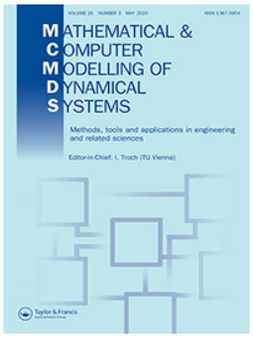

Taylor & Francis
Taylor & Francis Group

Mathematical and Computer Modelling of Dynamical Systems

Methods, Tools and Applications in Engineering and Related Sciences

ISSN: (Print) (Online) Journal homepage: https://www.tandfonline.com/loi/nmcm20

Robust object-oriented formulation of directed thermofluid stream networks

Dirk Zimmer

To cite this article: Dirk Zimmer (2020) Robust object-oriented formulation of directed thermofluid stream networks, Mathematical and Computer Modelling of Dynamical Systems, 26:3, 204-233, DOI: 10.1080/13873954.2020.1757726

To link to this article: https://doi.org/10.1080/13873954.2020.1757726

Published online: 16 Jun 2020.

MATHEMATICAL AND COMPUTER MODELLING OF DYNAMICAL SYSTEMS
2020, VOL. 26, NO. 3, 204–233
https://doi.org/10.1080/13873954.2020.1757726

Taylor & Francis
Taylor & Francis Group

ARTICLE

OPEN ACCESS Check for updates

Robust object-oriented formulation of directed thermofluid stream networks

Dirk Zimmer

Institute of System Dynamics and Control, German Aerospace Center (DLR), Oberpfaffenhofen, Germany

ABSTRACT

Object-oriented formulation of thermal fluid streams often yields large non-linear equation systems whose numerical solution is difficult to achieve. This paper revisits the fundamental equations for thermal fluid streams and introduces a new term: the steady mass flow pressure $\hat{p}$. Using this term, the equations can be brought into a form where all non-linear computations are explicit. This enables a robust and object-oriented formulation of even complex architectures. The modelling of aircraft environmental control systems is presented as one possible application example.

ARTICLE HISTORY
Received 19 July 2019
Accepted 15 April 2020

KEYWORDS
Thermal fluid systems; object-oriented modelling; robust computational methods

1. Motivation

Modelling and simulation of thermal fluid systems are in high demand for many engineering applications. Prominent challenges are the design and control of climate and cooling systems for aircraft [1] or ground vehicles [2]. Other applications are the process modelling and control of thermal power plants [3] and the simulation of building physics [4] for environment-friendly work places. Detailed simulations using Computational Fluid Dynamics (CFD) methods are often infeasible for an early design phase and hence other methods are used that impose a much lower computational burden and require a far less detailed modelling effort.

A methodology that is most prominently applied is the object-oriented modelling of fluid systems by means of differential-algebraic equation systems using languages like Modelica. For instance, all references for the application field above, refer to Modelica implementations and Figure 1 presents the Modelica model diagram for an air-cycle that is being part of an aircraft climate system. Modelica is an open and free modelling language [5] supported by various commercial and free tools. Also, there are free Modelica standard libraries supporting the common physical modelling base for different application fields: a Media library that supports various models of the fluid's thermodynamic properties [6] and a standard fluid library [7] with a common interface [8] for the modelling of fluid streams through various components between volume elements and system boundaries.

Yet despite all this support, there still remain reoccurring problems that pose major challenges for the end-user. Most of them involve the solvability of (larger) non-linear

CONTACT Dirk Zimmer Dirk.Zimmer@dlr.de Institute of System Dynamics and Control, German Aerospace Center (DLR), Oberpfaffenhofen, Germany

equation systems. Every so often, initialization or simulation of the fluid systems fails for reasons that are hard to detect for a non-specialist. From the end-user perspective, this is perceived as a lack of robustness significantly slowing down the development time of new fluid architectures.

It is important to understand how crucial robustness is and how costly a lack of it is. Our application field is the modelling and simulation of climate and cooling system for aircraft [1]. Whenever a new aircraft is designed, dozens of different architectures are being proposed and evaluated. For each architecture, a large set of failure cases need to be considered, combined with a large set of very different (often extreme) environmental conditions. Also, the parameters will be optimized by specialized algorithms and complete mission simulation is used for evaluation. In total, hundreds of thousands of simulations are performed on different architectures, configurations, and parameters. Ideally, none of these simulations should fail for numerical reasons but from the authors' experience, in practice, thousands of working hours are spent to manually fix occurring difficulties. Many other application fields share similar problems although these are typically little reported.

There is however a common root for this problem. These are large non-linear equation systems. The air cycle model of Figure 1 contains a non-linear equation system

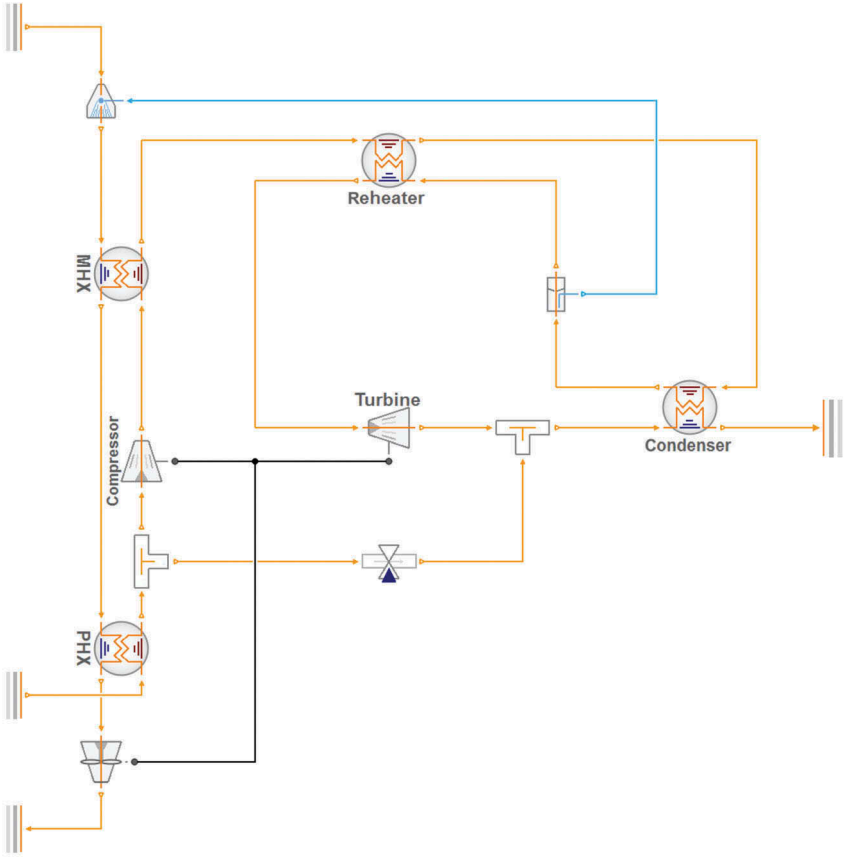

Figure 1. Modelica model diagram of a three-wheel bootstrap air cycle used in aircraft environmental control systems.

over more than 200 variables as reported in [9]. The Modelica models of ThermosysPro [3] may contain non-linear equation systems larger than 1000. Similar problems with non-linear equation systems are reported for building physics [10]. Difficult to solve non-linear equation systems are also reported for smaller hydraulic networks and heat exchangers in [11]. Although the simulation tools do their best to solve such systems, it is evident that a robust solution can hardly be expected. Hence, a significant effort has been undertaken to improve the situation by applying homotopy methods [12]. Although this led to some improvement, it ultimately was of limited success.

This paper presents a new approach to achieve a robust formulation for thermal fluid systems in an object-oriented environment. We achieve this targets by two means: first, we avoid the creation of any large non-linear equation system and second, we generate a set of DAEs that is robustly solvable by stiff-system ODE solvers after a structural transformation.

The contribution of this paper is thus of purely structural nature. We do not stipulate how to model individual components and media models or what parameters or level of detail to use. Instead, we present a structural format and when each component is modelled in this format, the modeller will know a priori that all possible systems composed under certain rules out of these components are solvable. This is the desired level of robustness.

