Francesco Casella and Francesco Schiavo
Dipartimento di Elettronica e Informazione, Politecnico di Milano:
Modelling and Simulation of Heat Exchangers in Modelica with
Finite Element Methods
pp. 343-352

Modelling and Simulation of Heat Exchangers in Modelica with Finite Element Methods

Francesco Casella and Francesco Schiavo*
Politecnico Di Milano
Dipartimento Di Elettronica Ed Informazione (DEI)
Via Ponzio 34/5, 20133 Milano, Italy
e-mail: {casella,schiavo}@elet.polimi.it

Abstract

The complete development of a Modelica model for 1-D heat exchangers is presented. The numerical method, termed *Finite Element Method*, is briefly reviewed and its application to heat exchangers partial differential equations is presented. Implementation issues are tackled as well, and the component developed is validated through simulation within the framework of *ThermoPower*, a recently released Modelica library for thermal power plants modelling, simulation and control. The component is included into such library which is publicly available through the Web [1].

1   Introduction

The process of heat exchange between two fluids that are at different temperatures and separated by a solid wall occurs in many engineering applications. The device used to implement this exchange is termed a *heat exchanger* (HE), and specific applications may be found in space heating and air-conditioning, power generation, waste heat recovery, and chemical processing [2].

In this paper it is presented a Modelica model of the fluid side of heat exchangers, developed using a numerical method known as *finite element method*; complete models of HEs are then obtained by suitably assembling such models with metal wall and heat exchange models.

Such model is included in the power generation Modelica library *ThermoPower* [1].

The goal of this research is twofold. First, to show how Modelica can be used effectively in the modelling of physical phenomena described directly by *Partial Differential Equations* (PDEs); this aim is

achieved through the application of a specific numerical method, namely the *Finite Element Method*, which can approximate a PDE with a set of *Ordinary Differential Equations* (ODEs). Second, to amply the library of models for thermal power generation plants which has been developed here at Politecnico di Milano, offering to potential users a broader choice for the complexity and accuracy with which they would like to model some specific physical phenomena; this second aim is achieved exploiting the Modelica features for object-oriented modelling and the standardized model interfaces which have been defined within the library *ThermoPower* [1].

The paper is organized as follows: Section 2 recalls the basic physical laws for HEs; Section 3 is a brief introduction to the numerical methods used, while Section 4 shows how such methods can be used to develop models for HEs; Section 5 deals with the Modelica implementation of the model and Section 6 shows some simulation results; Finally, conclusions and perspectives on future work are given in Section 7.

2   First Principle Model

Consider a compressible fluid flowing through a pipe-shaped volume (*V*) with rigid boundary area and exchanging thermal energy through such boundary (figure 1).
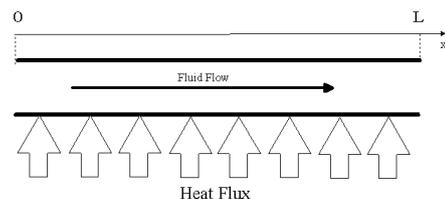

Figure 1: Heat Exchanger Scheme

---

*Corresponding author

Assume that

- the longitudinal dimension (x) is far more relevant than the other two;

- the volume V is "sufficiently" regular (that is V is such that the fluid motion along x is not interrupted);

- there are no phase-changes along the pipe (that is the fluid is always either single-phase or two-phase).

- the Reynolds number (Re) is such that turbulent fluid flow is assured along all the pipe, which in turn guarantees almost uniform velocity and thermodynamic state of the fluid across the radial direction;

When water or steam is assumed as the working fluid, the last hypothesis does not hold at very low flowrates; however, in practical system simulations, the plant never operates in such conditions for a long time.

It is then possible to define all the thermodynamic intensive variables with respect only to longitudinal abscissa (x) and time (t). Within this framework, the conservation equations for mass, dynamic momentum (neglecting the kinetic term) and energy (neglecting the diffusion term) can be formulated as follows:

$$A\frac{\partial \rho}{\partial t} + \frac{\partial w}{\partial x} = 0 \tag{1}$$

$$\frac{\partial w}{\partial t} + A\frac{\partial p}{\partial x} + \rho g A \frac{\partial z}{\partial x} + \frac{C_f}{2\rho A^2}\omega w|w| = 0 \tag{2}$$

$$\frac{\partial h}{\partial t} + w\frac{v}{A}\frac{\partial h}{\partial x} = v\frac{dp}{dt} + v\frac{\omega}{A}\phi_{ext} \ , \tag{3}$$

where A is the pipe cross-section, ρ the fluid density, w the mass flow-rate, p the fluid pressure, g the gravity acceleration, z the pipe height, C_f the Fanning friction coefficient, ω the wet perimeter, h the fluid specific enthalpy, v the fluid specific volume, ϕ_{ext} the heat flux entering the pipe across the lateral surface.

3 Finite Element Methods For Time-Dependant Advection Equation

Consider the following first-order linear *partial differential equation* (PDE):

$$\begin{cases} \dfrac{\partial u}{\partial t} + \beta \cdot \nabla u + \sigma u = f & \text{in } \Omega \times (0,T) \\ u = g & \text{on } \partial \Omega^{in} \times (0,T) \\ u = u_0 & \text{on } \Omega \text{ for } t = 0 \ , \end{cases} \tag{4}$$

where Ω denotes a bounded domain ($x \in \Omega$) in $\Re^m$ with boundary $\partial\Omega$, $T > 0$ is a prescribed time value ($t \in (0,T)$), $u(x,t)$ is the unknown (for example a temperature field), $f(x,t)$ is given function, $\beta(x,t)$ is a given velocity field , $\sigma(x,t)$ an adsorption coefficient, ∇ is the gradient operator; $u_0 = u_0(x)$ is the assigned initial datum and $g(x,t)$ is the assigned *Dirichlet* boundary condition defined on the inflow boundary $\partial\Omega^{in} = \{x \in \partial\Omega | \beta(x,t) \cdot \overrightarrow{n}(x) < 0\}$ ($\overrightarrow{n}$ is the unit outward normal vector on $\partial\Omega$).

