

**ADVANCES ON THE COUPLING BETWEEN A COMMERCIAL CFD PACKAGE  
AND A COMPONENT-BASED SIMULATION PROGRAM**

Diego A. Arias

University of Wisconsin – Madison, Madison, WI  
daarias@wisc.edu, sel.me.wisc.edu**ABSTRACT**

This paper presents the development of an automatic coupling between the component-based system simulation program TRNSYS and the general-purpose CFD package FLUENT. Due to the nature of the iterative solution strategy in TRNSYS, the coupling can be considered as a fully-dynamic coupling, as the CFD component iterates with all of the other components in the system in order to obtain a converged solution at each time step. The connection between the two programs takes place by running the CFD program from TRNSYS through a script file written by TRNSYS during the simulation. The results from the CFD simulation are then used by other component model within TRNSYS to simulate the behavior of the rest of the system.

**INTRODUCTION**

Several studies have indicated the increasing interest of having a robust tool that performs the coupling between component-based simulation packages and detailed computational fluid dynamics (CFD) software. In the case of building performance, multi-zone models and CFD programs provide complementary approaches to predicting building performance (Zhai et al. 2002). In the case of engine simulation packages, a coupled simulation between one-dimensional models and a CFD program can be used to simulate the general behavior of an entire engine, at the same time that it provides detailed three-dimensional information about a particular engine part (Gamma Technologies 2001).

Crawley et al. (2005) compared the capabilities of several building energy simulation programs and reported that, among the 20 software compared, only three allowed for a simultaneous solution of thermal loads or airflow models with CFD packages.

The objective of this paper is to present the developments on the coupling between the system simulation software TRNSYS and the commercial

CFD package FLUENT, and to demonstrate the capabilities of a user-friendly connection with a basic example.

TRNSYS is a transient system simulation program with a modular structure. A system is defined in TRNSYS as a set of components interconnected in such a manner as to accomplish a specific task. Each component can be represented as a black box that uses the inputs to calculate some outputs. The relation between the inputs and outputs can be determined analytically, numerically, or empirically. The modular nature of TRNSYS allows for great flexibility, as these components may be part of the standard library or they can be written by the user (Klein et al., 2000). TRNSYS has several standard components that perform an automatic connection with other programs, such as COMIS (Feustel 1999), CONTAM (Dols and Walton 2002), EES (Klein 2006), among others. Using the same basic calling protocols, a new component was developed in TRNSYS, which allows for the automatic data exchange with FLUENT.

FLUENT is a popular, general-purpose CFD environment for a wide range of applications (FLUENT Inc. 2005). The capability of running a simulation from a text file was used as the basic feature required for the coupling with TRNSYS. Other CFD programs that use the same feature could be linked with TRNSYS using the same protocols.

This paper presents the implementation of an automatic coupling between TRNSYS and FLUENT. First, a general description of the different strategies for coupling the two types of programs is presented. Second, the paper describes the new component in TRNSYS that facilitates the coupling. Finally, the paper presents an example of a simulation of a simple flow network where one of the components is modeled with FLUENT.

## COUPLING CFD AND ENERGY-BASED SYSTEM SIMULATION PROGRAMS

Zhai et al. (2005) classified the different approaches that can be taken when coupling a CFD and energy-based simulation software into two general groups, according to whether the simulations are performed at an instant of time (static), or if the simulation is carried on during a period of time (dynamic).

- **Static coupling:** This coupling involves a one-step or a two-step information exchange between the energy simulation software and the CFD program. When this process takes place a few times, it can be performed manually. This can be a good alternative when both solutions are not very sensitive to the information that is exchanged.
- **Dynamic coupling:** If the system is not very dependent on the exchanged variables, it is possible to repeat the static coupling at each time step. This strategy is known as *quasi-dynamic coupling*. When the solution is highly dependent on the exchanged variables, it is required that the two programs iterate until they achieve a coupled-converged solution. Two alternatives for this iteration are presented by Zhai et al: *full dynamic coupling*, where at each time step the CFD program is called and the iterations are performed until convergence, and *virtual dynamic coupling*, where the inputs to the CFD program during the current time step are compared with the results from previous solutions of the coupled system and interpolated. This last alternative may be used to reduce the computational expense.