In Section 2, we revisit the formulation of the underlying physical laws and derive a useful decomposition for the pressure potential. In Section 3, we then use this decomposition to define a structural format for thermofluid models and examine the resulting equation system structure. Corresponding practical implementations and use cases are presented in Section 4. Potential errors and lismitations are discussed in Section 5 whereas Section 6 discusses open points and concludes this paper with a positioning of the presented approach. Readers who are primarily interested in a comparison of this method with respect to other methods may hence start at Section 6.

A concluding remark for this section: this work mostly builds on and refers to the Modelica standard. This is because Modelica is an openly accessible standard with an active research community. However, there are many other tools (often described as 1D tools, since the fluid flows are regarded as one-dimensional) that may also profit from this work. We have hence formulated most content in general terms and confined the specifics of Modelica to Section 4.

2. Fundamental equations

What leads to the creation of large non-linear equation systems in fluid networks? Whereas a smaller non-linear equation system may occur within a component (such as a heat exchanger), larger non-linear systems are created by a network of such components that may contain branches, by-passes, and loops. Whenever fluid flows join, a (quasi-) static analysis will require an equivalence of pressure for each involved junction. In order to fulfil this equivalence, the corresponding mass flow rates become part of a non-linear equation system.

Figure 2 provides a simple academic example. A given mass flow $\dot{m}_0$ is split at (A) into $\dot{m}_1$ and $\dot{m}_2$. Both parts of the branch flow through black box components that arbitrarily manipulate the thermodynamic state before they rejoin at (B) and flow through another

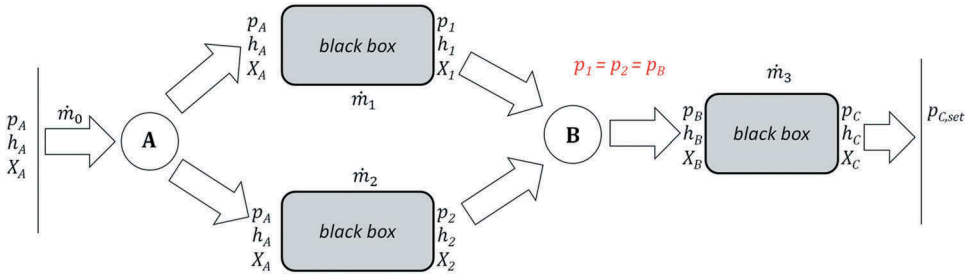

Figure 2. Splitting the mass flow rate in order to achieve equivalence of pressure at junction (B) may give rise to a non-linear equation system.

black box component to reach the sink. The thermodynamic state is in this example described by the pressure p, the specific enthalpy h and the mass-fraction content X. Junction (B) requires an equivalence of pressure: $p_1 = p_2$. In general, these two pressures will be non-linearly dependent on $\dot{m}_1$ or $\dot{m}_2$, respectively. This means that in order to maintain the pressure equivalence, a non-linear equation system has to be solved that computes the corresponding split of the mass flow rate. There might also be none or multiple solutions to this system.

For actual industrial examples, object-oriented formulations typically tend to bloat the size of the equation system and more complex examples create larger equation systems. It is then a priori unclear how many solutions exist and whether a generic non-linear equation solver will find any of them (and which one), especially at (re-)initialization.

2.1. Inertial pressure in fluid streams and steady mass flow pressure

In order to increase robustness, we shall hence not rely on a generic solver but rather provide differential equations that lead to the desired equivalence. Fortunately, the laws of physics offer a favourable way to formulate this. To this end, let us review the one-dimensional Euler equation for a stream [13] (or [14,15], respectively) with velocity v and density ρ through a pipe section as illustrated in Figure 3:

$$\rho \frac{\partial v_s}{\partial t} + \rho v_s \frac{\partial v_s}{\partial s} = -\frac{\partial p}{\partial s} - \frac{\partial p_{ext}}{\partial s} \tag{1}$$

where p_{ext} represents an additional term of pressure acting along the streamline s. It may be used to formulate gravity, friction, or other terms. Integrating along the streamline leads to:

$$\int_{s_1}^{s_2} \rho \frac{\partial v_s}{\partial t} ds + \rho \bar{v} \Delta v = -\Delta p - \Delta p_{ext} \tag{2}$$

where Δ expresses the difference between the endpoint of integration s_1 and s_2. The variable $\bar{v}$ expresses the mean velocity. The second term expresses the difference in dynamic pressure and is typically denoted by $\Delta q = \rho \bar{v} \Delta v$. For the first term, we can express v_s in terms of the mass flow rate $\dot{m}$:

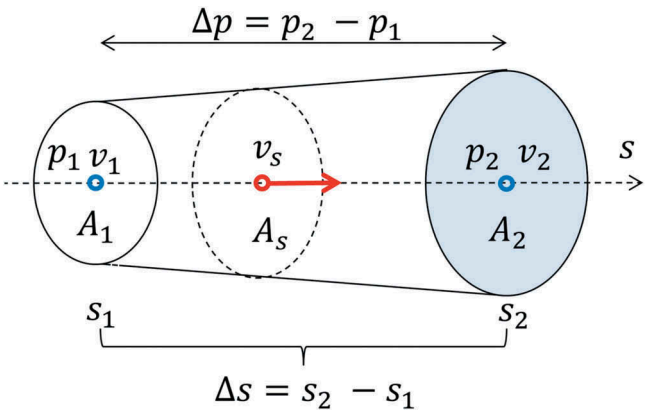

Figure 3. One-dimensional fluid stream along a pipe section of length *Δs*.

$$v_s = \frac{\dot{m}}{\rho A_s} \tag{3}$$

If we assume that each section of the streamline upholds the conservation of mass, the mass flow rate is invariant with respect to s and can be extracted from the integral in the first term:

$$\frac{d\dot{m}}{dt}\int_{s_1}^{s_2}\frac{1}{A_s}ds + \Delta q = -\Delta p - \Delta p_{ext} \tag{4}$$

The remaining integral term is denoted as *inertance* L [14,15]. In analogy to the mechanical inertia, it represents the reluctance of the mass flow rate to change. For a straight pipe with constant cross-section, the inertance simplifies to $L = \Delta s/A$. In any case, the inertance will be a constant value for constant geometries of the stream. Hence, we denote the corresponding pressure term as (negative) *inertial pressure* r:

$$-\Delta r = -(r_{s_2} - r_{s_1}) = L\frac{d\dot{m}}{dt} \tag{5}$$

The Euler equation reads now as a simple balance of pressures:

$$-\Delta r + \Delta q = -\Delta p - \Delta p_{ext} \tag{6}$$

We shall now define an unusual term, not present on textbooks on the matter: the *steady mass flow pressure* $\hat{p}$. It is denoted in this way because $\hat{p} = p$ if $d\dot{m}/dt = 0$. It is simply defined as the complement to the inertial pressure r:

$$p = \hat{p} + r \tag{7}$$

Using this definition to substitute p in the pressure balance from above yields:

$$\Delta\hat{p} = -\Delta p_{ext} - \Delta q \tag{8}$$

The terms of the right-hand side may actually be dependent (amongst other quantities) on the mass flow rate and pressure. For instance, the dynamic pressure q is dependent on the density which depends on the pressure and the friction component in p_{ext} may

depend on viscosity which may in turn depend on the pressure. Let us make this dependence explicit:

$$\Delta\hat{p} = -\Delta p_{ext}(p, \dot{m}, ...) - \Delta q(p, \dot{m}, ...) \tag{9}$$

If we are willing to sacrifice some precision in the case of unsteady mass flows, we may instead use $\hat{p}$ in the place of p:

$$\Delta\hat{p} \cong -\Delta p_{ext}(\hat{p}, \dot{m}, ...) - \Delta q(\hat{p}, \dot{m}, ...) \tag{10}$$

This form is very useful since we can compute now the pressure and all other thermodynamic properties such as specific enthalpy and mass fraction for a system explicitly in downstream direction assuming steady mass flow. In Section 5, we analyse the impact on the validity of this approximation in detail but the reasoning for using $\hat{p}$ instead of p is that

- for gases, r is often very small,
- for liquids, the thermodynamic properties are often not very sensitive to p,
- many formulas for friction (or related phenomena) are formulated under the assumption of a steady flow in the first place.

