Port-Based Modeling of Transport Phenomena in 1-D Thermofluid Systems and Object-Oriented Implementation With the Modelica Language

Francisco M. Márquez, Pedro J. Zufiria, and Luis J. Yebra

***Abstract*—In this article, we present the physical foundations and the development of a Modelica library with components for modeling 1-D thermofluid systems. Modelica was selected because it is an object-oriented modeling language that facilitates the incremental design of the library. Modelica also allows the modeling of acausal components, where the input/output relation is defined by the boundary conditions under which models are simulated. We model single-substance systems that are macroscopically homogeneous, isotropic, and uncharged, which we call "simple systems." To model the behavior of these systems, we assume the postulate of classical irreversible thermodynamics. For the graphical representation of complex systems, we utilize a modified bond graph symbology to enhance the readability of the diagrams. We also implement the equations for viscosity and thermal conductivity to complete the IAPWS-95 model of water, enabling its use as a fluid in the piping systems presented in our examples.**

***Index Terms*—Bond graphs, Modelica language, object-oriented modeling, port-based modeling, thermofluid systems.**

I. Introduction

THERMOFLUID systems are fundamental to the chemical, energy, and process industries, playing a central role in processes, such as energy conversion, heat recovery, fluid transport, and thermal regulation. Their importance is increasingly evident in the context of the global transition toward clean, sustainable, and economically viable energy solutions. Accurate modeling and control of these systems are essential for optimizing performance, improving energy efficiency, and reducing environmental impact. A notable example is solar thermal power plants, where accurately modeling the behavior of heat transfer fluids (HTFs) within absorber tubes is critical to ensuring proper control and efficient operation. Similarly, thermofluid modeling plays a vital role in the design of heating, ventilation, and air conditioning systems, which are essential for maintaining optimal indoor environments in both large buildings and residential settings. Moreover, the growing demand for high-performance thermal management in emerging applications (such as the cooling of data centers and cloud computing infrastructure) underscores the need for robust modeling tools. These facilities generate substantial heat loads and require precise control strategies to ensure reliability, energy efficiency, and environmental compliance.

In this article, we derive port-based thermofluid models considering the phenomena of convective transport of mass, momentum, and entropy, as well as their associated dissipative phenomena. For the thermodynamic part, we will follow the port-based approach to classical thermodynamics and classical irreversible thermodynamics, as introduced in [19].

A thermofluid system is generally composed of several subsystems with substances flowing through ducts, exchanging mass, momentum, and energy with each other and with their surroundings. Modeling and controlling the behavior of these systems is not an easy task and necessarily requires a multidisciplinary approach with contributions from thermodynamics, fluid dynamics, electromechanics, or control theory. This difficulty justifies the approach and tools we have chosen to address this problem: 1) the port-based formalism [8] for modeling complex dissipative systems composed of simple subsystems that store and exchange energy through interconnection ports, 2) the bond graph methodology [3] for their graphical representation, and 3) the Modelica language [20] for their implementation and simulation. Our approach integrates the three

Received 9 July 2025; accepted 5 August 2025. The work of Luis J. Yebra was supported by the Universidad de Alcalá through the "Programa propio Giner de los Ríos" Research Grant. The work of Pedro J. Zufiria was supported by the Spanish Ministry of Science and Innovation through MCIN/AEI/10.13039/501100011033 under Grant PID2023-146540OB-C42. This work was supported by MCIN/AEI/10.13039/501100011033 and European Union "NextGenerationEU"/PRTR under Grant TED2021-129189B-C21/TED2021-129189B-C22, Grant PCI2022-134974-2, and Grant PID2023-146540OB-C42. Paper no. TII-25-4546. *(Corresponding author: Francisco M. Márquez.)*

Francisco M. Márquez is with the Departamento de Automática, Universidad de Alcalá (UAH), 28801 Alcalá de Henares, Spain (e-mail: francisco.marquez@uah.es).

Pedro J. Zufiria is with the Departamento de Matemática Aplicada a las Tecnologías de la Información y las Comunicaciones, Information Processing and Telecommunications Center (IPTC), ETSI Telecomunicación, Universidad Politécnica de Madrid (UPM), 28040 Madrid, Spain (e-mail: pedro.zufiria@upm.es).

Luis J. Yebra is with the Plataforma Solar de Almería, CIEMAT, 04200 Tabernas, Spain (e-mail: luis.yebra@psa.es).

Digital Object Identifier 10.1109/TII.2025.3598496

abovementioned modeling tools, which until now have often been used separately.

A. Literature Review

The work of Paynter [23] on the modeling of complex multidomain systems, focusing on energy storage and transfer between system components through bonds, can be considered an early predecessor work to the idea of port-based systems. The formalization of these concepts in terms of differential geometry led to the definition of port-Hamiltonian systems [28]. In [3] and [14], we find a detailed explanation of the bond graph methodology and many examples of its application to model multidisciplinary engineering systems. For a general introduction to port-Hamiltonian systems, we refer to [27], while a more detailed explanation of the modeling and control of complex systems using the port-Hamiltonian approach, accompanied by illustrative examples, can be found in [8]. The modeling of physical systems using both bond graph representation and the port-Hamiltonian formalism shares fundamental elements centered on the concept of energy exchange through power ports. These power ports represent the interaction between components via conjugate variables (effort and flow) whose product corresponds to the exchanged power. Both approaches describe system behavior by tracking energy flow and storage, ensuring an explicit representation of interactions among components via these ports. However, the bond graph method is primarily a graphical tool that intuitively captures the structure and causality of energy transfer, facilitating visualization and model construction. In contrast, the port-Hamiltonian formalism provides a more rigorous mathematical framework based on differential geometry, where Dirac structures formalize the interconnection and energy exchange between system components. Notably, bond graphs can serve as a practical graphical means to represent port-Hamiltonian systems, combining rigorous theory with intuitive visualization. See [18] for a more comprehensive discussion on these modeling approaches.

The latest revision of the Modelica language specification can be found in [20]. The authors in [6] and [29] describe the development of a Modelica library for modeling systems with bond graphs. Two Modelica libraries dealing with the modeling of thermodynamic and fluid systems are Modelica.Fluid and ThermoPower [5], where the authors propose an applied orientation to the modeling of thermofluid systems and power plants, with an important number of examples ready to be simulated.

The axiomatic presentation of equilibrium thermodynamics can be found in [4] and the treatment of classical irreversible thermodynamics in [7]. Öttigner [22] presented the general equations for the nonequilibrium reversible–irreversible coupling (GENERIC) formalism, where he developed the nonequilibrium thermodynamics with Hamiltonian structure, and the authors in [16] and [17] related the GENERIC formalism to the port-Hamiltonian approach by starting from the concept of system exergy.

Regarding transport phenomena, the standard references considered are [2] and [25]. Tosun [26] studied these phenomena with a different approach since he started with the integral balances (of momentum, energy, and mass) and then elaborated on the microscopic balances (equations of change).

In order to address the modeling of complex nonequilibrium thermodynamic systems, the network theory of complex systems, the bond graph representation, and the lumped-parameter Hamiltonian formulation are explored in [21] and [24] (under the denomination of network thermodynamics). As we will show in this article, in the context of thermofluid systems, control volume networks play a similar role in modeling complex systems to those defined in network thermodynamics.

B. Contributions

In this article, we complement the thermodynamic part of the library pHlib described in [19] for modeling thermofluid systems by adding the transport phenomena related to fluid motion. To justify the inclusion in the library of new components, as well as the structure of some subsystems (e.g., pipes), we make a preliminary study of the classical theory of thermofluid systems, starting with the formulation of the evolution equations in differential form and continuing with the integral form (called integral balances). We use the latter equations to obtain lumped-parameter models of components (control volumes) that we will interconnect in order to model complex thermofluid systems.

In the thermal domain, port-Hamiltonian and bond graph modeling use entropy and entropy flow rate as state and flow variables, respectively. Therefore, as an alternative to considering the usual balances in the modeling of thermofluid systems (mass, momentum, and energy), we have paid attention to the entropy balance instead of the energy balance. We have thus obtained general expressions for entropy transported by convection, generated by viscous friction, and by heat conduction. These expressions, together with mass and momentum balance, have been incorporated into the generic model of a control volume with two ports for exchanging matter (with information related to mass, volume, momentum, and entropy) and one port for exchanging heat (with information related to entropy).

In this article, we extend the thermodynamic model for generic substances to the thermofluid domain by adding the transport properties (viscosity and thermal conductivity). The object-oriented design of the library pHlib has facilitated this task since it was sufficient to extend the initial class SingleSubstance defined in the package Thermodynamics.Substances.Partial and add two new equations to each substance. With this extension, we have completed the formulation 1995 for the thermodynamic properties of water for general and scientific use (IAPWS-95 [13]). The implementation of the IAPWS-95 model is particularly interesting because it allows establishing comparisons of accuracy and simulation execution time with the IAPWS-IF97 [10] model implemented with Modelica Standard Library (MSL).

We have also extended and adapted to the thermofluid domain the set of graphical symbols derived from the bond graph representation. These symbols facilitate the creation of models with the graphical user interface of Modelica.

