preservation equations

This section shows how mass, energy and momentum preservation equations have been discretised and implemented in Modelica. The starting point is the numerical integration with the finite volume approach of the CD Eq. (4), a well known subject in the context of fluid dynamics (see (Patankar 1980; Versteeg and Malalasekera 2007) for more information). The discretisation of such an equation leads to the following compact representation:

$$V_\rho \frac{\mathrm{d}\Phi_\mathrm{P}}{\mathrm{d}t} + a_\mathrm{P}\Phi_\mathrm{P} = a_\mathrm{E}\Phi_\mathrm{E} + a_\mathrm{W}\Phi_\mathrm{W} + a_\mathrm{N}\Phi_\mathrm{N} + a_\mathrm{S}\Phi_\mathrm{S} + S \tag{10}$$

where all the terms accounting for convective and diffusive exchanges are collected into the $a_{\mathrm{P,N,S,W,E}}$ coefficients (where N,S,W and E correspond to north, south, west and east), while the sources are merged into S. When taking into account the 2D case such coefficients are defined as

$$a_\mathrm{E} = D_\mathrm{e} A(|\, P_\mathrm{e}\, |) + \| -F_\mathrm{e}, 0 \| \tag{11a}$$

$$a_\mathrm{W} = D_\mathrm{w} A(|\, P_\mathrm{w}\, |) + \| F_\mathrm{w}, 0 \| \tag{11b}$$

$$a_\mathrm{N} = D_\mathrm{n} A(|\, P_\mathrm{n}\, |) + \| -F_\mathrm{n}, 0 \| \tag{11c}$$

$$a_\mathrm{S} = D_\mathrm{w} A(|\, P_\mathrm{s}\, |) + \| F_\mathrm{s}, 0 \| \tag{11d}$$

$$a_\mathrm{P} = a_\mathrm{W} + a_\mathrm{E} + a_\mathrm{N} + a_\mathrm{S} \tag{11e}$$

where $\| a,b \|$ is the maximum between a and b, while D and F are respectively the diffusion conductances and mass fluxes. Into Eq. (11) the Péclet numbers ($P_{\mathrm{n,s,w,e}}$) are introduced and defined in Eq. (12) as the ratio between the flows and the diffusive conductances of a given face of the CV.

$$P_x = \frac{F_x}{D_x} \text{ with } x \in \{\mathrm{n,s,w,e}\} \tag{12}$$

The Péclet number aims at comparing the effect of the diffusion of the scalar quantity Φ with respect to the transport of the same quantity by fluid motion. It comes into play as an argument for function $A(\cdot)$. This function, introduced in (Patankar 1980), allows to decouple the numerical discretisation from the employed convective scheme. In this way it is possible to define a unique structure for the constitutive equations, where the convection scheme can be adapted, in order to satisfy the given requirements (e.g., concerning accuracy). The functions listed in Table 4 are just some examples of how the possible schemes can be implemented. A general discretised formulation for the CD equation has been presented. The remaining part of the section shows how such a general form can fit the mass, energy and momentum preservation equations.

Table 4 The various functions $A(|P|)$

Scheme	Function for $A(\|P\|)$
Central differencing	$1–0.5\|P\|$
Upwind	1
Hybrid	$\|\|0,1–0.5\|P\| \|\|$

4.4 The continuity equation

The discretised continuity equation is obtained by integrating Eq. (1a) along a CV located in a P-cell (a grey cell in Fig. 3), leading to

$$V\frac{d\rho_P}{dt} + F_e - F_w + F_n - F_s = 0 \tag{13}$$

For such a CV, the mass flow on the east face can be computed as

$$F_e = (\rho u_x)_e = v_x^e \cdot \begin{cases} \rho_E & \text{if } v_x^e < 0 \\ \rho_P & \text{if } v_x^e > 0 \end{cases} \tag{14a}$$

where ρ_P is the density of the CV considered, ρ_E is the density of the CV that surrounds the considered one on its east side, while v_x^e is the x-velocity of the air crossing the eastern surface (other fluxes can be computed in a similar way). Velocities (due to the implementation of the staggered grid approach) are immediately readable from the surrounding x (and y) momentum CVs. Concerning the Modelica representation, taking as reference the numbering convention shown in Fig. 2, for the CV [i,k] the density and the velocities are the ones listed in Table 5. Since the flows defined above are useful for computing other quantities, they are saved into matrices Fe_M[*,*], Fw_M[*,*], Fn_M[*,*] and Fs_M[*,*] representing the four flows for each volume [i,k].

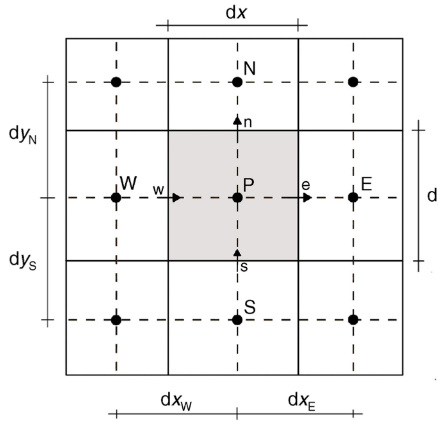


Fig. 3 Grid with dimensions and notations. The grey volume is the CV in which the continuity equation has been integrated. Arrows indicate the positive directions of the mass flows

Table 5 Density and velocities to be considered when discretising the continuity preservation equation over the corresponding CV (P-cell)

ρ_P	←	rho[i,k]
v_x^e	←	V$_x$[i,k]
v_x^w	←	V$_x$[i-1,k]
v_y^n	←	V$_y$[i,k]
v_y^s	←	V$_y$[i,k-1]

4.5 The energy equation

The discretised energy Eq. (1b), once integrated over a CV located in a P-cell (see grey cell in Fig. 3), can be rewritten as

$$V\rho c_v \frac{dT_P}{dt} + c_v a_P T_P = c_v\left(a_E T_E + a_W T_W + a_N T_N + a_S T_S\right) + Q_{\text{source}} \tag{15}$$

where T_P is the temperature of the CV analysed, and $T_{E,W,N,S}$ are the temperatures of the surrounding CVs. The definition of the coefficients $a_{P,E,W,N,S}$ provided for the general case in Eq. (11) have to be adapted. In particular, just the diffusive conductance has to be updated, since the CV taken into account is the same used for the continuity equation, the flows are the same (as defined in Eq. (14)). The diffusive conductances are thus computed as

$$D_e = \frac{\Gamma_e A_x}{dx_E} = \frac{\gamma_{\text{eff}}^e dy}{dx_E} \tag{16a}$$

$$D_w = \frac{\Gamma_w A_x}{dx_W} = \frac{\gamma_{\text{eff}}^w dy}{dx_W} \tag{16b}$$

$$D_n = \frac{\Gamma_n A_y}{dy_N} = \frac{\gamma_{\text{eff}}^n dx}{dy_N} \tag{16c}$$

$$D_s = \frac{\Gamma_s A_y}{dy_S} = \frac{\gamma_{\text{eff}}^s dx}{dy_S} \tag{16d}$$

where γ_{eff} is the effective thermal conductivity of the air. By implementing the general Eq. (15), it is possible to employ different convective schemes by choosing the proper $A(\cdot)$ function. When the CV `[i,k]` is considered, the distances that appear in Eq. (16) and shown in Fig. 3, can be computed as listed in Table 6.

