

ThermoBondLib – A New Modelica Library for Modeling Convective Flows

François E. Cellier
ETH Zürich
Switzerland
FCellier@Inf.ETHZ.CH

Jürgen Greifeneder
Technische Universität Kaiserslautern
Germany
Greifeneder@EIT.Uni-Kl.DE

Abstract

This paper describes a new library designed for modeling convective flows in physical systems. The library is based on bond graph technology. Thermo-bonds are introduced as a means to capture the convective flow of the internal energy of matter through a physical system. ThermoBondLib is a companion library to the BondLib and MultiBondLib libraries that were released in 2005 and 2006, respectively.

Keywords: Bond Graph, Thermo-bond Graph, Convective Flow

1 Introduction

1.1 Introduction to Thermo-bond Graphs

Bond graphs [1] describe the flow of power through a physical system. Each power flow can be written as the product of two variables, one extensive, the other intensive. For example, electrical power can be written as the product of voltage and current, whereas mechanical translational power can be written as the product of force and velocity.

Since all physical systems have to comply with energy conservation laws, a tool that balances all power flows in a physical system can, in principle, be used to model any such system.

A Modelica library based on bond graph technology, BondLib [2], was released in 2005 and presented at the 4th Modelica conference in Hamburg-Harburg. The library won the 1st price for a free Modelica library at that conference.

Whereas bond graphs are capable of describing

won the 1st price for a free Modelica library at that conference.

When dealing with 2D or 3D mechanical systems, the d'Alembert principle needs to be formulated several times, once for each spatial direction. Also, the principle needs to be formulated separately for translational and rotational motions. This calls for a vector representation of bonds, which is precisely the framework that MultiBondLib has been based upon.

MultiBondLib also offers several wrapped libraries for higher-level descriptions of 2D and 3D mechanical systems. These higher-level descriptions are internally implemented as multi-bond graphs. The multi-bond graphs are then directly translated to the equation layer using the matrix/vector notation of Modelica. This was simpler and more efficient than translating the multi-bond graphs first down to regular bond graphs in a graphical fashion.

When dealing with convective flows, there is yet another complication to be considered [4-7]. When considering a mass moving macroscopically from one place to another, that mass carries along with it its internal energy of matter, U :

$$U = T \cdot S - p \cdot V + g \cdot M \quad (1)$$

where T denotes temperature, S is the entropy, p represents pressure, V stands for the volume, g symbolizes the Gibbs potential (specific enthalpy), and M finally captures the mass.

The flow of internal energy, $Udot = dU/dt$, can be written as:

$$Udot = T \cdot Sdot - p \cdot q + g \cdot Mdot \quad (2)$$

where $q = Vdot$ represents the volumetric flow rate. Consequently, a mass flow is always accompanied by a heat flow and a volumetric flow.

For this reason, mapping convective flows directly down to regular bond graphs is once again cumbersome. The size of these bond graphs would grow too fast.

The ThermoBondLib library presented in this paper provides another vector bond graph representation, whereby each thermo-bond is composed of three regular bonds, one representing mass flow, a second representing volumetric flow, and a third representing heat flow. A thermo-bond can be envis-

aged as a parallel connection of three regular bonds, as shown in Fig.1:

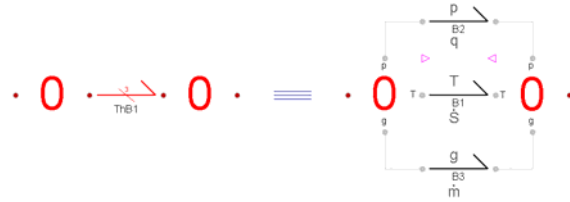


Figure 1: Representation of a thermo-bond

For reasons of efficiency, the thermo-bond model has not been composed in ThermoBondLib in a graphical fashion, but rather by equations directly. Hence the bond graph of Fig.1 offers only a conceptual decomposition of a thermo-bond.

1.2 The Thermo-bond Connectors

The thermo-bond connectors would, in analogy to the regular bond connectors have to carry at least seven variables: the three effort variables, T , p , and g ; the three flow variables, $Sdot$, q , and $Mdot$; and finally, the directional variable, d , that assumes a value of $d = -1$ at the connector at which the bond emanates and a value of $d = +1$ at the connector to which the bond leads.

In reality, the thermo-bond connector is an 11-tuple. It also carries (for convenience) the three state variables, S , V , and M , and in addition a Boolean variable, $Exist$, that is set true when there is mass to be transported, and is set false when the mass is close to zero. The Boolean variable is useful, because the models often operate on specific flows, i.e., flows per unit of mass, and Modelica becomes agitated when we attempt to divide a flow rate by zero.

The thermo-bond connector is depicted in Fig.2.

```
connector ThBondCon "Bi-directional thermo-bond graph connector"
  Modelica.SIunits.Temperature T "Temperature";
  Modelica.SIunits.Pressure p "Pressure";
  Modelica.SIunits.SpecificEnthalpy g "Gibbs potential";
  Modelica.SIunits.ThermalConductance Sdot "Entropy flow";
  Modelica.SIunits.VolumeFlowRate q "Volume flow";
  Modelica.SIunits.MassFlowRate Mdot "Mass flow";
  Modelica.SIunits.Entropy S "Entropy";
  Modelica.SIunits.Volume V "Volume";
  Modelica.SIunits.Mass M "Mass";
  Real d "Directional variable";
  Boolean Exist "True if substance exists";
end ThBondCon;
```

Figure 2: Thermo-bond connector

The measurement unit of entropy flow is currently set to *ThermalConductance* rather than the dimensionally compatible *EntropyFlowRate*, because the

latter unit is still missing in the Modelica Standard Library.