A static coupling can be achieved without the development of automatic data exchange between the two programs. Examples of studies that have used the static approach are Karlsson and Moshfegh (2006), Manz and Frank (2005), and Salom et al. (2005).

Examples of dynamic coupling are the studies performed by Jayaraman et al. (2005), Zhai and Chen (2005) and Farhanieh and Sattari (2006). These studies required the development of algorithms for the iterative solution between the two programs.

## AUTOMATIC COUPLING IN TRNSYS

The development of the automatic coupling with a CFD program took advantage of the following TRNSYS' features:

- **Modular nature of the simulation environment:** Any kind of system that can be defined as a

logical set of components where the outputs from one component are used as inputs to another component can be simulated in TRNSYS. One can think of those components as black boxes that calculate some values as function of their inputs. The component that connects to a CFD program was created with the same philosophy in mind: boundary conditions and additional parameters (e.g., transport properties, relaxation factors, solver parameters, etc.) are passed as inputs to the CFD program, and the results of the CFD simulation are used by the other components to simulate the performance of the entire system. Figure 1 shows a simple project layout that represents the modular characteristics of the simulations in TRNSYS.

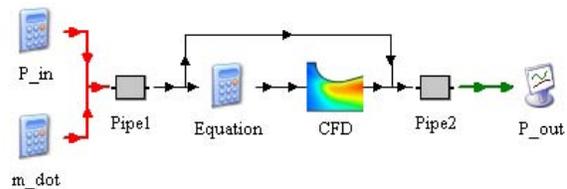


Figure 1. Project layout showing the connection with the new CFD component

- **Iterative solution algorithm:** TRNSYS uses a successive substitution method for the solution of the system of components (Stoecker 1989). This iterative solution strategy is required in order to ensure a converged solution when the information flow is cyclic or when the order in which the components are solved is not the same as the logical information flow.

The information flow diagram of a system may exhibit either cyclic or acyclic flow of information. Cyclic information flow occurs whenever there is a path in the information flow diagram where the outputs from a component influence its own inputs. Figure 1 shows a system with acyclic information flow, and Figure 2 a system with cyclic information flow.

The order of appearance of the components in the TRNSYS input file assigns the order in which the components are solved. This feature may cause the need for an iterative solution of a system that may not have required it, if the components were placed in the input file in different order. For example, in Figure 1, if the component labeled as 'Pipe2' appears first than 'Pipe1' in the input file, at least one iteration will be required to ensure that the outputs from 'Pipe1' are actually used as inputs to 'Pipe2' during the current time step.

This iterative nature of the TRNSYS simulations makes the coupling with the CFD component a *fully dynamic coupling* by default, i.e., TRNSYS will call the CFD component as many times as needed during a given time step until the values of the outputs from each component have converged within user-defined tolerances.

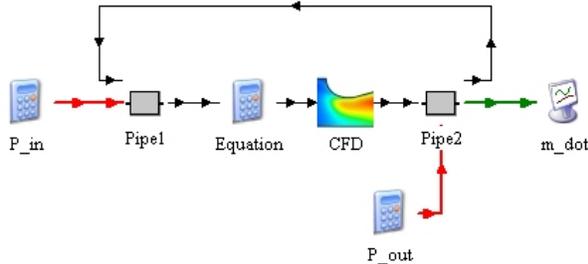


Figure 2. Project layout showing the cyclic connection between TRNSYS and FLUENT

The component was created in a very general fashion, in order to allow for any kind of simulation that can be performed in FLUENT. The component launches FLUENT as a separate process and shares information via intermediate text files. The communication between TRNSYS and FLUENT uses two text files:

- **Script file written by TRNSYS component**, which contains the instructions for running the FLUENT simulation. This script file takes advantage of FLUENT's capability to run in batch mode from a script file. The user should be knowledgeable in FLUENT's text commands in order to be able to write the appropriate script file (FLUENT Inc. 2005).
- **Results file written by FLUENT**, which contains the results from the CFD simulation. This file is read by TRNSYS in order to use the results determined by the CFD calculations and continue the simulation.