Table 6 Distances for computing diffusive conductances with respect to the continuity CV

dx	←	`dx[i-1]`
dy	←	`dy[k-1]`
dx_E	←	`0.5*dx[i-1]+0.5*dx[i]`
dx_W	←	`0.5*dx[i-2]+0.5*dx[i-1]`
dy_N	←	`0.5*dy[k-1]+0.5*dy[k]`
dy_S	←	`0.5*dy[k-2]+0.5*dy[k-1]`

4.6 *x*-momentum equation

The discretised x-momentum equation reads

$$V\rho\frac{\mathrm{d}v_x^{\mathrm{P}}}{\mathrm{d}t}+a_{\mathrm{P}}v_x^{\mathrm{P}}=a_{\mathrm{E}}v_x^{\mathrm{E}}+a_{\mathrm{W}}v_x^{\mathrm{W}}+a_{\mathrm{N}}v_x^{\mathrm{N}}+a_{\mathrm{S}}v_x^{\mathrm{S}}+A_x(P_{\mathrm{W}}-P_{\mathrm{E}}) \tag{17}$$

where v_x^{P} is the horizontal velocity of the air in the considered CV, $v_x^{\mathrm{E,W,N,S}}$ are the surrounding ones, A_x is the CV surface normal to the x-direction and $P_{\mathrm{W,E}}$ are the pressure in the surrounding P-cells. The CV employed for the discretisation, as shown in Fig. 4 is shifted along the x axis of half a volume (both left and right), with respect to adjacent P-cells. This has two consequences: First, the velocity located in the centre of the CV (x-cell) is exactly in the middle of the faces of the continuity CV (P-cell). Second, the pressure on the east and west boundaries of the x-momentum CV, appearing in Eq. (17), are exactly the ones computed in the continuity CV (no interpolation for retrieving the value on the faces is needed). As in the previous cases, the flows and the diffusive conductances have to be defined. The flows are defined according to Fig. 4. The east face of the CV (the grey one) is surrounded by the horizontal flows computed for the continuity CV on the right (the green one in Fig. 4—left). The same is valid for the west boundary (the yellow one in Fig. 4—center). For such a reason, the flows on the east (e) and west (w) faces can be computed as the mean of the flows of the surrounding continuity CVs

$$F_{\mathrm{e}}=(\rho v_x)_{\mathrm{e}}=\frac{F_{\mathrm{e}}^{\mathrm{E}}+F_{\mathrm{w}}^{\mathrm{E}}}{2} \tag{18a}$$

$$F_{\mathrm{w}}=(\rho v_x)_{\mathrm{w}}=\frac{F_{\mathrm{e}}^{\mathrm{W}}+F_{\mathrm{w}}^{\mathrm{W}}}{2} \tag{18b}$$

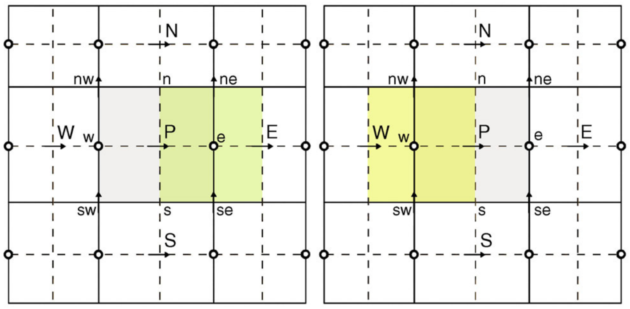


Fig. 4 Grid for the x-momentum discretisation. In grey the CV for the discretised x-momentum cell, with other colors the surrounding P-cells

where $F_{\mathrm{e,w}}^{\mathrm{E}}$ indicate the fluxes on the east and west faces of the continuity CV located on the right of the x-momentum CV (the same is valid for the opposite face). For computing the flows over the south (s) and north (n) faces of the x-momentum CV, the vertical velocities that surround the volume have to be involved. Doing so, four y-velocities located exactly on the four corners of the volume respectively $v_y^{\mathrm{NE,NW,SE,SW}}$ come into play. For each one of these velocities, crossing the north/south faces of the continuity CVs that surround the one considered here, flows are associated. The two remaining flows can be defined through mean values and computed as

$$F_{\mathrm{n}}=(\rho v_y)_{\mathrm{n}}=\frac{F_{\mathrm{n}}^{\mathrm{E}}+F_{\mathrm{n}}^{\mathrm{W}}}{2} \tag{19a}$$

$$F_{\mathrm{s}}=(\rho v_y)_{\mathrm{s}}=\frac{F_{\mathrm{s}}^{\mathrm{E}}+F_{\mathrm{s}}^{\mathrm{W}}}{2} \tag{19b}$$

Concerning the Modelica implementation, if the x-momentum CV `[i,k]` is considered, the surrounding flows are listed in Table 7. The next step is the definition of the diffusive conductances as shown in Eq. (20)

$$D_{\mathrm{e}}=\frac{\Gamma_{\mathrm{e}}A_x}{\mathrm{d}x_{\mathrm{E}}}=\frac{\mu_{\mathrm{eff}}^{\mathrm{e}}\mathrm{d}y}{\mathrm{d}x_{\mathrm{E}}} \tag{20a}$$

$$D_{\mathrm{w}}=\frac{\Gamma_{\mathrm{w}}A_x}{\mathrm{d}x_{\mathrm{W}}}=\frac{\mu_{\mathrm{eff}}^{\mathrm{w}}\mathrm{d}y}{\mathrm{d}x_{\mathrm{W}}} \tag{20b}$$

$$D_{\mathrm{n}}=\frac{\Gamma_{\mathrm{n}}A_y}{\mathrm{d}y_{\mathrm{N}}}=\frac{\mu_{\mathrm{eff}}^{\mathrm{n}}\mathrm{d}x}{\mathrm{d}y_{\mathrm{N}}} \tag{20c}$$

$$D_{\mathrm{s}}=\frac{\Gamma_{\mathrm{s}}A_y}{\mathrm{d}y_{\mathrm{S}}}=\frac{\mu_{\mathrm{eff}}^{\mathrm{s}}\mathrm{d}x}{\mathrm{d}y_{\mathrm{S}}} \tag{20d}$$

Since the grid is staggered, also the distances between the centre of the cell and the surroundings CVs differ. Considering an `[i,k]` x-momentum CV, its dimensions are defined in Table 8.

