Fluid Flow Modelling with Modelica

Marco Bonvini[1*], Mirza Popovac[2]

[1]Dipartimento di Elettronica e Informazione, Politecnico di Milano, Via Ponzio 34/5, 20133 Milano, Italia;
[*]*bonvini@elet.polimi.it*

[2]Austrian Institute of Technology, Sustainable Building Technology, Gieffinggasse 2, 1210, Vienna, Austria

Simulation Notes Europe SNE 23(1), 2013, 25 - 30
DOI: 10.11128/sne.23.tn.10167
Received: July 3, 2012 (Selected MATHMOD 2012 Postconf. Publ.); Revised Accepted: March 20, 2013;

Abstract. This work shows how the problem of modelling fluids motion can be addressed in Modelica. This innovative approach makes possible to face such a problem in a multi-physic modelling language as Modelica is. In this way it is possible to simulate together the fluid and the system that interacts with it, without any additional effort and taking advantage of the Modelica libraries representing buildings, power plants, water treatment systems, HVAC and so forth.

Introduction

Modelling fluid flows is extremely important in simulating many engineering processes. When the fluid is constrained to move in ducts or pipes strong assumptions/simplifications can be taken into account without affecting the description of the fluid properties (e.g. temperatures, pressures, densities,...) and their distributions. The mentioned simplification for such cases where a spatial coordinate prevails the others leads to zero or one dimensional models where the spatial dependence is respectively disregarded or limited to just one coordinate (e.g. pipes). However there are elements like tanks (in the context of hydraulic systems) or rooms (in the context of HVAC systems) where zero or one dimensional models are not appropriate. The standard practice when simulating such more complicated scenery, is to employ CFD codes. Despite this approach is capable of representing in a very detailed way the fluid thermal dynamics, it has some drawback. The main one is in its modularity. CFD cannot be easily integrated with other models in order to represent the entire system, the only way for doing such a task is to employ the so called co-simulation techniques, that introduce a communication overhead and some non trivial convergence problem as shown by Trčka et al. [8].

The aim of this work is to provide a general methodology for modelling 2D or 3D fluid flows with Modelica. Modelica is a multi-physic Object-Oriented modelling language [**?**]. In Modelica several modelling libraries, representing a variety of systems are already available [6], and new ones can be developed. Thanks to the modularity of the language and the deffinition of standard interfaces, models belonging to different physical domains can be coupled together. Providing a way for modelling fluid flows in such an environment is a step ahead in the direction of a real integrated multi-domain simulation tool, thus avoiding co-simulation and its drawbacks [8], [10]. The proposed modelling approach aims at representing simple scenery in cases where the powerful capabilities of CFD software are not needed. More precisely, complex geometries and high velocities are not taken into account , however a wide range of application like rooms, portion of buildings, storage tanks can be modelled. As consequence, despite the apparent simplicity of the proposed approach a widespread set of relevant cases can be investigated.

The structure of the paper is the following: an introductory section where the governing equations are shown is followed by a section in which the discretisation approach is presented. Then, the implementation in the Modelica language of the discretised equation is discussed. The last section concerns the validation of the models, and more in detail, a comparison between experimental data coming from a natural convection case is reported. Some conclusions as well future works complete the paper.

1 The Governing Equations

The motion of fluid is described by the equations of mass, energy and momentum balance, and this set of equations is often referred to as the Navier Stokes equations (NS). In the case of the Newtonian fluid they can be written as:

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

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

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

where the scalars p, T, e, h, ρ, k and μ are respectively the fluid pressure, temperature, specific energy, specific enthalpy, density, thermal conductivity and dynamic viscosity; the vectors $\mathbf{v}$ and f are the fluid velocity and the external forces only, such as gravity, acting on the fluid.

