Strong Coupling of Modelica System-Level Models with Detailed CFD Models for Transient Simulation of Hydraulic Components in their Surrounding Environment

Conference Paper · March 2011

DOI: 10.3384/ecp11063256

1 author:

Antoine Viel

Siemens

14 PUBLICATIONS **501** CITATIONS

SEE PROFILE

Strong Coupling of Modelica System-Level Models with Detailed CFD Models for Transient Simulation of Hydraulic Components in their Surrounding Environment

Antoine Viel
LMS Imagine
7 Place des Minimes, 42300 Roanne, France
antoine.viel@lmsintl.com

Abstract

Strong coupling with a CFD software is usually suited to the coupled transient simulation of an hydraulic component (like a valve, a pump, ...) with its surrounding environment. Due to the nature of the solvers used by CFD code, co-simulation is generally the best way to couple a Modelica system-level simulator and a CFD solver. This article describes a methodology and the associated technology for establishing a co-simulation between a Modelica model simulated with an ODE/DAE solver like the one encountered in LMS Imagine.Lab AMESim, and a 3D model of flow computed by a CFD software. The physical, numerical, and computer-related aspects of co-simulation handled by this methodology are exemplified on an application test case in fluid power.

Keywords: Tool Coupling; Co-simulation; Hydraulic Component Modeling; CFD

1 Introduction

The detailed design of an hydraulic component like a compound relief valve [1] is a complex task and requires a model including many details. The methodology to model this component implies starting with a simple model and after analysis and comparison with experimental results, the model is made more complex step by step. With this approach it is possible to understand the influence of different parts of the valve on the overall system transient behavior. However experimental results might not be available for all subsystems. In such cases involving complex 2D or 3D geometries with turbulent flows and distributed phenomena like cavitation, the use of Computational Fluid Dynamics simulation is of great help. CFD modeling is easy to setup when the fluid domain to be studied can be decoupled from its surrounding environment. In such a case, the boundary conditions are well-known, and act as real sources with no reactions. If the flow is coupled to its environment, and this is frequently the case when performing transient simulation of a complete hydraulic circuit, the whole coupled system has to be simulated. This can be done by coupling the CFD model with the system-level model, the latter providing the boundary conditions of the meshed fluid domain. Coupling a system-level simulator with a CFD code can be performed through different approaches [7]:

- Weak coupling is well suited to the case of hydraulic models (or a part of it) which can be reduced to a static relationship between a small number of lumped variables. This relationship is usually characterized by performing batch runs in the CFD code, and gathering the results in lookup tables.

- Strong coupling with a CFD software is usually suited to the coupled transient simulation of an hydraulic component with its surrounding environment (remaining part of the hydraulic circuit, thermal exchanges, mechanical work done by the fluid, ...). Due to the nature of the solvers used by CFD code, co-simulation is generally the only way to couple a Modelica system-level simulator and a CFD solver [10].

The following sections describes a methodology for establishing a co-simulation between a Modelica simulation environment with an ODE/DAE solver and a 2D/3D model of flow computed by a CFD software. The physical, numerical, and computer-related aspects of co-simulation handled by this methodology are exemplified step-by-step on an application test case.

2 Application test case: modelling of a compound relief valve

2.1 Overall system description

The hydraulic system considered as test case is sketched out on Figure 1. This type of pilot-operated valve is usually used for limiting the pressure in an hydraulic circuit, by releasing a fraction of the flow from the inlet of the valve to its outlet. On the contrary of the direct-acting valves like the check-valves, operation of a compound relief valve is not affected by the flow going through it. A small pressure difference across the ball valve is enough to open the pilot valve indenpendently of the main flow rate from inlet to outlet. The corresponding Modelica system-level model is shown on Figure 2.

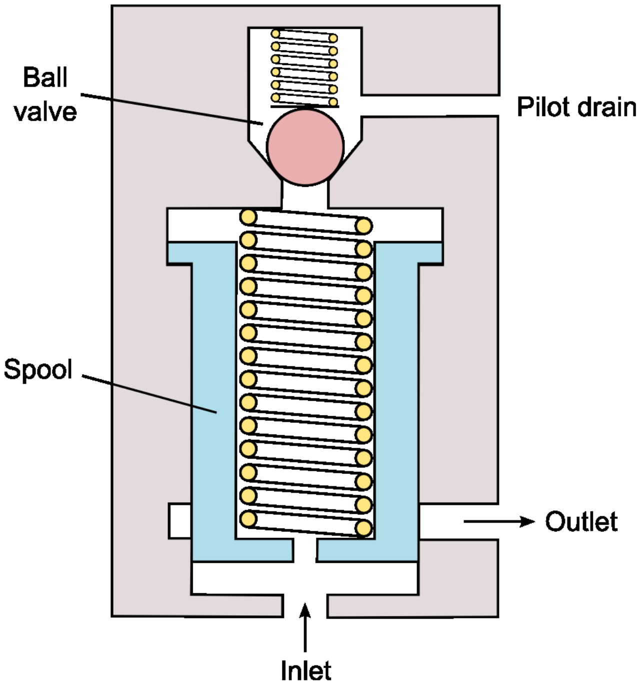

Figure 1: Sketch of the compound relief valve

The model tries to reproduce the transient behavior of the valve operating in a simplified hydraulic circuit made of a varying-flowrate pump and a load, on the left and right sides of the circuit sketch. At the center of the sketch, each component of the valve has a functional modeling counterpart on the Modelica model, using the hydraulic component design approach [2] [3]. The spool valve model is made of two parts:

1. The hydraulic part deals with the flow through the outer orifices of the spool, which areas depend on the spool position. It also describes the pressure dynamics in the top and bottom spool chambers using two hydraulic capacitive elements, which are explicitly materialized on the sketch. The corresponding pressures are acting on the spool body through the two piston-like elements.