The equation (4) is called *time-dependant advection equation* [3] and it can represent the energy equation (3) for heat exchangers.

In the following, for the sake of simplicity, the equation (4) will have the form $\frac{\partial u}{\partial t} + Lu = f$, where L is the proper differential operator.

The approximated solution of the PDE (4) can be obtained through several numerical methods; on the other side, only methods that allow to transform a PDE into a set of ordinary differential equations (ODEs) or differential-algebraic equations (DAEs) with respect to time are suitable to use within the Modelica framework. Within this paper the focus is on a numerical method termed *Finite Element Method* (FEM) [3],[4]. Other interesting methods for the approximation of PDE (4) are the *Finite Difference Method* (FDM) and *Finite Volume Method* (FVM) [3], [5]. The advantage of using FEM instead of FVM or FDM is that it can provide more accurate solution or, in specific cases, the exact nodal values for the unknown [3].

The FEM is based on the discretization of the solution region into elementary elements. The unknown variable u is expressed in terms of assumed *approximating* or *interpolation* functions within each element. The interpolation functions are local, i.e. functions defined over smaller sub-domains, where these sub-domains extend over a few elements, and are zero everywhere else. The local interpolation functions are ordinarily very simple functions, such as low-degree polynomials. The interpolation functions are defined in terms of the values of the variable at specified points called *nodes*. Nodes usually lie on the element boundaries where adjacent elements are considered to be connected. In addition to boundary nodes, an element may also have a few interior nodes. The nodal values u_i of the variable and the interpolation functions for the elements completely define the behavior of the variable within the elements. For the finite element representation of a problem, the nodal values of the variable become the new unknowns. Once these unknowns are found, the interpolation functions define the variable

throughout the assemblage of elements. Clearly, the nature of the solution and the degree of approximation depend not only on the size and number of the elements used, but also on the interpolation functions selected [3].

3.1   The Method of Weighted Residual

The *Method of Weighted Residual* (MWR) is a mathematical technique for obtaining finite element equations from linear and non-linear PDEs. Referring to (4), the problem solved by the MWR is to find the nodal values of an approximated solution ($u_h(x,t)$) so as to make an error (called residual)

$$R_h(x,t) = \frac{\partial u_h(x,t)}{\partial t} + Lu_h(x,t) - f(x,t) \qquad (5)$$

small over the entire solution domain Ω, i.e.

$$\int_\Omega R_h v_h d\Omega \approx 0, \quad \forall v_h \in V_h \ , \qquad (6)$$

where $v_h(x)$ are linearly independent *weighting functions* (as many as the nodal points) belonging to an appropriate finite dimensional space V_h. The *Petrov-Galerkin* methods used in the HE model development belong to this family.

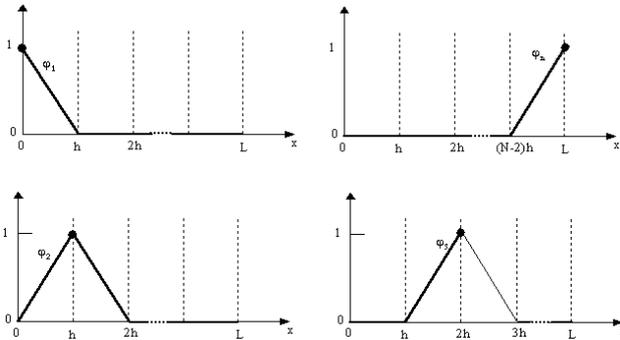## 3.2   Finite Element Basis Function and Space

Figure 2: The "triangular" basis functions $\varphi_j(x)$

The solution domain Ω is decomposed into elements K of mesh size h_K. The *finite element space* X_h^k is the finite dimension space of continuous piecewise-polynomial functions of degree k defined within each element K. The basic idea of the FEM is therefore to approximate the infinite dimensional solution, belonging to a infinite dimension space X, with a finite dimensional one, belonging to X_h^k (whose size will be

called N). In figure 2 the piece-wise linear ($k = 1$) interpolating functions are depicted. The space of *interpolating* functions will be called hereafter W_h and its interpolating functions $\varphi_i(x)$. Then the *approximated* solution $u_h(x,t)$ of u is expressed as

$$u_h(x,t) = \sum_{i=1}^{N} u_i(t)\varphi_i(x) \quad \text{for } t > 0$$

$$u_{0,h}(x) = \sum_{i=1}^{N} u_{0,i}\,\varphi_i(x) \quad \text{for } t = 0 \qquad (7)$$

3.3   Petrov-Galerkin Methods

In the following, for the sake of simplicity, the inner functional product notation $(u,v) = \int_\Omega uvdx$ is used. In addition the way boundary conditions are enforced into the approximated equation is not included, since it is presented separately later.

By expanding (6) and properly choosing the weighting function space, the *Petrov-Galerkin* (PG) approximation of the PDE problem (4) consists in finding $u_h \in W_h$ such that

$$\frac{d}{dt}(u_h, v_h) + (Lu_h, v_h) = (f, v_h) \ \forall v_h \in V_h \qquad (8)$$

with $W_h \neq V_h$ but $\dim(W_h) = \dim(V_h) = N, \ \forall h > 0$.