For performing simulations and comparing pHlib with MSL, we have implemented a pipeline model working in turbulent flow regime. This model extends the general two-port control

volume model and is parameterized with the pipe's geometrical properties and position. To calculate the integral entropy balances, we assume a perfect mixing of the fluid (this assumption is usual in turbulent flows), and we have also applied the customary correlations for the friction factor (Colebrook correlation [26, eq. (4.5-9)]) and the heat transfer coefficient (Dittus and Boelter correlation [26, eq. (4.5-26)]).

II. Evolution Equations for Mass, Momentum, and Entropy in Thermofluid Systems

For a system modeled as a network of connected control volumes, the conservation, transport, and production of the main physical quantities must be considered. In single-substance thermofluid systems, these quantities are typically mass, momentum, and energy. However, in the context of energy-based port modeling approaches, such as port-Hamiltonian and bond graph formalisms, the thermal domain is described using the entropy S as a state variable and the conjugate pair $\langle T|\dot{S}\rangle = \langle\text{temperature}|\text{entropy flow rate}\rangle$ to define ports for energy transfer. This approach contrasts with common industrial practice, where entropy modeling is often avoided due to its mathematical complexity, leading instead to the use of pseudobond graphs based, for example, on heat and heat flow rate (see [3, Sect. 10.1] and [14, Sect. 12.5]). While such approximations may simplify implementation, they sacrifice the formal structure of the underlying port-based framework. We, therefore, present the evolution equations for mass, momentum, and entropy in a unified and port-compatible formulation, employing the standard notation of the transport phenomena literature (see [2, Appendix A]).

Transport phenomena are molecular and are modeled through partial differential equations in three independent space variables and time, called *evolution equations*. To complete the model, we need the *constitutive equations*, "which serve to distinguish different types of material behavior" [9, p. 115].

The evolution equation of the extensive variable $F(r,t)$, $(r \in \Omega \subset \mathbb{R}^3, t \in \mathbb{R})$, expressed from its density $f(r,t) = \mathrm{d}F/\mathrm{d}V$ is

$$\frac{\partial f}{\partial t} + \nabla \cdot \boldsymbol{\Phi}_f = q_f + s_f \tag{1}$$

where $\boldsymbol{\Phi}_f$ is the flux of F (amount of F which flows per unit surface and unit time), $q_f(r,t)$ is the rate of production/consumption per unit volume of the quantity F (e.g., density of energy produced or consumed by chemical reaction), and $s_f(r,t)$ is a *long-range process* density from the surroundings (e.g., radiation, gravity, etc.).

From the quantities defined in Table I, our proposal consists in deriving the evolution equations for mass (2), momentum (3) (see [2, Ch. 11]), and entropy (4) (see [25, Ch. 5]), as an alternative port-based formulation to the standard ones that explicitly consider the energy

$$\frac{\partial \varrho}{\partial t} + \nabla \cdot (\varrho \mathbf{v}) = 0 \tag{2}$$

$$\frac{\partial \varrho \mathbf{v}}{\partial t} + \nabla \cdot (\varrho \mathbf{v}\mathbf{v} + p\mathbf{I} + \boldsymbol{\tau}) = \varrho \mathbf{g} \tag{3}$$

TABLE I
Quantities for the Evolution Equations (2)–(4) of Single Substance Thermofluids Systems

	F	f	$\boldsymbol{\Phi}_f$	q_f	s_f
Mass	m	ϱ	$\varrho\mathbf{v}$	0	0
Momentum	$m\mathbf{v}$	$\varrho\mathbf{v}$	$\varrho\mathbf{v}\mathbf{v} + p\mathbf{I} + \boldsymbol{\tau}$	0	$\varrho\mathbf{g}$
Entropy	ms	ϱs	$\varrho\mathbf{v}s + \frac{1}{T}\mathbf{j}_q$	$-\frac{1}{T^2}(\mathbf{j}_q \cdot \nabla T))$ $-\frac{1}{T}(\boldsymbol{\tau} : \nabla\mathbf{v})$	$\frac{1}{T}q_r$

$\{\mathbf{v}\mathbf{v}\}_{ij} = [\mathbf{v}]_i[\mathbf{v}]_j, 1 \le i \le N, 1 \le j \le N$ (diadic product), $(\nabla \cdot \mathbf{v}) = \sum_{i=1}^{N} \partial[\mathbf{v}]_i/\partial x_i$ (divergence of a vector field), $\{\nabla\mathbf{v}\}_{ij} = \partial[\mathbf{v}]_j/\partial x_i, 1 \le i \le N, 1 \le j \le N$ (grad. of a vector field), $(\boldsymbol{\tau} : \nabla\mathbf{v}) = \mathrm{trace}\{\boldsymbol{\tau} \cdot \nabla\mathbf{v}\} = \sum_{i=1}^{N}\sum_{j=1}^{N}\{\boldsymbol{\tau}\}_{ij}\partial[\mathbf{v}]_i/\partial x_j$. The symbols are defined in Appendix.

TABLE II
Constitutive Equations for Single Substance Thermofluid Systems [15, pp. 45 and 192]

Newton's Law	$\boldsymbol{\tau} = -\mu\{\nabla\mathbf{v} + (\nabla\mathbf{v})^\intercal\} + (\frac{2}{3}\mu - \kappa)(\nabla \cdot \mathbf{v})\mathbf{I}$	Pa
Fourier's Law	$\mathbf{j}_q = -\lambda\nabla T$	W/m^2

$$\frac{\partial \varrho s}{\partial t} + \nabla \cdot \left(\varrho\mathbf{v}s + \frac{1}{T}\mathbf{j}_q\right) = -\frac{1}{T^2}(\mathbf{j}_q \cdot \nabla T) - \frac{1}{T}(\boldsymbol{\tau} : \nabla\mathbf{v}) + \frac{1}{T}q_r. \tag{4}$$

Table II gives the constitutive equations that complete the model of the system. If we do not have enough constitutive equations, the model will have more unknowns than equations (indeterminate system). In this case, experimental information usually referred to as "experimental correlations" is used; it is rather common that such experimental information is limited to a specific geometry of the control volume $\Omega(t)$ where the equations are defined.

To study transport phenomena at a "macroscopic level" in practical engineering applications, the evolution equations are integrated in each control volume that exchanges mass, momentum, and entropy with the environment. In the following section we derive the "integral balances" that allow to obtain lumped-parameter models of dynamic systems described by ordinary differential equations (see Sections III-A–III-C). From these equations, we will identify the state variables, as well as the effort and flow variables, of a port-based model of the thermofluid system (see Sect. III-D).

III. Integral Balances for Thermofluid Systems

The evolution (2)–(4), the constitutive equations of the involved materials (see Table II), and the boundary conditions altogether characterize a thermofluid system, but their solution has to be found usually by numerical methods, either because the geometry of the volume Ω is so complex that it is not easy to define the conditions at the boundary $\partial\Omega$, or because the constitutive equations are nonlinear.

Fortunately, for many engineering applications, it is not necessary to know all variables at each point of the volume Ω,

so it is sufficient to decompose the system into a network of connected control volumes and characterize each volume with a set of lumped parameters.

The integration of (2)–(4), by applying the Ostrogradsky–Gauss theorem and the Reynolds' transport theorem [9, p. 78] to a time-dependent volume $\Omega(t)$ (with a boundary $\partial\Omega(t)$ that moves at velocity $\mathbf{v}_A$), leads to a set of ordinary differential equations that provides a macroscopic description of the system

$$\dot{m} = \frac{\mathrm{d}}{\mathrm{d}t}\int_{\Omega}\varrho \mathrm{d}V = \int_{\partial\Omega}\varrho[\mathbf{v}_A - \mathbf{v}]\cdot\mathrm{d}\mathbf{A} \tag{5}$$

$$\begin{aligned}\dot{\mathbf{P}} = \frac{\mathrm{d}}{\mathrm{d}t}\int_{\Omega}\varrho\mathbf{v}\mathrm{d}V &= \int_{\partial\Omega}\varrho\mathbf{v}\left([\mathbf{v}_A - \mathbf{v}]\cdot\mathrm{d}\mathbf{A}\right)\\ &- \int_{\partial\Omega}\{p\mathbf{I}+\boldsymbol{\tau}\}\cdot\mathrm{d}\mathbf{A} + \int_{\Omega}\varrho\mathbf{g}\mathrm{d}V\end{aligned} \tag{6}$$

$$\begin{aligned}\dot{S} = \frac{\mathrm{d}}{\mathrm{d}t}\int_{\Omega}\varrho s\mathrm{d}V &= \int_{\partial\Omega}\varrho s[\mathbf{v}_A - \mathbf{v}]\cdot\mathrm{d}\mathbf{A}\\ &- \int_{\partial\Omega}\frac{1}{T}\mathbf{j}_q\cdot\mathrm{d}\mathbf{A} - \int_{\Omega}\frac{1}{T^2}(\mathbf{j}_q\cdot\nabla T)\mathrm{d}V\\ &- \int_{\Omega}\frac{1}{T}(\boldsymbol{\tau}:\nabla\mathbf{v})\mathrm{d}V + \int_{\Omega}\frac{1}{T}q_r\mathrm{d}V.\end{aligned} \tag{7}$$

In the following sections we compute the vector fields for each one of these rate equations.