The operations performed by the new component are:

1. During the initial time step manipulations, the component calls FLUENT in order to initialize the CFD solution.
2. At each time step, the inputs to the component are used to update the boundary conditions of the CFD simulation.
3. The new component writes a script file with the modified boundary conditions and parameters required by FLUENT.
4. The component calls FLUENT with the script file.
5. FLUENT runs the scrip file and writes a text file with the desired results.

6. After the CFD simulation has finished, the TRNSYS component reads the results file and updates the outputs from the component.
7. TRNSYS continues the simulation. If required, it will iterate during the current time step until all components reach a converged solution.

## A SIMPLE FLOW NETWORK EXAMPLE

The implementation of the coupled CFD component is illustrated by the following example. It is based on the simulation of a simple flow network that consists of two pipes and an accessory that connects the two pipes.

### Basic equations for pipe flow

The momentum balance equation for incompressible flow inside a pipe of constant cross-sectional can be written as

$$P_1 = P_2 + f(L/D)\rho v^2/2, \quad (1)$$

where  $P_1$  and  $P_2$  are the static pressures at the inlet and outlet of the pipe, respectively,  $L$  is the length and  $D$  is the diameter of the pipe,  $v$  is the average velocity and  $\rho$  is the fluid density. A second equation for the Darcy friction factor,  $f$ , is required to close this system of equations:

$$f = F(Re, \varepsilon), \quad (2)$$

where the  $Re$  is the Reynolds number and  $\varepsilon$  is the relative surface roughness. The solution of this system of equations depends on the known information. Classical textbooks on fluid dynamics (e.g., White 1999) present the following two different cases, depending on whether the mass flow rate is known or not.

- **Case 1: Mass flow rate and inlet pressure known; solve for outlet pressure.** If the mass flow rate is known, the solution of the above system of equations can be obtained sequentially without any kind of iteration.
- **Case 2: Inlet and outlet pressures known; solve for mass flow rate.** If the mass flow rate is not known, the solution of the system of equations requires an iterative procedure.

### Flow networks

These basic cases for the solution of the momentum balance equation in a pipe represent the foundations for more complex systems. For example, Figure 3 shows a simple flow network with two pipes interconnected by a device, which could be a pump, valve, accessory, etc.

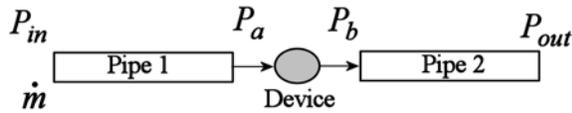


Figure 3. Simplified flow network

For example, if the device is an orifice, it may be represented as

$$\dot{m} = C_D A \sqrt{2\rho \Delta P} \quad (3)$$

where the mass flow rate,  $m$ , across the device of cross sectional area  $A$  is a function of the pressure drop across the device,  $\Delta P$ . A correction factor, commonly known as discharge coefficient,  $C_D$ , can be used to account for all of the deviations from an ideal flow. This correction factor is usually determined from experiments, and is written as function of the Reynolds number  $Re$ ,

$$C_D = F(Re) \quad (4)$$

The above classification of pipe flow problems into two cases can be extended to this flow network, i.e., if the mass flow rate is known, the solution of the flow network can be performed by solving the components in the direction of the flow. If the mass flow rate is unknown, the solution of the system requires an iterative procedure.

In some situations, the characteristics of the device are not well known, or it is of interest to study the behavior of a device when it is integrated into a system of components. In this situation, the representation of the device using a set of equations like Eqn. (3) and (4) may not be possible. On the contrary, it may be desired to use Computational Fluid Dynamics to study the characteristics of the component and connect it directly with the system.