With the scalar projection brought in, and for simplicity analysing only 2D case, the momentum equation (1c) is decomposed into two scalar equations:

$$\begin{aligned} &\tfrac{\partial \rho v_x}{\partial t} + \tfrac{\partial \rho v_x v_x}{\partial x} + \tfrac{\partial \rho v_x v_y}{\partial y} = \\ &f_x - \tfrac{\partial p}{\partial x} + \tfrac{\partial}{\partial x}\left(\mu \tfrac{\partial v_x}{\partial x}\right) + \tfrac{\partial}{\partial y}\left(\mu \tfrac{\partial v_x}{\partial y}\right) \end{aligned} \tag{2a}$$

$$\begin{aligned} &\tfrac{\partial \rho v_y}{\partial t} + \tfrac{\partial \rho v_y v_x}{\partial x} + \tfrac{\partial \rho v_y v_y}{\partial y} = \\ &f_y - \tfrac{\partial p}{\partial y} + \tfrac{\partial}{\partial x}\left(\mu \tfrac{\partial v_y}{\partial x}\right) + \tfrac{\partial}{\partial y}\left(\mu \tfrac{\partial v_y}{\partial y}\right) \end{aligned} \tag{2b}$$

where the subscripts x, y denote the components of the 2D Cartesian coordinate system. In the case of natural convection having as vertical axis the y one, clearly $f_x = 0$ and $f_y = -\rho g$, with g being the gravity acceleration. In order to solve numerically the momentum equation (as well the continuity and the energy ones), each term appearing has to be properly represented.

2 The Discretised Model

The conservation equation for mass (refeqn:mass), energy (refeqn:energy) and momentum (1c) can be represented in the standard Convection-Diffusion (CD) form, which (again for simplicity) in the 2D case reads:

$$\begin{aligned} &\underbrace{\frac{\partial \rho \Phi}{\partial t}}_{\text{local}} + \underbrace{\frac{\partial \rho v_x \Phi}{\partial x} + \frac{\partial \rho v_y \Phi}{\partial y}}_{\text{convective}} = \\ &\underbrace{\frac{\partial}{\partial x}\left(\Gamma_\Phi \frac{\partial \Phi}{\partial x}\right) + \frac{\partial}{\partial y}\left(\Gamma_\Phi \frac{\partial \Phi}{\partial y}\right)}_{\text{diffusive}} + \underbrace{S_\Phi}_{\text{source}} \end{aligned} \tag{3}$$

where the generic quantity Φ is the scalar quantity transported by the fluid moving with velocity $\mathbf{v} = (v_x, v_y)$, and Γ_Φ is the diffusivity coefficient. The time dependent variable Φ can be either one velocity component, the internal energy or the mass fraction of a chemical species. The source term S_Φ is the generation rate of the scalar quantity Φ per unit volume.

The generic CD equations (3) states that the (unsteady) local change of the scalar quantity Φ is equal to the sum of the convective change, the diffusive change, and the generation from a source. For example, replacing Φ with v_y, Γ_Φ with the viscosity μ and collecting in the source term S_Φ both the gravity $-\rho g$ and the pressure gradient dp/dy, the y-momentum equation (2b) is obtained. Following the numerical procedure for solving CD, described in [7], the general CD equation (3) is integrated over a grid of Control Volumes (CV) as shown in Figure 1.

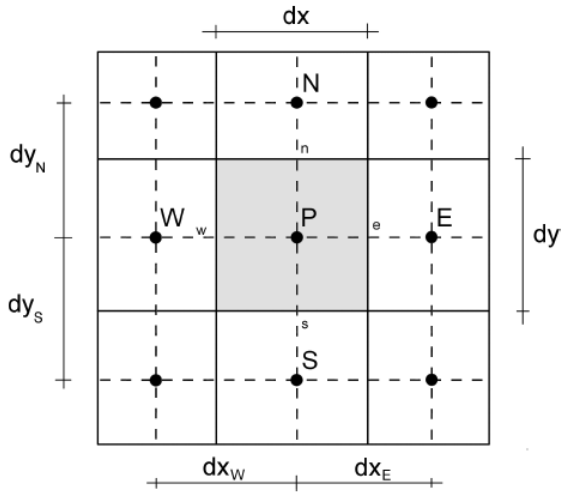


Figure 1: Grid employed for the spatial discretisation of the CD equation.

Applying the Gauss' theorem, the volume integrals are replaced with surface ones:

$$\begin{aligned} &\textstyle\int_V \frac{\partial \rho \Phi}{\partial t} dV + \int_S (\partial \rho v_x \Phi)\mathbf{i} \cdot \mathbf{n} dA + \int_S (\partial \rho v_y \Phi)\mathbf{j} \cdot \mathbf{n} dA = \\ &\textstyle\int_S \left(\Gamma_\Phi \frac{\partial \Phi}{\partial x}\right)\mathbf{i} \cdot \mathbf{n} dA + \int_S \left(\Gamma_\Phi \frac{\partial \Phi}{\partial y}\right)\mathbf{j} \cdot \mathbf{n} dA + \int_V S_\Phi dV \end{aligned} \tag{4}$$

where $\mathbf{i}$ and $\mathbf{j}$ are respectively the x and y components of the unit vector, while $\mathbf{n}$ is the outgoing unit normal vector from the surface element dA. The surface integrals that appear in (4) can be approximated via sums over the faces of the considered control volume.