2. The mechanical part directly represents the rigid body dynamics of the spool submitted to the pressure, spring and viscous damping forces.

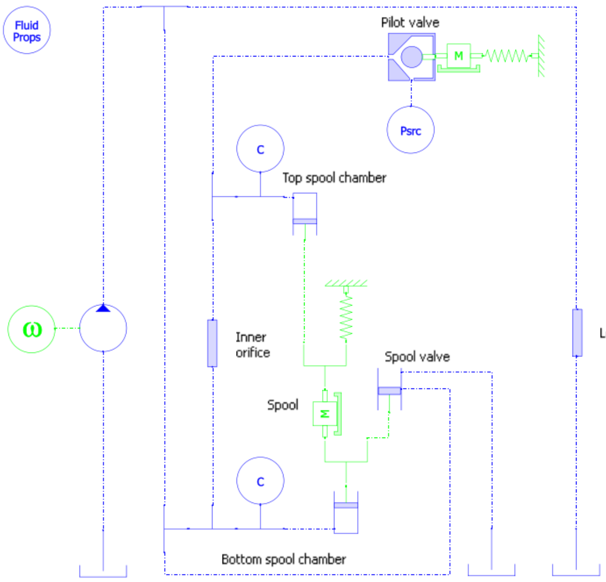

Figure 2: Sketch of the Modelica model of a compound relief valve.

At the top of the model sketch lies the pilot valve, which is a ball poppet valve. The hydraulic part of this model is aimed at being replaced by the detailed CFD model shown on Figure 3. A simplified Modelica model of the flow in the valve seat is described in the next section. The remaining mechanical part is the rigid body dynamics of the ball submitted to the static and dynamic fluid forces and to the spring force. In the model, the motion of the ball is limited in its travel by an ideal endstop which corresponds to the pilot valve being closed.

2.2 System-level model of the pilot valve

To study the stability of the coupled system resulting from the co-simulation of the detailed CFD model of the ball valve with the system-level of the hydraulic cicruit and mechanical part of the valve, a simplified equivalent system-level model of the ball valve is needed.

The model tries to reproduce the transient behavior of the ball poppet submitted to the pressure and viscosity forces arising from the turbulent flow in the valve

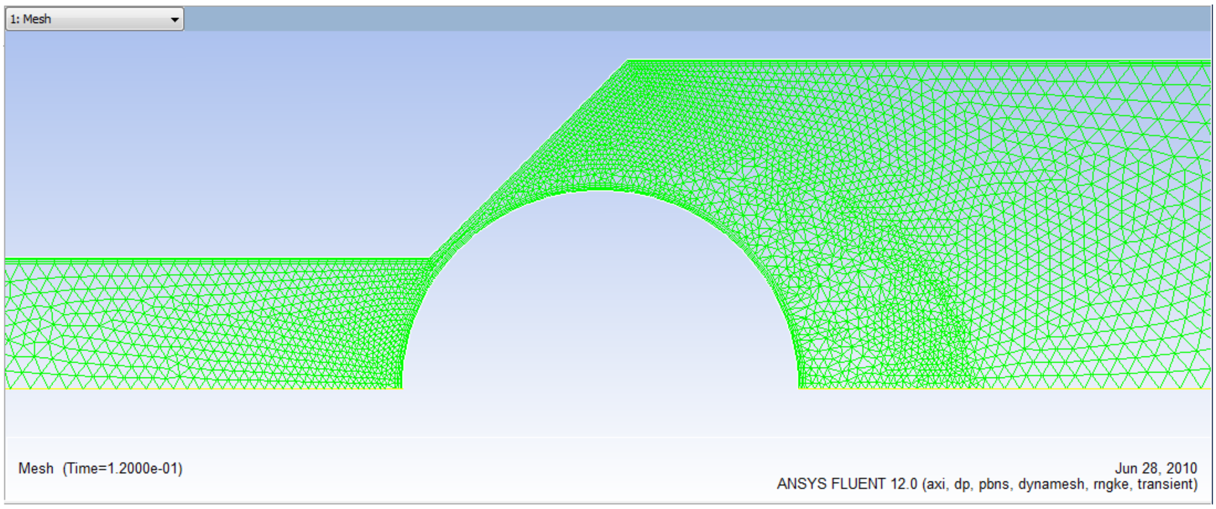
1: Mesh
Mesh (Time=1.2000e-01)
Jun 28, 2010
ANSYS FLUENT 12.0 (axi, dp, pbns, dynamesh, rngke, transient)

Figure 3: 2D axisymmetric fluid domain considered for detailed flow modeling.

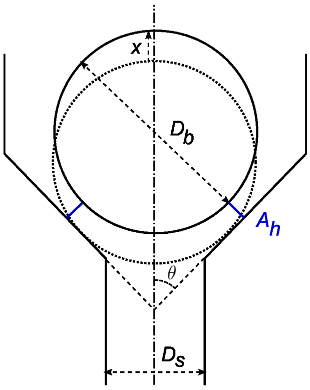

Figure 4: Geometry of the ball poppet valve.