#### Solution of the flow network in TRNSYS

Figure 1 shows an implementation of a system that represents the **Case 1**, where the mass flow rate is known. In this situation, the inlet pressure,  $P_{in}$ , is known as well as the mass flow rate,  $\dot{m}$ . These two values are sent as inputs to the component that represents the first pipe. The pressure at the outlet of the pipe and mass flow rate are then used as inputs to the CFD representation of the flow device. Finally, the pressure at the outlet of the CFD component is used as inlet pressure to the second pipe. The pressure

downstream of the system is the output of the simulation.

Figure 2 shows a situation that resembles **Case 2**, i.e., the mass flow rate is not known. The inputs to the system are the upstream and downstream pressures,  $P_{in}$  and  $P_{out}$ , respectively. The pressure at the outlet of the first pipe and mass flow rate from the first pipe are used as inlets to the CFD component. The pressure at the outlet of the CFD component and the pressure downstream of the system are used as inputs to the second pipe, which solves for the mass flow rate. Finally, the mass flow rate calculated in the second pipe is used as the additional input to the first pipe.

#### CHARACTERISTICS OF THE CFD REPRESENTATION OF THE DEVICE

In this example, a venturi was modeled in FLUENT and connected with two pipes in the flow network configurations indicated by Figures 1 and 2. In order to create a simulation with the CFD component connected to the other simulation components, the CFD component had to be validated in terms of grid size sensitivity and verification with experimental results.

#### Characteristics of the numerical model

FLUENT was used to model the steady-state airflow through a typical venturi of same dimensions as those used in carburetors for small utility engines (see Figure 4).

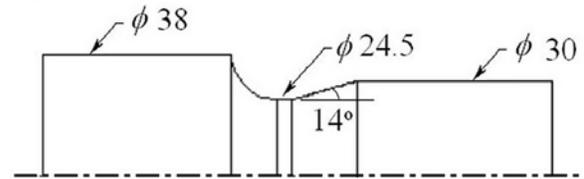


Figure 4. Dimensions of venturi (in mm)

A 2-D axi-symmetric model was used to simulate the venturi. An unstructured mesh was used to discretize the geometry of the converging-diverging nozzle. This geometry required the use of a second-order discretization scheme of the flow equations in order to avoid numerical diffusion. The convergence criteria were set to  $10^{-6}$  for the residual of the energy equation, and  $10^{-4}$  for the other conserved quantities.

#### Turbulence models

Four turbulence models for Reynolds-averaged Navier-Stokes equations were compared in order to find the most appropriate model: Standard k- $\epsilon$ , Realizable k- $\epsilon$ , Renormalization Group (RNG) k- $\epsilon$  and Reynolds Stress model (RSM) (FLUENT 2005).

Standard wall functions were used for the near-wall treatment.

In order to compare the results from the different turbulence models, a uniform inlet velocity was set to 20 m/s (Reynolds number based on the venturi throat  $=8 \times 10^4$ ). The results showed that, in general, the pressure and velocity fields predicted by the different turbulent models were very similar. The value of the discharge coefficient was approximately 0.91 for all of the turbulence models, with the standard  $k-\epsilon$  giving a difference of approximately 10% with respect to the other turbulence models. Figure 5 shows the differences in turbulent kinetic energy, where it can be seen that the standard  $k-\epsilon$  model predicted higher dissipation of turbulent kinetic energy, which explained the lower discharge coefficient. This figure also shows that the RNG  $k-\epsilon$  model gave similar results to the RSM model, requiring fewer iterations and computer resources. Based on these results, the RNG  $k-\epsilon$  model was selected for the following simulations.

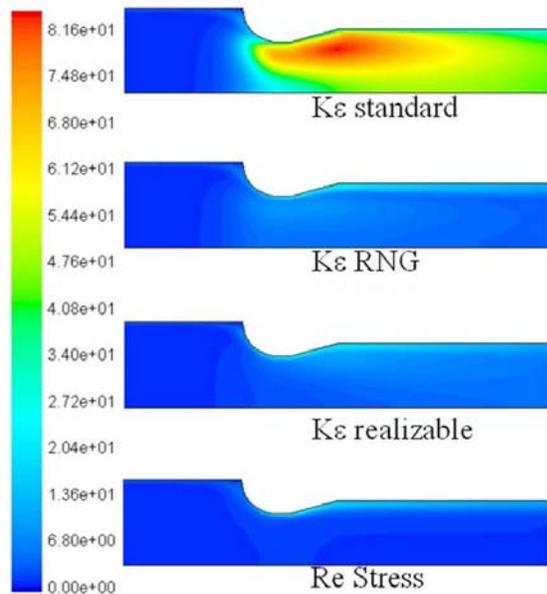


Figure 5. Comparison of turbulence kinetic energy between turbulence models ( $m^2/s^2$ )