When streamlines meet a boundary or each other, gradients in $\hat{p}$ will occur. In order to uphold equivalence of pressure p, we can account for these gradients by the inertial pressure r that goes along with a corresponding change in mass flow rate. Fortunately, the law for the inertial pressure is a simple linear differential equation. As for the inertance L, it is independent of the thermodynamic state of the fluid and time-invariant for systems with a fixed geometry.

This is the basic idea. We now have to go into detail and study how to set up the equations for a junction of two (or more) streamlines.

2.2. Ideal mixing of fluid streams

The decomposition

$$p = r + \hat{p} \tag{11}$$

enables us to formulate a mixing law in an explicit form in downstream direction for the thermodynamic state. This state is hereby based on the steady mass flow pressure $\hat{p}$, the corresponding specific enthalpy $\hat{h}$ and the mass fraction content X (a vector with $n-1$ positive entries for n substances and norm < 1). We thereby follow the convention that inflow rates are positive and outflow rates are negative while i is an index for inflowing streams. outflowing stream is denoted by the index mix. First, we formulate the conservation of mass (and mass fraction correspondingly):

$$\dot{m}_{mix} + \sum_i \dot{m}_i = 0 \tag{12}$$

$$\dot{m}_{mix} X_{mix} + \sum_i \dot{m}_i X_i = 0 \tag{13}$$

Now, let us formulate the conservation of energy under the assumption that the thermodynamic state can be approximated by steady-mass flow conditions. The flow of energy is expressed by $\dot{m}h$. Using the decomposition of pressure, we may also decompose h correspondingly into $h = \hat{u} + (\hat{p} + r)\hat{\rho}^{-1} = \hat{h} + r\hat{\rho}^{-1}$, with $\hat{u}$ being the specific internal energy and $\hat{\rho}^{-1}$ being the specific volume; both approximations based on $\hat{p}$ for the thermodynamic state of steady mass flow. The energy balance hence also splits into two equations: Equation 14 expresses the balance of convective energy transport (in terms of $\hat{h}$); Equation 15 expresses the balance of volume work $r\dot{V} = r\dot{m}\hat{\rho}^{-1}$ performed by the different fluid flows accelerating and decelerating each other.

$$\dot{m}_{mix}\hat{h}_{mix} + \sum_i \dot{m}_i\hat{h}_i = 0 \tag{14}$$

$$r_{mix}\dot{V}_{mix} + \sum_i r_i\dot{V}_i = 0 \tag{15}$$

Figure 4 illustrates the mixing of two fluid streams into one and it displays that the three corresponding steady mass flow pressures $\hat{p}_1$, $\hat{p}_2$, and $\hat{p}_{mix}$ all differ across the junction. To ensure pressure equivalence, these differences are compensated by the inertial pressure that accelerates or decelerates the corresponding fluid streams. This is illustrated by the moving discs in Figure 4 and expressed by the formulation of pressure balance:

$$\forall_i : \hat{p}_i + r_i = \hat{p}_{mix} + r_{mix} \tag{16}$$

If we base the thermodynamic state on the steady mass flow pressure $(\hat{p}, \hat{h}, X)$ and assume it to be known for all inflowing streams, we can now compute the state of the outflowing mixture. X_{mix} results directly out of the conservation of mass (Equation 13). $\hat{h}_{mix}$ results directly out of the conservation of enthalpy (Equation 14). $\hat{p}_{mix}$ must then be determined such that Equations (15) and (16) are upheld.

The pressure balance enables us to express r_i in terms of steady mass flow pressures and r_{mix}: $r_i = \hat{p}_{mix} + r_{mix} - \hat{p}_i$. Plugging this into Equation (15) yields:

$$r_{mix}\dot{V}_{mix} + \sum_i \dot{V}_i(\hat{p}_{mix} + r_{mix} - \hat{p}_i) = 0 \tag{17}$$

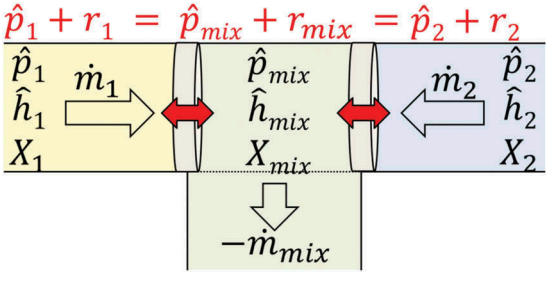

Figure 4. Illustration of the thought experiment for the mixing of two fluids.

which can be transformed to:

$$\hat{p}_{mix} \sum_i \dot{V}_i = \sum_i \hat{p}_i \dot{V}_i - r_{mix}(\dot{V}_{mix} + \sum_i \dot{V}_i) \tag{18}$$

In case the isenthalpic mixing of the fluid streams also conserves the volume ($\dot{V}_{mix} + \sum_i \dot{V}_i = 0$) we can explicitly compute $\hat{p}_{mix}$:

$$\hat{p}_{mix} = \frac{\sum \hat{p}_i \dot{V}_i}{\sum \dot{V}_i} \tag{19}$$

When the volume is not conserved in the mixing process (for instance because there occurs a phase transition), we need to relax Equation 18 to uphold this useful explicit formula. This means we accept an error when the mass flow of the mixture accelerates or decelerates. For steady mass flows, the error is zero anyway since r_{mix} will be zero in this case.

In case the fluids have almost constant density (such as liquids), the following approximation using mass flow rates instead of volume flow rates is helpful:

$$\hat{p}_{mix} \cong \frac{\sum \dot{m}_i \hat{p}_i}{\sum \dot{m}_i} \tag{20}$$

This approximation is also useful, when the difference in densities is small or one does not overly care about the correctness of the mass flow dynamics at transient behaviour.

3. Equation systems for unidirectional networks

3.1. Introductory example

Since we have found an explicit computational form for a mixing junction, the decomposition of p into $\hat{p}$ and r helps to restructure the overall equation system of fluid networks into a favourable form. This is best explained by means of an example. Figure 5 repeats the example of Figure 2 where split fluid streams rejoin. On each branch, there is a simple black box component that manipulates the thermodynamic state of the fluid by algebraic equations in an arbitrary way. For each section of each branch, the law for the inertial pressure applies as in Equation 5.

This law is applied to all thermofluid components, causing all mass-flows of the system to become potential state variables. This means that if the thermodynamic state of the inlet is defined, each component can compute the thermodynamic state of the outlet in a straightforward manner. All required variables represent knowns. This forward computation is symbolized by the bold-faced variables in Figure 5. All of them are computed simply from source to sink.

Given these variables, we can now compute the inertial pressure r and the mass-flow dynamics. We start at the boundaries. For each source, r is stipulated to be 0. At each sink, $\hat{p} + r$ equals the desired outlet pressure. The pressure balance law is applied for junction (A) and (B). The black box components contain the law for the inertial pressure.