A. Mass Flow Rate

The vector $[\mathbf{v}_A - \mathbf{v}]$ represents the velocity of the boundary $\partial\Omega$ relative to the fluid. The fluid enters the control volume if $[\mathbf{v}_A - \mathbf{v}]\cdot\mathrm{d}\mathbf{A} > 0$, and leaves it if $[\mathbf{v}_A - \mathbf{v}]\cdot\mathrm{d}\mathbf{A} < 0$. If we consider the specific quantity $x = X/m$ and decompose the boundary $\partial\Omega$ into N parts $\partial\Omega_{1\ldots N}$ through which the fluid enters or exits, then the integral $\int_{\partial\Omega}\varrho x[\mathbf{v}_A - \mathbf{v}]\cdot\mathrm{d}\mathbf{A}$ represents the flow rate of X transported by the incoming and outgoing fluid (convective flow rate). This is the case for mass transport (5) in a thermofluid system with N ports with boundaries $\partial\Omega_{1\ldots N}$, such that $\mathbf{v}_A - \mathbf{v} = 0$ at the boundary $\partial\Omega_0 = \partial\Omega\backslash\bigcup\partial\Omega_{1\ldots N}$ [Assumption (i)]

$$\begin{aligned}\dot{m} &= \int_{\partial\Omega}\varrho[\mathbf{v}_A - \mathbf{v}]\cdot\mathrm{d}\mathbf{A}\\ &= \sum_{i\in 1\ldots N}\int_{\partial\Omega_i}\varrho[\mathbf{v}_A - \mathbf{v}]\cdot\mathrm{d}\mathbf{A} = \sum_{i\in 1\ldots N}\dot{m}_i.\end{aligned} \tag{8}$$

If we call $\mathbf{v}_{\partial\Omega_i}$ the average of the velocity $[\mathbf{v}_A - \mathbf{v}]$ at the boundary $\partial\Omega_i$ (more details on area averaging of quantities in [25, Ch. 4 and 7]), and assume that: (ii) $\partial\Omega_i$ is a plane with constant cross section $\mathbf{A}_i$, (iii) the fluid enters or leaves orthogonally each plane $\partial\Omega_i$, and (iv) the density ϱ_i is uniform in the boundary $\partial\Omega_i$, then (see Fig. 1)

$$\dot{m} = \sum_{i\in 1\ldots N}\varrho_i\mathbf{v}_{\partial\Omega_i}\cdot\mathbf{A}_i. \tag{9}$$

B. Rate of Change for Momentum

The first term of (6) is the convective part $\dot{\mathbf{P}}_{\text{conv}}$ of the rate of change for momentum $\dot{\mathbf{P}}$. If we call $\mathbf{v}_i$ the average of the fluid velocity at the boundary $\partial\Omega_i$ and we keep the assumptions (i)–(iv) of the previous paragraph, then

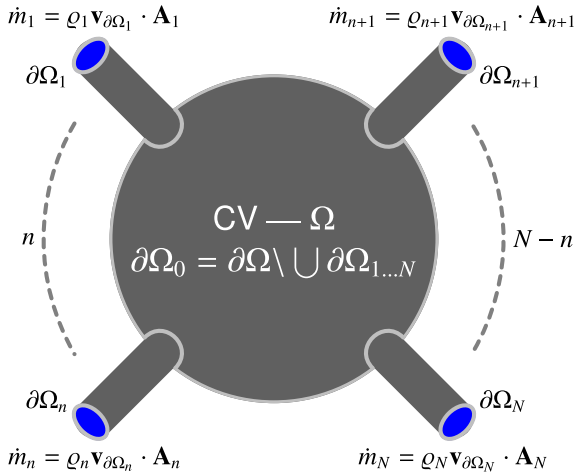


Fig. 1. Control volume Ω with N ports. It should be noted that $\mathbf{v}_A - \mathbf{v} = 0$ in $\partial\Omega_0$ (outside the ports).

$$\begin{aligned}\dot{\mathbf{P}}_{\text{conv}} &= \int_{\partial\Omega}\varrho\mathbf{v}\left([\mathbf{v}_A - \mathbf{v}]\cdot\mathrm{d}\mathbf{A}\right)\\ &= \sum_{i\in 1\ldots N}\dot{m}_i\mathbf{v}_i = \sum_{i\in 1\ldots N}\dot{\mathbf{P}}_{\text{conv},i}.\end{aligned} \tag{10}$$

The second term of (6) accounts for the molecular transport contribution $\dot{\mathbf{P}}_{\text{molec}}$ to the rate of change for momentum. In practical applications, it is common to assume: (v) area-averaged pressure p_i at cross section of each port $\partial\Omega_{1\ldots N}$ and (vi) zero viscous forces (or neglected in comparison with the pressure forces) on $\partial\Omega_{1\ldots N}$. Under assumptions (i)–(vi), we can write

$$\begin{aligned}\dot{\mathbf{P}}_{\text{molec}} &= \int_{\partial\Omega}\{p\mathbf{I}+\boldsymbol{\tau}\}\cdot\mathrm{d}\mathbf{A}\\ &= \sum_{i\in 1\ldots N}\int_{\partial\Omega_i}p\mathrm{d}\mathbf{A} + \int_{\partial\Omega_0}\{p\mathbf{I}+\boldsymbol{\tau}\}\cdot\mathrm{d}\mathbf{A}\\ &= \sum_{i\in 1\ldots N}p_i\mathbf{A}_i + \int_{\partial\Omega_0}p\mathrm{d}\mathbf{A} + \int_{\partial\Omega_0}\boldsymbol{\tau}\cdot\mathrm{d}\mathbf{A}\\ &= \sum_{i\in 1\ldots N}\dot{\mathbf{P}}_{\text{press},i} + \dot{\mathbf{P}}_{\text{press},0} + \dot{\mathbf{P}}_{\text{visc},0}.\end{aligned} \tag{11}$$

The third term of (6)

$$\dot{\mathbf{P}}_{\text{grav}} = \int_{\Omega}\varrho\mathbf{g}\mathrm{d}V = m\mathbf{g} \tag{12}$$

accounts for gravitational contribution to the total rate of change for momentum, leading to

$$\begin{aligned}\dot{\mathbf{P}} &= \dot{\mathbf{P}}_{\text{conv}} - \dot{\mathbf{P}}_{\text{molec}} + \dot{\mathbf{P}}_{\text{grav}}\\ &= \sum_{i\in 1\ldots N}\dot{m}_i\mathbf{v}_i - \left[\sum_{i\in 1\ldots N}p_i\mathbf{A}_i + \dot{\mathbf{P}}_{\text{press},0} + \dot{\mathbf{P}}_{\text{visc},0}\right] + m\mathbf{g}.\end{aligned} \tag{13}$$

C. Entropy Flow Rate

The first term of (7) is the convective part of the entropy flow rate. If we keep the previous assumptions (i)–(vi), and assume area-averaged specific entropies $s_{1\dots N}$ at boundaries $\partial\Omega_{1\dots N}$ [Assumption (vii)], then

$$\begin{aligned}\dot{S}_{\text{conv}} &= \int_{\partial\Omega} \varrho s[\mathbf{v}_A - \mathbf{v}]\cdot \mathrm{d}\mathbf{A} = \sum_{i\in 1\dots N} s_i\left(\varrho_i \mathbf{v}_{\partial\Omega_i}\cdot \mathbf{A}_i\right)\\ &= \sum_{i\in 1\dots N} \dot{m}_i s_i = \sum_{i\in 1\dots N} \dot{S}_{\text{conv},i}. \end{aligned} \tag{14}$$

The second term of (7) is the entropy flow rate by heat conduction through the system boundary. If we assume area-averaged temperatures $T_{1\dots N}$ on ports $\partial\Omega_{1\dots N}$ [Assumption (viii)] in addition to assumptions (i)–(iv), then

$$\begin{aligned}\dot{S}_{\text{cond},\partial\Omega} &= -\int_{\partial\Omega} \frac{1}{T}\mathbf{j}_q\cdot \mathrm{d}\mathbf{A}\\ &= \sum_{i\in 1\dots N}\int_{\partial\Omega_i}\frac{\lambda}{T_i}\nabla T\cdot \mathrm{d}\mathbf{A} + \int_{\partial\Omega_0}\frac{\lambda}{T}\nabla T\cdot \mathrm{d}\mathbf{A}\\ &= \sum_{i\in 1\dots N}\dot{S}_{\text{cond},i} + \dot{S}_{\text{cond},0}. \end{aligned} \tag{15}$$

The third and fourth terms of (7) correspond to the generation of entropy inside the control volume, and both terms are positive. Indeed, the third term is due to heat conduction, and the integrand is a positive quantity

$$\dot{S}_{\text{cond},\Omega} = -\int_\Omega \frac{1}{T^2}(\mathbf{j}_q\cdot\nabla T)\mathrm{d}V = \int_\Omega \lambda\frac{|\nabla T|^2}{T^2}\mathrm{d}V \geq 0 \tag{16}$$

and the fourth term is the entropy produced by viscous dissipation, and the integrand $-\frac{1}{T}(\boldsymbol{\tau}:\nabla\mathbf{v})$ is also positive (see [2, p. 82])

$$\dot{S}_{\text{visc}} = -\int_\Omega \frac{1}{T}(\boldsymbol{\tau}:\nabla\mathbf{v})\mathrm{d}V \geq 0. \tag{17}$$

The last term of (7), $\dot{S}_{q_r} = \int_\Omega \frac{1}{T}q_r \mathrm{d}V$, is the entropy flow rate by the long-range processes from the surroundings, and no further simplification will be made at this time. In summary

$$\dot{S} = \sum_{i\in 1\dots N}\dot{m}_i s_i + \dot{S}_{\text{cond},\partial\Omega} + \dot{S}_{\text{cond},\Omega} + \dot{S}_{\text{visc}} + \dot{S}_{q_r}. \tag{18}$$

We can now identify the state, effort, and flow variables of a port-based model of the thermofluid system, as shown in the next section.