Equation (8) has to be satisfied for any $v_h \in V_h$, that is it has to be satisfied for all the functions of any basis of the space V_h itself; the basis functions of V_h will be denoted as $\{\psi_i | i = 1\dots N\}$. The functional space V_h is termed the space of *test* or *weighting* functions. Then being $\{\varphi_j | j = 1\dots N\}$ a basis for the space W_h, and substituting (7) into (8), it can be obtained a set of N ODEs for the unknown vector $U(t)$:

$$\mathbf{M}\frac{dU(t)}{dt} + \mathbf{A}U(t) = \mathbf{F}(t), \ U(0) = U_0, \qquad (9)$$

Where $U(t) = [u_i(t)]$, $F(t) = [(f, \psi_i)]$, $U_0 = [u_{0,i}]$, $A_{ij} = (L\varphi_j, \psi_i)$, $M_{ij} = (\varphi_j, \psi_i)$, for $i, j = 1\dots N$. The matrix $\mathbf{M}$ and $\mathbf{A}$ are called the *mass* and *stiffness* matrix respectively.

General assumptions guarantee the existence and uniqueness of a solution [3].

The (standard) *Galerkin* method is a particular case of the PG one, where the test functions space (V_h) is chosen to be the same as the approximating functions space (W_h), therefore $M_{ij} = (\varphi_j, \varphi_i)$, $A_{ij} = (L\varphi_j, \varphi_i)$, $F_i(t) = (f, \varphi_i)$.

The application of the standard Galerkin method to advection dominated problems (as the one considered) could lead to solutions with oscillatory behavior due

to numerical instabilities. To overcome such a problem it is possible to use a *stabilized Petrov-Galerkin method.*

The basic idea of stabilization methods is to relate the functional space V_h to W_h through a differential operator L_h somewhat related to the differential operator L, that is $V_h = \{w_h + L_h w_h | w_h \in W_h\}$. The equation (8) thus becomes

$$\frac{d}{dt}(u_h, w_h + L_h w_h) + (L u_h, w_h + L_h w_h) = \qquad (10)$$
$$= (f, w_h + L_h w_h) \quad \forall w_h \in W_h$$

3.4 Treatment of Boundary Conditions

The boundary conditions (BCs) can be imposed in two different ways:

1. *Strong formulation (sf)*: the the boundary conditions are enforced directly in the definition of the space W_h of the admissible solutions, while the *test functions* vanish on the boundary. The boundary conditions are satisfied at all nodes lying on $\partial\Omega^{in}$.

2. *Weak formulation (wf)*: the boundary conditions are enforced indirectly in the unknown nodal values of the approximated equation. The boundary conditions is not imposed exactly at all nodes of $\partial\Omega^{in}$, but a suitable linear combination between them and the residual of the PDE is enforced. Therefore the problem formulation becomes: for any $t \in [0, T]$ find $u_h \in W_h$ such that

$$\begin{cases} \dfrac{d}{dt}(u_h, v_h) + (L u_h, v_h) - \displaystyle\int_{\partial\Omega^{in}} \beta \cdot \overrightarrow{n} u_h v_h d\gamma \\ = (f, v_h) - \displaystyle\int_{\partial\Omega^{in}} \beta \cdot \overrightarrow{n} g_h v_h d\gamma \quad \forall v_h \in V_h \\ u_h(0) = u_{0,h} \end{cases}$$
$$(11)$$

It is important to note that the additional integral terms can be easily computed for the one-dimensional case since $\partial\Omega^{in}$ is a finite set of points (at most two: $x = 0$ and $x = L$).

The main differences of the two boundary condition formulations are:

- In the *wf* the nodal values on the boundary are unknown and therefore the number of finite element equations to be solved is higher than that obtained from the strong formulation.

- In the case of flow reversal (change of β sign in equation 4) the inflow boundary changes. In the *wf* the state variables (i.e. the nodal values) are always the same since the nodal values on the boundary are also problem unknowns. Instead, in the *sf*, the nodal values on the boundary are known, so that the state variables depend on the flow direction.

In the model developed the choice has been to adopt the *wf* since it can be accurate as *sf* while providing easier implementation in the case of flow reversal [5].

4 FEM Model for Heat Exchangers

In this section it will be shown how the numerical methods introduced can be applied to the balance equations so to transform them into a set of ODEs that can be used directly in Modelica models.

The spatial domain ($[0, L]$) has been divided into a grid of uniformly spaced elements with size $l = L/(N-1)$, where N (≥ 2)is the number of finite elements that are going to be used.

The interpolating functions have been chosen to be linear (figure 2); their analytical expression is

$$\varphi_1(x) = \begin{cases} \dfrac{l-x}{l} & 0 < x \leq l \\ 0 & otherwise \end{cases}$$

$$\varphi_N(x) = \begin{cases} \dfrac{x-(N-2)l}{l} & (N-2)l < x \leq L \\ 0 & otherwise \end{cases}$$

$$\varphi_i(x) = \begin{cases} \dfrac{x-(i-2)l}{l} & (i-2)l < x \leq (i-1)l \\ \dfrac{il-x}{l} & (i-1)l < x \leq il \\ 0 & otherwise \end{cases}$$
$$(12)$$

In the following the notation $\overline{\varphi} = [\varphi_1 \cdots \varphi_N]^T$ will be used.

The *stabilized Petrov-Galerkin Method* termed *GALS (Galerkin/Least-Squares)*, which has been proven to be the most suitable one for the advection dominated case [6], has been used to obtain the test functions:

$$\psi_j(x) = \varphi_j(x) + \alpha \frac{l}{2} \frac{d\varphi_j(x)}{dx}, \quad j = 1 \ldots N \qquad (13)$$

where α is a stabilization coefficient ($0 \leq \alpha \leq 1$); for $\alpha = 0$ the standard (i.e. non stabilized) method can be obtained.