Table 7 Flows that surround the x-momentum CV

F_e^E	←	`Fe_M[i,k-1]`
F_w^E	←	`Fw_M[i,k-1]`
F_n^E	←	`Fn_M[i,k-1]`
F_s^E	←	`Fs_M[i,k-1]`
F_e^W	←	`Fe_M[i-1,k-1]`
F_w^W	←	`Fw_M[i-1,k-1]`
F_n^W	←	`Fn_M[i-1,k-1]`
F_s^W	←	`Fs_M[i-1,k-1]`

Table 8 Distances for computing diffusive conductances with respect to the x-momentum CV

dx	←	`0.5*dx[i-1]+0.5*dx[i]`
dy	←	`dy[k-1]`
dx_E	←	`dx[i]`
dx_W	←	`dx[i-1]`
dy_N	←	`0.5*dy[k-1]+0.5*dy[k]`
dy_S	←	`0.5*dy[k-2]+0.5*dy[k-1]`

4.7 y-momentum equation

The discretised version of the y-momentum preservation equation reads

$$V\rho\frac{dv_y^P}{dt} + a_P v_y^P = a_E v_y^E + a_W v_y^W + a_N v_y^N + a_S v_y^S + A_y(P_S - P_N) - \rho_P g V \quad (21)$$

and it differs from the x-momentum one (Eq. (17)) because it contains the gravity term $-\rho_P g V$ (where g is the gravity acceleration). The only part of this term that is notable, is the density ρ_P, since the other quantity are constants. Of course such a density is not defined in the centre of the considered CV, but it is on the continuity CVs that surround the considered one (green and yellow volumes in Fig. 5). As for the other quantities it is possible to obtain its value by a weighted mean

$$\rho_P = \frac{\rho_s + \rho_n}{2} \quad (22)$$

where ρ_s and ρ_n are the densities computed in the centre of the continuity CVs shown in Fig. 5 (respectively the yellow and the green CVs). The remaining terms of the y-momentum equation do not differ significantly from the ones introduced in the discretised x-momentum equation, except for the indices, thus omitted.

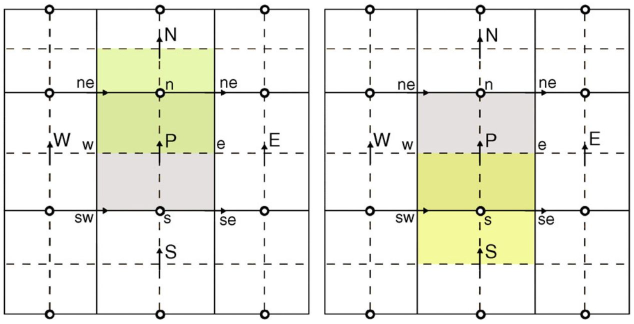


Fig. 5 Grid for the y-momentum discretisation. In grey the CV for the discretised y-momentum cell, with other colours the surrounding ones

4.8 Turbulence model

The turbulence model is present in each x and y-momentum CV. For a given `[i,k]` velocity CV, the turbulence model is implemented as shown in Table 9, where `mu` is the air dynamic viscosity, `rho` is the air density (here assumed to be constant), `Cu` is a constant parameter of the turbulence model, `karm` is the von-Karman constant, `y_plus` is the normalised minimal distance from the wall and *y_wall* is the distance between the centre of the CV and the nearest wall. In the model is present also a boolean flag, `laminar`, that neglects the turbulence model when the laminar flow assumption is assumed to be valid (e.g. very low velocities or high viscosity). The quantities `muY30` and `gammaY30` are respectively the viscosity and the thermal conductivity in the near wall region when `y_plus` is assumed to be equal to 30. These values have been used for a linear interpolation between the various cases, as shown in Table 9.

Table 9 Modelica implementation of the turbulence model

Quantity	Equation
y^+	`y_plus = MUx[i,j]/karm/mu`

μ_{eff}	`MUeff= if laminar`
	`mu`
	`else`
	`mu + karm*Cu*Sqrt(Vx[i,j]^2)`
	`*rho*y_wall`

μ_w	`MUwall= if y_plus < 5 then`
	`mu`
	`else if 5 <= y_plus < 30 then`
	`mu + (y_plus -5)*(muY30 -mu)/25`
	`else if y_plus >= 30 then`
	`rho*karm*Cu*Sqrt(Vy[i,j]^2)`
	`*y_wall/log(E*y_plus)`

k_{eff}	`K = if laminar then`
	`gamma`
	`else`
	`gamma + (MUeff-mu)*gamma/mu`

k_w	`Kwall = if y_plus < 5 then`
	`gamma`
	`else if 5 <= y_plus < 30 then`
	`gamma + (y_plus -5)*(gammaY30`
	`-gamma)/25`
	`else if y_plus >= 30 then`
	`gamma*(MUeff-mu)/MUwall`

4.9 State equations

To complete the set of equations, the fluid state ones are introduced and coded in Modelica. These relationships have to be written for each continuity CV (P-cell). A brief summary of the state equations for the `[i,k]` CV is reported in Table 10, according to their definition in Eqs. (2b) and (3a).

Table 10 State equations included in the continuity CV

Scalar	Expression

Density ρ	`rho[i,k] = rho_o + ComprCoeff*P[i,k]`
	`-ThermExpCoeff*(T[i,k]-To)`

Specific energy e	`e[i,k] = cv*T[i,k]`

4.10 Interfaces

The Modelica model has been written in a way that makes it possible to define the boundary conditions through standard interfaces. The adoption of such interfaces, in Modelica named connectors (Mattsson et al. 1998), allows the model to describe a specific part of the physical system. The model of the room is not the solution of a specific problem (e.g. the natural convection in a square room), it describes the interaction between the boundaries and the interior domain, so the same model can be used for representing various scenarios, each one characterised by its specific boundary conditions.

The boundaries of the domain, as defined in Section 4.2, are a specific subset of the domain variables. More in detail, the domain is represented by 2D array variables while a specific boundary (e.g. the east one) is represented by 1D array variables. Such boundary variables will be defined through connectors. The model of the room, incomplete because without specific values for its boundaries, will be completed by connecting it to surrounding models. The equations belonging to the surrounding models, describing various physical phenomena do not come into play except for the value provided to the connector linked together to the one of the room.