Just like in the case of the regular bond connectors, the thermo-bond connectors come in three varieties, one used for a-causal thermo-bonds, and the other two used for the two types of causal thermo-bonds.

1.3 Advantages of Bond Graph Modeling

Why is it useful to represent component models of a Modelica library internally by means of bond graphs?

To demonstrate the usefulness of this approach to modeling physical systems, let us consider the model of a heat conduction element offered as a component model of the heat transfer sub-library of the Modelica standard library. The model is shown in Fig.3.

```

model ThermalConductor
  "Lumped thermal element transporting heat without storing it"
  extends Interfaces.Element1D;
  parameter SI.ThermalConductance G "Constant thermal conductance of material";
equation
  Q_flow = G*dT;
end ThermalConductor;

```

Figure 3: Conduction model of Standard Library

The thermal conductor is modeled in the same way as a regular electrical resistor. Unfortunately, this model is incorrect.

To demonstrate the problems with this model, let us look at the bond graph representation of a resistor, as shown in Fig.4.

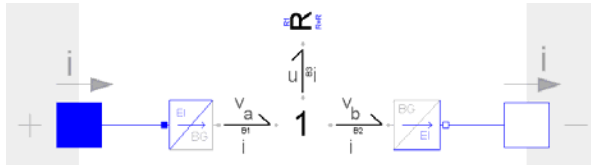


Figure 4: Bond graph representation of a resistor

The electrical power leaving the positive terminal equals $v_a \cdot i$, whereas the power arriving at the negative terminal equals $v_b \cdot i$. At the 1-junction, the potential drop is being calculated, and the difference in power, $u \cdot i$, gets sent to the resistor.

What happens to the power as it arrives at the resistor? It cannot vanish. It gets dissipated into heat.

In an electrical resistor model, it may make sense to ignore (exclude from the model) the thermal phenomena. However in a thermal resistor, this makes no sense whatsoever.

Thus, the dissipated heat (generated entropy) needs to be routed back into the thermal network.

Hence a correct model of the thermal conductor would have to look as depicted in Fig.5:



Figure 5: Bond graph model of thermal conductor

where the heat exchanger, HE, is internally represented as shown in Fig.6.

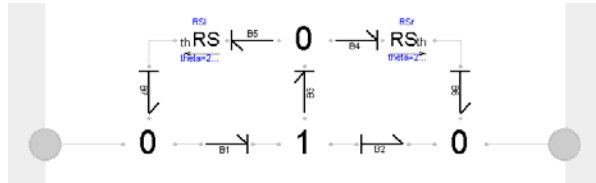


Figure 6: Bond graph model of heat exchanger

The generated heat is divided in two at the top 0-junction, and the two halves of the generated entropy flows are rerouted to the nearest 0-junctions to the left and the right of the 1-junction.

Using the bond graph approach, the shortcomings of the model contained in the Modelica Standard Library became obvious at once.

2 Potential Equilibration

In convective flows, each of the three sub-flows can equilibrate its potential variable separately.

When two bodies are in contact with each other, heat diffusion takes place between them. Heat diffusion is modeled by the thermal conductor discussed earlier. However in the new model, the two connectors to the left and the right are now thermo-bond connectors. The enhanced heat exchanger model is shown in Fig.7.

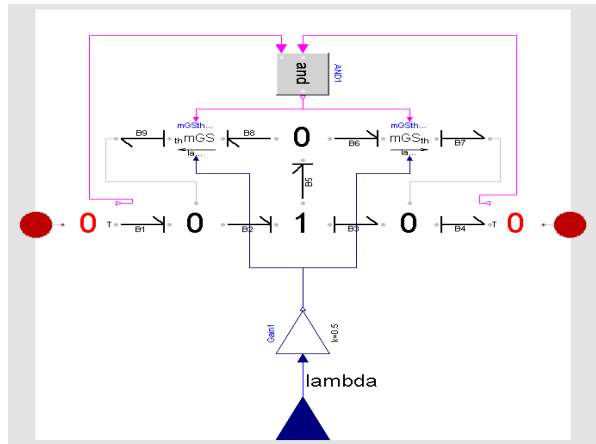


Figure 7: Thermo-bond graph model of heat exchanger

The resistive source, RS, was replaced by a conductive source, GS. The thermal conduction, λ , is imported as a modulating signal, rather than being treated as a parameter value, and finally, heat conduction only takes place if both neighboring substances exist.

The red 0-junctions are special junctions that take the red vector bond apart and make the individual component flows available.

The second type of potential equilibration is the volume work, represented in ThermoBondLib by the pressure/volume exchanger, PVE. The pressures in two neighboring volumes will equilibrate, if the two volumes are separated by a movable membrane. The corresponding model is shown in Fig.8.