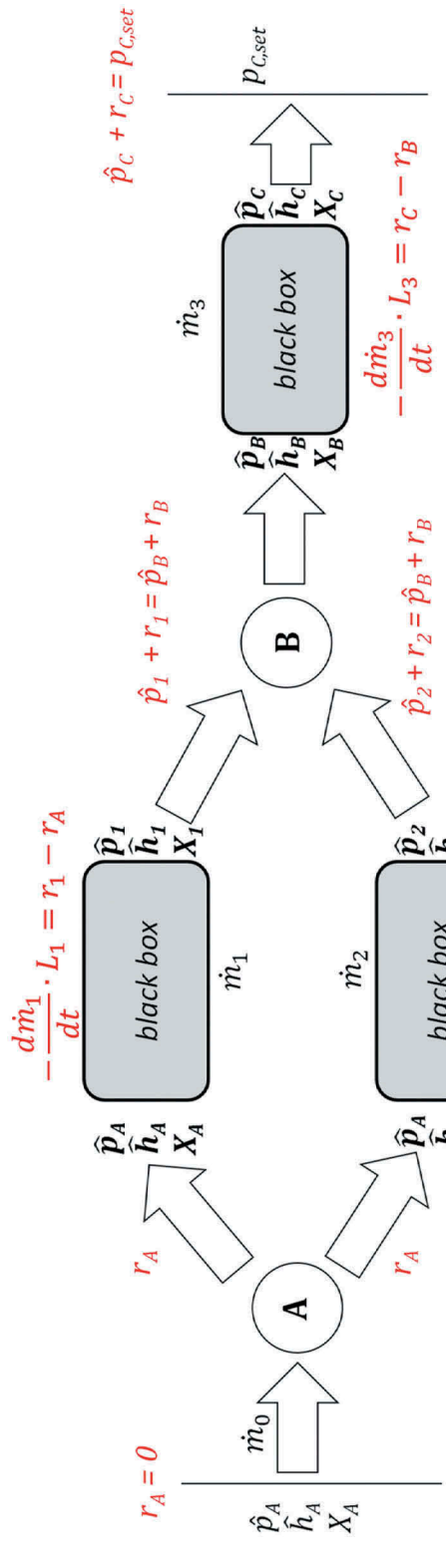

Figure 5. Computation of a simple directed fluid flow. The bold variables can be computed straightforward from source to sink. The equations next to the blocks describe the mass-flow dynamics and form a linear system of equations.

In total, we can setup the following equation system for the mass-flow dynamics:

$$-\frac{d\dot{m}_1}{dt}L_1 = r_1 \tag{21}$$

$$-\frac{d\dot{m}_2}{dt}L_2 = r_2 \tag{22}$$

$$-\frac{d\dot{m}_3}{dt}L_3 = r_C - r_B \tag{23}$$

$$\hat{p}_1 + r_1 = \hat{p}_B + r_B \tag{24}$$

$$\hat{p}_2 + r_2 = \hat{p}_B + r_B \tag{25}$$

$$\hat{p}_C + r_C = p_{C,set} \tag{26}$$

$$\frac{d\dot{m}_3}{dt} = \frac{d\dot{m}_1}{dt} + \frac{d\dot{m}_2}{dt} \tag{27}$$

Equation 27 thereby results from the application of the dummy-derivative method [16] in order to reduce the set of state variables: it represents the time-derivative of the mass-flow constraint $\dot{m}_1 + \dot{m}_2 = \dot{m}_3$. We can rewrite the equations above in matrix form:

$$\begin{bmatrix} -1 & & & & -L_1 & & \\ & -1 & & & & -L_2 & \\ & & 1 & -1 & & & -L_3 \\ 1 & & -1 & & & & \\ & 1 & -1 & & & & \\ & & & & -1 & -1 & 1 \end{bmatrix} \begin{bmatrix} r_1 \\ r_2 \\ r_B \\ r_C \\ d\dot{m}_1/dt \\ d\dot{m}_2/dt \\ d\dot{m}_3/dt \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ \hat{p}_B - \hat{p}_1 \\ \hat{p}_B - \hat{p}_2 \\ \hat{p}_{C,set} - \hat{p}_C \\ 0 \end{bmatrix} \tag{28}$$

The resulting equation system is not only linear. The matrix elements are either integers or describe the pipe geometry and hence likely form invariants with respect to simulation time. This means that the equation system can be inverted upfront and the evaluation of the corresponding variables is performed as simple matrix-vector multiplication during the simulation. When implementing such a form of equations in Modelica, a corresponding tool for the generation of simulation code like Dymola [17] will perform this automatically.

3.2. Informal explanation

Let us recapitulate the computations from above once more in order to better understand the arising computational scheme. Based on the assumption of a steady mass flow, we simply compute the thermodynamic state straightforward from source to sink for each component. We can do so because the thermodynamic state at the sources is given and all mass flows are known since they form state-variables of the system. Also, the mixing laws have been formulated in such a way that the thermodynamic state of the outlet is

a function of the inlets and the mass-flows. At each boundary, pressure differences in $\hat{p}$ may occur and are attributed to r. This also applies to mixing junctions. To compute the change of mass-flows that corresponds to the occurring inertial pressure r, a linear system of equations needs to be solved.

3.3. Formal explanation

Now, where we have studied one example and acquired an informal understanding, let us more formally analyse whether we can make general claims for the structure of the resulting equation system. To this end, we will define a structural format for all components and we will define how these components are connected to form a full thermal fluid system. Based on these definitions, we derive general statements on the structure of the resulting equation system. In the last part, we shall then show that the defined format is indeed applicable to model thermal fluid systems.

3.3.1. Definitions

We define θ as a vector of dimension d large enough to fully describe the thermodynamic state of the fluid it represents.

We then define a component $C = (V, E)$ as pair of a set of variables V and a set of equations E.

For a component with n inlets and o outlets, the variable set consists of

$$V_C := \left\{ \mathbf{\Theta}_{in} := [\theta_{in,1}, \ldots, \theta_{in,n}], \mathbf{\Theta}_{out} := [\theta_{out,1}, \ldots, \theta_{out,o}], \mathbf{u}, \mathbf{y}, \mathbf{x}, \dot{\mathbf{x}}, \dot{\mathbf{m}}, \ddot{\mathbf{m}}, \mathbf{r}, \mathbf{p}_r, r_C \right\} \tag{29}$$

where $\mathbf{\Theta}_{in}$ is an $n \times d$ matrix of thermodynamic inflow vectors and $\mathbf{\Theta}_{out}$ the $o \times d$ outflow counterpart. The vectors for mass-flows $\dot{\mathbf{m}}$, its time derivative $\ddot{\mathbf{m}}$, the inertial pressures $\mathbf{r}$ and the pressure set points $\mathbf{p}_r$ are all of dimension $n + o$. The variable r_C is an auxiliary variable and acts as a common point of reference for the inertial pressure. The signal input $\mathbf{u}$, the signal output $\mathbf{y}$ are of arbitrary dimension (often zero), $\mathbf{x}$ and its time derivative $\underline{\mathbf{x}}$ are vectors of arbitrary continuous-time states of arbitrary but equal dimension (also often zero). A component has thus $(d + 4)(n + o) + 2\dim(\mathbf{x}) + \dim(\mathbf{u}) + \dim(\mathbf{y}) + 1$ variables.

Figure 6 depicts such a component as a rectangular box and illustrates the sets of fluid inlets and outlets, signal inputs and outputs, and internal states. Naturally, degenerate forms of components with no inlets, no outlets, or no internal states are all covered by the above definition.

The corresponding set of equations is then defined by:

$$E_C = \left\{ \begin{array}{c} \mathbf{\Theta}_{out} = f_\Theta(\mathbf{\Theta}_{in}, \mathbf{x}, \dot{\mathbf{m}}, \mathbf{u}), \\ \dot{\mathbf{x}} = f_x(\mathbf{\Theta}_{in}, \mathbf{\Theta}_{out}, \mathbf{x}, \dot{\mathbf{m}}, \mathbf{u}), \\ \mathbf{y} = f_y(\mathbf{x}, \dot{\mathbf{m}}), \\ \mathbf{p_r} = f_p(\mathbf{\Theta}_{in}, \mathbf{\Theta}_{out}, \mathbf{x}, \dot{\mathbf{m}}, \mathbf{u}), \\ \mathbf{r} - \hat{\mathbf{p}}_r + L\ddot{\mathbf{m}} = r_C, \\ \sum \dot{m}_i = 0 \mid r_C = 0, \\ d\mathbf{x}/dt = \dot{\mathbf{x}}, \\ d\dot{\mathbf{m}}/dt = \ddot{\mathbf{m}} \end{array} \right\} \tag{30}$$

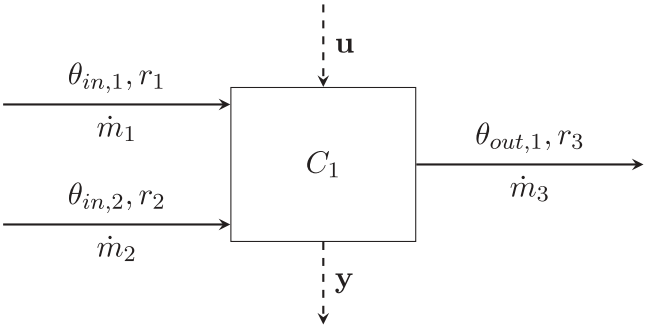

Figure 6. Interface representation of an exemplary component with its interface variables. The fluid flows are represented by solid lines and signal flows by dashed lines. The non-listed variables $\mathbf{x}, \dot{\mathbf{x}}, \dot{\mathbf{m}}, \mathbf{p}_r, r_C$ form internal variables of the component.

where f_Θ, f_x, f_y, and f_p are arbitrary non-linear functions and $\mathbf{L}$ is a linear diagonal matrix with the corresponding inertances. In case f_Θ is of the form: $f_\Theta(\mathbf{x}, \dot{\mathbf{m}})$ and $r_C = 0$ applies, we call C a loop-breaker component and mark this by a tilde, such as $\tilde{C}$. Loop breakers are needed to model cyclical fluid flows. Usually, they represent elements of a fixed volume or mass.