The structure of the Modelica connectors (Modelica, 2013) linked to the variables and volumes on the boundaries of the room is shown in Table 11. With such an approach a variety of cases can be represented without additional effort. For example a wall that surrounds the room will impose the mass flow equal to zero and a given temperature to the boundary temperature nodes (without taking into account if the model of the wall has a given number of layers). If the wall has an air inlet the temperature and velocities will be defined as in the previous case, except for the volume in which the air is injected. Here the velocity e.g. will be imposed and the condition of the incoming air (specific enthalpy, density) will be defined by the surrounding component.

Table 11 Variables contained within the Modelica interfaces

Name	Type	Description
`T`	Effort	Temperature (K)
`Q`	Flow	Heat flow rate (W)
`P`	Effort	Pressure (Pa)
`m_flow`	Flow	Mass flow rate (kg/s)
`h`	Stream	Specific enthalpy (J/(kg·K))
`rho`	Stream	Density (kg/m^3)

Connectors can be used also for accounting internal heat gains such as a person (or an appliance) heating a specific volume. In this case the interaction will be represented by imposing the heat flow rate (appearing in the energy equation as a source term). Figure 6 shows an example in which the model of a pipe, through connectors, can be connected together with the model of a room. In this case, heat flux flowing through the connection and coming into play as a source term in the energy equation will be defined according to the temperature difference between the pipe surface and the air.

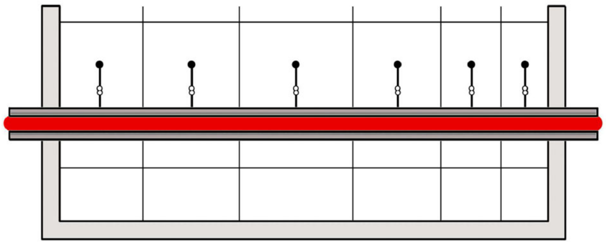


Fig. 6 Example of interaction between different models through connectors: a pipe fed with hot water heating the interior of the room

4.11 The numerical solver

The aim of this subsection is to help the reader familiarise with the numerical techniques used to solve the problem. One of the advantages of the equation-based modelling approach (which Modelica adopts) is that the problem expression is separated from its numerical solution. In accordance with that, in the previous sections there are not explicit references to any algorithm or numerical solution techniques. In other words, adopting an equation-based approach makes solver-related issues transparent to the user. The problem, and thus the equations, are described as they are, without explicitly introducing e.g. any time discretisation facts. The Modelica-based tool manipulates the set of equation (e.g., for index reduction) in order to get a system of Ordinary Differential Equations (ODE). Once the ODE system is available, a suitable numerical (either implicit/explicit fixed/variable time step) solver is applied. The solver that has been used in the reported examples is an implicit variable step one (namely DASSL).

5 Applications

In this section, some examples show the capabilities of the developed models to correctly represent the temperature distribution and the airflow pattern in three main cases: natural, forced and mixed convection. For each case a comparison between experimental results and data obtained via simulation (using Dymola as Modelica tool) has been done. Results have also been compared with those provided by fine-scale CFD models, developed with Fluent (2006).

5.1 Natural convection

The introductory example is a validation of the presented model, investigating the case of natural convection in a tall cavity (see Fig. 7 (left)) where the right wall is heated while the left one is cooled. The cavity dimensions are 0076×.21.8×05.2[m]. Experimental results for such a cavity are taken from (Betts and Bokhari 2000). The shape of the cavity, as well as its symmetry, allows to describe the fluid with a 2D grid, without reducing the accuracy in the description of the temperature distribution and the airflow field. For such a reason a non-uniform grid of 11×21 volumes has been used. The comparisons between simulation data and experimental results are listed in Figs. 8 and 9. In particular, both temperature and vertical velocity profiles at different heights ($y = \{0.1,0.4,0.6,0.9\}Y$, where Y is the height of the cavity) are shown. The agreement between results provided by Modelica models and both CFD and experimental data is very good as can be seen in Figs. 8 and 9. For the considered example, the simulation of 200 s takes 5.45 s on a standard PC.

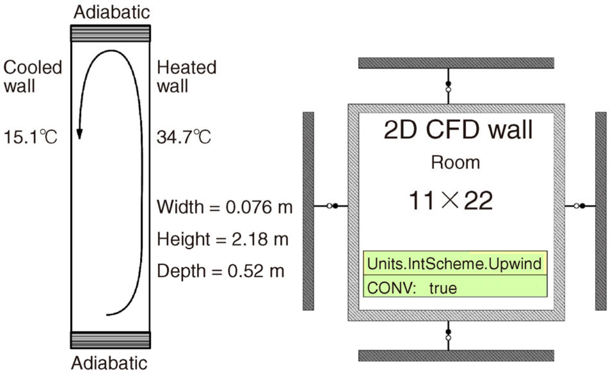
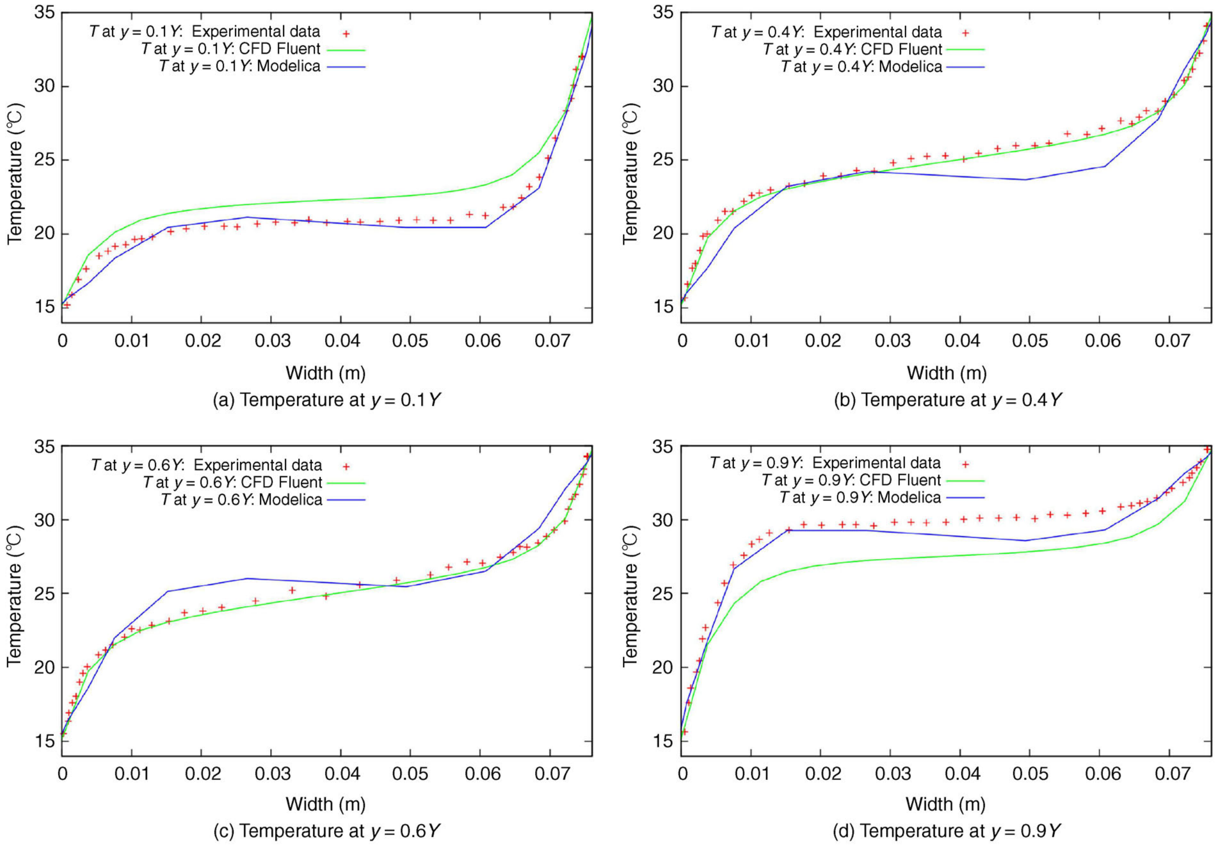