The unsteady term and the sources that represent respectively the variation of the scalar quantity Φ over the time, and the rate of generation into the CV, are replaced as follows:

$$\int_V \frac{\partial(\rho\Phi)}{\partial t} dV \simeq V \frac{d(\rho\Phi)}{dt} \tag{5a}$$

$$\int_V S_\Phi dV \simeq V S_\Phi \tag{5b}$$

where V is the volume of the CV. In particular, implementing the Final solution in Modelica, there is no need for explicitly implement a time discretisation method. Indeed, using the capabilities of the Modelica solvers (e.g. [**?**]) several adaptive time step solvers can be employed without any additional effort required.

When the CD equation aim at representing the NS equations, the pressure gradients appearing in (2) are included into the source terms. Once integrated over the CV and converted into surface integrals they read

$$\int_S P\mathbf{i}\cdot\mathbf{n}dS \simeq A_x(P_e - P_w) \tag{6a}$$

$$\int_S P\mathbf{j}\cdot\mathbf{n}dS \simeq A_x(P_n - P_s) \tag{6b}$$

where $A_{x,y}$ are the surfaces of the CV normal to the x and y direction respectively, and $P_{e,w,n,s}$ are the pressures on the boundaries of the CV. The diffusive term of the equation (4) can be approximated as:

$$\begin{aligned}&\textstyle\int_S \left(\Gamma_\Phi \frac{\partial\Phi}{\partial x}\right)\mathbf{i}\cdot\mathbf{n}dA \simeq \\ &D_{\Phi_e}(\Phi_E - \Phi_P) - D_{\Phi_w}(\Phi_P - \Phi_W)\end{aligned} \tag{7a}$$

$$\begin{aligned}&\textstyle\int_S \left(\Gamma_\Phi \frac{\partial\Phi}{\partial y}\right)\mathbf{j}\cdot\mathbf{n}dA \simeq \\ &D_{\Phi_n}(\Phi_N - \Phi_P) - D_{\Phi_s}(\Phi_P - \Phi_S)\end{aligned} \tag{7b}$$

where $D_{\Phi_{e,w,n,s}}$ are the diffusivity coefficients evaluated at the CV faces. For the east face (omitting the others for brevity) it is computed as:

$$D_{\Phi_e} = \Gamma_{\Phi_e}\frac{A_x}{dx_E} \tag{8}$$

where dx_E is the distance between the center of the CV and the neighbour close to the E face. The diffusivity of the fluid is a property that may vary between adjacent CVs (e.g. the fluid viscosity or the thermal conductivity vary in time and space). For such a reason the diffusivity Γ_{Φ_e} is computed on the boundaries of the CV as a weighted mean of the fluid properties in the cell P and the cell E.

The most influential in a fluid flow is the convective term. Assuming that the velocities are normal to the surfaces of the CV, (v_x normal to faces E and W, while v_y normal to faces N and S), they can be approximated as:

$$\int_S (\rho v_x \Phi)\mathbf{i}\cdot\mathbf{n}dA \simeq F_e\Phi_e - F_w\Phi_w \tag{9a}$$

$$\int_S (\rho v_y \Phi)\mathbf{j}\cdot\mathbf{n}dA \simeq F_n\Phi_n - F_s\Phi_s \tag{9b}$$

where $F_{e,w,s,n}$ are the mass fluxes over the faces of the control volume. Again for brevity, for the e face the mass flux is computed as

$$F_e = (\rho v_x)_e A_x \tag{10}$$

where the subscript indicates that the value is computed on the e boundary of the CV, while A_x is the surface of the CV normal to x directions. The values $\Phi_{e,w,n,s}$ introduced in (9) are the values of the scalar variable Φ on the boundaries of the CV. The way these values are computed has a strong impact on the numerical solution. The standard practice is to employ the first order accurate UPWIND scheme (11), where the scalar value on the boundary is taken from the one computed in the cell from where the fluid is flowing (hence the name upwind). Other methods, extensively described in [7], have been implemented.