D. Port-Based Variables

According to the definition of *power port* (see [18, Def. 1]), for thermofluids systems, joining the thermodynamic part (see [19, Sect. IV.A]) and the transport part [see (5)–(7)], the state variables used to define the lumped port-based models are $x = (\mathbf{P}, S, V, m) \in \mathbb{R}^3\times\mathbb{R}^3_+$ (where $\mathbb{R}_+ = \{z\in\mathbb{R} : z > 0\}$). Based on the previous analysis, the total energy E of a system, the flow variable $\dot{x}$, the effort variable $\mathrm{d}E$, and the energy flow rate $\dot{E}$ are

$$\begin{aligned} E &: \mathbb{R}^3\times\mathbb{R}^3_+ \to \mathbb{R}\\ \dot{x} &= \dot{\mathbf{P}}\cdot\partial_{\mathbf{P}} + \dot{S}\partial_S + \dot{V}\partial_V + \dot{m}\partial_m \in T(\mathbb{R}^3\times\mathbb{R}^3_+)\\ \mathrm{d}E &= \frac{\partial E}{\partial \mathbf{P}}\cdot\mathrm{d}\mathbf{P} + \frac{\partial E}{\partial S}\mathrm{d}S + \frac{\partial E}{\partial V}\mathrm{d}V + \frac{\partial E}{\partial m}\mathrm{d}m\\ &= \mathbf{v}\cdot\mathrm{d}\mathbf{P} + T\mathrm{d}S - p\mathrm{d}V + g\mathrm{d}m \in T^*(\mathbb{R}^3\times\mathbb{R}^3_+)\\ \dot{E} &= \langle \mathrm{d}E|\dot{x}\rangle = \mathbf{v}\cdot\dot{\mathbf{P}} + T\dot{S} - p\dot{V} + g\dot{m}. \end{aligned} \tag{19}$$

Listing 1: Thermofluids.Interfaces.

```
connector KineticPort
  extends Port (dimPort=1,
    redeclare SI.Velocity e,
    redeclare SI.MomentumFlux f);
end KineticPort;

connector ThermofluidPort
  extends Port(dimPort = 4);
end ThermofluidPort;

class KineticBond
  SI.Velocity v = from.e[1];
  SI.MomentumFlux dot_P = from.f[1];
  extends OneDimensionalBond (
    redeclare connector BasePort = KineticPort);
end KineticBond;

class ThermofluidBond
  SI.Velocity v = from.e[4];
  SI.MomentumFlux dot_P = from.f[4];
  extends ThermodynamicBond (
    redeclare connector BasePort = ThermofluidPort);
end ThermofluidBond;
```

IV. Modelica Implementation of 1-D Thermofluid Systems

In this section, we develop and add the package Thermofluids to the library pHlib for modeling 1-D thermofluid systems. Taking advantage of the inheritance mechanism offered by the Modelica language, we have declared new classes that extend the classes defined in the packages Dirac, pHS [18], and Thermodynamics [19].

A. Power Ports and Bonds

To model the energy transport in a thermofluid system, we need ports and bonds related to the mechanical energy, in addition to those already defined in thermodynamic systems for the internal energy [see (19)]. Listing 1 shows these new ports and bonds defined in the package Thermofluids for modeling 1-D thermofluid systems.

Fig. 2 shows the symbols used for graphical representation of bonds, sources, storage elements, and other elements defined in the package Thermofluids. These symbols facilitate the composition of complex systems with the Modelica drag-and-drop graphical user interface.

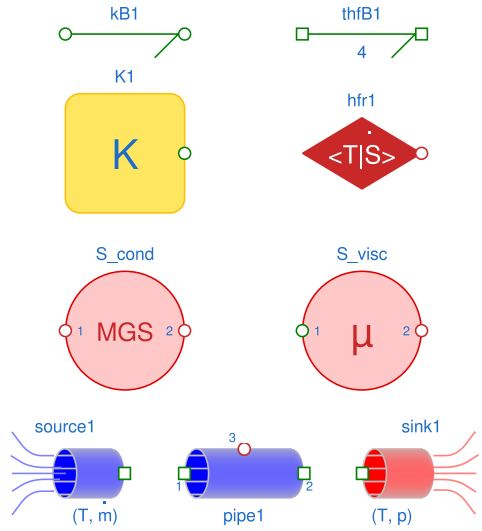


Fig. 2. Symbols added to pHlib: kB1 (kinetic bond), thfB1 (4-D thermofluid bond), K1 (kinetic energy storage element), hfr1 (heat flow rate source), S_cond (thermal conductor), S_visc (viscous resistor), source1 (fluid source), pipe1 (pipe), and sink1 (fluid sink).

Listing 2: Thermofluids.Substances.Partial.SingleSubstance.

```
partial class SingleSubstance
  replaceable class BaseSubstance =
    Thermodynamics.Substances.Partial.SingleSubstance;
  extends BaseSubstance;
  // Transport properties.
  replaceable SI.DynamicViscosity mu "Viscosity";
  replaceable SI.ThermalConductivity lambda "Thermal conductivity";
  // Dimensionless numbers
  SI.PrandtlNumber Pr = c_p*mu/lambda "Prandtl number";
end SingleSubstance;
```

B. Transport Properties of Single Substances

The class SingleSubstance (see Listing 2) incorporates the transport properties (viscosity μ and thermal conductivity λ), in addition to the thermodynamic properties that are modeled in the class Thermodynamics.Substances.Partial.SingleSubstance, from which it is derived. Both classes are partial classes that we will use as templates to define derived classes for modeling the properties of specific substances.

C. Transport Properties of Water

The class WaterIAPWS95 (see Listing 3) extends SingleSubstance and Termodynamics.Substances.WaterIAPWS95 (see [19, Listing 12]), and models water transport properties described in [11] for viscosity and in [12] for thermal conductivity. This implementation has been validated with the computer program verification test values given in [11, Table 4] for viscosity with $\bar{\mu}_2 = 1$ and [12, Table 4] for thermal conductivity with $\bar{\lambda}_2 = 0$.

D. Storage Elements

Thermodynamics.Storage.InternalEnergy [19, Listing 4] is the base class for Thermofluids.Storage.InternalEnergy and, as we can see in Listing 4, the set of substances we can choose to model the internal energy behavior of the control volumes that make up a thermofluid system is constrained by the class SingleSubstance (see Section IV-B).

To model the kinetic energy, we define the class KineticEnergy (see Listing 5), which extends pHS.Storage.C [18, Def. 23].