Fig. 7 (Left) scheme of the tall cavity; (right) Modelica connection scheme of the natural convection experiment



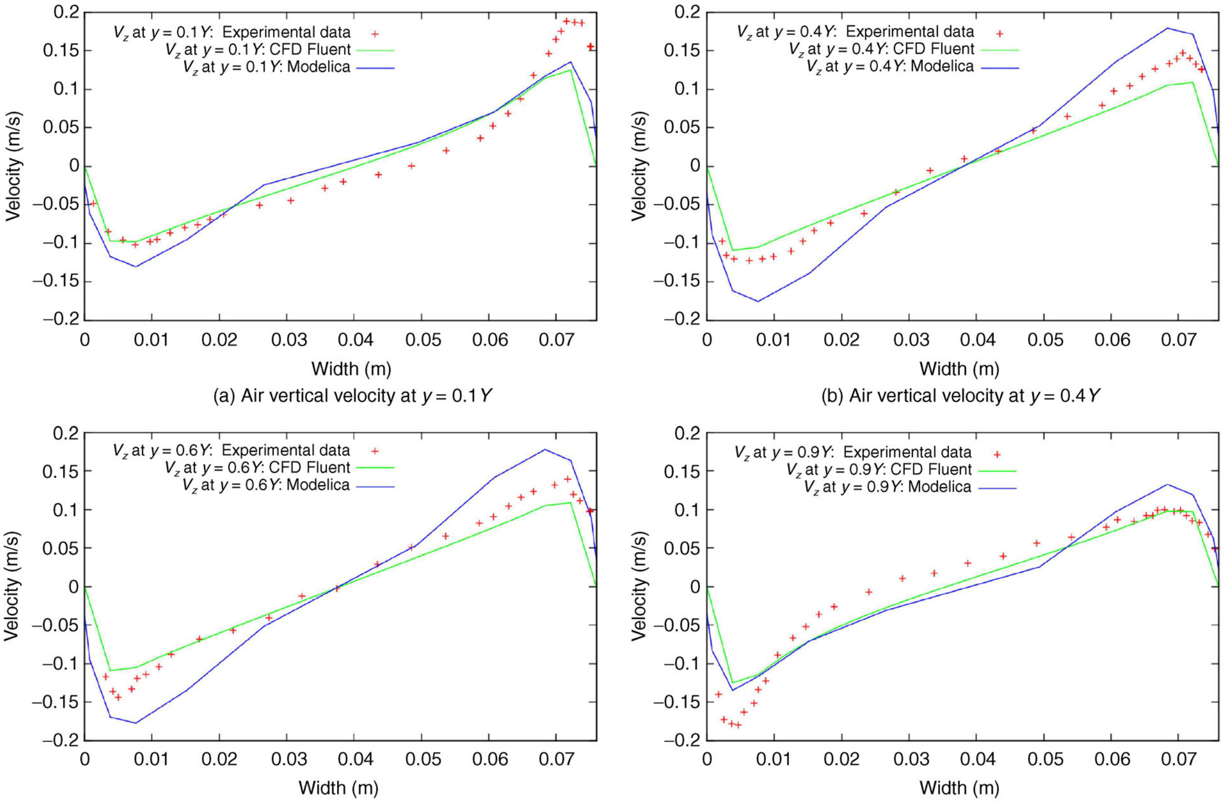
(a) Temperature at $y = 0.1Y$

(b) Temperature at $y = 0.4Y$

(c) Temperature at $y = 0.6Y$

(d) Temperature at $y = 0.9Y$

Fig. 8 Temperature distributions at different heights—Natural convection in a tall cavity



(a) Air vertical velocity at $y = 0.1Y$

(b) Air vertical velocity at $y = 0.4Y$

(c) Air vertical velocity at $y = 0.6Y$

(d) Air vertical velocity at $y = 0.9Y$

Fig. 9 Air vertical velocity distributions at different heights— Natural convection in a tall cavity

5.2 Forced convection

The second example is the forced convection experiment described in (Restivo 1979), for which physical measured data are available. The considered room is of rectangular shape, as shown in Fig. 10, and its size is 3×9 [m]. In this case, the airflow pattern within the room is mainly driven by the air inlet injected in the upper left corner and flowing out from the lower right one. The height of the air inlet is $h_{\mathrm{IN}} = 0.056H$ (where H is the height of the room), while the height of the outlet is $h_{\mathrm{OUT}} = 0.16H$. The air velocity at the inlet is $V = V_x = 0.455$ m/s and horizontally directed. In this case, since the velocity of the air jet injected in the upper left corner of the room is remarkable, it strongly affects the airflow motion. Contrary to the previous case, in which the unique (small) driving force was the buoyancy, here the convective terms appearing in the momentum equations play a crucial rule.