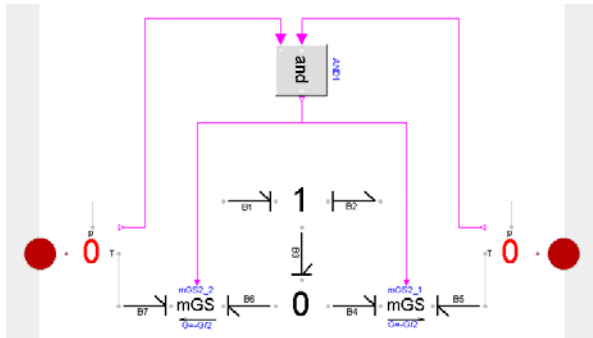


Figure 8: Thermo-bond graph model of volume work

Volume work is a dissipative phenomenon. In the process of pressure equilibration, entropy is being generated that is fed back into the thermal ports of the neighboring 0-junctions.

The conductance values of the volume work element are negative, because a positive pressure difference leads to a negative volume flow: if the pressure on the left side is larger than that on the right side, the membrane is pushed to the right, thereby making the volume to the left larger and not smaller.

The two bonds to the left and the right of the conductive sources are pointing in opposite direction, because a negative volume flow leads to a positive entropy flow in accordance with Eq.(2).

The third type of potential equilibration is the mixing element, represented in ThermoBondLib by the mass exchanger, ME. The Gibbs potentials in two neighboring volumes may equilibrate, if the two masses are able to mix. The corresponding model is shown in Fig.9.

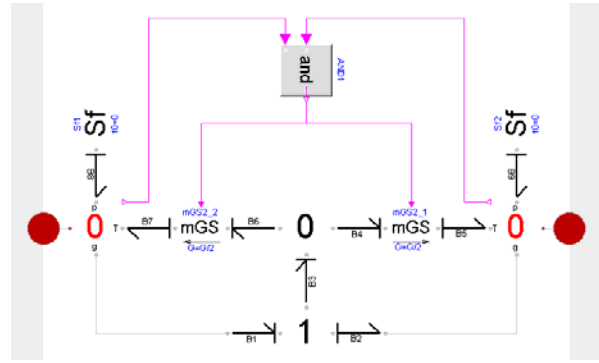


Figure 9: Thermo-bond graph model of mixing

Also mixing is a dissipative phenomenon. In the process of mixing two fluids, mixing entropy is being generated.

The ME model is a bit more problematic than the other two potential equilibration models, because mass cannot really flow without taking its volume and heat along. The model can be used to explain the cause of mixing entropy, but in practice, more complex models will be needed in most cases to describe mixing phenomena.

The three potential equilibration elements have in common that they don't require state information. All three elements equilibrate one potential only, i.e., they don't lead to associated flows of the other two types. All three phenomena are dissipative in nature, i.e., generate entropy in the process of potential equilibration.

3 The Substance Models

Whereas the transport models, such as the potential equilibration models, are responsible for computing the flows in and out of volumes, the substance models compute the three potentials of a volume. They also compute the three state variables by means of integrating over the difference between inflows and outflows. In bond graph terminology, the substance models are capacitive fields, CF.

Unfortunately, the substance models cannot be coded in a fully substance-independent fashion. Separate models need to be created for different types of substances. Of course, these models have a lot in common with each other, and inheritance schemes can be set up that restrict the recoding between similar models to the bare minimum, but this hasn't been accomplished yet. At the current time, substance models have only been made available for air, water, and water vapor.

The air model shall serve here as an example. It is depicted in Fig.10. The model is fully coded by means of equations. It computes the three potentials, T , p , and g , and determines the values of the three state variables, S , V , and M , by integrating the three flow variables, $Sdot$, q , and $Mdot$.

```

model Air "Capacitive field representing air"
  extends Interfaces.PassiveOnePort;
  parameter Modelica.SIunits.Entropy S0=6.81010184 "Entropy if no air";
  parameter Modelica.SIunits.Volume V0=0.83112221e-3 "Volume if no air";
  parameter Modelica.SIunits.Mass M0=1e-3 "Mass if no air";
  parameter Modelica.SIunits.SpecificHeatCapacity cp=1004.0
    "Heat capacity of air at constant pressure";
  parameter Modelica.SIunits.SpecificHeatCapacity R=287.2 "Gas constant";
  parameter Modelica.SIunits.Mass epsM=0.5e-6
    "Smallest mass distinguishable from zero";
  parameter Boolean fict=false "True is fictitious values are used";
  parameter Modelica.SIunits.Temperature T_fict=298.53
    "Fictitious temperature is no air";
  parameter Modelica.SIunits.Pressure p_fict=1e5
    "Fictitious pressure if no air";
  Modelica.SIunits.Entropy S_int "Entropy of air";
  Modelica.SIunits.Volume V_int "Volume of air";
  Modelica.SIunits.Mass M_int "Mass of air";
  Modelica.SIunits.SpecificHeatCapacity cv
    "Heat capacity of air at constant volume";
  Modelica.SIunits.SpecificVolume v "Specific volume";
  Modelica.SIunits.SpecificEntropy s "Specific entropy";
  Real ln_v "Natural logarithm of specific volume";
equation
  der(M_int) = Mdot;
  der(S_int) = Sdot;
  der(V_int) = q;
  Exist = M_int > epsM;
  cv = cp - R;
  v = if Exist then V_int/M_int else 0;
  s = if Exist then S_int/M_int else 0;
  ln_v = Modelica.Math.log(V_int/M_int);
  p = if Exist or not fict then T*R*M_int/V_int else p_fict;
  T = if Exist or not fict then 293.15*exp((s - 6813.7 - R*(ln_v + 0.17245))
    /cv) else T_fict;
  g = T*(cp - s);
  M = if Exist then M_int else M0;
  V = if Exist then V_int else V0;
  S = if Exist then S_int else S0;
end Air;

```

Figure 10: Model of air storage

Since air can be considered an ideal gas, the pressure may be computed from the ideal gas equation. All fluids are subject to an *equation of state*, and that equation is being used to determine the pressure.