Listing 3: Thermofluids.Substances.WaterIAPWS95.

```
class WaterIAPWS95 "IAPWS R6–95(2008) + R12–08 + R15–11"
  // IAPWS R6-95 (2018) - Thermodynamic properties [19, Listing 12]
  extends Partial.SingleSubstance (redeclare class BaseSubstance =
      Thermodynamics.Substances.WaterIAPWS95);
  //
  // IAPWS R12-18 (2008) - VISCOSITY
  // Reference constants.
  constant SI.Temperature T_asterisk = T_c; // T*
  constant SI.Density rho_asterisk = rho_c; // ϱ*
  constant SI.Pressure p_asterisk = 22.064*MPa; // p*
  constant SI.DynamicViscosity mu_asterisk = 1E–06; // μ*
  // Dimensionless variables.
  Real bar_T = T/T_asterisk; // T̄ = T/T*
  Real bar_rho = rho/rho_asterisk; // ϱ̄ = ϱ/ϱ*
  Real bar_p = p/p_asterisk; // p̄ = p/p*
  Real bar_mu = mu/mu_asterisk; // μ̄ = μ/μ*
  Real bar_mu_0; // μ̄0(T̄)
  Real bar_mu_1; // μ̄1(T̄, ϱ̄)
  // Coefficients for Hi in μ̄0(T̄).
  Real H_0[4] = {1.67752, 2.20462, 0.6366564, –0.241605};
  // Coefficients for Hij in μ̄1(T̄, ϱ̄) (see [11, Table 2, p.5]).
  Real H_1[6,7] = [5.20094E–01,..., 0;
                   ...
                   0,..., –5.93264E–04];
  constant Real bar_mu_2 = 1; // μ̄2(T̄, ϱ̄)
  // IAPWS R15-11 (2011) - THERMAL CONDUCTIVITY
  // Reference constants.
  SI.ThermalConductivity lambda_asterisk = 1E–03; // λ*
  // Dimensionless variables.
  Real bar_lambda = lambda/lambda_asterisk; // λ̄ = λ/λ*
  Real bar_lambda_0; // λ̄0(T̄)
  Real bar_lambda_1; // λ̄1(T̄, ϱ̄)
  // Coefficients for Li in λ̄0(T̄).
  Real L_0[5] = {2.443221E–03, 1.323095E–02, 6.770357E–03,
    –3.454586E–03, 4.096266E–04};
  // Coefficients for Lij in λ̄1(T̄, ϱ̄) (see [12, Table 2, p.6]).
  Real L_1[5,6] = [1.60397357,..., 0.00609859258;
                   ...
                   –2.7203370,..., 0.012913842];
  constant Real bar_lambda_2 = 0; // λ̄2(T̄, ϱ̄)
equation
  // Correlating equation for VISCOSITY μ̄(T̄, ϱ̄), see [12, Eq.10].
  bar_mu = bar_mu_0 * bar_mu_1 * bar_mu_2;
  // μ̄o(T̄), viscosity in the dilute-gas limit, see [12, Eq.11].
  bar_mu_0 = 100*sqrt(bar_T)/sum(H_0[i+1]/bar_T^i for i in 0:3);
  // μ̄1(T̄, ϱ̄), contribution to viscosity due to finite density, [12, Eq.12].
  bar_mu_1 = exp(bar_rho * sum((1/bar_T–1)^i * sum(H_1[i+1,j+1]*(
    bar_rho–1)^j for j in 0:6) for i in 0:5));
  // Correlating equation for THERMAL CONDUCTIVITY λ̄(T̄, ϱ̄),
  // see [11, Eq.15].
  bar_lambda = bar_lambda_0 * bar_lambda_1 + bar_lambda_2;
  // λ̄o(T̄), thermal conductivity in the dilute-gas limit, see [11, Eq.16].
  bar_lambda_0 = sqrt(bar_T)/sum(L_0[i+1]/bar_T^i for i in 0:4);
  // λ̄1(T̄, ϱ̄), contribution to λ̄ due to finite density, see [11, Eq.17].
  bar_lambda_1 = exp(bar_rho * sum((1/bar_T–1)^i * sum(L_1[i+1,j+1]*(
    bar_rho–1)^j for j in 0:5) for i in 0:4));
  assert(not(T > 645.91 and T < 650.77 and rho > 245.8 and rho <
    405.5),
    "WARNING (Thermofluids.Substances.WaterIAPWS95):\n
    (T,rho)=("+String(T)+","+String(rho)+") near the critical point.\n"
    +"See IAPWS R12–08 (2008), Sections 2.7 and 2.8",
    AssertionLevel.warning);
end WaterIAPWS95;
```

Listing 4: Thermofluids.Storage.InternalEnergy.

```
class InternalEnergy "Thermofluid internal energy storage element"
  extends Thermodynamics.Storage.InternalEnergy(
    redeclare replaceable
      Thermofluids.Substances.Partial.SingleSubstance substance
    constrainedby
      Thermofluids.Substances.Partial.SingleSubstance);
end InternalEnergy;
```

Listing 5: Thermofluids.Storage.KineticEnergy.

```
class KineticEnergy "Kinetic energy storage element"
  SI.Momentum P "State variable";
  SI.Mass m; // m = ϱV is calculated in the control volume.
  SI.KineticEnergy K = P^2/(2*m) "Kinetic energy"; // K(P) = 1/2 P^2/m
  extends pHS.Storage.C (redeclare KineticPort port, x0={0});
equation // State equations
  x = {P}; // x = [P] ∈ 𝒳 = ℝ.
  dH = {P/m}; // dH(x) = ∂K/∂P = [P/m] ∈ T*𝒳.
end KineticEnergy;
```

E. Rigid Two-Port Control Volumes

Ducts are essential components for transporting substances in a thermofluid system. For example, in a parabolic trough solar thermal power plant, the absorber tubes are the elements through which the HTF flows and where the electromagnetic energy from solar radiation is transformed into the internal energy of the fluid. Pipelines are the conduits used in these installations and can be modeled as control volumes with two ports called port 1 (or inlet port) and port 2 (or outlet port).

Fig. 3 shows the components and bonds of the partial class TwoPortsKU that models a CV with a fixed boundary and, therefore, constant volume.

TwoPortsKU implements the integral balance (8)–(13) and (14)–(18), and consists of the following elements.

1) The storage elements U1 and K1 for internal and kinetic energies, respectively.
2) The four-port slicers slc1 and slc2 that separate the 4-D thermofluid port_1 and port_2 into four 1-D ports (thermal port #1, pressure port #2, mass port #3, and kinetic port #4).
3) The 000-junction j1 used to connect internal energy storage elements.
4) The 0-junctions j6, j8, and j12, and the gyrators gY1 and gY2 that take into account the terms $\sum_{i\in 1\ldots N}\dot{\mathbf{P}}_{\text{press},i}$ [see (11)] and $\dot{\mathbf{P}}_{\text{grav}}$ [see (12)]. For a two-port CV

$$-(\dot{\mathbf{P}}_{\text{press},1}+\dot{\mathbf{P}}_{\text{press},2}) = -(p_1\mathbf{A}_1+p_2\mathbf{A}_2) = (p_1A_1-p_2A_2)\mathbf{n}_{1\to 2}$$

where $\mathbf{n}_{1\to 2}$ is a unit vector in the flow direction $\text{port}_1 \to \text{port}_2$. We do not consider the term $\dot{\mathbf{P}}_{\text{press},0}=\int_{\partial\Omega_0} p\mathrm{d}\mathbf{A}$ in the model because it has a zero value in a CV with radial symmetry and equal pressure in each section of Ω. This is the case for an ideal pipe but it must be taken into account in elements where there is a change of flow direction, such as bends and T-junctions.

The term $\dot{\mathbf{P}}_{\text{grav}}$ is calculated by the modulated source p_grav, depends on the position of the CV, and will be evaluated when extending TwoPortsKU in a derived class.

5) The modulated mass flow rate generator mmfr1 that calculates $\dot{m}_2$. In ducts, such as pipes, $\dot{m}_2$ is calculated as a function of $\Delta p_{\text{equiv}} = p_1 - p_2 - \varrho L|\mathbf{g}|\sin\alpha$ [2, p. 50], and this value is available in the pressure sensor delta_p.
6) The modulated gyrator mGY1 that calculates the fluid outlet velocity $|\mathbf{v}_2| = \dot{m}_2/(\varrho A_2)$ as a function of the mass flow rate $\dot{m}_2$.
7) The dissipative element S_visc that calculates $\dot{S}_{\text{visc}}$ [see (7)] based on the mass flow rate $\dot{m}_2$.
8) The modulated thermal conductors S_cond_0 and S_cond_Omega that implement (15) $\dot{S}_{\text{cond},0}$ and (16) $\dot{S}_{\text{cond},\Omega}$, respectively. The conductance of both conductors depends on the CV geometry and will be defined when extending TwoPortsKU in a derived class. Port_3 is a thermal port for connecting an external heat source. The implementation of class MGS is shown in [19, Listing 15].
9) The modulated transformers mTF1 and mTF2 that calculate the convective part $\dot{S}_{\text{conv},2} = \dot{m}_2 s_2$ of the outlet entropy flow rate [see (14)].
10) 0-junctions, 1-junctions, and bonds for connecting the previous components.

F. Rigid Pipes

Pipes can be modeled as circular cross-section, two-port control volumes. We have implemented the Modelica code of class PipeTurbulentKU that extends class TwoPortsKU and models the behavior of a turbulent flow through a constant volume pipe. We implement the turbulent flow model because it is the most frequent in industrial thermofluid installations. In the case of solar thermal power plants, it is necessary to work in a turbulent regime to favor heat transfer to the HTF.

The parameters that define the geometry and position of a pipe are diameter D, length L, roughness ε of the surface $\partial\Omega_0$, and angle α of horizontal elevation. In each instance of the class PipeTurbulentKU, it is also necessary to specify the substance and the initial state (T_0, S_0, and ϱ_0) of the storage element U1.

For completing the model of a pipe, it is necessary to extend the partial class TwoPortsKU with equations for $\dot{m}_2$, f (friction factor), $\dot{S}_{\text{visc}}$, $\dot{S}_{\text{cond},0}$, $\dot{S}_{\text{cond},\Omega}$, and $\dot{P}_{\text{grav}}$. The last one depends on the pipe horizontal elevation α and produces the equivalent pressure

$$p_{\text{grav}} = -\varrho L|\mathbf{g}|\sin\alpha. \tag{20}$$

To calculate $\dot{m}_2$, we use the correlations for a steady-state turbulent flow in pipes (see [26, Sect. 4.5])

$$\dot{m}_2 = \varrho\sqrt{\frac{\pi^2 D^5|\Delta p_{\text{equiv}}|}{32\varrho L f}} \tag{21}$$