The sum of equations for the component is formed by a term for each element in order of appearance: $od + dim(\mathbf{x}) + dim(\mathbf{y}) + (n + o) + (n + o) + 1 + dim(\mathbf{x}) + (n + o)$. This means that $nd + (n + o) + dim(\mathbf{u})$ further equations are needed to match the number of variables of V_C. These equations will be added when we connect the component with other components.

We shall now define how components have to be connected. Figure 7 illustrates a fully connected system. As depicted, there are two possible connections: the connection of a fluid outlet to a fluid inlet and the connection of a signal output to a signal input.

The connection of the ith outlet of component A to the jth inlet of component B (or vice versa) yields the following equations:

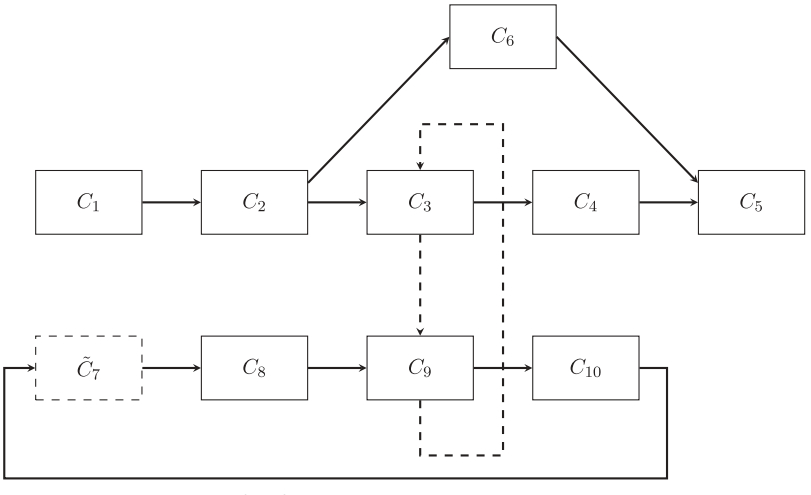

Figure 7. System composed out of different components. C_7 is marked by dashed lines since it meets the structural requirements for a loop-breaker.

$$\begin{aligned} \Theta_{in,j,B} &= \Theta_{out,i,A} \\ r_{j,B} &= r_{n+i,A} \\ \dot{m}_{j,B} + \dot{m}_{n+i,A} &= 0 \end{aligned} \tag{31}$$

The connection of the ith output of component A to the jth input of component B yields the following equation:

$$u_{j,B} = y_{i,A} \tag{32}$$

A system is fully connected if and only if all inlets are connected to an outlet and vice versa, and all inputs are connected to an output or are defined as a function of time or total system input. No inlet shall be connected to multiple outlets and vice versa, and no input variable shall be connected to more than one output.

In fully connected systems, the $2(o+n)$ connection equations for r and $\dot{m}$ are thus always shared by two components. Hence, in a fully connected system, the following number of equations are added for each component: $nd + (n+o) + dim(\mathbf{u})$. This means that each component now has the same number of variables and equations and consequently also the whole system. This is a prerequisite for a structurally regular equation system.

3.3.2. Derivation of the resulting structure

The connection equations are all linear equations. Each component contains a linear and a non-linear part. The non-linear part is expressed by the functions f_Θ, f_x, f_y, and f_p of Equation (30). What can we say about the computation of this non-linear part?

Since $\mathbf{x}$ and $\dot{\mathbf{m}}$ are state variables, we can assume them to be known. This means that the function $f_y()$ always can be computed straight-forward for all components since it only depends on known variables. Furthermore, since the inputs $\mathbf{u}$ consist of other component's outputs also all input variables can be assumed to be known. We can first determine $\mathbf{y}$ of all components and then $\mathbf{u}$.

A straightforward solution of $\mathbf{\Theta}_{out} = f_\Theta(\mathbf{\Theta}_{in}, \mathbf{x}, \dot{\mathbf{m}}, \mathbf{u})$ is hence possible if also $\mathbf{\Theta}_{in}$ is known. $\mathbf{\Theta}_{in}$ of a component will be determined by $\mathbf{\Theta}_{out}$ of another component. The question whether we can straight-forwardly compute $f_\Theta()$ of all components hence depends on the system topology.

We can represent the connections of components as directed graph. The components C form the vertices and the edges point from the outlets to the connected inlets. An exception are loop-breaker components $\tilde{C}$. These components form two disjoint vertices. One for all its inlets and one for all its outlets. When the resulting graph is acyclic, the directed graph can be topologically sorted and gives rise to a partial order of its vertices or components, respectively. We can then compute the functions $f_\Theta()$ in the partial order of their corresponding component.

Once $f_\Theta()$ of all components has been evaluated, we can also evaluate $f_x()$ and $f_p()$ of all components simply because these functions share the same set of inputs in addition to the output of f_Θ.

This means that the complete non-linear part of the equation system can be solved by a straightforward evaluation of all components, when the connection graph determined by the inlet to outlet connection of the components is acyclic and taking the special case of loop-breaker components into account. Or in other terms: the partial order of the

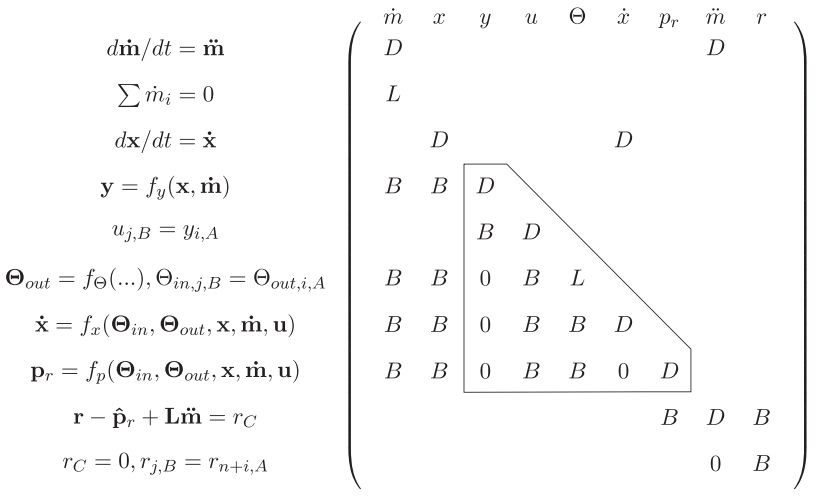

Figure 8. The structure incidence matrix permuted close to a Block-Lower-Triangular (BLT) form [19]. This matrix is expressed block-wise: D denotes a fully diagonal block, L an lower-triangular block with full diagonal, B a block with some non-zero entries, and 0 as well as empty space equally express a zero matrix block. The variables $\mathbf{y}$, $\mathbf{u}$, ... represent the corresponding variables of all components in the system. The equations in the marked lower-triangular part are potentially non-linear.

components determined by the connections from outlet to inlet enables to permute the structural incidence matrix [18] for the variables $\mathbf{y}, \mathbf{u}, \mathbf{\Theta}_{in}, \mathbf{\Theta}_{out}, \dot{\mathbf{x}}, \mathbf{p}_r$ into a lower-triagonal form with full diagonal. This is marked with a triangular-like shape in Figure 8.