All fluids are also subject to a *caloric equation of state*. That equation is being used to determine the temperature. Once the temperature and the specific entropy are known, the Gibbs potential can be determined as well.

Additional code has been added to prevent the model from dividing by zero in case the volume gets completely emptied out.

4 Evaporation and Condensation

A second class of transport phenomena beside from potential equilibration is the phase change of a fluid. Water can boil off (evaporate), and it can condensate out, either in the bulk or on a cold surface.

Whereas some transport phenomena are dissipative, like the potential equilibration phenomena, oth-

ers are reversible. Evaporation and condensation are reversible transport phenomena. In the process of evaporation, the activation energy is taken out of the thermal domain. Sensible heat gets converted to latent heat. However in the reverse process of condensation, the previously borrowed latent heat gets converted back to sensible heat¹.

How much evaporation/condensation takes place, i.e., where the flow equilibrium is between the liquid and the gaseous forms of a fluid, is determined by minimizing the overall energy of the system. However in practice, this energy minimization problem is hardly ever solved on-line. The ThermoBondLib code, like most other such programs, computes the amount of evaporation/condensation using steam tables.

5 The Pressure Cooker

We are now able to code a first example using the ThermoBondLib library. To this end, we shall model a pressure cooker.

Inside the pressure cooker, there are three types of substances: water, air, and water vapor. As the pressure cooker is being heated, more water boils off, producing additional water vapor. The water vapor would like to occupy more space than the water in its liquid form, but it cannot, because the total volume of the pressure cooker is fixed. Consequently, the pressure values inside the pressure cooker are rising. Fig.11 depicts a preliminary model representing the pressure cooker.

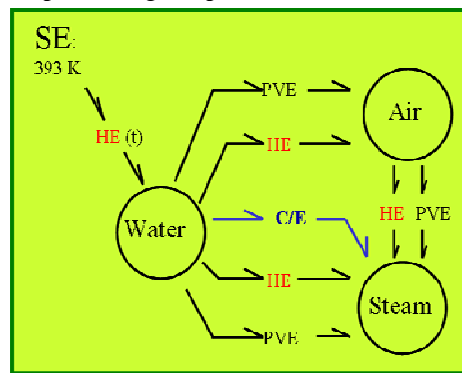


Figure 11: Basic pressure cooker model

There are three different volumes containing three different substances. Potential equilibration in the form of heat exchange and volume work is al-

¹ In the classical model, sensible heat is taken out of the liquid phase during evaporation, but gets added to the gaseous phase during condensation.

lowed to occur. Furthermore, there is evaporation and condensation taking place between the water volume and the steam volume.

An improved model takes into account that heating and cooling take place not only through the bottom of the pot, but also through the metal walls, i.e., we need to add boundary layers representing the air and the steam in the vicinity of the metal walls. The enhanced model is shown in Fig.12.

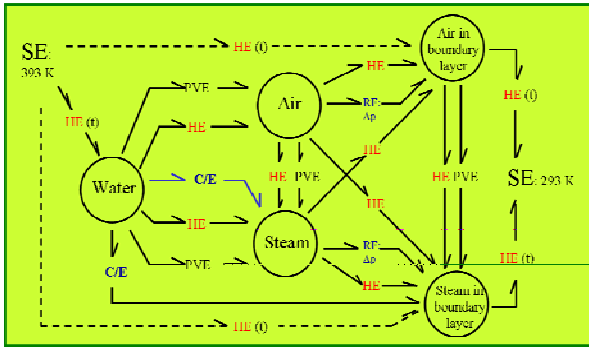


Figure 12: Enhanced pressure cooker model

Special volume work elements are used between the bulk and the two boundary layers. These models ensure that the volume of the boundary layers doesn't change.

We are now ready to encode the model using ThermoBondLib, as shown in Fig.13.

There are five separate 0-junctions representing the five different substances. In some cases, individual 0-junctions had to be split into two or even three 0-junctions to provide for a sufficiently large number of connectors.

At each of these 0-junctions, a capacitive field is attached, modeling the properties of the five substances. Between the 0-junctions, there are placed all the transport models representing the exchange of mass, volume, and heat between the five substances.

Whereas this model is still fairly simple, it already occupies an entire screen. Model wrapping techniques ought to be used to represent the model at a yet higher level, below which the thermo-bond graphs can be hidden.