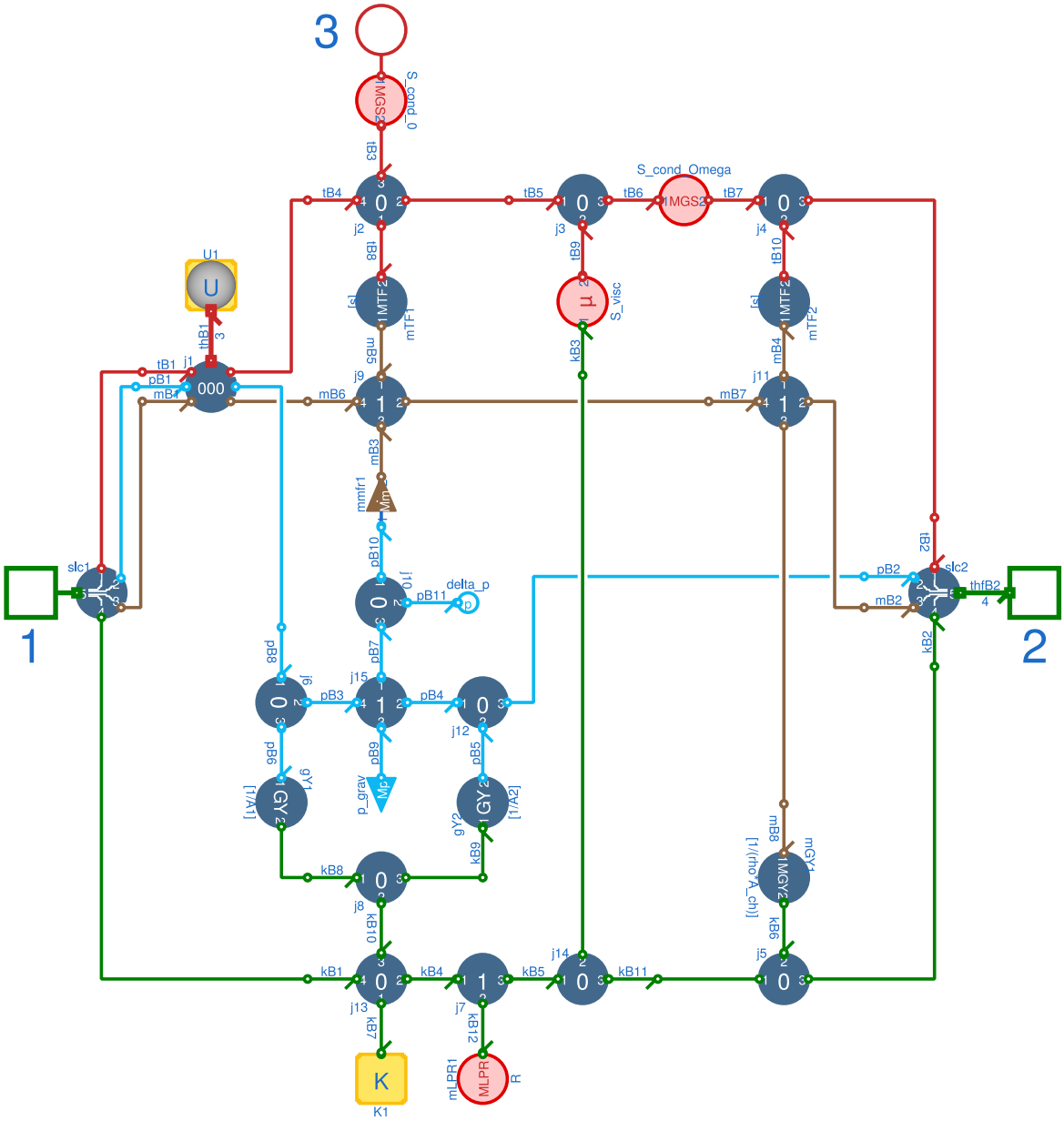


Fig. 3. Port-based model of a rigid two-port control volume. In the case of pipes with circular cross section, this model is associated with the pipe icon shown in Fig. 2. This helps users to build systems, such as the one represented in Fig. 5, using the drag-and-drop functionality of Modelica graphical editors.

and the friction factor f is computed from the Colebrook correlation [26, eq. (4.5-9)]

$$\frac{1}{\sqrt{f}} = -4\log\left(\frac{\varepsilon/D}{3.7065} + \frac{1.2612}{\mathrm{Re}\sqrt{f}}\right), \quad \mathrm{Re} = \frac{4\dot{m}_2}{\pi\mu D}. \tag{22}$$

S_visc is an RSmu dissipative component that extends the class pHS.Dissipative.RS (see [18, Remark 26]) and adds to the thermal domain the entropy flow rate $\dot{S}_{\mathrm{visc}}$ calculated as a function of $\dot{m}_2$ and the friction factor f (see [1, eq. (14)])

$$\dot{S}_{\mathrm{visc}} = \frac{32L\dot{m}_2^3 f}{\varrho^2 T D^5}. \tag{23}$$

In thermofluid systems, the temperature is a quantity with a very slow transient compared to pressure, fluid velocity, or mass flow rate. Therefore, assuming steady-state conditions for the mass flow rate is suitable for modeling the transient temperature response.

For calculating the conductance $G(\dot{S}_{\mathrm{cond},\Omega})$ of the component S_cond_Omega, we assume perfect mixing conditions in the fluid, which simplifies the evaluation of (16) with the following approximation:

$$\dot{S}_{\mathrm{cond},\Omega} = \int_\Omega \lambda \frac{|\nabla T|^2}{T^2}\mathrm{d}V$$

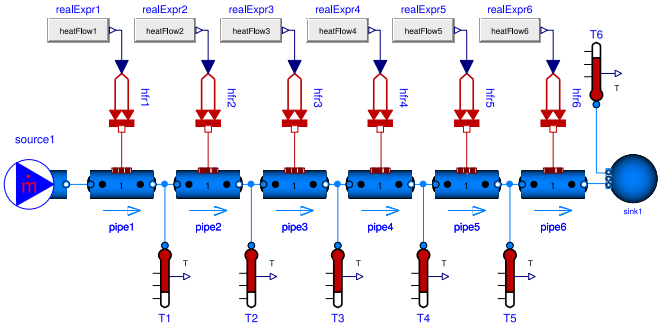


Fig. 4. MSL model of a system with one source of flow (source1), six heat sources (hfr1...6), six pipes (pipe1...6), and one sink (sink1).

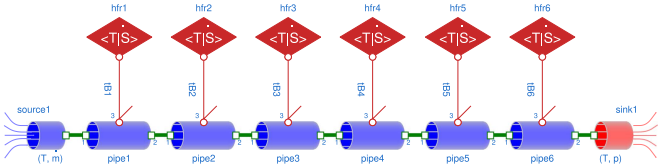


Fig. 5. Implementation with pHlib of the piping system previously modeled with MSL (see Fig. 4).

$$\approx \int_\Omega \lambda \frac{|(T_1 - T_2)/L|^2}{T_1 T_2} \mathrm{d}V = \frac{\lambda V}{L^2} \frac{(T_1 - T_2)^2}{T_1 T_2}$$

and comparing with [19, eq. (96)]

$$G(\dot{S}_{\mathrm{cond},\Omega}) = \frac{\lambda V}{L^2} = \frac{\lambda \pi D^2}{4L}. \tag{24}$$

To calculate the conductance $G(\dot{S}_{\mathrm{cond},0})$ of the component S_cond_0, we apply the Dittus–Boelter correlation [26, eq. (4.5-26)] to determine the heat transfer coefficient for turbulent flows in pipes

$$\Delta T_{\mathrm{ch}} = T_{01} - T_1 (see\,[2, eq.14.1 - 2])$$

$$\mathrm{Nu} = 0.023\,\mathrm{Re}^{4/5}\,\mathrm{Pr}^{0.4}, \quad \mathrm{Pr} = c_p \mu / \lambda$$

$$h = \frac{\lambda\,\mathrm{Nu}}{\ell_{\mathrm{ch}}} = \frac{\lambda\,\mathrm{Nu}}{D} \tag{25}$$

$$\dot{U}_{\mathrm{cond},0} = h A_0 \Delta T_{\mathrm{ch}} (see\,[2, eq.14.1 - 1]) \tag{26}$$

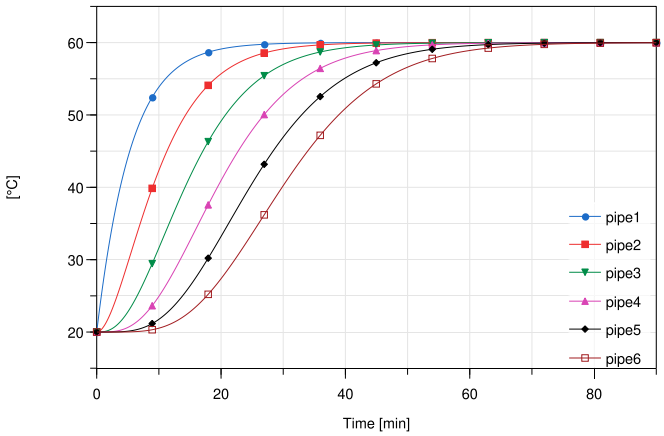
and comparing with [19, eq. (95)],

$$G(\dot{S}_{\mathrm{cond},0}) = h\,A_0. \tag{27}$$

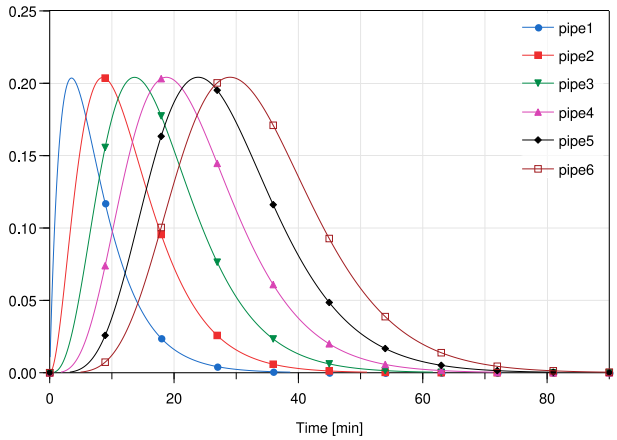
V. Examples

The examples in this section have been processed and simulated with Dymola running in a Windows 10 PC computer configured with a 3.40 GHz Intel Core i5-7500 CPU and 32 GB of memory.