### Mesh sensitivity

The convergence of the solution with respect to mesh size was assessed by running the simulations with three meshes of different sizes. The result for an infinitely small mesh was approximated using the Richardson Extrapolation (Roache 1994, and Celik and Karatekin 1997). It was found that the results of the intermediate-size mesh were within a 5% agreement with an infinitely small mesh.

### Comparison with experimental results

An experimental setup, shown in Figure 6, was built in order to verify the results from the simulations. A low pressure zone was created downstream of the venturi with a flow-amplifier; this low pressure region drove air into the system, forcing it to flow across the venturi. The air velocity was measured with a hot film anemometer at the inlet of the venturi, and the static pressure was measured at the inlet and at the outlet of the venturi with an electronic differential pressure transducer.

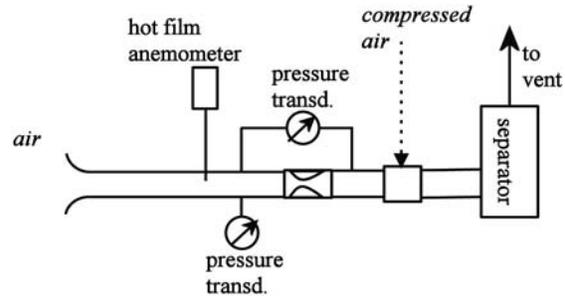


Figure 6. Experimental setup for the study of venturi

Figure 7 shows the comparison between the experimental and FLUENT-calculated discharge coefficient, over a range of airflow rates. It was found that the FLUENT results were within 10% of the experimental values. The final trend was captured, but the increase at low mass flow rate was not captured. This discrepancy might be due to the measurement of the velocity at a single point and using it as the average velocity for calculating the total mass flow rate.

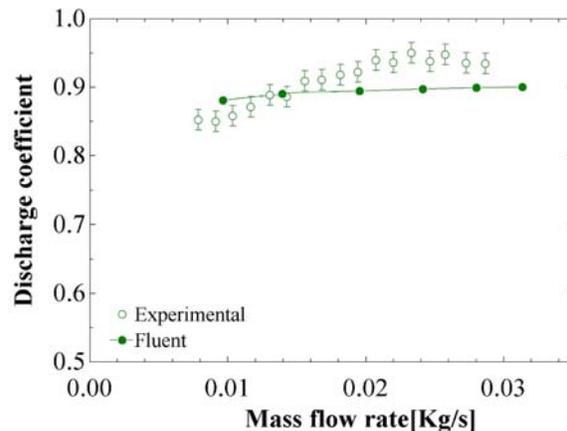


Figure 7. Comparison between experimental results and Fluent-calculated  $C_D$

### Results of the axi-symmetric model

Figure 8 shows an example of the static pressure and velocity fields obtained with the simulation of the flow

inside the clear venturi model used in the experiments. This figure indicates that the two-dimensional simulation is able to capture effects that a one-dimensional solution cannot capture: the change in momentum at the inlet of the venturi experienced by the fluid produces a lower static pressure at the walls of the throat than at the centerline.

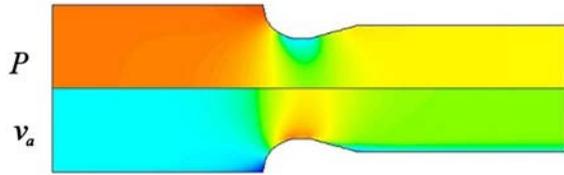


Figure 8. Static pressure and air velocity fields in venturi, using an axi-symmetric geometry.