seat. The modeling assumptions and analysis follow roughly [4]. The flow rate through the valve is given by the elementary orifice law:

$$q = c_d A_h(x) \sqrt{\frac{2\,(p_{up} - p_{dn})}{\rho(\bar{p})}} \tag{1}$$

where p_{up}, p_{dn}, and $\bar{p} = \frac{1}{2}(p_{up} + p_{dn})$ are the upstream, downstream and mean pressures, ρ the fluid density depending mainly on pressure, A_h the throat area, and c_d the discharge coefficient. The throat area, shown on Figure 4 depends geometrically on the ball lift x:

$$A_h(x) = \frac{\pi}{2} x \sin\theta \,(x \sin\theta + D_b) \tag{2}$$

with D_b the ball diameter and θ the half chamfer angle. The discharge coefficient is usually modeled as a smooth function of the flow number:

$$c_d = c_{d,max} \tanh\!\left(\frac{2\lambda}{\lambda_{crit}}\right)$$

the flow number λ being given by:

$$\lambda = \frac{D_h}{\mu(\bar{p})} \sqrt{2\rho(\bar{p})\,(p_{up} - p_{dn})}$$

λ_{crit} is the critical value of the flow number, corresponding to the laminar-turbulent transition. The hydraulic diameter D_h is directly linked to the ball lift by a geometrical relation:

$$D_h = 2x \sin\theta \tag{3}$$

When the ball is rising, the actual flow rate is bounded by the inlet area rather than the throat area. This is taken into account in the Modelica model by computing a maximum lift x_{max} such that in equation (2) the area becomes equal to the inlet area:

$$A_h(x_{max}) = \frac{\pi}{4} D_s^2$$

where D_s is the seat diameter. The effective lift value, bounded by x_{max} is then used to compute the throat area (2) and the hydraulic diameter (3).

The hydromechanical part of the model deals with the fluid forces acting on the ball. Static pressure forces are usually distinguished from dynamic forces stemming from the acceleration of fluid through the orifice. Under varying flow conditions resulting from varying lift, the pressure distribution along the ball can have large variations. The dependency of the pressure distribution, and hence of the active area used in the computation of the static pressures forces, could be studied using a preliminary *uncoupled* CFD computation [6]. The pressure field is computed at steady-state for different value of the ball lift. Some results are shown on Figure 5.

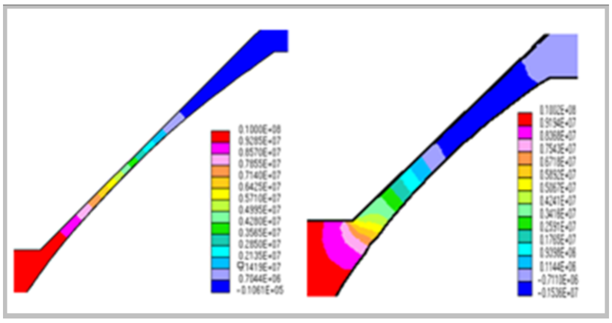

Figure 5: Pressure distribution in the orifice for two different fixed ball lifts.

For laminar flows, the pressure drop occurs at the minimum geometrical cross section

$$A_l = \frac{\pi}{4} D_b^2 \cos^2\theta$$

As it is shown on Figure 5 when transiting from small ball lift to high lift, the flow becomes turbulent and separates from the ball, and the corresponding pressure distribution moves towards the inlet [5]. A model, proposed by [4], takes this into account by considering the laminar-turbulent transition. The active area A_p is thus a fraction of the geometrical pressurized area:

$$\frac{A_p}{A_l} = \begin{cases} 1+\sqrt{\frac{\lambda}{\lambda_{crit}}}\,(f-1) & \text{if} \quad \lambda \le \lambda_{crit} \\ f & \text{if} \quad \lambda > \lambda_{crit} \end{cases} \qquad (4)$$

where $0 < f < 1$ is a turbulent active area factor, fitted on the steady-state CFD computations. Finally, the fluid forces acting on the ball are given by the sum of the static pressure forces and the dynamic or jet forces:

$$F = A_p\,(p_{up} - p_{dn}) + F_{jet}$$

with

$$\begin{aligned} F_{jet} &= \rho(\bar{p})\,q\,(v_{up} - v_{dn}\cos\theta) \\ &= 2\,c_d\,A_h(x)\,(p_{up} - p_{dn})\,(c_d\,\frac{A_h(x)}{A_h(x_{max})} - \cos\theta) \end{aligned}$$

3 Coupling principles and methodology

When trying to couple simulators, many different issues arise (types of physical coupling, numerical methods, software and hardware implementations) which seem intricate at a first glance. A general methodology [7] is required to prioritize these issues, thus avoiding suboptimal choices based only on computer-related contingencies. The decision flow chart shown on Figure 6 synthesizes this methodology used to couple a system-level simulator with third-party simulation software. The choices made for coupling the 1D system-level model of an hydromechanic system with a CFD model are emphasized on the flow chart. The next sections explain these choices.

3.1 Type of physical coupling

The first choice to be made is between weak coupling or strong coupling. Weak coupling means that the model could be reduced to a dynamic part, cascaded with a non-linear static part, obtained by statically characterizing the system on some operating points, typically using some batch run functionality of the CFD software. Weak coupling has many advantages from the numerical and the software point of view: the external model is generally reduced to data tables that are evaluated directly in the system-level simulator, using a table lookup library [7]. However, as we are interested in the study of the coupled system in general transient conditions, the weak coupling seems inappropriate. Static characterization - if it makes sense - would require a lot of batch run computations with varying boundary conditions. As the model reduction assumptions cannot be always made about a CFD model, strong or full coupling between the two models is considered here.