$$\Phi_e = \begin{cases}\Phi_E & \text{if } F_e > 0 \\ \Phi_P & \text{if } F_e < 0\end{cases} \tag{11}$$

3 Implementation in Modelica

Having all the terms appearing in the CD equation (3) discretised over a given CV, the last step is to transform such an equation in a compact form that can be written in Modelica. The standard form, employed in all CFD tools and carefully explained in [7] has been adapted in order to be straightforwardly implemented in Modelica. The equation reads

$$V\rho\frac{d\Phi_P}{dt} + a_P\Phi_P = a_E\Phi_E + a_W\Phi_W + a_N\Phi_N + a_S\Phi_S + S \tag{12}$$

where the coefficients $a_{E,W,N,S,P}$ are a compact representation of both the diffusive and convective terms, while S are the possible sources (e.g. gravity, heat sources, external forces ... depending on the nature of Φ) or in the case of the momentum equation the

pressure gradients. Coefficients $a_{E,W,N,S}$ are defined as:

$$a_E = D_e A\left(\left|\frac{F_e}{D_e}\right|\right) + \|-F_e, 0\| \tag{13a}$$

$$a_W = D_w A\left(\left|\frac{F_w}{D_w}\right|\right) + \|F_w, 0\| \tag{13b}$$

$$a_N = D_n A\left(\left|\frac{F_n}{D_n}\right|\right) + \|-F_n, 0\| \tag{13c}$$

$$a_S = D_s A\left(\left|\frac{F_s}{D_s}\right|\right) + \|F_s, 0\| \tag{13d}$$

with $a_P = a_E + a_W + a_N + a_S$. In (13), $\|a,b\|$ is the maximum between a and b while $A(\cdot)$ is a function that represents the convective scheme employed (e.g. UPWIND or Central Difference) as described in [7].
Such a generalised version of the CD equation can be used for discretising the mass, the energy and the momentum balance equations. These equations have been spatially discretised over a staggered grid as suggested by Versteeg and Malalasekera ([9]), an example of such a grid is shown in Figure 2. The basic idea behind the staggered grid is to integrate the balance equations over CVs that differs, in order to avoid numerical problems as evidenced in [7], [9].

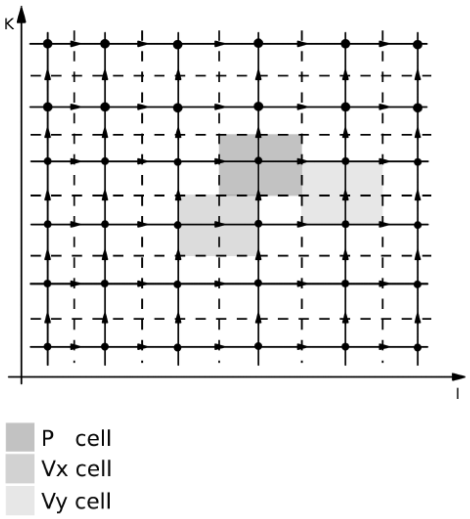


Figure 2: Staggered grid – Mass and Energy balance equations are discretised over P cells, while x-Momentum and y-Momentum equations are discretised over Vx and Vy cells.

For the boundary conditions, either the value (e.g. temperatures, velocities) or the gradients (e.g. heat fluxes) can be described. The grid of CVs is implemented as a matrix of nodes. The value of each node represents one of the scalar variable for which the CD equation is solved (e.g. the temperature, or velocity components). Boundary conditions are a given subset of values of these matrices. Boundary conditions can be extended by employing connectors, in such a way the values of a particular quantity (e.g. the temperature) instead of being defined a priori can be assigned by an other model. It is important to underline that Connectors are standard interfaces between the model and its neighbours. If the behaviour of the model has been properly described, just by knowing the information provided by its connectors, the model can be linked together with any model that implements the same connectors. This is a crucial point in the context of OO modelling, and this is the key that allows a real and powerful multi-physic simulation. The model is completed with a description of the fluid (the fluid state equation) that introduce a relationship between the temperature, pressure and density of the fluid. Such a relationship has been kept intentionally simplified, in order to reduce the complexity of the model. In particular a linearised version of the ideal gas relationship has been introduced.