The behavior of the static pressure along the centerline of the venturi is shown in Figure 9. It shows the static pressure for different outlet boundary conditions: -50 to -1000 Pa. It follows the classical behavior of a compressible nozzle described in compressible flow textbooks (e.g., Anderson 2003). The static pressure is almost constant at the inlet and outlet of the venturi, showing a slight decrease at the outlet due to frictional losses in the straight pipe section. In all of the cases the pressure decreases at the inlet of the converging section and reaches a minimum at the throat. As the inlet boundary condition was set to isentropic stagnation conditions, the different flow rates achieved by changing the outlet static pressure resulted in different inlet static pressures.

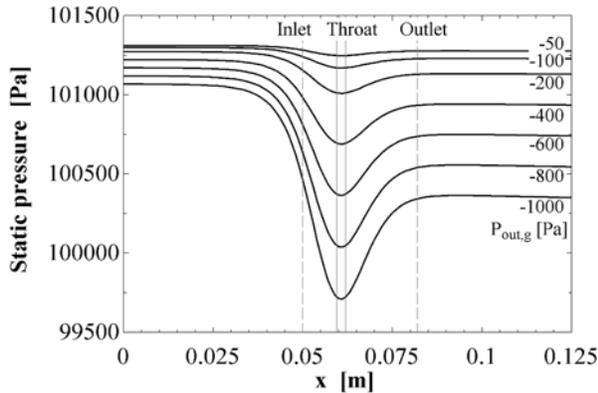


Figure 9. Static pressure along the venturi centerline.

## RESULTS OF THE COUPLED CFD FLOW NETWORK

Once the CFD model of the venturi was validated with the mesh size sensitivity analysis and with experimental results, it was simulated in TRNSYS with the two additional pipes in the flow network configuration shown in Figure 1.

Figure 10 shows the result of the static pressure at the outlet of the venturi. It can be seen that the initial values (time=0) of static pressure did not provide good estimates for the intermediate pressures at the inlet of the venturi ( $P_a$ ) and at the outlet of the venturi ( $P_b$ ). However, at the first time step, time=1h, the iteration of the system finds the right solution where the pressure at the inlet of the venturi is greater than at the outlet.

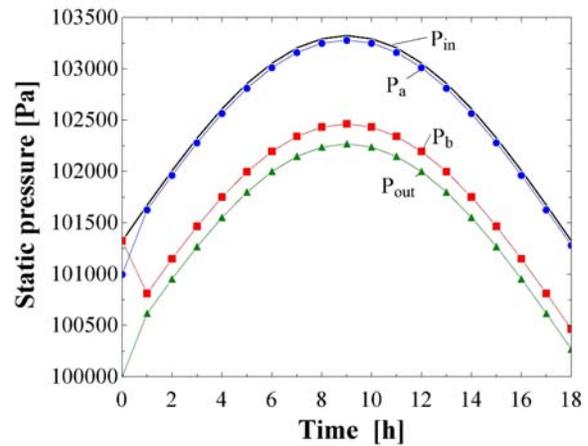


Figure 10. Static pressure at different locations in the flow network shown in Figure 3

## SUMMARY AND POSSIBILITIES FOR FUTURE SIMULATIONS

A connection between the commercial CFD package FLUENT and the system simulation package TRNSYS is now available with the development of a standard TRNSYS component that controls the data exchange between the two programs. Due to the nature of the iterative solution methods in TRNSYS, this connection can be considered as a *fully-dynamic coupling*.

This data exchange between TRNSYS and the CFD program takes place by taking advantage of FLUENT's capabilities to run in batch mode from a text script file.

An example of air flow across a converging-diverging nozzle was created in order to demonstrate the capabilities of the developed coupling. It required the validation of the CFD model with mesh sensitivity analysis and experimental results. Although this example is basic in its nature, it represents the possibilities for future implementations. Examples of the future applications are dynamic airflow, temperature distribution and pollutant dispersion under complex boundary conditions, like a complex

interaction of solar radiation, ventilation and shading on external facades.