3.2 Solver interaction

The two models being strongly coupled, the next choice to be done deals with the numerical solvers used to perform the transient simulation of the coupled system. The best way is to use only one numerical integration solver, simulating a unique mathematical model, obtained by importing one of the two involved model into the other one. This way, the numerical issues are only related to the capabilities of the numerical integration scheme to handle the physically-originating stiffnesses of the coupled system, which could typically be handled by variable-timestep variable-order linear multistep methods. Importing the external model into the system-level simulator implies obviously that this model could be exported from its simulation environment, either the mathematical structure of the equations, or the value of the parameters to be fed to an equivalent model. However, with most of the CFD software [9], the mathematical structure of the fluid model (including conservation laws like the Navier-Stokes equations and constitutive relations like fluid properties or rheological

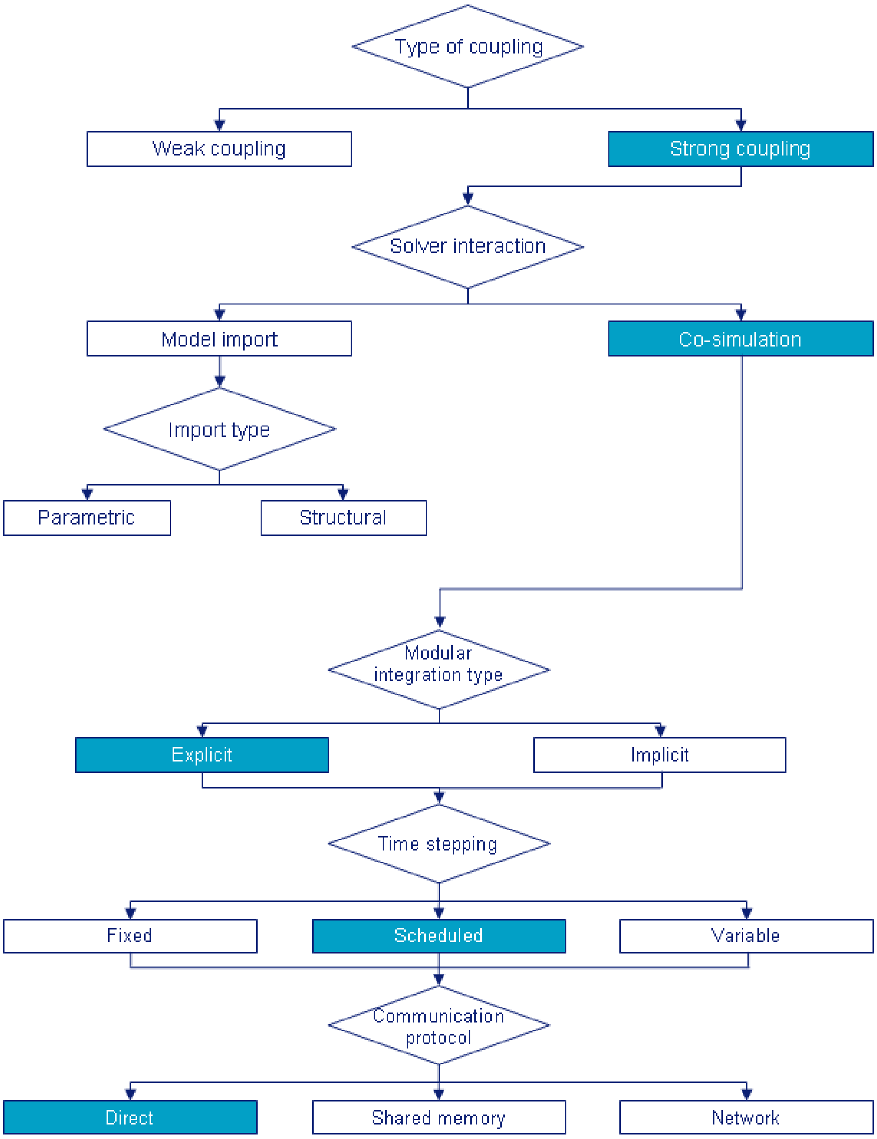

Figure 6: Decision flow chart for a general simulators coupling methodology.

models) is tightly linked with the numerical methods used to discretize it. Exporting the whole fluid model from a CFD software seems very difficult to realize, and the solver capabilities that would be required to solve such a model are very specific, and are not available in general system-level simulators. That is why the two solvers must be retained, and the transient simulation is performed through co-simulation.

3.3 System partitioning for co-simulation

In co-simulation, the two involved solvers exchange only a predefined set of variables at some communication time point. Thus, the whole coupled system must be partitioned in two subsystems, and the exchanged variables on the boundary have to be precisely defined. The Modelica model described on Figure 2 is modified in order to delegate the detailed modelling of the flow in the seat to the CFD code. The modified part of the AMESim-Modelica model is shown on Figure 7. A non-standard construction called *external connector* is introduced to enforce the causality at the boundary of the Modelica model. A Modelica model with such connectors is processed by the AMESim Modelica compiler in the usual a-causal way, except for the variables declared as external inputs which have their causality imposed by the outside world. The part of the model corresponding to the valve seat is removed from model and three causal/a-causal gateway models are introduced at the boundary of the model:

```
external connector ConnectForceInPosOut
  output SIunits.Velocity v;
  output SIunits.Position x;
```