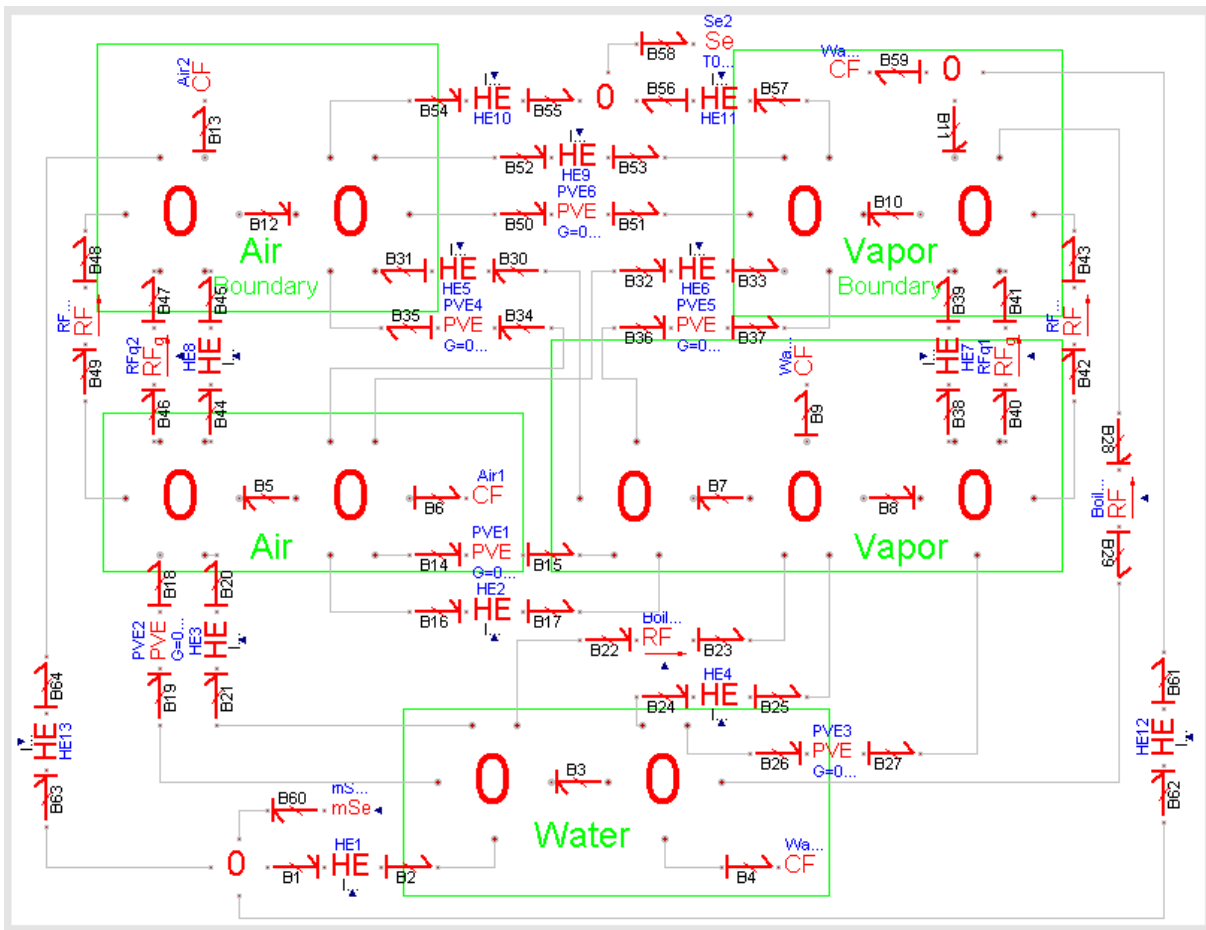


Figure 13: Coded pressure cooker model

Such a wrapped model might represent the control volumes as containers, and the transport models between them as pipe segments. Yet, the wrapping of thermo-bond graphs representing convective flows hasn't been accomplished yet. Wrapped sub-libraries shall be added to ThermoBondLib at a later time.

Some simulation results are shown in Figs.14 and 15. The pressure cooker is placed on a hot surface at time 0. Cold water is poured over it after 10,000 seconds. Whereas the five temperature values are significantly different, the pressure values are almost indistinguishable.

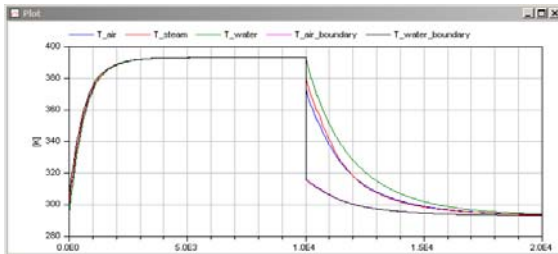


Figure 14: Temperature values of pressure cooker

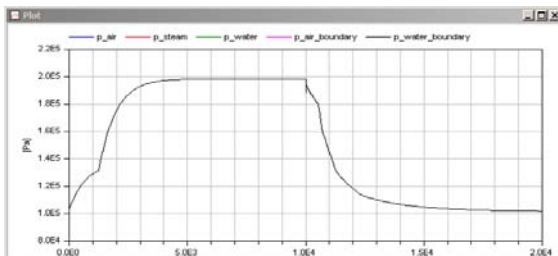


Figure 15: Pressure values of pressure cooker

After approximately 1300 seconds, the water begins to boil, which leads to a knee in the pressure curves.