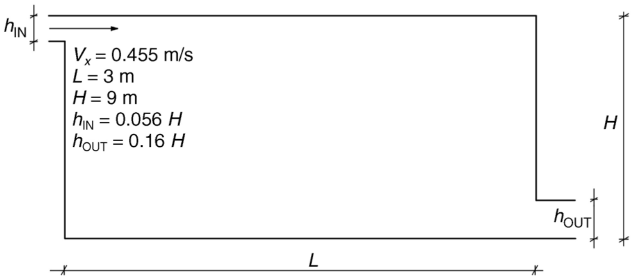


Fig. 10 Scheme of forced convection room

Figures 11, 12 and 13 show the horizontal air velocity distribution along two vertical sections of the room, located respectively at x=H and x=$2H$. Figure 11 shows that there is a good agreement between experimental data and the one obtained with Modelica models, using a mesh of 28×20 volumes. Figure 13 shows that despite has been employed a coarser mesh, the airflow pattern is correctly represented. Previous works, facing the problem with the so called sub-zonal approach (Mora et al. 2003), poorly represents such a recirculation in the room. In Fig. 12 given a mesh of 20×15 volumes, the suitable convective schemes: UpWind (UW), Hybrid (Hyb) and Power Law (PL) have been compared. Despite all the solutions are good enough since they correctly represent the airflow pattern, the higher order method (PL) gives the best results. The simulation performances of this example are listed in Table 12.

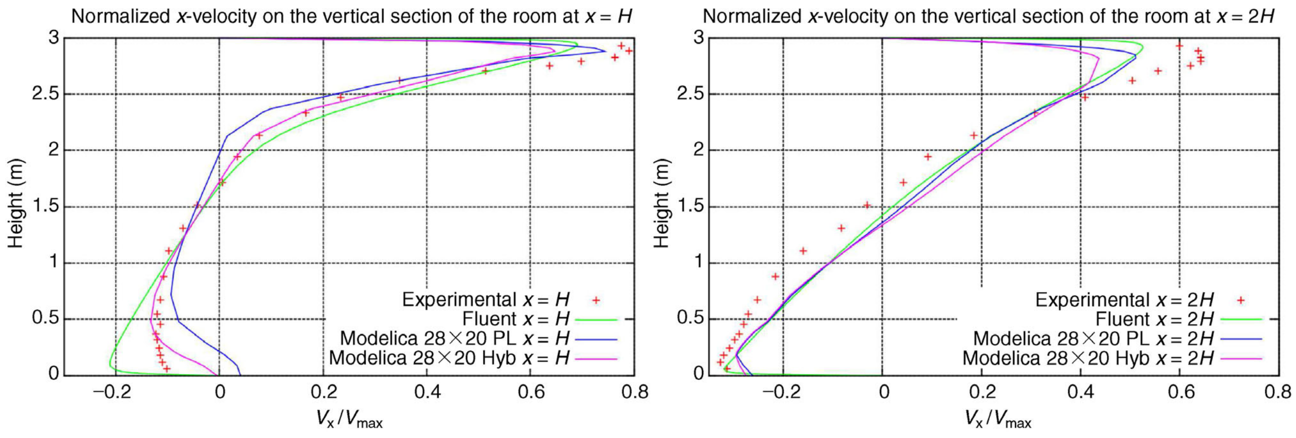


Fig. 11 Air horizontal velocity distributions at *x* = *H* and *x* = 2*H*—Comparison between experimental data, CFD Fluent and Modelica with a grid of 28×20 volumes

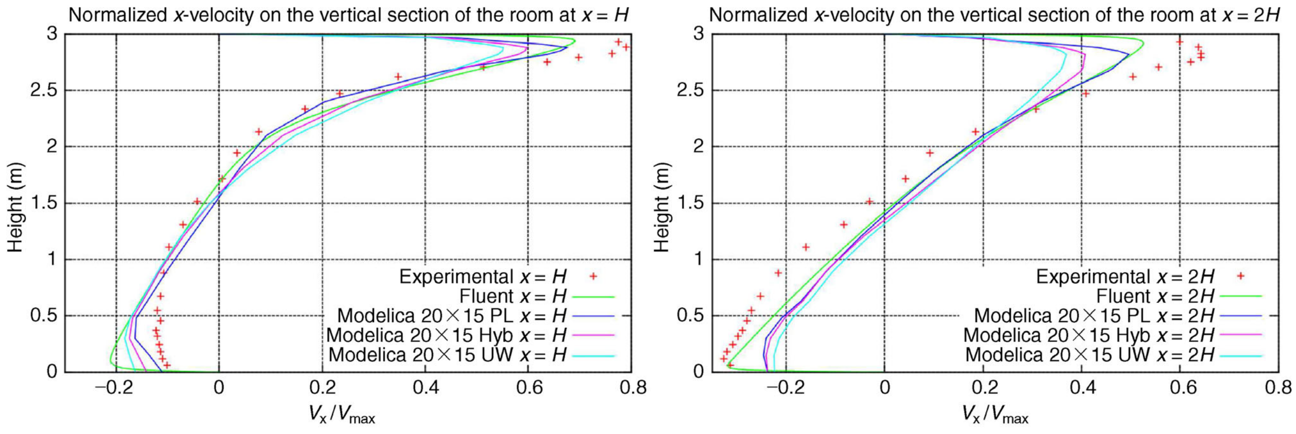


Fig. 12 Air horizontal velocity distributions at *x*=*H* and *x*=2*H*— Comparison between experimental data, CFD Fluent and Modelica with a grid of 20×15 volumes using different convective scheme (UpWind, Hybrid and Power Law)

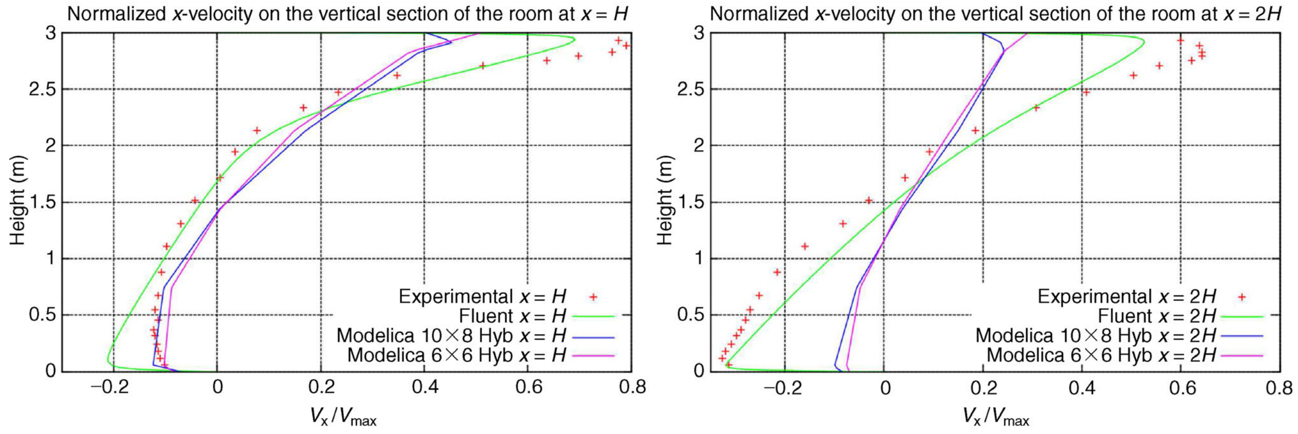


Fig. 13 Air horizontal velocity distributions at *x*=*H* and *x*=2*H*— Comparison between experimental data, CFD Fluent and Modelica with coarse meshes (10×8 and 6×6)