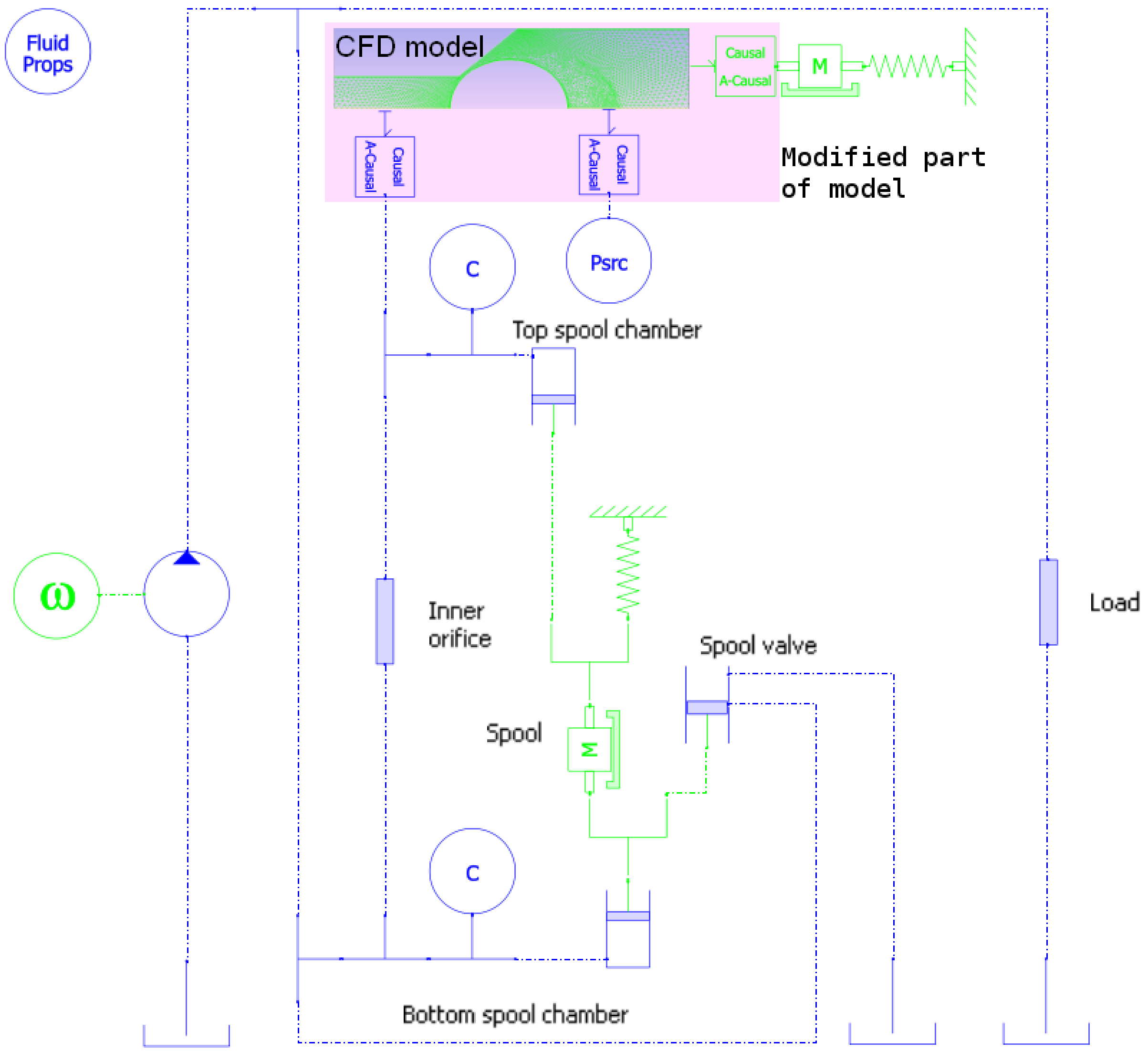

Figure 7: Modified Modelica model with external connectors.

```
  input   SIunits.Force     f;
end ConnectForceInPosOut;

model ForceInputPosOutput
  ConnectForceInPosOut c "Causal connector";
  Flange                 a "A-causal connector";
equation
  a.f + c.f = 0;
  a.x = c.x;
  a.v = c.v;
end ForceInputPosOutput;

external connector ConnectFlowInPressureOut
  output SIunits.Pressure_bar   p;
  input  SIunits.VolumeflowRate q;
end ConnectFlowInPressureOut;

model FlowrateInputPressureOutput
  ConnectFlowInPressureOut c  "Causal connector";
  FluidPort                  fp "A-causal connector";
equation
  fp.q + c.q = 0;
  fp.p = SIunits.from_bar(c.p);
end FlowrateInputPressureOutput;
```

These models act as sources or sensors from the system-level point of view and they carry the exchanged variables shown on Table 1.

In the system-level to CFD direction, the variable is directly read from the Modelica model, and applied as a space-constant boundary condition (pressure, wall position and velocity) to the CFD model. In the other direction, the variable is computed by integrating the related quantity (pressure giving force, fluid velocity giving flow rate) on the boundaries. Some additional unit conversions are performed in the external causal/a-causal connectors, since the Modelica library upon which the model is built works with SI units, whereas the CFD software user functions used for specifying the boundary conditions could be written in trade units.

Modelica Model	*Variable*	*Causality*
`ForceInputPosOutput`	x (ball position)	to CFD
`ForceInputPosOutput`	v (ball velocity)	to CFD
`ForceInputPosOutput`	f (fluid forces acting on ball)	from CFD

`FlowrateInput`
`PressureOutput`
(instance 1)
	p
	(pressure
	at inlet)
		to CFD

`FlowrateInput`
`PressureOutput`
(instance 1)
	q
	(flow rate
	at inlet)
		from CFD

`FlowrateInput`
`PressureOutput`
(instance 2)
	p
	(pressure
	at outlet)
		to CFD

`FlowrateInput`
`PressureOutput`
(instance 2)
	q
	(flow rate
	at outlet)
		from CFD

Table 1: Variables exchanged during co-simulation.

3.4 Modular integration type

The two subsystems being coupled through the state variables listed on Table 1, there is no need to set up a full algebraic coupling method, and hence only the explicit modular integration method is implemented on each simulator, as depicted on Figure 8. The simulation time is partitioned in macro time step, in which the integration process is strictly cascaded from a simulator to another, whatever the actual order. Inside a macro time step, the exchanged variables are held constant.

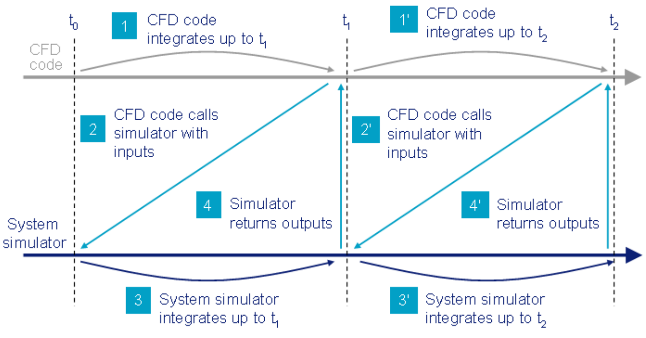
t$_0$
1 CFD code integrates up to t$_1$
t$_1$
1' CFD code integrates up to t$_2$
t$_2$
CFD code
2 CFD code calls simulator with inputs
2' CFD code calls simulator with inputs
4 Simulator returns outputs
4' Simulator returns outputs
System simulator
3 System simulator integrates up to t$_1$
3' System simulator integrates up to t$_2$

Figure 8: Explicit modular integration scheme.