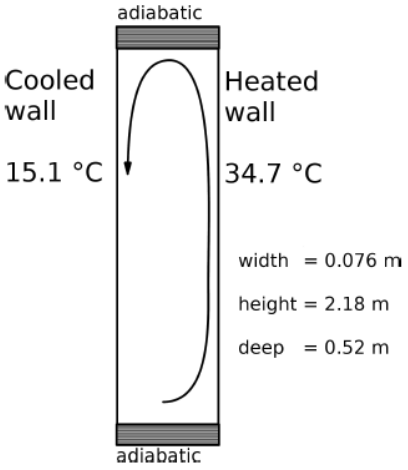
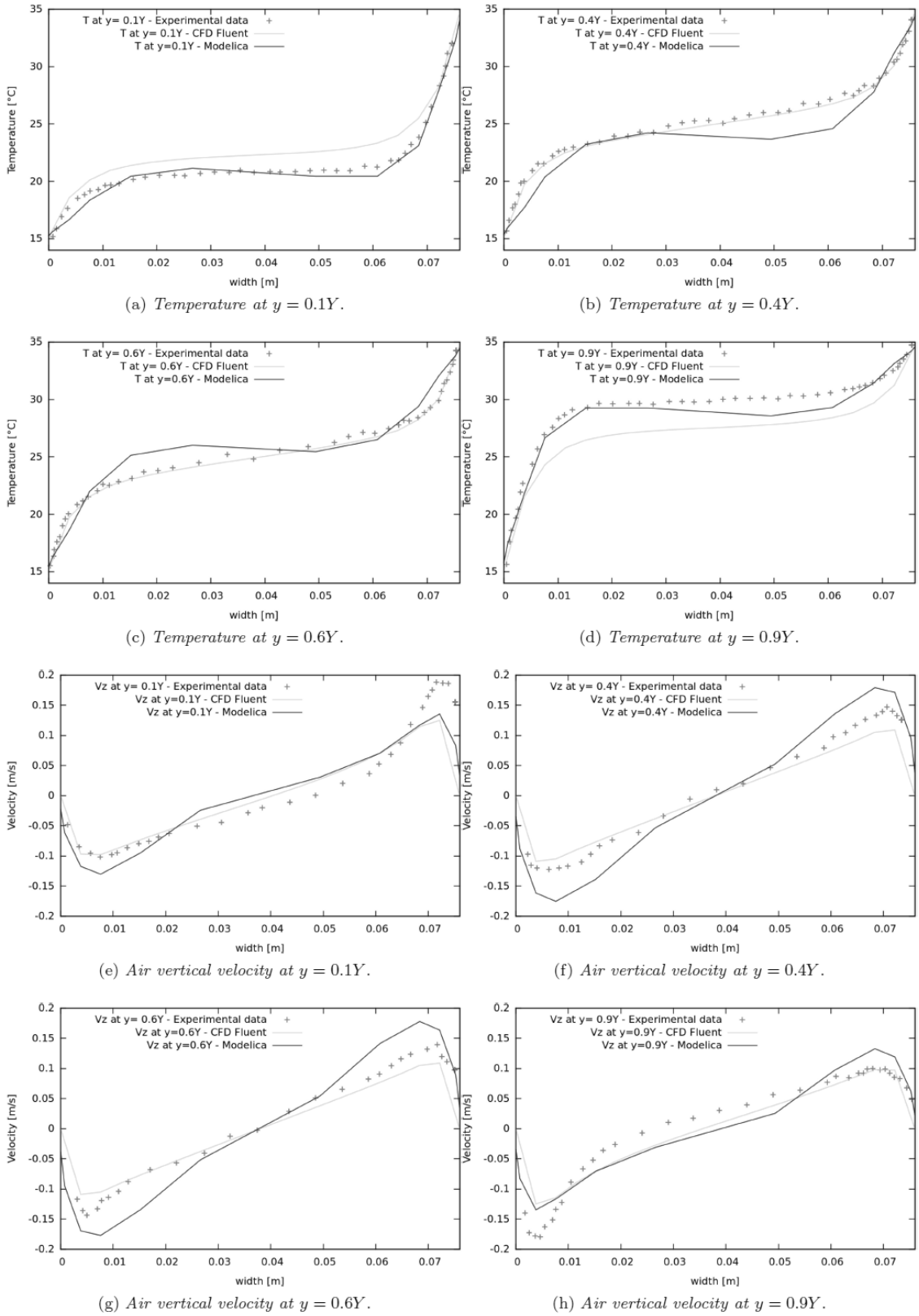


Figure 3: Scheme of the tall cavity.