The following hypothesis have been taken into account in the finite element formulation:

- h linear on each element

- T linear with h

- v linear with h

- ϕ_{ext} linear on each element

- w uniform along the HE

- p uniform along the HE

- p, h, w are the state variables

That means h, T, v, ϕ_{ext} can be expressed as

$$h(x,t) = \sum_{i=1}^{N} h_i(t)\varphi_i(x) = \overline{h}(t)^T \, \overline{\varphi}(x), \ \ \overline{h} = [h_1 \cdots h_N]^T$$

$$T(x,t) = \sum_{i=1}^{N} T_i(t)\varphi_i(x) = \overline{T}(t)^T \, \overline{\varphi}(x), \ \ \overline{T} = [T_1 \cdots T_N]^T$$

$$v(x,t) = \sum_{i=1}^{N} v_i(t)\varphi_i(x) = \overline{v}(t)^T \, \overline{\varphi}(x), \ \ \overline{v} = [v_1 \cdots v_N]^T$$

$$\phi_{ext}(x,t) = \sum_{i=1}^{N} \phi_i(t)\varphi_i(x) = \overline{\phi}(t)^T \, \overline{\varphi}(x), \ \ \overline{\phi} = [\phi_1 \cdots \phi_N]^T \tag{14}$$

The considered hypotheses do not affect the generality of the model, at least if there aren't any phase changes along the HE.

In the balance equations both the fluid density specific volume are involved, and their relation is well known to be $\rho = 1/v$; since v has been assumed to be linear with h (which is linear on each element), it should result $\rho = \left(\sum_{i=1}^{N} v_i\varphi_i\right)^{-1}$, that is ρ is not linear with h. As a matter of fact, for the sake of simplicity, it has been assumed that also ρ can be expressed as

$$\rho(x,t) = \sum_{i=1}^{N} \rho_i(t)\varphi_i(x) = \overline{\rho}(t)^T \, \overline{\varphi}(x), \ \ \overline{\rho} = [\rho_1 \cdots \rho_N]^T$$

$$\text{with } \rho_i = (v_i)^{-1} \quad \forall i = 1 \cdots N \tag{15}$$

It can be shown that the error introduced by this approximation (computed as $\int_0^h (v^{-1} - \rho)dx$) is $O(h)$.

Among the balance equations, the mass and dynamic momentum ones describe the fast pressure and flow rate dynamics, while the energy one describes the slower dynamics of heat transport with the fluid velocity; the most relevant phenomenon, for power generation plant modelling, is the latter one, so that the equation (3) has been discretized with a fine approximation through FEMs, while equations (1)-(2) have been treated with a coarser approximation.

4.1   Energy Balance Equation

Consider the energy balance equation for the HE:

$$\frac{\partial h}{\partial t} + w\frac{v}{A}\frac{\partial h}{\partial x} = v\frac{dp}{dt} + v\frac{\omega}{A}\phi_{ext} \tag{16}$$

with reference to the advection equation (4) used in the finite element formulation, it results $\beta = w\frac{v}{A}$ and $\sigma = 0$, while the term f is simply the right hand side of the equation.

The application of a PG method, with weakly imposed boundary conditions, leads to a set of N ODEs:

$$\int_0^L \left(\sum_{i=1}^{N} \dot{h}_i\varphi_i\right)\psi_j dx + \int_0^L \left(\frac{w}{A}\sum_{i=1}^{N} v_i\varphi_i \sum_{i=1}^{N} h_i\frac{d\varphi_i}{dx}\right)$$

$$\psi_j dx + \int_{\partial\Omega^{in}} \left(\frac{w}{A}\sum_{i=1}^{N} v_i\varphi_i \sum_{i=1}^{N} h_i\varphi_i\right)\psi_j dx =$$

$$\int_0^L \sum_{i=1}^{N} v_i\varphi_i \left(\dot{p} + \frac{\omega}{A}\sum_{i=1}^{N} \phi_i\varphi_i\right)\psi_j dx +$$

$$+ \int_{\partial\Omega^{in}} \left(\frac{w}{A}\sum_{i=1}^{N} v_i\varphi_i \, h_{in}\right)\psi_j dx, \quad \forall\psi_j \in V_h \tag{17}$$

where h_{in} is the fluid specific enthalpy at the inflow boundary. Such set of ODEs can be easier represented with the following differential matrix equation:

$$M\dot{\overline{h}} + \frac{w}{A}B\,\overline{h} + \frac{w}{A}C\,\overline{h} = \dot{p}M\,\overline{v} + \frac{\omega}{A}Y\,\overline{\phi} + \frac{w}{A}K\overline{v}, \tag{18}$$

where M, B, C, Y, K are defined as

$$M_{ji} = \int_0^L \varphi_i\psi_j dx$$

$$B_{ji} = \int_0^L \left(\sum_{k=1}^{N} v_k\,\varphi_k\right)\frac{d\varphi_i}{dx}\psi_j dx$$

$$C_{ji} = \int_{\partial\Omega^{in}} \left(\sum_{k=1}^{N} v_k\,\varphi_k\right)\varphi_i\psi_j dx \tag{19}$$

$$Y_{ji} = \int_0^L \left(\sum_{k=1}^{N} v_k\,\varphi_k\right)\varphi_i\psi_j dx$$

$$K_{ji} = \int_{\partial\Omega^{in}} h_{in}\,\varphi_i\psi_j dx$$

The detailed expressions for the matrices M, B and Y are reported in appendix A, while the matrices C and K (which express the BCs) will be analyzed thoroughly in the next section.