6 The Air Balloon

As a second example, we shall consider a bottle containing compressed air, from which an air balloon is to be filled.

Here, we are facing a new complication. If we were to model the bottle containing compressed air by means of the same capacitive field as before, the volume of the bottle would shrink as air is allowed to escape from it. Hence we need to create a new capacitive field that models the storage of air under conditions of constant volume. This model is shown in Fig.16.

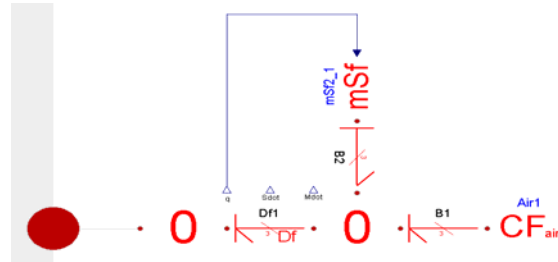


Figure 16: Air storage under isochoric conditions

The isochoric air storage model is built hierarchically from a regular air storage model. The volumetric flow coming out of the storage is measured using a flow sensor, Df. A counter-flow of equal magnitude is generated by the modulated flow source, mSf. It takes its energy from the thermal domain. Hence the bottle keeps a constant volume. Instead of shrinking in size, the bottle cools down.

We are now ready to model the air balloon system. The model is shown in Fig.17.

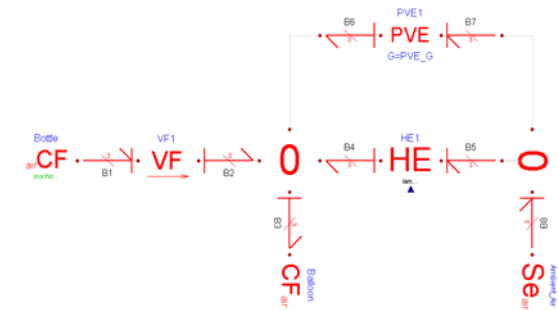


Figure 17: Air balloon model

The CF element to the left of Fig.17 represents the bottle. It is modeled using the isochoric air storage element of Fig.16. The CF element in the center represents the balloon. It is modeled using the regular air storage element of Fig.10.

The effort source, Se, to the right represents the ambient air. Between the balloon and the ambient air, potential equilibration (heat exchange and volume work) are allowed to take place.

The transport of a fluid across a pipe satisfies the wave equation. When cutting the pipe into individual segments, each segment can be modeled using a container, a capacitive field, representing the compressibility of the fluid, and using a transporter, an inductive field, representing the inertia of moving mass.

In the model of Fig.17, the volume flow element, VF, represents the transporter. A volume flow is obtained that is caused by the pressure difference between the two connectors. The volume flow induces a proportional mass flow and a proportional heat flow.

The volume flow element is depicted in Fig.18. The top portion of the bond graph represents the volumetric flow. The volumetric flow is being computed inside the inductor, I. A flow sensor, Df, measures the flow and induces proportional mass and entropy flows using internal flow sources, mSf. The center portion of the bond graph represents the entropy (heat) flow portion of the model, whereas the lowermost part calculates the mass flow.

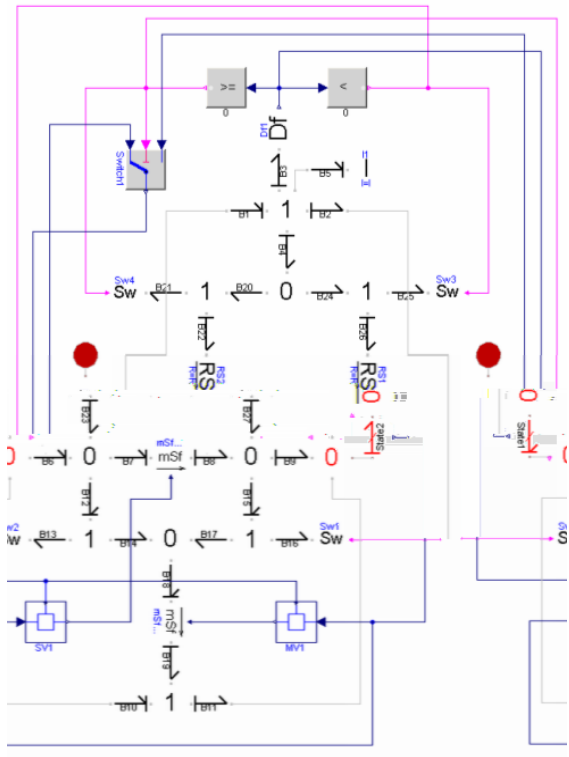


Figure 18: Inductive field representing transporter

As is to be expected, the temperature, pressure, and mass in the bottle decrease, whereas the volume remains constant. In contrast, mass and volume in the balloon increase, whereas temperature and pressure remain almost constant.

Some simulation results are shown in Figs.19-21. Fig.19 shows the temperature values in the bottle and the balloon; Fig.20 depicts the volumes of the two storages; and Fig.21 presents the two air masses.