3.5 Time stepping

The choice of a co-simulation time step T (or macro time step size) is determined by numerical stability of the co-simulated system with respect to the continuous case. Co-simulating two systems with explicit modular integration method introduces a sample with zero-order hold on the exchanged variables. The whole system obtained is a loop sampled system, which numerical stability may differ from the intrinsic physical stability of the fully coupled continuous system. Stability study of such loop sampled system is carried using the usual stability criteria from linear control system theory [8]. The coupled system comprising the seat and the mechanical part of the valve has many non-linearities, arising from geometry or pressure-dependent fluid properties. To study the stability of the loop sampled system, the system has to be linearized around some operating point. This can be performed using the Linear Analysis Tools from LMS Imagine.Lab, once the most dimensioning operating point has been recognised. At high ball lift, the flow in the seat becomes turbulent, and flow separation occurs. This means that the pressure drop moves towards the inlet, and the active area of the ball is decreased, while the influence of the momentum forces is increased. Stability study of the coupled system is thus carried by considering a linearized model of the mechanical part of the valve subjected to the sole jet or momentum forces of the fluid. In this simplified linearized system, co-simulation is taken into account by considering a sample and zero-order hold at rate T of the ball position. The hydraulic stiffness k_h represents the sensitivity of the jet forces with respect to the valve lift, at the desired operating point (typically high lift). The bloc-diagram of the resulting loop sampled system is depicted on Figure 9.

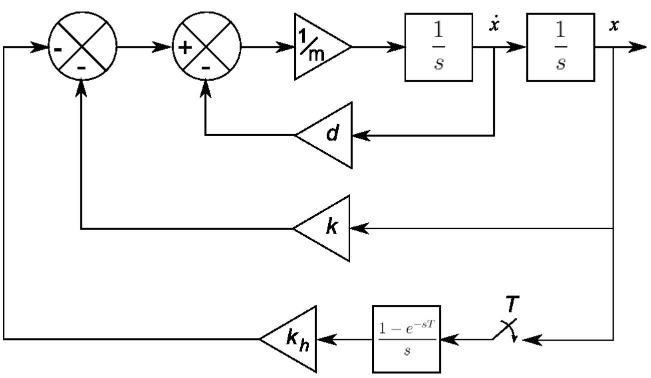

Figure 9: Bloc diagram of the loop sampled system considered for studying co-simulation stability.

The discrete transfer function of this loop sampled system is then given by:

$$\begin{aligned} H_{loop}(z) &= \mathbf{Z}\left[\frac{1}{m s^2 + d s + k}\,\frac{1-e^{-Ts}}{s}\,k_h\right] \\ &= (1-z^{-1})\,K_{loop}\,\mathbf{Z}\left[\frac{\omega_0^2}{s\,(s^2+2\,\zeta\,\omega_0\,s+\omega_0^2)}\right] \end{aligned}$$

with $K_{loop} = \frac{k_h}{m\,\omega_0^2}$ the loop gain, $\omega_0^2 = \frac{k}{m}$ the undamped natural frequency, and $\zeta = \frac{d}{2\sqrt{km}}$ the mechanical damping ratio. Finally the loop transfer function is given by:

$$H_{loop}(z) = \frac{K_{loop}\,(a\,z+b)}{z^2 - 2\,z\,e^{-\zeta\,\omega_0\,T}\,\cos\omega_d T + e^{-2\,\zeta\,\omega_0\,T}} \qquad (5)$$

with $\omega_d = \sqrt{1-\zeta^2}\,\omega_0$,

$$a = 1 - e^{-\zeta\,\omega_0\,T}\,\cos\omega_d T - \frac{\zeta\,e^{-\zeta\,\omega_0\,T}}{\sqrt{1-\zeta^2}}\,\sin\omega_d T$$

and

$$b = e^{-2\,\zeta\,\omega_0\,T} - e^{-\zeta\,\omega_0\,T}\,\cos\omega_d T + \frac{\zeta\,e^{-\zeta\,\omega_0\,T}}{\sqrt{1-\zeta^2}}\,\sin\omega_d T$$

At the operating point corresponding to high ball lift, the Linear Analysis Tool of LMS Imagine.Lab gives the following values for the loop transfer function coefficients: $K_{loop} = 3.85$, $\frac{\omega_0}{2\pi} = 50$ Hz and $\zeta = 0.47$. For a co-simulation macro-time step $T = 0.1\mu$s, the Bode diagram of H_{loop} is shown on Figure 10. Applying the Nyquist stability criterion to (5) with these values, the crossover frequency is $f_{cross} = 390$ Hz, giving a gain margin value $GM = 10\,\log_{10}(|H_{loop}(f_{cross})|) = 12$ dB, and a phase margin $PM = 28°$. These values are enough for ensuring the numerical stability of co-simulation, and hence the time step size may be scheduled up to 0.1μs at each macro step.

4 Communication protocol and computer implementation

The last choice to be done is related to the communication protocol implemented between the two simulators. Computational performance is usually considered for choosing the communication channel. Distributed co-simulation (two software running on two different processors or two different computers) leading to parallel processing speedup is interesting only if computational load is well balanced between the two software. This is clearly not the case when co-simulating a system-level model with a CFD code.