4.2 Mass Balance Equation

Consider the mass balance equation for the HE:

$$A\frac{\partial \rho}{\partial t} + \frac{\partial w}{\partial x} = 0 \qquad (20)$$

Since pressure (p) and specific enthalpy (h) have been chosen as the thermodynamic state variables, it results

$$\frac{\partial \rho}{\partial t} = \frac{\partial \rho}{\partial h}\frac{\partial h}{\partial t} + \frac{\partial \rho}{\partial p}\frac{\partial p}{\partial t} \qquad (21)$$

Substituting in such equation the expression reported in (14) for h and ρ, it follows

$$\frac{\partial \rho}{\partial t} = \overline{\rho_h}\left(\overline{\phi}\overline{\phi}^T\right)\dot{\overline{h}} + \dot{p}\,\overline{\rho_p}\,\overline{\phi} \qquad (22)$$

where $\overline{\rho_h} = [\frac{\partial \rho_1}{\partial h}|_{h_1,p} \cdots \frac{\partial \rho_N}{\partial h}|_{h_N,p}]$ and $\overline{\rho_p} = [\frac{\partial \rho_1}{\partial p}|_{h_1,p} \cdots \frac{\partial \rho_N}{\partial p}|_{h_N,p}]$
Then, integrating the mass balance equation along the spatial domain, it results

$$\int_0^L \frac{\partial \rho}{\partial t}dx = -\frac{1}{A}\int_0^L \frac{\partial w}{\partial x}dx, \qquad (23)$$

leading to the ODE

$$\overline{\rho_h}^T E\,\dot{\overline{h}} + \dot{p}\,\overline{\rho_p}^T D = \frac{1}{A}(w_0 - w_L), \qquad (24)$$

where w_0 and w_L are the fluid mass flow-rate at abscissa 0 and L respectively; E and D are a matrix and a vector (details can be found in appendix A):

$$E_{ji} = \int_0^L \phi_i\phi_j dx, \quad D_i = \int_0^L \phi_i dx \qquad (25)$$

4.3 Dynamic Momentum Equation

Consider the dynamic momentum balance equation for the HE:

$$\frac{\partial w}{\partial t} + A\frac{\partial p}{\partial x} + \rho gA\frac{dz}{dx} + v\frac{C_f\omega}{2A^2}w|w| = 0 \qquad (26)$$

Substituting the expression reported in (14) for ρ and v and integrating along the spatial domain, the following expressions result (dz/dx is assumed as a constant parameter):

$$\int_0^L \frac{\partial w}{\partial t}dx + \int_0^L A\frac{\partial p}{\partial x}dx + \int_0^L gA\frac{dz}{dx}\sum_{i=1}^N \rho_i\phi_i dx +$$
$$+ \int_0^L \frac{C_f\omega}{2A^2}w|w|\sum_{i=1}^N v_i\phi_i dx = 0, \qquad (27)$$

leading to the ODE

$$L\dot{w} + A(p_L - p_0) + gA\frac{dz}{dx}\overline{\rho}^T D +$$
$$+ \frac{C_f\omega}{2A^2}w|w|\overline{v}^T D = 0, \qquad (28)$$

Assuming the Reynolds number is sufficiently high, C_f is approximately constant; for medium-range values of Re, it can be computed with Colebrook's equation. When dealing with water/steam flow in industrial plants, the transition and laminar regimes correspond to very low pressure drops, which need not be computed with high accuracy; therefore, a minimum value of $Re = 2100$ is assumed. Last, but not least, a small linear friction term is added to enhance numerical stability at low or zero flowrate; the parameter w_0 should be much smaller than the nominal flowrate, so that the added term is negligible during normal operation. Thus equation 28 becomes

$$L\dot{w} + A(p_L - p_0) + gA\frac{dz}{dx}\overline{\rho}^T D +$$
$$+ \frac{C_f\omega}{2A^2}w(|w| + w_0)\overline{v}^T D = 0. \qquad (29)$$

5 Modelica Implementation

The developed model has been implemented in a component called `Flow1Dfem` (figure 3) which is part of the library *ThermoPower* [1].
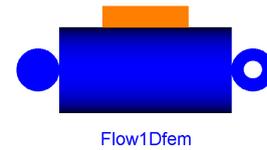

Figure 3: The Modelica Model

For the present model, it has been assumed that the fluid inside the HE is a water/steam mixture. The medium models used for water and steam are provided by the free "ThermoFluid" library [7].
The component is perfectly interchangeable with the actual *ThermoPower* component for 1-D HEs, since it has the same connectors: two flanges for water/steam flow and a terminal for heat flux. Here the definition of such interfaces is reported, for further details see [1]:

```
connector WaterFlangeA
        Pressure          p;
    flow MassFlowRate     w;
    input SpecificEnthalpy hBA;
```

```
    output SpecificEnthalpy hAB;
       end WaterFlangeA;

connector WaterFlangeB
        Pressure         p;
    flow MassFlowRate    w;
   input SpecificEnthalpy hAB;
  output SpecificEnthalpy hBA;
       end WaterFlangeA;

connector DHT
  parameter Integer N;
  Temperature T[N];
  flow HeatFlux phi[N];
end DHT;
```

In the code `hAB` and `hBA` are the fluid specific enthalpies in case its direction is from an A-type flange to a B-type one and viceversa. Such connectors support flow reversal. In the model there is a connector `infl` of type `WaterFlangeA` (corresponding to $x = 0$) and a connector `outfl` of type `WaterFlangeB` corresponding to $x = L$.

The model offers many customization possibilities through parameters: the HE geometry can be fully specified (length, diameter, height); the dynamic momentum term $\partial w/\partial t$ can be switched off to avoid fast pressure oscillations; the C_f coefficient can be either constant or computed by the Colebrook equation; the compressibility effect deriving from the discretization of equation (1) can be associated to either the upstream or downstream pressure; the numerical stabilization coefficient α can be chosen in the interval $[0, 1]$.