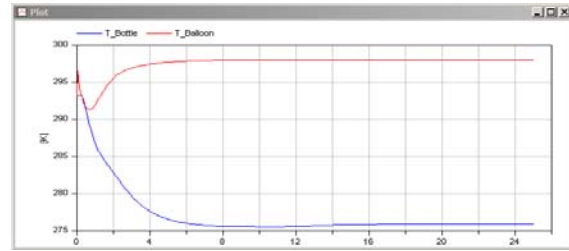


Figure 19: Temperature values of bottle and balloon

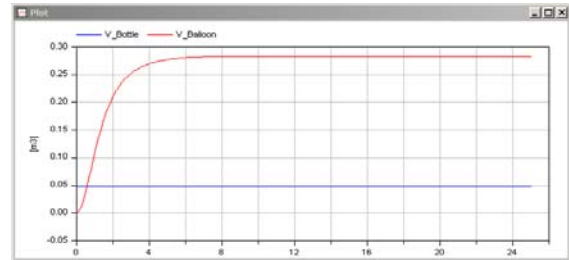


Figure 20: Volumes of bottle and balloon

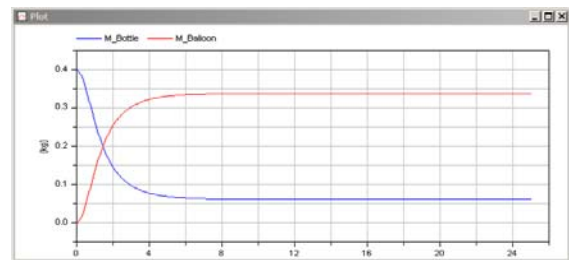


Figure 21: Air masses of bottle and balloon

The temperature in the balloon changes only temporarily, whereas the bottle cools down significantly. The volume of the balloon grows, whereas that of the bottle remains constant. Finally, mass is being transported from the bottle to the balloon. The sum of masses in the two containers remains constant.

7 The Water Loop

As a third example, we shall consider a closed water loop. Such water loops are found frequently in heating systems. They may represent either water that circulates between a water heater and a set of radiators; alternatively, such a loop may represent water (or glycol) circulating between a water heater and a set of (thermal) solar collectors; and finally, it may represent glycol circulating between a heat pump and the well of a geothermal system.

The water loop is represented in the model by four pipe segments and four water storages. The

water storages are modeled as isochoric capacitive fields. Three of the four pipe segments are specified as inductive fields, whereas the fourth one represents the pump.

The overall model is shown in Fig.22. The pipe segment containing the pump, depicted in the model as a forced volume flow, FVF, is built up internally in essentially the same way as the free volume flow, VF, used to represent regular pipe segments. The only difference is that the inductor is eliminated from the model and replaced by an external (regular) bond connector at which the mechanical pump can be connected. The pump itself is represented by a small DC motor.

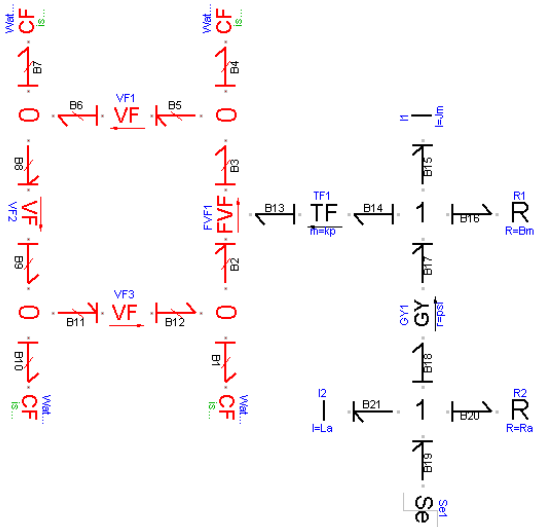


Figure 22: Water-loop model

The (black) effort source at the lower right corner of Fig.22 represents the armature voltage. The I and R elements next to it model the armature inductance and the armature resistance. The GY element higher up describes the transformation of electrical to mechanical (rotational) power. The I and R elements yet higher up represent the inertia of the rotor and the friction of the mechanical bearings. The TF element to their left transforms mechanical into hydraulic power.

The black (right) portion of Fig.22 is a regular bond graph, modeled using BondLib, whereas the red (left) portion of Fig.22 is a thermo-bond graph, modeled using ThermoBondLib.

Some simulation results are shown in Figs.23 and 24. Fig.23 shows the pressure values in the four water storages. The pump generates a high pressure value at its exit that is then successively reduced again by the subsequent regular pipe segments. The highest pressure is p_2 , the pressure of

the capacitive field immediately above (i.e., at the exit of) the pump, whereas the lowest pressure is p_1 , the pressure of the capacitive field below (i.e., at the entrance of) the pump. After the pump is turned on, the pressure values oscillate for a little while, before they stabilize at new values.

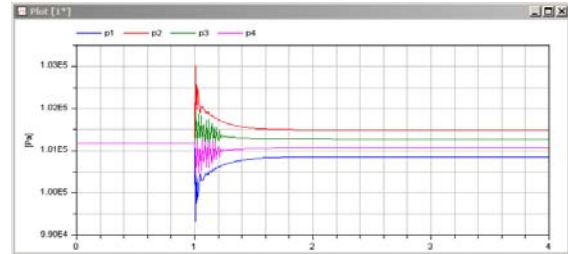


Figure 23: Pressure values in the four water storages

Fig.24 shows the mass flows in the three regular pipe segments. They are virtually indistinguishable one from another. At time 1 sec, the pump is switched on. It takes roughly 1 sec for the mass flow to fully build up.