The columns of a structure incidence matrix express the unknown variables of the system. We denote them here by their scalar representatives. The rows express the complete set of equations. We show their corresponding form by taking examples from E_C or the connection equations. The matrix element k, l of the structure incidence matrix is non-zero if the kth equation contains the lth variable.

The structure incidence reveals a structural singularity: the vector of mass flow rates $\dot{\mathbf{m}}$ is over-constrained and hence a part of the diagonal remain zero. The equations of the form $\sum \dot{m}_i = 0$ express a constraint between potential state variables. Let us resolve this singularity by the dummy derivative method [16] that has its origins from Pantelides [20]. To this end, we add the time derivatives of the constraint equations to the system $\sum \ddot{m}_i = 0$ and remove correspondingly the same number of differential equations of the form $d\dot{m}/dt = \ddot{m}$. After this removal, only a subset $\dot{\mathbf{m}}_S \subset \dot{\mathbf{m}}$ will represent states whereas all elements $\dot{\mathbf{m}} \backslash \dot{\mathbf{m}}_S$ are determined directly by the linear constraint equations. The good news is that the dimension of the state vector is hence typically significantly reduced. The dummy derivative method can be successfully applied as long as the set of constraints is non-redundant. Since a loop-breaker component does not contain the constraint $\sum \dot{m}_i = 0$, this condition will be fulfilled. It is not possible to formulate a redundant constraint for the mass flows in a cycle-free graph with the given equation structure. This is because an acyclic graph gives rise to a partial order of the mass flow variables (or in concrete terms: downstream mass flow rates can only depend on upstream mass flow rates). The sum equations that relate the mass flows with each other can hence be put into a lower-triangular form with a full diagonal.

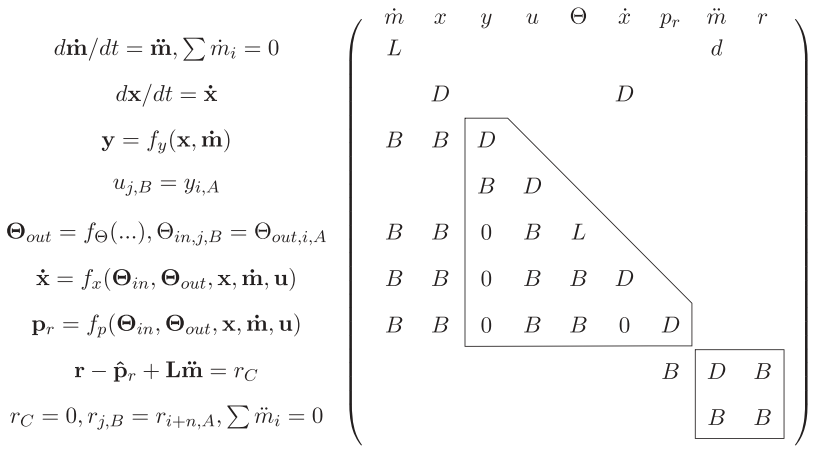

Figure 9. The structure incidence matrix brought into BLT form. This matrix is expressed block wise with the same notation as in Figure 8. *d* now denotes a non-full diagonal matrix. Note, the equations in the marked quadratic block are all linear.

Figure 9 shows the structure after the application of the dummy derivative method. We see that the only entries above the diagonal correspond to the differential equations. Hence, these will be taken care of by the numerical ODE-solver. What can we state about the structure of the remaining linear equation system? This part is marked by a rectangle in Figure 9. All the equations in this block are linear. Even better, when the parameters of the inertance $\mathbf{L}$ in equations of the form $\mathbf{r} - \hat{\mathbf{p}}_r + L\ddot{\mathbf{m}} = r_C$ are invariant to time, the linear equation system can be inverted upfront since all other coefficients are integer constants. $\mathbf{L}$ will be invariant to time when the system is of fixed geometry: a reasonable assumption for many technical systems.

In summary, we retrieve the finding of our initial illustration example of Figure 5: acyclic thermal fluid networks can be computed by straight-forward evaluation of non-linear computations from source to sink and the solution of a linear system. Since the dummy derivative method and matrix-inversion can be performed upfront, the linear part can be solved by a matrix-vector multiplication during simulation time.

Cyclic flows can also be computed in this way, as long as each cycle contains at least one loop-breaker component. The outlets of the loop-breaker act then as a source and the inlets act as a sink.

3.3.3. *Implementation of components*

We shall now show that the defined structure of equations is indeed feasible to model thermal fluid systems. To this end, we will briefly sketch the most important components in this sub-section. In the next section, we present a Modelica implementation with a corresponding use case.

To model thermal fluid systems, we roughly need the following kinds of components:

- Boundaries (e.g. sources and sinks)
- Flow distribution and mixing (such as junctions and splitters)
- Volumes
- Flow manipulation (pipes, valves, compressors, turbines, etc.)

Let us quickly go through them one by one taking an ideal gas as medium, so that the thermodynamic state is defined by pressure and enthalpy for steady mass flow $\Theta = (\hat{p}, \hat{h})$. We also define that input parameters of the model are denoted by the suffix *set* and that the inertance L results out of geometric parameters of the corresponding component. ϵ shall represent a small number >0. All these variables can hence be assumed to be known.

A typical source boundary has the following variables and equations:

$$V_{Source} := \left\{ \mathbf{\Theta}_{out} = \begin{bmatrix} \hat{p}_1 \\ \hat{h}_1 \end{bmatrix}, \dot{m}_1, \ddot{m}_1, r_1, p_{r1}, r_C \right\} \tag{33}$$

$$E_{Source} = \left\{ \begin{array}{c} \begin{pmatrix} \hat{p}_1 \\ \hat{h}_1 \end{pmatrix} = \begin{pmatrix} p_{set} \\ h_{set} \end{pmatrix}, \\ p_{r1} = 0 \\ r_1 - p_{r1} + L\ddot{m}_1 = r_C, \\ r_C = 0, \\ d\dot{m}_1/dt = \ddot{m}_1 \end{array} \right\} \tag{34}$$

A typical sink is defined by the following variables and equations:

$$V_{Sink} := \left\{ \mathbf{\Theta}_{in} = \begin{bmatrix} \hat{p}_1 \\ \hat{h}_1 \end{bmatrix}, \dot{m}_1, \ddot{m}_1, r_1, p_{r1}, r_C \right\} \tag{35}$$

$$E_{Sink} = \left\{ \begin{array}{c} p_{r1} = p_{set} - \hat{p}_1 \\ r_1 - p_{r1} + L\ddot{m}_1 = r_C, \\ r_C = 0, \\ d\dot{m}_1/dt = \ddot{m}_1 \end{array} \right\} \tag{36}$$

Such a modelling of source and sink makes the sink the energy source for the mass-flow acceleration. We can also formulate a junction of two flows with index 1 and 2 including mixing into 3 in the following form:

$$V_{Junction} := \left\{ \mathbf{\Theta}_{in} = \begin{bmatrix} \hat{p}_1 & \hat{p}_2 \\ \hat{h}_1 & \hat{h}_2 \end{bmatrix}, \mathbf{\Theta}_{out} = \begin{bmatrix} \hat{p}_3 \\ \hat{h}_3 \end{bmatrix}, \begin{bmatrix} \dot{m}_1 \\ \dot{m}_2 \\ \dot{m}_3 \end{bmatrix}, \begin{bmatrix} \ddot{m}_1 \\ \ddot{m}_2 \\ \ddot{m}_3 \end{bmatrix}, \mathbf{r}, \mathbf{p}_r, r_C \right\} \tag{37}$$

$$E_{Junction} = \left\{ \begin{array}{c} \begin{pmatrix} \hat{p}_3 \\ \hat{h}_3 \end{pmatrix} = \begin{pmatrix} (\dot{m}_1 v_1 \hat{p}_1 + \dot{m}_2 v_2 \hat{p}_2)/(|\dot{m}_1| v_1 + |\dot{m}_2| v_2 + \varepsilon) \\ (\dot{m}_1 \hat{h}_1 + \dot{m}_2 \hat{h}_2)/(|\dot{m}_1| + |\dot{m}_2| + \varepsilon) \end{pmatrix}, \\ \mathbf{p}_r = -\hat{\mathbf{p}}, \\ \mathbf{r} - \mathbf{p}_r + L\ddot{\mathbf{m}} = r_C, \\ \dot{m}_1 + \dot{m}_2 + \dot{m}_3 = 0, \\ d\dot{\mathbf{m}}/dt = \ddot{\mathbf{m}} \end{array} \right\} \tag{38}$$

where v represents the specific volume being a non-linear function of the thermodynamic state $(\hat{p}, \hat{h})$. The variables $\mathbf{p}_r$ and r_C are here used to express pressure equivalence. Equations for general n to m junctions can be set up accordingly.