It should be noted that the matrices M, B, Y, E and the vector D are completely defined once the parameter α has been chosen; thus they can be computed once for all before the simulation starts by efficient Modelica compilers. The definition of such matrices is made thought some loops, as showed below:

```
M[1, 1] = l/3 - l*alfa/4;
  M[N, N] = l/3 + l*alfa/4;
  M[1, 2] = l/6 - l*alfa/4;
  M[N, (N - 1)] = l/6 + l*alfa/4;
  if N > 2 then
    for i in 2:N - 1 loop
      M[i, i - 1] = l/6 + l*alfa/4;
      M[i, i] = 2*l/3;
      M[i, i + 1] = l/6 - l*alfa/4;
      M[1, i + 1] = 0;
      M[N, i - 1] = 0;
      for j in 1:(i - 2) loop
        M[i, j] = 0;
      end for;
      for j in (i + 2):N loop
        M[i, j] = 0;
      end for;
    end for;
  end if;
```

It can be noticed that many of the matrices entries are zeros, so it could appear that the use of a matrix notation for the balance equations could increase the computational burden; nevertheless, it has been discovered (by direct inspection of the generated C code) that efficient compilers can simplify the set of ODEs obtained expanding the differential matrix equations in the Modelica code, removing the terms corresponding to the zero entries in the matrices.

5.1 Boundary Conditions and Flow Reversal

One of the most relevant features of the model is the capability to handle not only flow reversal in the HE, but also the most "unusual" transients for what concerns flow, that is the model is able to handle also transient where the fluid is entering or exiting from both the extremities (which are operating conditions which can be experienced when suddenly decreasing or increasing the heat-flux).

The matrices C and K, enforcing the boundary conditions into equation (18), depend on the inflow boundary $\partial\Omega^{in}$. It can be noted that, in the 1-D case, the inflow boundary can be constituted at most by the points $x = 0$ and $x = L$, depending on the fluid mass-flow rate direction in that specific direction.

Suppose, for example, that the inflow boundary is just $x = 0$ (that means `infl.w` > 0 and `outfl.w` < 0). Considering the analytical expression for C and K and for the interpolating and weighting function, it results

$$C_{ij} = \int_{x=0} \left(\sum_{k=1}^{N} v_k \, \varphi_k \right) \varphi_i \psi_j dx =$$
$$= \begin{cases} (1 - \frac{\alpha}{2})v_1 & \text{if } i = j = 1 \\ 0 & \text{otherwise} \end{cases}$$

The same happens if the inflow boundary is $x = L$: the only non-zero entries for the matrices C and K can be $(1, 1)$ and (N, N). The code for such entries is obtained through simple conditional equations:

```
C[1, 1] = if (infl.w >= 0) then
          (1 - alfa/2)*v[1, 1] else 0;
C[N, N] = if (outfl.w >= 0) then
          (1 + alfa/2)*v[N, 1] else 0;

K[1, 1] = if (infl.w >= 0) then
          (1 - alfa/2)*infl.hBA else 0;
K[N, N] = if (outfl.w >= 0) then
          (1 + alfa/2)*outfl.hAB else 0;
```

6 Simulations

The component has been tested with other models from the library *ThermoPower* using *Dymola* simulation environment [8]; specific configurations have

been set up in order to investigate the model behaviour with respect to the single balance equations and to their interactions in the most common layouts found in power plants. Many simulations have been carried out but, for the sake of brevity, only the most significative ones are reported here; all the test set ups are included in the library and are available on-line [1].

In all the reported simulations, the HE has a length of $10\,m$ and radius $1\,cm$. All the simulations use $N = 20$ nodes.

The first simulation reported is aimed at testing the energy balance equation; the experimental layout is depicted in figure 4: the HE (*hex*) is connected with a mass flow rate source, an external source of heat flow, a valve (which accounts for head losses) and a sink with fixed pressure.
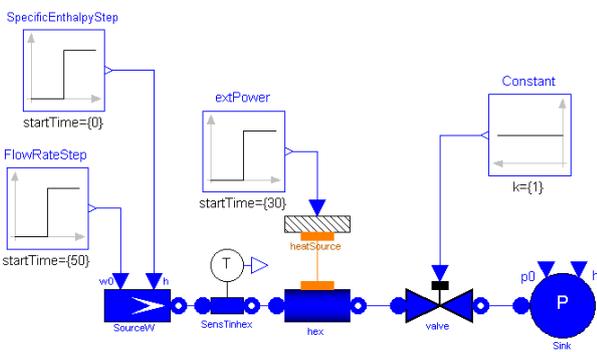

Figure 4: First Experiment Layout

The fluid involved in the experiment is liquid water at temperatures between $297\,K$ and $322\,K$, the pressure inside the HE during the experiment varies from $1.6^5\,Pa$ to $2^5\,Pa$ and the mass flow rate is comprised in the interval $0.2 - 0.3\,Kg/s$.

At the initial time instant there is a step variation from $10^5\,J/m^3$ to $1.42 \cdot 10^5\,J/m^3$ of the specific enthalpy for the fluid of the flow rate source; at time $30\,s$ there is a step variation of the energy flux entering the HE from 0 to $1.25 \cdot 10^4\,W/m^2$; at time $50\,s$ there is a step variation in the source mass flow rate from $0.3\,Kg/s$ to $0.2\,Kg/s$.

The temperature of the fluid at the end of the HE is reported in figure 5. The exact solution (assuming ρ constant) for the underlying PDE would lead to a temperature step variation at time $t = 10\,s$ and ramp variations at time $t = 30\,s$ and $t = 50\,s$; the simulation results show good accordance with such behavior.