Table 12 Simulation time spent for simulating the forced convection case: 1000 s

Grid	Method	Time (s)
6×6	Hybrid	0.1
10×8	Hybrid	0.36
20×15	Hybrid	16.3
20×15	Power Law	19.6
20×15	UpWind	18.7
28×20	Hybrid	160
28×20	PowerLaw	182

5.3 Mixed convection

In the case of mixed convection, both the driving forces analysed in the previous cases (air inlet and buoyancy) are present at the same time. The room addressed in this example is of square shape and its size is 1.04×1.04 [m], as shown in Fig. 14. The inlet height, located in the left upper corner, is $h_{\mathrm{IN}}=0.018\mathrm{m}$, while the outlet height located in the opposite one is $h_{\mathrm{OUT}}=0.024$ m. The inlet air is horizontally injected at a temperature of 15℃ at a velocity of 0.455 m/s. The surrounding walls except for the floor are kept at a constant temperature of $T_{\mathrm{wall}}=15$℃, the floor is heated and kept at a temperature of $T_{\mathrm{h}}=35$℃. Experimental data, namely temperature and velocity profile across the middle vertical section of the room, are given in (Blay et al. 1992).

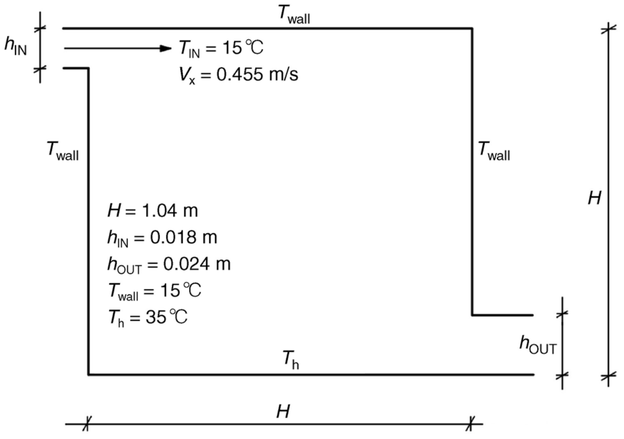


Fig. 14 Scheme of the mixed convection room

In Fig. 15 (left) the temperature profile across the vertical section of the room is shown, while Fig. 15 (right) shows the horizontal velocity profile over the same vertical section. Both figures compare experimental results with CFD and Modelica ones. More in detail, Modelica results have been obtained using various non-uniform grids, respectively of 25×25, 20×20, 15×10 and 12×8 volumes. Figure 15 does not include the results of the grid with 25×25 volumes since they cannot be distinguished by the results obtained with the grid 20×20. The sensitivity of the velocity profile with respect to the number of volumes contained into the grid is low, while the same is not true for the temperature profile. The Modelica model using the finer mesh reproduces almost correctly the temperature profile, while the others underestimate the value. To note that even with the coarse meshes the shape of the temperature profile is almost the same. Concerning the velocity profile, CFD gives better results with respect to Modelica models; they correctly represent the profile shape but they give an underestimation. Table 13, contains the simulation performances and the sum of the power flows exchanged through the walls at steady state. As the number of volumes increases the power exchanged through the walls converges to a single value, evidencing the correcness of the model.

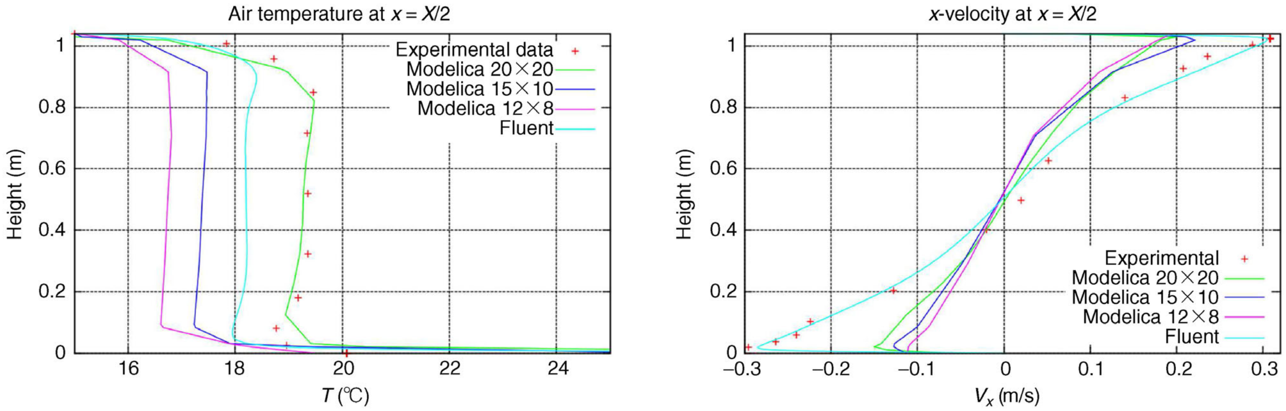


Fig. 15 Temperature (left) and horizontal velocity distributions (right) at x=X/2—Comparison between experimental results, CFD (Fluent) and Modelica models with different grid

Table 13 Simulation time spent for simulating the mixed convection case (3600 s) and power flowing through the walls

Grid	Method	Time (s)	Power (W)
12×8	Hybrid	2.36	140.5
15×10	Hybrid	4.02	123.18
20×20	Hybrid	288	96.709
25×25	Hybrid	810	95.6314

6 Conclusion

A modelling framework has been proposed that plugs the NS equations, suitably formulated in a unitary manner and spatially managed with a sub-zonal discretisation, into an OO environment (namely, by resorting to the Modelica language). This results in CFD models that are coarse-scale with respect to those typically employed, allow to catch the relevant energy-related phenomena in the building context, and yield efficiency advantages.