Fig. 4 shows the MSL model of a system with one source of flow (source1), six heat sources (hfr1...6), six pipes (pipe1...6), and a sink (sink1). Fig. 5 shows the same piping system modeled with pHlib. The system is parameterized to perform simulations where we can define the following:



(a)



(b)

Fig. 6. Evolution of (a) water temperature and (b) relative error in pipe1...6 during a one-and-a-half-hour simulation for the pHlib model of Fig. 5. Source1 injects a flow of 1 kg/s at 60 °C into pipe1, and hfr1...6 do not inject heat.

1) pipes geometry, position, and flowing substance;
2) source mass flow rate and temperature;
3) sink temperature and pressure;
4) heat flow rate of the heat sources.

Fig. 6 shows the temperature evolution of water in the aforementioned six-pipe test system during one-and-a-half hour simulation of the MSL and the pHlib models. The simulation parameters are as follows:

1) source1 injects a flow of 1 kg/s of water at 60 °C into pipe1;
2) pipe1...6 are ideal pipes ($\varepsilon = 0$) in horizontal position ($\alpha = 0$), 20 cm diameter, 10 m long, and initially filled of water at 20 °C;
3) sink1 is at 20 °C, 1 bar pressure;
4) the sources hfr1...6 do not inject heat into the pipes;
5) water properties have been modeled with IAPWS-IF97 in the MSL model and IAPWS-95 in the pHlib model.

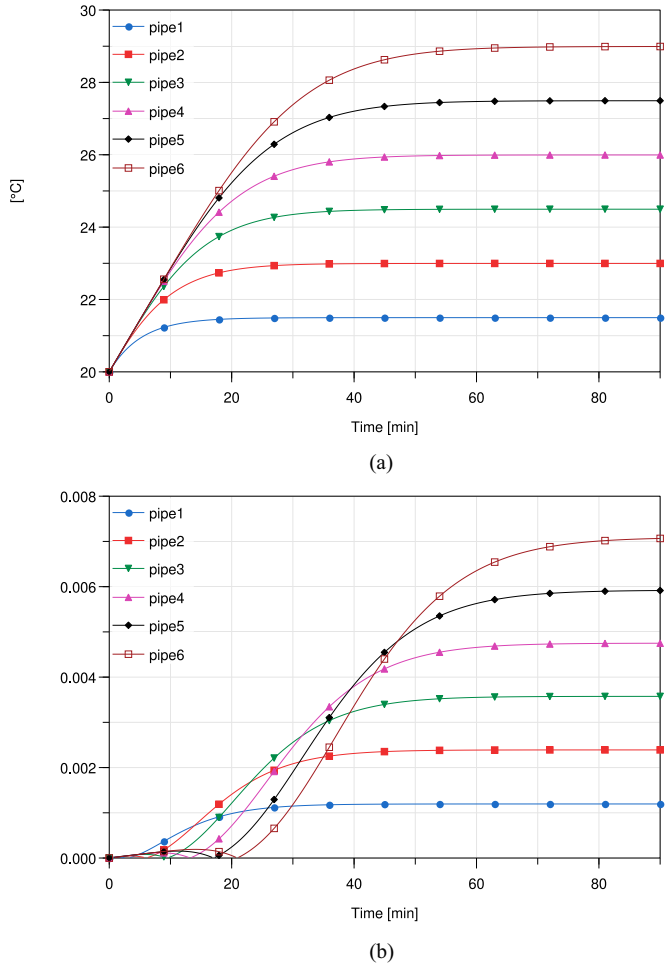


Fig. 7. Evolution of (a) water temperature and (b) relative error in pipe1...6 during a one-and-a-half-hour simulation for the pHlib model of Fig. 5. Source1 injects a flow of 1 kg/s at 20 °C into pipe1, and hfr1...6 inject a 1000 W/m^2 heat flux each one.

Fig. 7 shows the results of the simulation with the following changes in the boundary conditions:

1) source1 injects the same flow but at 20 °C;
2) hfr1...6 inject a 1000 W/m^2 heat flux into pipe1...6.

As we can see, both models show qualitatively similar behavior. For quantitative comparisons, we have taken MSL as a reference and calculated the relative errors of the pHlib model with the expression

$$\text{error}_{\%}(T_{\text{pHlib}}) = \frac{|T_{\text{MSL}} - T_{\text{pHlib}}|}{|T_{\text{MSL}}|} \times 100. \tag{28}$$

Table III summarizes the results obtained when measuring the simulation times for models developed using both libraries across two scenarios while varying the number N of pipes in the system. In the first scenario, where water is injected at 60 °C without any applied heat flux, the average simulation time for models built with pHlib is approximately 3.12 times higher than that of the equivalent MSL-based models. In the second scenario, involving water injection at 20 °C along with a heat flux of 1000 W/m^2, the computational overhead decreases, with pHlib-based models taking on average 1.62 times longer than those using MSL. The observed performance difference can be attributed to the fact that pHlib relies on the IAPWS-95 formulation for the fundamental equation of water, whereas MSL employs the simpler IAPWS-IF97 standard. While IAPWS-95 is computationally more demanding, it offers a significant modeling advantage: it provides a single, continuous, and differentiable equation valid over the entire range of thermodynamic states. This contrasts with IAPWS-IF97, which is segmented into five regions and may experience chattering phenomena, especially in systems with dynamic region changes.

TABLE III
EXECUTION TIMES (IN SECONDS) COMPARING PHLIB WITH MLS

	TSF†=60°C, HF‡=0 W/m^2			TSF=20°C, HF=1000 W/m^2		
N¶	t_{pHlib}	t_{MSL}	$t_{\text{pHlib}}/t_{\text{MSL}}$	t_{pHlib}	t_{MSL}	$t_{\text{pHlib}}/t_{\text{MSL}}$
1	0.24	0.10	2.40	0.13	0.09	1.44
2	0.52	0.16	3.25	0.27	0.16	1.69
3	0.82	0.26	3.15	0.42	0.25	1.68
4	1.15	0.36	3.19	0.57	0.34	1.67
5	1.55	0.46	3.37	0.74	0.43	1.72
6	1.85	0.55	3.36	0.84	0.53	1.58

†Temperature of the source of flow. ‡Heat flux. ¶Number of pipes.

VI. CONCLUSION AND FUTURE WORKS

In this work, we have developed components that model the transport properties of simple substances. These properties, together with their thermodynamic properties, allow us to build port-based models of thermofluid systems for industrial applications. In a particular case, we have modeled the behavior of water as a heat transfer fluid and used it in a piping system operating in a turbulent regime.

When comparing pHlib models (using the IAPWS-95 implementation for water) with MLS models (using the IAPWS-IF97 implementation), we have seen that they exhibit the same qualitative behavior and that the relative errors depend on the dominant excitation. For example, when injecting a water mass flow at 60 °C in a six-pipe system, transient temperature errors are less than 0.2%, and steady-state errors tend to zero, while when injecting a heat flux of 1000 W/m^2, transient errors are almost zero, and steady-state errors do not exceed 0.008% . With these results and accounting for the average execution time ratio of both simulations ($t_{\text{pHlib}}/t_{\text{MSL}} \approx 3.12$ for the first scenario and $t_{\text{pHlib}}/t_{\text{MSL}} \approx 1.62$ for the second one), we can conclude that our models with IAPWS-95 can be used in real-time applications for control or implementation of digital twins. Regarding model complexity, we note that both implementations are comparable in terms of the number of state variables in the computational models generated by Dymola. The results obtained encourage us to continue working on the library enhancement to add new phenomena to thermofluid systems modeling (e.g., radiation effects, external convection, bend resistance, and entropy generation in T-junctions) and thus undertake the modeling of complex systems.

APPENDIX
LIST OF SYMBOLS

Symbol	Quantity	SI unit
c_p	Isobaric heat capacity	J/(kg·K)
$\langle e \vert f \rangle$	Dual product $\langle$effort$\vert$flow$\rangle$	W
E	Total energy	J
f	Friction factor	1
$\mathbf{g}$	gravity acceleration	m/s^2
h	Heat transfer coefficient	W/(m^2·K)
$\mathbf{I}$	Identity tensor	1
$\mathbf{j}_q$	Heat flux	W/m^2
K	Kinetic energy	J
m	Mass	kg
Nu	Nusselt number	1
$\mathbf{P}$	Momentum	N·m
p	Pressure	Pa
Pr	Prandtl number	1
q_r	Long-range process power density	W/m^3
Re	Reynolds number	1
S	Entropy	J/K
s	Specific entropy	J/(kg·K)
T	Temperature	K
ΔT_{ch}	Characteristic temperature difference	K
$T\mathcal{X}$	Tangent bundle of $\mathcal{X}$	—
$T^*\mathcal{X}$	Cotangent bundle of $\mathcal{X}$	—
U	Internal energy	J
$\mathbf{v}$	Velocity	m/s
V	Volume	m^3
$\mathcal{X}$	Smooth manifold $\mathcal{X}$	—
α	Horizontal elevation angle	1
ε	Wall roughness	m
κ	Dilatational viscosity	Pa·s
λ	thermal conductivity	W/(m·K)
μ	viscosity	Pa·s
ϱ	Density	kg/m^3
$\boldsymbol{\tau}$	Viscous stress (tensor)	Pa

REFERENCES

[1] A. Bejan, "A study of entropy generation in fundamental convective heat transfer," *J. Heat Transfer*, vol. 101, no. 4, pp. 718–725, 1979.