A turbulence model and also a wall function representing the interaction between the fluid and the domain boundaries have been introduced. The complexity of the above mentioned models has been kept as low as possible, in order to reduce the computational effort. Therefore, following the idea of Prandtl ([**?**]), a zero-equation turbulence model is used, with the wall function (Launder and Spalding, [4]) used for imposing the wall boundary conditions for momentum equation.



(a) *Temperature at* $y = 0.1Y$.

(b) *Temperature at* $y = 0.4Y$.

(c) *Temperature at* $y = 0.6Y$.

(d) *Temperature at* $y = 0.9Y$.

(e) *Air vertical velocity at* $y = 0.1Y$.

(f) *Air vertical velocity at* $y = 0.4Y$.

(g) *Air vertical velocity at* $y = 0.6Y$.

(h) *Air vertical velocity at* $y = 0.9Y$.

Figure 4: Temperature (a - d) and air vertical velocity (e - h) distributions at different heights.

4 Validation

The validation is performed investigating the case of natural convection in a tall cavity (see Figure 3) where the right wall is heated while the left one is cooled. The sizes of the cavity are $0.076 \times 2.18 \times 0.52[m]$.

Experimental results for such a cavity are taken from [2]. The shape of the cavity, as well its symmetry, allows to describe the fluid with a 2D grid, without reducing the accuracy in the description of the temperature distribution and the air flow 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 Figures 4. 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. More in detail, Figures 4 (a-d) are the temperature profiles, while 4 (e-h) are the vertical velocity ones. In each plot experimental data are compared against simulation data provided by a standard CFD code ([1]) and simulation data obtained with Modelica models. The agreement between results provided by Modelica models and both CFD as well as experimental data is very good as can be seen in the various Figures.

To stress that the aim of Modelica models is not to give more accurate results with respect to CFD ones, but to give comparable ones by using a modelling paradigm that offers the possibility to integrate not only the fluid motion but also the interaction with other systems (e.g. the walls that surround the ambient, the environmental conditions as well as a suitable representation of the heat sources acting on the system).

5 Conclusion

A model capable of simulating fluid flows with an approach which is not the standard CFD has been proposed. Such a model makes possible to face the fluid-flows problem in a multi-domain modelling language, such as Modelica.

Despite the simplicity of the numerical scheme employed as well the geometry description taken into account, the big advantage is that now it is possible to simulate together the fluid and the system that interacts with it, without any additional effort and taking advantage of the Modelica libraries and avoiding co-simulation.

References

[1] Ansys fluent home page. URL http://www.ansys.com.

[2] Betts P L, Bokhari I H. Experiments on turbulent natural convection in an enclosed tall cavity. *International Journal of Heat and Fluid Flow*. 2000; 21(6): 675 - 683. doi:10.1016/S0142-727X(00)00033-3

[3] Dymola 7.4. URL http://www.3ds.com/products/catia/portfolio/dymola.

[4] Launder B, Spalding D. The numerical computation of turbulent flows. *Comput. Meth. Appl. Mech. Engrg.*. 1974; 2(3): 269 - 289. doi:10.1016/0045-7825(74)90029-2

[5] Mattsson S, Elmqvist H, Otter M. Physical system modeling with Modelica. *Control Engineering Practice*. 1998; 6(4), 501 - 510. doi:10.1016/S0967-0661(98)00047-1

[6] Modelica home page. URL http://www.modelica.org/.

[7] Patankar S. *Numerical heat transfer and fluid flow*. Edition 1. Washington, DC: Hemisphere Publishing Corp.; 1980. 197 p.

[8] Trčka M, Hensen J L, and Wetter M. 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*. 2010; 18(7): 957 - 970. doi:10.1016/j.simpat.2010.02.011

[9] Versteeg H, Malalasekera W. *An introduction to computational fluid dynamics: the finite volume method*. Edition 2. Upper Saddle River, NJ, USA: Pearson Prentice Hall; 2007. 503 p.

[10] Zhai Z, Chen Q Y. Numerical determination and treatment of convective heat transfer coefficient in the coupled building energy and cfd simulation. *Building and Environment*. 2004; 39(8): 1001 - 1009. doi:10.1016/j.buildenv.2004.01.023