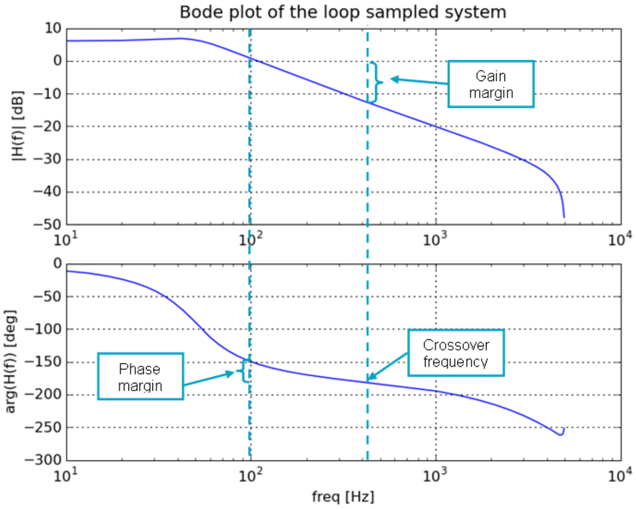

Figure 10: Bode diagram of the loop transfer function for a co-simulation time step $T = 0.1\mu$s.

The load is strongly unbalanced: typically, the complete transient simulation of the step response with the system-level model takes less than 1 minute, whereas the valve simulated with the detailed CFD model of the flow in the seat takes about 1 day on a quad core CPU computer. Distributed co-simulation being useless, there is no need for multiprocessor or multicomputer communication protocol. A direct local communication link is therefore used on the same processor. The Modelica model with embedded LMS Imagine.Lab AMESim solver and Modelica libraries is compiled as a shared library using the Generic Co-simulation Interface of LMS Imagine.Lab, and then is linked locally with the CFD software, namely ANSYS Fluent. At each macro time step, ANSYS Fluent is acting as the master simulator: it schedules the next macro step size, and calls the AMESim solver with the input variables, according to the modular integration scheme depicted on Figure 8. Output variables are exchanged at the end of the current macro time step, and a new macro step can take place.

5 Some co-simulation results

As an example, we consider computing the step response of the check valve, by increasing the pressure at inlet from 0 to 10 bar in 1 ms. Some of the exchanged variables (ball lift, hydraulic forces, upstream pressure, flow rate) are monitored in LMS Imagine.Lab during co-simulation (Figure 11), while the other flow quantities - static pressure, velocity, Reynolds number, etc - are displayed directly in the CFD software (Figure 12).

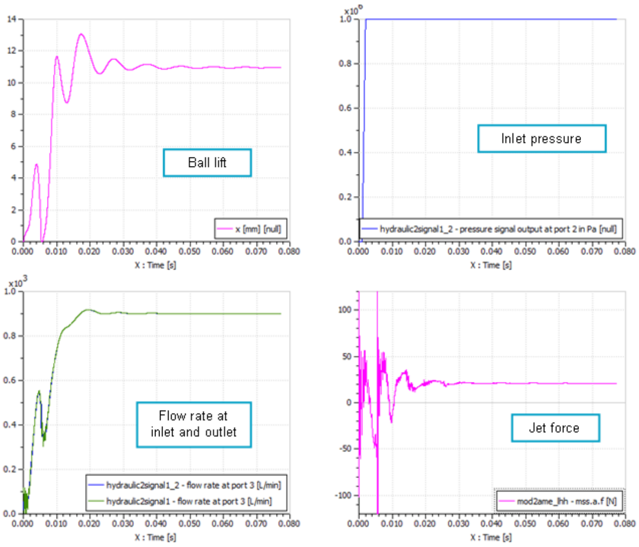

Figure 11: Exchanged variables monitored in LMS Imagine.Lab.

The variables monitored in LMS Imagine.Lab AMESim are the typical responses to such an input of pressure. It is interesting to note that the jet force values (i.e. hydrodynamic forces) can be reached, since it is a key factor in the design of such and hydro-mechanical systems.

The map of the static pressure can be represented for the whole geometry in the CFD software. It allows accessing detailed results that can not be reached in usual lumped-parameter models. The map of static pressure obtained here at different simulation times corresponds to the expected results for such a use-case of poppet with ball on conical seat [4].

6 Conclusions

We can finally conclude that the strong coupling of some Modelica components with a CFD model combining two different software such as LMS Imagine.Lab AMESim for the system-level tool and ANSYS Fluent for the CFD software was successful. Beside the fluid power test case exemplified in this paper, the methodology was also applied to the modeling of a full direct diesel injection system [10] using the 3D CFD code Principia Eole to accuratly predict the cavitation transients in a nozzle under multiple injection conditions. It would require some additional works to apply this methodology outside fluid power to confirm that it brings a real added-value for other types of industrial applications. Another promising way of research lies in the Functional Mock-up Interface (FMI, [11]), which standardises the coupling be-