There are several areas for future developments: as mentioned by Zhai et al., a fully-dynamic coupling may be very computational intensive. A virtual-dynamic coupling could be implemented by creating a new component that checks if the current inputs to the CFD component are close to a previous converged case, and then decides to run a full new CFD case or interpolate within the stored results.

In addition, the implementation of the CFD-coupling component described in this paper has no limitations on the number of CFD components present in a simulation. Future work will show the feasibility of running a system in which two or more components require a full CFD solution and are coupled with additional components.

## REFERENCES

- Anderson, J.D., 2003. *Modern Compressible Flow: With Historical Perspective*. McGraw Hill, 3<sup>rd</sup> edition.
- Crawley, D.B., Hand, J.W., Kummert, M., and Griffith, B.T., 2005. Contrasting the capabilities of building energy performance simulation programs. <http://www.eere.energy.gov/buildings>
- Celik, I. and Karatekin, O., 1997. Numerical experiments on the application of Richardson extrapolation with nonuniform grids. *Journal of Fluids Engineering*, 119(3), 584-590.
- Dols W.S., and Walton, G.W., 2002, *CONTAMW 2.0 User Manual*, NISTIR 6921.
- Farhanieh, B., and Sattari, S., 2006. Development of an integrated model for airflow in building spaces. *Renewable Energy*, (31), 401-416.
- Feustel, H.E., 1999, COMIS-An international multizone air-flow and contaminant transport model, *Energy and Buildings*, (30), 3-18.
- FLUENT Inc., 2005. *FLUENT 6.2 User's Guide*. Vol. 1. FLUENT Inc., Lebanon, NH.
- Gamma Technologies, 2001. *GT Power: User's Manual*, Version 5.2.
- Jayaraman, B., Lorenzetti, D.M., Gadgil, A.J. 2005. Coupled building pollutant transport simulation. *Dynamics*, (24), 18-19. <http://www.cd-adapco.com/products/brochures/dynamics/dynamics24.pdf>. Last checked: 21 June 2006.
- Karlsson, J.F., Moshfegh, B. 2006. Energy demand and indoor climate in a low energy building – changed control strategies and boundary conditions. *Energy and Building* (38), 315-326.
- Klein, S.A., 2006, Engineering Equation Solver (EES). <http://www.fchart.com>. F-Chart Software.
- Klein, S.A., Beckman, W.A., Mitchell, J.W., Duffie, J.A., Duffie, N.A., Freeman, T.L., Mitchell, J.C., Braun, J.E., Evans, B.L., Kummer, J.P., Urban, R.E., Fiksel, A., Thornton, J.W., Blair, N.J., Williams, P.M. and Bradley, D.E., 2000. TRNSYS: *A transient system simulation program*. Vol. 1. Solar Energy Laboratory, University of Wisconsin-Madison.
- Manz, H., Frank, Th. 2005. Thermal simulation of buildings with double-skin façades. *Energy and Buildings*, (37) 114-1121.
- Roache, P.J., 1994. Perspective: A method for uniform reporting of grid refinement studies. *Journal of Fluids Engineering*, 116(3), 405-413.
- Salom, J., Pascual, J., Houzeaux, G. 2005. Analysis and optimization of the thermal comfort of a non-air conditioned 16 000 m<sup>2</sup> tent, with transient and CFD simulations. Ninth International IBPSA Conference, Montréal, 1075-1082.
- Stoecker, W.F., 1989. *Design of thermal systems*. McGraw Hill, 3<sup>rd</sup> edition.
- White, F.M., 1999. *Fluid Mechanics*. McGraw Hill, 4<sup>th</sup> edition.
- Zhai, Z., Chen, Q., Haves, P., and Klems, J.H., 2002. On approaches to couple energy simulation and computational fluid dynamics programs. *Building and Environment*, (37), 857-864.
- Zhai, Z.J., and Chen, Q.Y., 2005. Performance of coupled building energy and CFD simulations. *Energy and Buildings*, (37), 333-344.