A component for a finite volume with ideal mixing has an internal state. Here is a volume component with one inlet and one outlet:

$$\tilde{V}_{Volume} := \left\{ \mathbf{Q}_{in} = \begin{bmatrix} \hat{p}_1 \\ \hat{h}_1 \end{bmatrix}, \mathbf{Q}_{out} = \begin{bmatrix} \hat{p}_2 \\ \hat{h}_2 \end{bmatrix}, \mathbf{x} = \begin{bmatrix} M \\ u \end{bmatrix}, \underline{\mathbf{x}} = \begin{bmatrix} \dot{M} \\ \dot{u} \end{bmatrix}, \begin{bmatrix} \dot{m}_1 \\ \dot{m}_2 \end{bmatrix}, \begin{bmatrix} \ddot{m}_1 \\ \ddot{m}_2 \end{bmatrix}, \mathbf{r}, \mathbf{p}_r, r_C \right\} \tag{39}$$

and

$$\tilde{E}_{Volume} = \left\{ \begin{array}{c} \begin{pmatrix} \hat{p}_2 \\ \hat{h}_2 \end{pmatrix} = \begin{pmatrix} Mu(\gamma - 1)/V \\ \gamma u \end{pmatrix}, \\ \begin{pmatrix} \dot{M} \\ \dot{u} \end{pmatrix} = \begin{pmatrix} \dot{m}_1 + \dot{m}_2 \\ (\dot{m}_1(\hat{h}_1(1 + \frac{\hat{p}_2 - \hat{p}_1}{\hat{p}_1}(1 - \gamma^{-1})) + u) + \dot{m}_2(\hat{h}_2 + u))/M \end{pmatrix}, \\ \mathbf{p}_r = \begin{pmatrix} \hat{p}_2 - \hat{p}_1 \\ 0 \end{pmatrix}, \\ \mathbf{r} - \mathbf{p}_r + \mathbf{L}\ddot{\mathbf{m}} = r_C, \\ r_C = 0, \\ d\dot{\mathbf{m}}/dt = \ddot{\mathbf{m}}, d\mathbf{x}/dt = \dot{\mathbf{x}} \end{array} \right\} \tag{40}$$

where M is the total mass content of the volume V and u is the specific internal energy. The parameter γ is the adiabatic constant: the fraction of enthalpy over internal energy. The volume model is modelled such that it behaves like a sink for its inlet and like a source for its outlet; hence the implementation of $\mathbf{p}_r$. The explicit formula for $\dot{u}$ requires explanation. It is derived out of the following energy balance:

$$\dot{u}M + \dot{M}u = \dot{m}_1\hat{h}_1 + \dot{m}_2\hat{h}_2 + \dot{m}_1\hat{h}_1\frac{\hat{p}_2 - \hat{p}_1}{\hat{p}_1}(1 - \gamma^{-1}) \tag{41}$$

where the term $\dot{m}_1 h_1(\hat{p}_2 - \hat{p}_1)/\hat{p}_1(1 - \gamma^{-1}) = (\hat{p}_2 - \hat{p}_1)\dot{V}_1$ represents the flow of volume work given or taken from the enthalpy of the volume by making the inlet account for the pressure difference between $\hat{p}_1$ and $\hat{p}_2$. By convention, the energy to accelerate (or decelerate) the fluid stream is taken out of the sink and if the sink is the inlet of a volume, the energy has to be taken from (or given to) the volume.

As we can see, the function f_Θ now fulfils the requirements of a loop-breaker component. Hence, all volume components can be used to break up cyclic flows into acyclic ones. This is a very useful property.

The last and most diverse set of components is denoted as flow manipulation. These are components where the thermodynamic state of a medium flowing through the component is manipulated. Many classical components such as compressors, turbines, valves, etc., fall in this category. Here, we represent the non-linear pressure drop in a pipe as one representative example. A simple linear-quadratic resistance law is being used:

$$V_{Pipe} := \left\{ \mathbf{\Theta}_{in} = \begin{bmatrix} \hat{p}_1 \\ \hat{h}_1 \end{bmatrix}, \mathbf{\Theta}_{out} = \begin{bmatrix} \hat{p}_2 \\ \hat{h}_2 \end{bmatrix}, \begin{bmatrix} \dot{m}_1 \\ \dot{m}_2 \end{bmatrix}, \begin{bmatrix} \ddot{m}_1 \\ \ddot{m}_2 \end{bmatrix}, \mathbf{r}, \mathbf{p}_r, r_C \right\} \tag{42}$$

and

$$E_{Pipe} = \left\{ \begin{array}{c} \begin{pmatrix} \hat{p}_2 \\ \hat{h}_2 \end{pmatrix} = \begin{pmatrix} \hat{p}_1 - k_1\dot{m}_1 - k_2\dot{m}_1|\dot{m}_1| \\ \hat{h}_1 \end{pmatrix}, \\ \mathbf{p}_r = 0, \\ \mathbf{r} - \mathbf{p}_r + \mathbf{L}\ddot{\mathbf{m}} = r_C, \\ \dot{m}_1 + \dot{m}_2 = 0, \\ d\dot{\mathbf{m}}/dt = \ddot{\mathbf{m}} \end{array} \right\} \tag{43}$$

As we can see, function $f_\Theta()$ is used to implement standard text-book equations of a component. Here, a law for pressure drop and the conservation of enthalpy is used. For different components, different textbook equations (or empirical formula) might be used.

When using textbook equations, a few things need to be considered to enable a robust solution of the resulting DAE system. First of all, $f_\Theta()$ must be well defined for its complete range of possible inputs. Typical critical points here are that media models often have a limited range of validity and that many textbook equations become singular (or invalid) when $\dot{m} \to 0$. Hence, care must be taken when using textbook equations. It is also favourable if $f_\Theta()$ is a continuous function with its partial derivatives well defined. This enables the application of stiff-stable implicit solvers.

For many components, such textbook equations represent static formulations and hence, $\mathbf{x}$ will be empty. However, certain component models may contain internal states. The rotational velocity of a turbine wheel may be such an example. This is one purpose of $\mathbf{x}$ and $f_x()$.

Sometimes, it is also beneficial to interact with the internal states of other components. Models for heat-exchangers are such an example. Highly simplified, a heat exchanger can be regarded as consisting in two pipes that have the wall temperature as an internal state. The heat exchange can then be formulated as a law between these two wall temperatures. Such laws can be implemented using the variables $\mathbf{u}$ and $\mathbf{y}$ with their corresponding functions $f_x()$ and $f_y()$.

It is important to note that all examples from above are not prescriptions but demonstrations of feasibility. Components can (and are, in our case) also be differently implemented in the same structural format. We will now continue demonstrating that this format is useful for the object-oriented modelling of complex thermo-fluid systems by sketching one possible implementation in Modelica.

4. Implementation in Modelica and exemplary use case

In order to implement the proposed computational scheme, the open and free language Modelica is a natural choice. It enables to directly formulate the system using differential-algebraic equations in an object-oriented way and frees the modeller from the task of computational realization. This means: if the modeller adheres to the proposed structure of the equation system, all further processing is performed by one of the Modelica tools. This includes the ordering of equations into a BLT-form, the symbolic index-reduction, the extraction of the linear equation system and the application of an ODE-solver.