Also, the OO approach significantly fosters modularity, which permits to join airflow, ES, control and other models so as to solve everything together, without the need for co-simulation. This (again) favours efficiency, but also results in a more structured and thus simpler management of simulation models, and makes the numerical solution transparent to the analyst. Such a feature is important for system-level studies, that are often needed also at early design stages, thus where full information is not (yet) available, or some design data are *de facto* the subject of the investigation. The proposed solution, in this respect, relies on a convenient abstraction of model connectors, allowing to encapsulate the internal dynamic behaviour of each component, thus to have descriptions of the same object at different detail levels, and all interchangeable with one another.

Simulation results and comparisons with both other simulation tools and experimental measurements have been presented to support the proposal.

Future work will be devoted to extending the proposed framework from airflows to other similar problems, and to further extend the developed model library. Also, validation of said models versus industrial regulations (e.g., the ASHRAE 140 standard) is underway, as well as the use of the developed models in studies aimed at both the design of new buildings or neighbourhoods, and the energy-oriented refurbishing of existing ones.

References

Allard F, Dorer V, Feustel H (1990). Fundamentals of the multizone airflow model—COMIS. Technical Note 29, Air Infiltration and Ventilation Centre, Coventry, UK.

Arias D (2006). Advances on the coupling between a commercial CFD package and a component-based simulation program. In: Proceedings of 2nd National IBPSA-USA Conference, Cambridge, MA, USA, pp. 231–237.

Beausoleil-Morrison I (2000). The adaptive coupling of heat and air flow modelling within dynamic whole-building simulation. PhD thesis, University of Strathclyde, UK.

Betts PL, Bokhari IH (2000). Experiments on turbulent natural convection in an enclosed tall cavity. *International Journal of Heat and Fluid Flow*, 21: 675–683.

Blay D, Mergui S, Niculae C (1992). Confined turbulent mixed convection in the presence of horizontal buoyant wall jet. *Fundamentals of Mixed Convection, ASME HTD*, 213: 65–72.

Bonvini M, Leva A, Zavaglio E (2012). Object-oriented quasi-3d sub-zonal airflow models for energy-related system-level building simulation. *Simulation Modelling Practice and Theory*, 22: 1–12.

Cellier FE, Kofman E (2006). Continuous System Simulation. New York: Springer.

Chen Q (2009). Ventilation performance prediction for buildings: A method overview and recent applications. *Building and Environment*, 44: 848–858.

Chen Q, Xu W (1998). A zero equation turbulence model for indoor airflow simulation. *Energy and Buildings*, 28: 137–144.

Crawley DB, Lawrie LK, Winkelmann FC, Buhl WF, Erdem AE, Pedersen CO, Liesen RJ, Fisher DE (1997). What next for building energy simulation—A glimpse of the future. In: Proceedings of IBPSA International Conference, Prague, Czech Republic, pp. 395–402.

Fluent (2006). Fluent 6.3 User's Guide.

Ljubijankic M, Nytsch-Geusen C, Rädler J, Löffler M (2011). Numerical coupling of Modelica and CFD for building energy supply systems. In: Proceedings of 8th International Modelica Conference, Dresden, Germany.

Mattsson S, Elmqvist H, Otter M (1998). Physical system modeling with Modelica. *Control Engineering Practice,* 6: 501–510.

Modelica (2013). Modelica Association, Available: http://www.modelica.org.

Mora L, Gadgil A, Wurtz E (2003). Comparing zonal and CFD model predictions of isothermal indoor airflows to experimental data. *Indoor Air*, 23: 77–85.

Ostlund P, Stavaker K, Fritzson P (2010). Parallel simulation of equation-based models on CUDA-enabled GPUs. In: Proceedings of 9th Workshop on Parallel/High-Performance Object-Oriented Scientific Computing, Reno, NV, USA, pp: 5:1–5:6.

Patankar S (1980). Numerical Heat Transfer and Fluid Flow. London, UK: Taylor and Francis.

Pedersen CO, Fisher DE, Liesen RJ, Strand RK, Taylor RD, Buhl WF, Winkelmann FC, Lawrie LK, Crawley DB (1997). Energybase: The merger of BLAST and DOE-2. In: Proceedings of IBPSA International Conference, Prague, Czech Republic, Volume III, pp. 1–8.

Ren Z, Stewart J (2003). Simulating air flow and temperature distribution inside buildings using a modified version of COMIS with sub-zonal divisions. *Energy and Buildings*, 35: 257–271.

Restivo A (1979). Turbulent flow in ventilated room. PhD Thesis, University of London, UK.

Sahlin P (2000). The methods of 2020 for building envelope and HVAC systems simulation—Will the present tools survive? In: Proceedings of CIBSE Conference, Dublin, Ireland.

Sahlin P, Eriksson L, Grozman P, Johnsson H, Shapovalov A, Vuolle M (2004). Whole-building simulation with symbolic DAE equations and general purpose solvers. *Building and Environment*, 39: 949–958.

Trčka M, Hensen JL, Wetter M (2009). Co-simulation of innovative integrated HVAC systems in buildings. *Journal of Building Performance Simulation*, 2: 209–230.

Trčka M, Hensen JL, Wetter M (2010). Co-simulation for performance prediction of integrated building and HVAC systems—An analysis of solution characteristics using a two-body system. *Simulation Modelling Practice and Theory*, 18: 957–970.

Versteeg H, Malalasekera W (2007). An Introduction to Computational Fluid Dynamics: The Finite Volume Method. Upper Saddle River, NJ, USA: Pearson Prentice Hall.

Wetter M (2009). Modelica library for building heating, ventilation and air-conditioning systems. In: Proceedings of 7th International Modelica Conference, Como, Italy.

Wetter M, Haugstetter C (2006). Modelica versus TRNSYS—A comparison between an equation-based and a procedural modeling language for building energy simulation. In: Proceedings of 2nd National IBPSA-USA Conference, Cambridge, MA, USA.

Wetter M, Zuo W, Nouidui TS (2011). Recent developments of the Modelica buildings library for building heating, ventilation and air-conditioning systems. In: Proceedings of 8th International Modelica Conference, Dresden, Germany.

Zhai Z (2003). Developing an integrated building design tool by coupling building energy simulation and computational fluid dynamics programs. PhD Thesis, Massachusetts Institute of Technology, USA.

Zhai Z, Chen Q (2004). Numerical determination and treatment of convective heat transfer coefficient in the coupled building energy and CFD simulation. *Building and Environment*, 39: 1001–1009.

Zuo W, Chen Q (2010). Fast and informative flow simulations in a building by using fast fluid dynamics model on graphics processing unit. *Building and Environment*, 45: 747–757.

Zuo W, Hu J, Chen Q (2010). Improvements in FFD modeling by using different numerical schemes. *Numerical Heat Transfer, Part B: Fundamentals*, 58: 1–16.