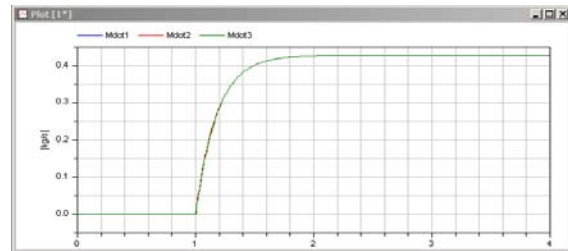


Figure 24: Mass flows in pipe segments

The three examples presented in this paper are included in the ThermoBondLib library, a free Modelica library downloadable from the Modelica website.

8 Conclusions

In this paper, a new free Modelica library for modeling and simulating convective flows in physical systems has been introduced.

ThermoBondLib has been designed as a graphical modeling library based on the thermo-bond graph methodology. Thermo-bond graphs were formally introduced by Greifeneder in his Diploma thesis [4].

Contrary to other types of systems, convective flows cannot be modeled in a completely abstract fashion. Different storages (capacitive fields) will look slightly different one from another. At the current time, storage (substance) models have only been made available for air, water, and water

vapor. Urgently needed are storage models for different types of glycol as well as for different types of industrial oils.

Although thermo-bond graphs are quite well readable, once a user has gotten familiarized with the methodology, bond graphs in general offer a fairly low-level interface to modeling. Bond graphs offer the simplest (lowest-level) interface to modeling physical systems that is still fully object-oriented. Hence wrapped thermo-bond graphs are more suitable for the description of complex systems involving convective flows. Wrapped thermo-bond graph sub-libraries shall be added to ThermoBondLib in due course.

Finally, thermo-bond graphs can also be used for the thermodynamic description of chemical reaction systems. First thermo-bond graph models of a few chemical reaction systems (hydrogen bromide and ammonia synthesis) were successfully coded in [4]. However, these models have not yet been added to the officially released version of ThermoBondLib.

References

- [1] Cellier, F.E.: [Continuous System Modeling](#). Springer-Verlag, New York, 1991
- [2] Cellier, F.E., Nebot, A.: [The Modelica Bond Graph Library](#). In: *Proceedings of the 4th International Modelica Conference*, Hamburg-Harburg, Germany (2005) Vol. 1, 57-65
- [3] Cellier, F.E., Clauß, C., Urquía, A.: [Electronic Circuit Modeling and Simulation in Modelica](#). In: *Proceedings of the Sixth Eurosim Congress on Modelling and Simulation*, Ljubljana, Slovenia (2007) Vol. 2, 1-10
- [4] Greifeneder, J.: [Modellierung thermodynamischer Phänomene mittels Bondgraphen](#). Diploma Thesis, University of Stuttgart, Germany, 2001
- [5] Greifeneder, J., Cellier, F.E.: [Modeling Convective Flows Using Bond Graphs](#). In: *Proceedings of ICBGM'01, 5th SCS Intl. Conf. on Bond Graph Modeling and Simulation*, Phoenix, Arizona (2001) 276-284
- [6] Greifeneder, J., Cellier, F.E.: [Modeling Multiphase Systems Using Bond Graphs](#). In: *Proceedings of ICBGM'01, 5th SCS Intl. Conf. on Bond Graph Modeling and Simulation*, Phoenix, Arizona (2001) 285-291
- [7] Greifeneder, J., Cellier, F.E.: [Modeling Multi-element Systems Using Bond Graphs](#). In: *Proceedings of ESS'01, 13th European Simulation Symposium*, Marseille, France (2001) 758-766
- [8] Zimmer, D., Cellier, F.E.: [The Modelica Multi-bond Graph Library](#). In: *Proceedings of the 5th International Modelica Conference*, Vienna, Austria (2006) Vol. 2, 559-568



François E. Cellier received his BS degree in electrical engineering in 1972, his MS degree in automatic control in 1973, and his PhD degree in technical sciences in 1979, all from the Swiss Federal Institute of Technology (ETH) Zurich. Dr. Cellier worked at the University of Arizona as professor of Electrical and Computer Engineering from 1984 until 2005. He recently returned to his home country of Switzerland. Dr. Cellier's main scientific interests concern modeling and simulation methodologies, and the design of advanced software systems for simulation, computer aided modeling, and computer-aided design. Dr. Cellier has authored or co-authored more than 200 technical publications, and he has edited several books. He published a textbook on Continuous System Modeling in 1991 and a second textbook on Continuous System Simulation in 2006, both with Springer-Verlag, New York.



Jürgen Greifeneder received a diploma degree in Engineering Cybernetics from the University Stuttgart, Germany in 2002. Dr. Greifeneder then switched to the University of Kaiserslautern, where he received a Ph.D. in Electrical and Computer Engineering in 2007 with a dissertation on the formal analysis of temporal behavior of Networked Automation Systems (NAS) by use of probabilistic model checking. Scientific stays at the University of Arizona (USA), at the Ecole Normale Supérieure de Cachan (F), and at the Universidade de Brasília (BR) completed his education. Dr. Greifeneder's primary research interests concern modeling, simulation, and control methodologies.