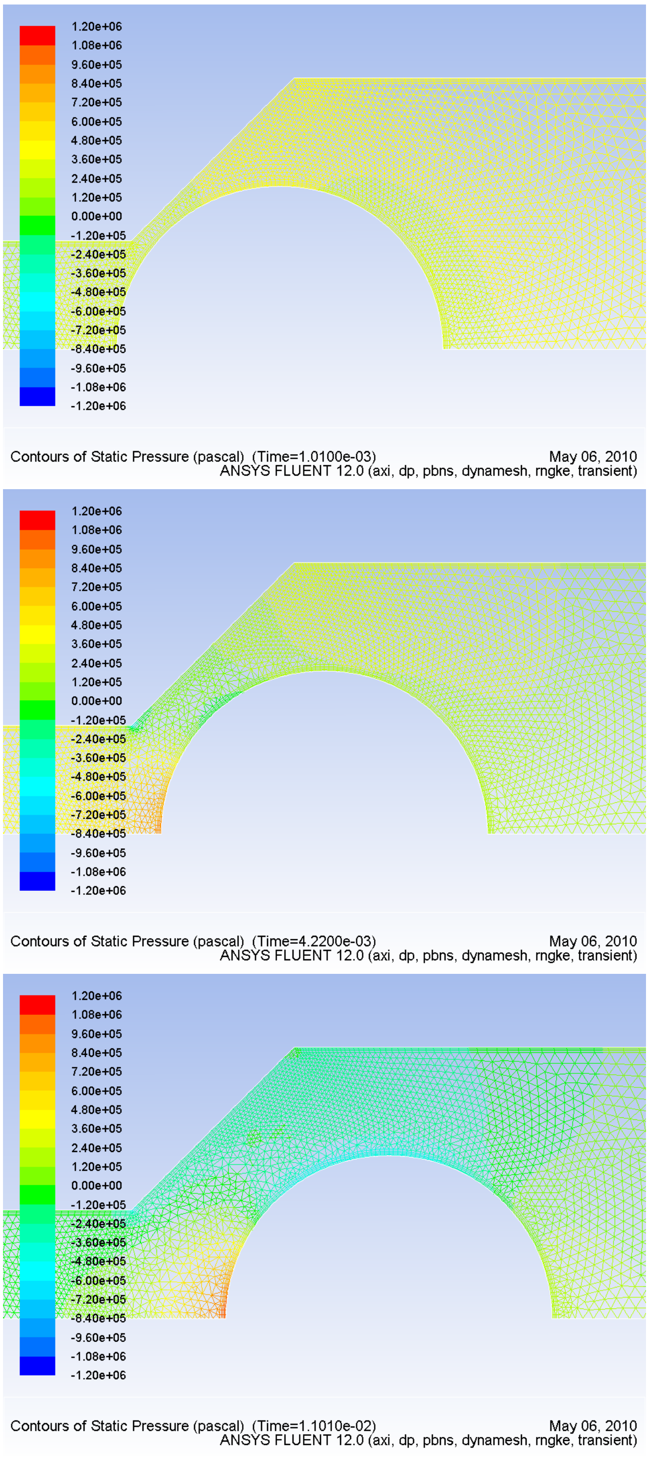
1.20e+06
1.08e+06
9.60e+05
8.40e+05
7.20e+05
6.00e+05
4.80e+05
3.60e+05
2.40e+05
1.20e+05
0.00e+00
-1.20e+05
-2.40e+05
-3.60e+05
-4.80e+05
-6.00e+05
-7.20e+05
-8.40e+05
-9.60e+05
-1.08e+06
-1.20e+06
Contours of Static Pressure (pascal) (Time=1.0100e-03)
May 06, 2010
ANSYS FLUENT 12.0 (axi, dp, pbns, dynamesh, rngke, transient)
Contours of Static Pressure (pascal) (Time=4.2200e-03)
May 06, 2010
ANSYS FLUENT 12.0 (axi, dp, pbns, dynamesh, rngke, transient)
Contours of Static Pressure (pascal) (Time=1.1010e-02)
May 06, 2010
ANSYS FLUENT 12.0 (axi, dp, pbns, dynamesh, rngke, transient)

Figure 12: Pressure distributions at t = 1 ms, 4.2 ms and 11 ms.

tween two solvers through co-simulation. The general simulator coupling methodology described here could benefit from this FMI specifications, especially regarding the implementation of implicit modular integration and time stepping techniques by CFD software editors.

Acknowledgments

This work was in part supported by DGE in the ITEA2 Eurosyslib (06020) project under contract number 07.2.93.0146.

References

[1] Akers A., Gassman M., Smith R., Hydraulic Power System Analysis, CRC Press, 2006.

[2] Mc Cloy D., Martin H.R., Control of Fluid Power: Analysis and Design, Wiley, 1980.

[3] Lebrun M., Richards C.W., How to create good models without writing a single line of code, 4th Scandinavian Int. Conference on Fluid Power, Linköping, Sweden, June 1997, vol. 1.

[4] Mittwollen N., Michl T., Breit R., Parametric hydraulic valve model including transitional flow effects. In: Proceedings of the 2nd MATHMOD Vienna, IMACS Symposium on Mathematical Modelling, Febr. 5-7, 1997, TU Vienna, Austria, Editors: I. Troch, F. Breitenecker, ARGESIM Report No.11.

[5] Clavier A., Alirand M., Vernat F., Sagot B., Local Approach to Improve the Global Approach of Hydraulic Forces in Ball Poppet Valves, 4th In. Symposium on Fluid Power, Wuhan, China, April 2003.

[6] Baudry X., Mare J.C., Linking CFD and lumped parameters analysis for the design of flow compensated spool valve, 1st Fluid Power Net Int. PhD Symposium, Hamburg, Germany, June 2000.

[7] Neyrat S., Viel A., Strong Coupling LMS Imagine.Lab Modelica with CFD Software, Report of the sub work package 2.4 of the ITEA2 Eurosyslib Project, LMS Imagine, June 2010.

[8] Franklin G.F., Powell J.D., Workman M.L., Digital Control of Dynamic Systems, Addison Wesley, 1997.

[9] Chung T.J., Computational Fluid Dynamics, Cambridge University Press, 2002.

[10] Marcer, R., Audiffren C., Viel A., Bouvier B., Walbott A., Argueyrolles B., Coupling 1D System AMESim and 3D CFD EOLE models for Diesel Injection Simulation. In: Proceedings of the ILASS Europe 23rd Annual Conference on Liquid Atomization and Spray Systems, Brno, Sept. 2010.

[11] Functional Mock-Up Interface for Co-Simulation 1.0 Specification, MODELISAR project (ITEA2 - 07006) and consortium, October 12th, 2010.