The second experiment is aimed at testing the mass balance equation; the experimental layout, similar to the first one, is depicted in figure 6.
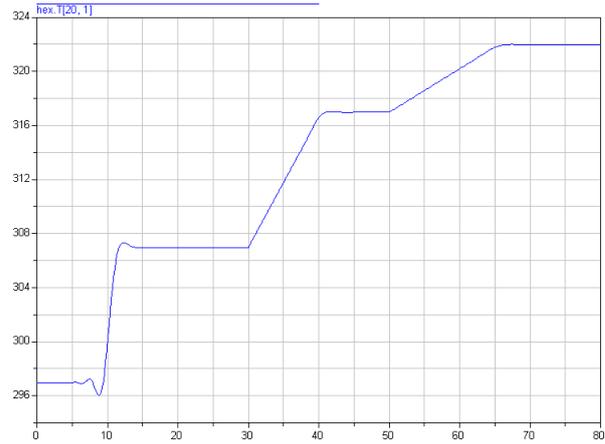

Figure 5: HE Outlet Temperature

The fluid involved in this experiment is superheated vapor with temperature and pressure at about $536\,K$ and $10^5\,Pa$ respectively; the mass flow rate flowing through the HE is about $10^{-2}\,Kg/s$.
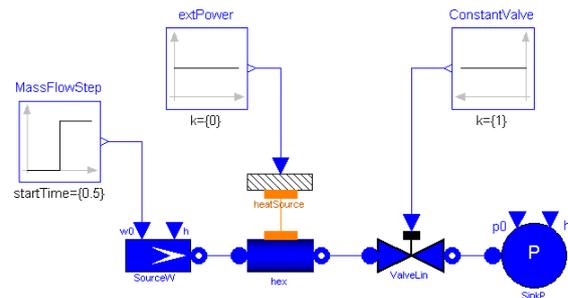

Figure 6: Second Experiment Layout

At time $0.5\,s$ there is a 10% step increment of the mass flow rate; the consequent HE pressure transient is depicted in figure 7.

The solution of the equations for such experimental setup, assuming uniform gas properties and ideal gas content, would lead to a first order transient whose time constant is in good accordance with the simulation results.

The last test reported here involves a two side HE (*hexA* and *hexB*) in counterflow configuration (figure 8). The two fluid sides are separated by a metal wall $1\,mm$ thick.

The operating fluid is liquid water with temperature in the range $296\,K - 321\,K$ and pressure about $3 \cdot 10^5\,Pa$. The experiment setup is such that the mass flow rates for the two HE sides have the same value ($0.31\,Kg/s$) with residence time $9.9\,s$.

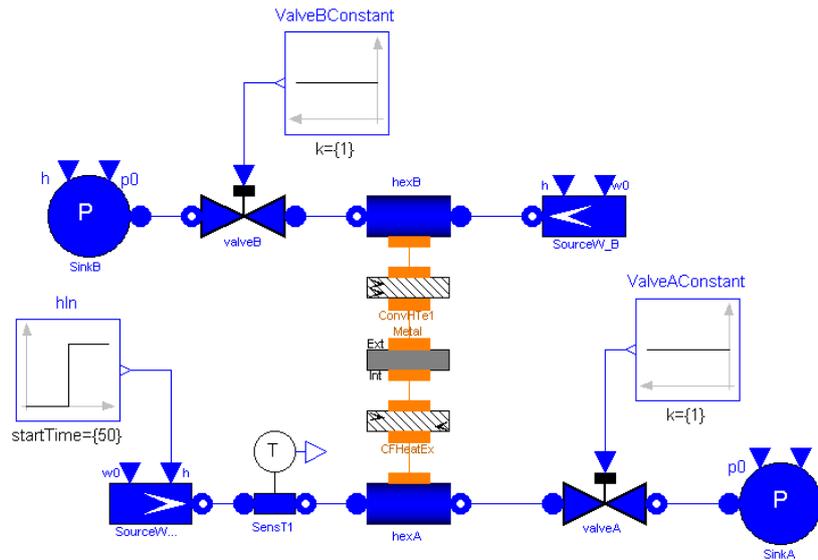At time $50\,s$ there is a step variation from $10^5\,J/m^3$ to
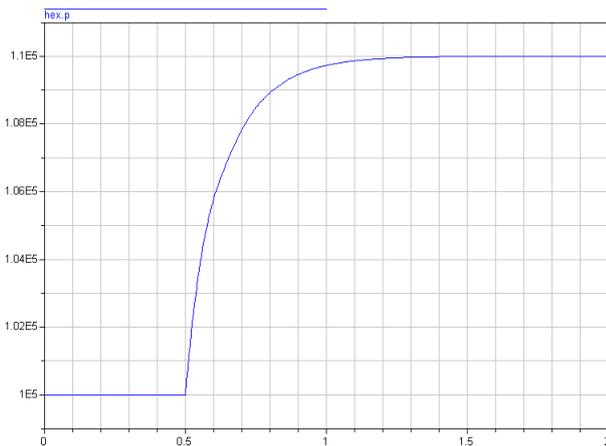
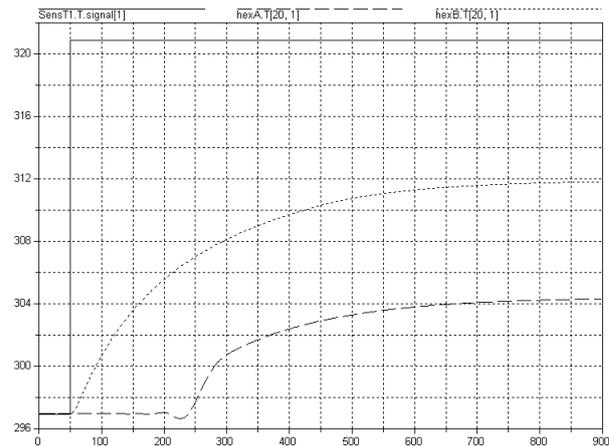Figure 8: HE Counterflow Configuration

Figure 7: HE Pressure

Figure 9: HE Temperatures: hexA inlet (continuous), hexB outlet (dotted) and hexA outlet (dashed)

$2 \cdot 10^5 \ J/m^3$ of the specific enthalpy for the fluid of the flow rate source for *hexA*.

In figure 9 are reported the inlet temperature for *hexA* (continuous line), the outlet temperature for *hexB* (dotted line) and the outlet temperature for *hexA* (dashed line).