[2] R. B. Bird, W. E. Stewart, and E. N. Lightfoot, *Transport Phenomena*. New York, NY, USA: Wiley, 2007.

[3] W. Borutzky, *Bond Graph Methodology–Development and Analysis of Multidisciplinary Dynamic System Models*. Berlin, Germany: Springer-Verlag, 2010.

[4] H. B. Callen, *Thermodynamics and an Introduction to Thermostatistics*. Hoboken, NJ, USA: Wiley, 1985.

[5] F. Casella and A. Leva, "Modelling of thermo-hydraulic power generation processes using modelica," *Math. Comput. Modelling Dynamical Syst.*, vol. 12, no. 1, pp. 19–33, 2006.

[6] F. E. Cellier and A. Nebot, "The Modelica Bond Graph Library," in *Proc. 4th Int. Modelica Conf.*, Tech. Univ. Hamburg, Harburg, Germany, 2005, pp. 57–65.

[7] S. R. de Groot and P. Mazur, *Non-Equilibrium Thermodynamics*. Amsterdam, The Netherlands: North-Holland Pub. Co., 1962.

[8] V. Duindam, A. Macchelli, S. Stramigioli, and H. Bruyninckx, *Modeling and Control of Complex Physical Systems. The Port Hamiltonian Approach*. Berlin, Germany: Springer, 2009.

[9] M. E. Gurtin, *An Introduction to Continuum Mechanics, volume 158 of Mathematics in Science and Engineering*. San Diego, CA, USA: Academic Press, 1981.

[10] *IAPWS*, "Revised release on the IAPWS industrial formulation 1997 for the thermodynamic properties of water and steam," Tech. Rep. IAPWS R7-97 (2012), International Association for the Properties of Water and Steam, Lucerne, Switzerland, 2007.

[11] *IAPWS*, "Release on the IAPWS formulation 2008 for the viscosity of ordinary water substance," Tech. Rep. IAPWS R12-08, Int. Assoc. Properties Water Steam, Berlin, Germany, 2008.

[12] *IAPWS*, "Release on the IAPWS formulation 2011 for the thermal conductivity of ordinary water substance," Tech. Rep. IAPWS R15-11, Int. Assoc. Properties Water Steam, Plzeň, Czech Republic, 2011.

[13] *IAPWS*, "Revised release on the IAPWS formulation 1995 for the thermodynamic properties of ordinary water substance for general and scientific use," Tech. Rep. IAPWS R6-95, Int. Assoc. Properties Water Steam, Prague, Czech Republic, 2018.

[14] D. Karnopp, D. Margolis, and R. Rosenberg, *System Dynamics: Modeling, Simulation, and Control of Mechatronic Systems*. Hoboken, NJ, USA: Wiley, 2012.

[15] L. D. Landau and E. M. Lifshitz, *Fluid Mechanics, Volume 6 of Course of Theoretical Physics*. Oxford, U.K.: Pergamon, 1987.

[16] M. Lohmayer, P. Kotyczka, and S. Leyendecker, "Exergetic port-Hamiltonian systems: Modelling basics," *Math. Comput. Modelling Dynamical Syst.*, vol. 27, no. 1, pp. 489–521, 2021.

[17] M. Lohmayer and S. Leyendecker, "Exergetic port-Hamiltonian systems: Navier-stokes-Fourier fluid," *IFAC-PapersOnLine*, vol. 55, no. 18, pp. 74–80, 2022.

[18] F. M. Márquez, P. J. Zufiria, and L. J. Yebra, "Port-Hamiltonian modeling of multiphysics systems and object-oriented implementation with the modelica language," *IEEE Access*, vol. 8, pp. 105980–105996, 2020.

[19] F. M. Márquez, P. J. Zufiria, and L. J. Yebra, "Port-Hamiltonian modeling of thermofluid systems and object-oriented implementation with modelica I: Thermodynamic part," *IEEE Access*, vol. 9, pp. 131496–131519, 2021.

[20] H. Olsson, *Modelica- A Unified Object-Oriented Language for Systems Modeling. Language Specification. Version 3.6*. Modelica Association, Linköping, Sweden, 2023. [Online]. Available: https://specification.modelica.org/maint/3.6/MLS.pdf

[21] G. Oster, A. Perelson, and A. Katchalsky, "Network thermodynamics," *Nature*, vol. 234, no. 5329, pp. 393–399, 1971.

[22] H. C. Öttinger, *Beyond Equilibrium Thermodynamics*. Hoboken, NJ, USA: Wiley, 2005.

[23] H. M. Paynter, *Analysis and Design of Engineering Systems*. Cambridge, MA, USA: MIT Press, 1961.

[24] A. S. Perelson, "Network thermodynamics. An overwiew," *Biophysical J.*, vol. 15, pp. 667–685, 1975.

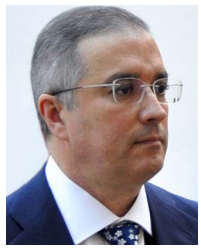
[25] J. C. Slattery, *Advanced Transport Phenomena*. New York, NY, USA: Cambridge Univ. Press, 1999.

[26] I. Tosun, *Modeling in Transport Phenomena: A Conceptual Approach*. Amsterdam, The Netherlands: Elsevier, 2007.

[27] A. van der Schaft and D. Jeltsema, "Port-Hamiltonian systems theory: An introductory overview," *Found. Trends Syst. Control*, vol. 1, no. 2/3, pp. 173–378, 2014.

[28] A. van der Schaft and B. Maschke, "The Hamiltonian formulation of energy conserving physical systems with external ports," *AEÜ - Archiv für Elektronik und Übertragungstechnik*, vol. 49, no. 5/6, pp. 362–371, 1995.

[29] D. Zimmer and F. E. Cellier, "The Modelica Multi-Bond Graph Library," in *Proc. 5th Int. Modelica Conf.*, Vienna, Austria, 2006, pp. 559–568.



Francisco M. Márquez received the B.Sc. degree in electronics engineering from the Universidad de Alcalá, Alcalá de Henares, Spain, in 1990, and the first M.Sc. degree in telecommunication engineering, the second M.Sc. degree in statistical and computational information processing, and the Ph.D. degree in mathematical engineering from the Universidad Politécnica de Madrid, Madrid, Spain, in 2010, 2011, and 2022, respectively.

From 1989 to 1991, he was the Head of a laboratory with INDRA, Madrid. Since 1991, he has been with the Departamento de Automática, Universidad de Alcalá, where he is currently an Associate Professor. His research interests include system modeling, industrial automation, and digital twins technologies.

Pedro J. Zufiria received the B.Sc. and M.Sc. degrees in telecommunication engineering from the Universidad Politécnica de Madrid (UPM), Madrid, Spain, in 1986, the first M.Sc. degree in mechanical engineering, the second M.Sc. degree in electrical engineering, and the Ph.D. degree in mechanical engineering from the University of Southern California, Los Angeles, CA, USA, in 1989, the Doctor Ingeniero de Telecomunicación degree in telecommunications engineering from the Ministry of Education and Science, Madrid, in 1991, and the Licenciado en Ciencias Matemáticas degree in mathematics from the Universidad Complutense de Madrid, Madrid, in 1997.

He is a Full Professor with the Telecommunications Engineering School, UPM, where he was the Chairman with the Department of Applied Mathematics for ICT, responsible for the Office of International Relations, Vice-Dean of Research and Graduate Studies, Codirector of the Orange and Cabify Chairs, and Founder of the Neural Networks and the Dynamical Systems, Learning, and Control Research Groups. He has authored over 100 publications on the analysis, control, and fault diagnosis of dynamical systems, the applications of statistics, machine learning paradigms, and complex network theory on system modeling and data processing.

Luis J. Yebra received the B.Sc. degree in telecommunications technical engineering from the Universidad de Alcalá, Alcalá de Henares, Spain, in 1993, the B.Sc. and M.Sc. degrees in physics from the Universidad Nacional de Educación a Distancia (UNED), Madrid, Spain, in 1997, and the Ph.D. degree in automatic control and industrial computing from the Department of Automatic Control and Computer Science, UNED, in 2006.

Since 1999, he has been with CIEMAT, Plataforma Solar de Almería Center. In 2007, he obtained the position of Tenured Scientist, working in dynamic modeling and automatic control in concentrated solar thermal plants. He has coauthored five books, more than 30 scientific articles published in journals (JCR), and more than 70 international conference papers. He has supervised five Ph.D. degree students, participated in more than 30 publicly funded research projects, and the creation of a spin-off company.

Dr. Yebra was the recipient of three prizes for the academic curriculum in UNED.