It should be pointed out that the last experiment has been conceived also to test the numerical robustness for the model: the results have shown that the coupling of two FEM-based components (*hexA* and *hexB*) does not affect the numerical stability, even for large values of the heat exchange coefficient. Further tests with different stabilization coefficients, not reported for the sake of brevity but available on-line, have confirmed the absence of numerical instabilities.

7   Conclusions and Work in Progress

A Modelica FEM-based model for heat exchangers has been presented. The model has been implemented into a specific component (`Flow1Dfem`) which is included in the *ThermoPower* library, developed for thermal power plants modelling, simulation and control.

The component, whose internal implementation is completely shielded from the connectors, has been validated through simulations for specific plants configurations.

The possibility to effectively use Modelica to model physical systems that are originally described by PDEs has been shown in the specific case of the advection

equation.

Current work is headed essentially in two directions:

- the further improvement of the developed model with particular emphasis on extensions to handle also phase changes along the spatial domain;

- the development of Modelica models for other systems described by PDEs, such as flexible robot links.

References

[1] F. Casella and A. Leva. Modelica open library for power plant simulation: Design and experimental validation. In *Modelica Conference*, Linköping, Sweden, 2003. www.elet.polimi.it/upload/casella/thermopower/.

[2] F. P. Incropera and D. P. DeWitt. *Fundamentals of Heat and Mass Transfer*. John Wiley & Sons, 1985.

[3] A. Quarteroni and A. Valli. *Numerical Approximation of Partial Differential Equations*. Springer Verlag, 1997.

[4] C. Johnson. *Numerical Solution of Partial Differential Equations by the Finite Element Method*. Cambridge University Press, 1987.

[5] M. L. Aime. *Engineering Methods And Tools For Modeling And Simulation Of Power Generation Plants*. PhD thesis, Politecnico Di Milano, 1999.

[6] C. Baiocchi and F. Brezzi. Stabilization of unstable numerical methods. In *Problemi Attuali dell'Analisi e della Fisica Matematica*, Taormina, Italy, 1992.

[7] H. Tummescheit and J. Eborn. Design of a thermo-hydraulic model library in Modelica. In *ESM (European Simulation Multiconference)*, Manchester, UK, 1998.

[8] Dymola 5.1b. *Dynasim AB*. Lund, Sweden.

A   Matrices Expression

$$M = l \begin{bmatrix} \frac{1}{3} - \frac{\alpha}{4} & \frac{1}{6} - \frac{\alpha}{4} & & \\ \frac{1}{6} + \frac{\alpha}{4} & \frac{2}{3} & \ddots & \\ & \ddots & \ddots & \frac{1}{6} + \frac{\alpha}{4} \\ & & \frac{1}{6} + \frac{\alpha}{4} & \frac{1}{3} + \frac{\alpha}{4} \end{bmatrix}$$

$$B = l \begin{bmatrix} B_{11} & B_{i,i+1} & & \\ B_{i,i-1} & B_{ii} & \ddots & \\ & \ddots & \ddots & B_{i,i+1} \\ & & B_{i,i-1} & B_{NN} \end{bmatrix}$$

$B_{11} = v_1 \left(-\frac{1}{3} + \frac{\alpha}{4}\right) + v_2 \left(-\frac{1}{6} + \frac{\alpha}{4}\right)$
$B_{i,i+1} = v_i \left(\frac{1}{3} - \frac{\alpha}{4}\right) + v_{i+1} \left(\frac{1}{6} - \frac{\alpha}{4}\right)$
$B_{i,i-1} = v_i \left(-\frac{1}{3} - \frac{\alpha}{4}\right) + v_{i-1} \left(-\frac{1}{6} - \frac{\alpha}{4}\right)$
$B_{i,i} = v_{i-1} \left(\frac{1}{6} + \frac{\alpha}{4}\right) + v_i \frac{\alpha}{2} + v_{i+1} \left(-\frac{1}{6} + \frac{\alpha}{4}\right)$
$B_{NN} = v_{N-1} \left(\frac{1}{6} + \frac{\alpha}{4}\right) + v_N \left(\frac{1}{3} + \frac{\alpha}{4}\right)$

$$Y = l \begin{bmatrix} Y_{11} & Y_{i,i+1} & & \\ Y_{i,i-1} & Y_{ii} & \ddots & \\ & \ddots & \ddots & Y_{i,i+1} \\ & & Y_{i,i-1} & Y_{NN} \end{bmatrix}$$

$Y_{11} = v_1 \left(\frac{1}{4} - \frac{\alpha}{6}\right) + v_2 \left(\frac{1}{12} - \frac{\alpha}{12}\right)$
$Y_{i,i+1} = v_i \left(\frac{1}{12} - \frac{\alpha}{12}\right) + v_{i+1} \left(\frac{1}{12} - \frac{\alpha}{12}\right)$
$Y_{i,i-1} = v_i \left(\frac{1}{12} + \frac{\alpha}{12}\right) + v_{i-1} \left(\frac{1}{12} + \frac{\alpha}{6}\right)$
$Y_{i,i} = v_{i-1} \left(\frac{1}{12} + \frac{\alpha}{12}\right) + v_i \frac{1}{2} + v_{i+1} \left(\frac{1}{12} - \frac{\alpha}{12}\right)$
$Y_{NN} = v_{N-1} \left(\frac{1}{12} + \frac{\alpha}{12}\right) + v_N \left(\frac{1}{4} + \frac{\alpha}{6}\right)$

$$E = l \begin{bmatrix} \frac{1}{3} & \frac{1}{6} & & \\ \frac{1}{6} & \frac{2}{3} & \ddots & \\ & \ddots & \ddots & \frac{1}{6} \\ & & \frac{1}{6} & \frac{1}{3} \end{bmatrix}$$

$$D = \begin{bmatrix} \frac{1}{2} & 1 \cdots 1 & \frac{1}{2} \end{bmatrix}^T$$