4.1. Interfaces and base components

In Modelica, individual components are collected in a package and interact with an interface class denoted as connector. In our case, the connector defines the fluid plug between components:

```
connector Inlet
  replaceable package Medium = Modelica.
    Media.Interfaces.PartialMedium;

  SI.Pressure r;
  flow SI.MassFlowRate m_flow;
  input SI.AbsolutePressure p_hat;
  input SI.SpecificEnthalpy  h_hat;
  input SI.MassFraction Xi[Medium.nXi];
 end Inlet;
```

respectively:

```
connector Outlet
  replaceable package Medium = Modelica.
    Media.Interfaces.PartialMedium;

  SI.Pressure r;
  flow SI.MassFlowRate m_flow;
  output SI.AbsolutePressure p_hat;
  output SI.SpecificEnthalpy  h_hat;
  output SI.MassFraction Xi[Medium.nXi];
 end Outlet;
```

The inertial pressure r and the mass flow rate m_flow form a pair of potential and flow variable. The thermodynamic state is expressed by steady mass flow pressure p_hat the corresponding enthalpy h_hat and the mass fraction vector Xi. It is handled as a signal going in the nominal flow direction and hence marked by input/output prefixes. Please note that the prefixes input and output per se do not imply causality. Modelica tools treat equations between such variables just as any other equation. Since the interface shall support various media, the medium type is a parameter and determines the dimension of the mass fraction vector.

Using this connector, we can setup the equations for a junction where two inflows portA, and portB meet and result in an outflow portC:

```
model Junction
  Inlet portA;
  Inlet portB;
  Outlet portC;
  SI.MassFlowRate m_flowA;
  SI.MassFlowRate m_flowB;
  SI.Pressure r_C;
  parameter Real (unit = 'm-1') L;
```

```
equation

  m_flowA = abs(portA.m_flow)+eps;
  m_flowB = abs(portB.m_flow)+eps;
  portC.p_hat = (portA.p_hat*m_flowA + portB.p_hat*m_flowB)
                          /(m_flowA+m_flowB);
  portC.h_hat = (portA.h_hat*m_flowA + portB.h_hat*m_flowB)
                          /(m_flowA+m_flowB);
  portC.Xi = (portA.Xi*m_flowA + portB.Xi*m_flowB)
                          /(m_flowA+m_flowB);

  portA.r + portA.p_hat + L*der(portA.m_flow) = r_C;
  portB.r + portB.p_hat + L*der(portB.m_flow) = r_C;
  portC.r + portC.p_hat + L*der(portC.m_flow) = r_C;

  portA.m_flow + portB.m_flow + portC.m_flow = 0;
end Junction;
```

This particular junction model uses the simplified equation for the steady mass flow pressure (Equation 20) based on the mass flow rate and not on the volume flow rate. Furthermore, a regularization scheme is applied to the (nominal) inflows in order to make the model numerically robust against zero mass flow.

Many components of the category flow manipulation have two ports. Here we present a generic two port model that implements the law for the inertial pressure and the conservation of mass:

```
partial model TwoPort
  Inlet portA;
  Outlet portB;
  parameter SIunits.Area A = world.A
  parameter SIunits.Length L = world.L
  SIunits.MassFlowRate m_flow;
  SIunits.Pressure dr;

equation
  0 = portA.m_flow + portB.m_flow;
  m_flow = portA.m_flow;
  portA.r – portB.r = dr;
  dr = der(m_flow)*L/A;
end TwoPort;
```

Actual implementations of pipes, pumps, valves, etc., can inherit this base model and extend it with the corresponding textbook equations.

4.2. Development of the HEXHEX library and its components

Starting from these base-models, we have developed the Modelica HEXHEX library for aircraft environmental control and cooling systems. It contains components for pipes, valves, pumps, fans, compressors, turbines, heat-exchangers, water extractors, etc. Also, there are boundary models to set the environmental conditions. Since this library represents a proprietary development for Airbus, its source or other details cannot be published.

The proposed structure of this paper ensures that if the components robustly compute, a total system composed out of them will do as well. On the other side, a single bad component that is not robust can still corrupt the complete system. Hence, each component must undergo a check-list for quality control:

- The equations of the component must fit into the proposed structural format.
- The component equations shall be well-matured and plausible for zero mass flow rates and mass flow rates against the nominal flow direction.
- The medium model for the thermodynamic properties must be robust enough to handle all occurring pressures and temperatures.
- Any continuous change in the mass flow rate shall lead to a continuous change in the inertial pressure. And hence the equations for $\hat{p}$ shall be formulated such that $\partial\hat{p}/\partial\dot{m}$ is always well defined.
- Further criteria regarding coding standards and usability must be met.

The check of these criteria is currently performed manually, involves also extra test-models and hence is labour prone. Yet the number of required components is finite and manageable. Given that the number of possible system compositions is significantly larger and potentially infinite, the effort undertaken for component development ultimately pays off.

To check whether the equations of a component fit into the proposed structural format, the modeller checks whether he can assign the equations of the Modelica model to the definition of C. Deviations (such as substitutions) that avoid bulky code are accepted. The functions f_Θ, f_x, f_y, f_p are not explicitly stated because this would impair the natural readability of the code. Auxiliary variables (such as m_flowA, m_flowB in the previous listing) needed to compute these functions are allowed.

It is important that the code remains naturally readable since a user of the library shall remain able to check the underlying model assumptions. Hence, the implementation of the proposed structure is not fully strict but with a sense of what is essential and appropriate.

4.3. Use case example

To demonstrate the application of HEXHEX, a complex electric architecture for an aircraft environmental control system has been modelled. In future, such system designs may replace the air cycle that is sketched in Figure 1.

Figure 10 shows the model diagram of an electric driven vapour cycle pack (eVCP). The function of this pack is to compress outside air, dehumidify it when necessary and cool (or occasionally heat) it to the desired outlet temperature. For cooling, the aircraft

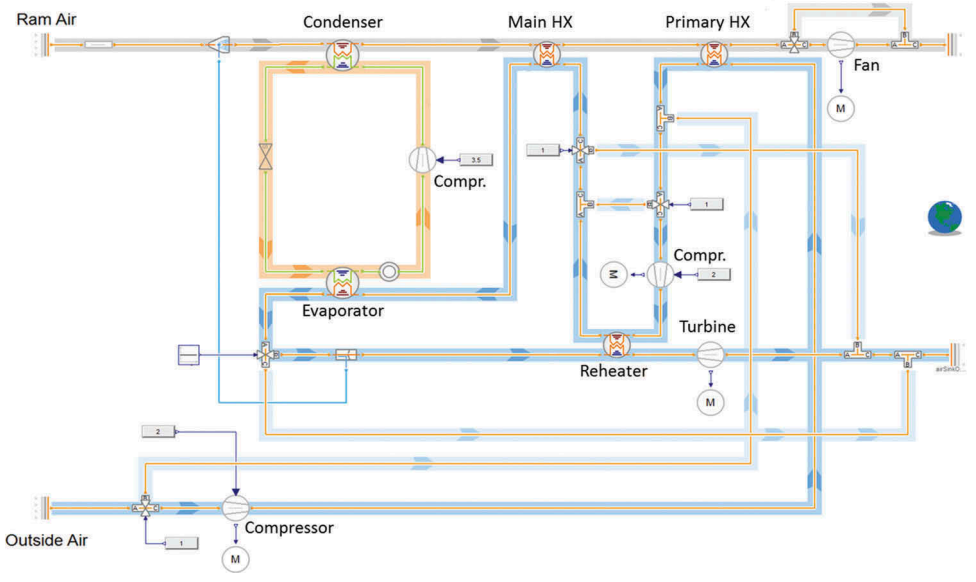

Figure 10. Modelica diagram of electric driven vapour cycle pack architecture using the HEXHEX library.

provides a ram-air channel where heat can be dissipated. To support the cooling effort, a vapour cycle with a refrigerant is provided. The architecture of this pack is derived from a patent [21]. Different from the originally published architecture, the vapour cycle was simplified. The original vapour cycle has an additional evaporator connected to recirculated air from the cabin.

Unlike conventional bleed-air driven air cycle packs [22], unconditioned outside air instead of bleed air from the engine enters the eVCP. The cold and low pressure air is compressed in a first stage before it passes the primary heat exchanger (PHX). A second compressor further raises the pressure and temperature. The dense air is then being cooled partially in order to extract the water. Finally, it is expanded in a turbine to the desired pressure level. To protect the turbine from icing and to reach the desired temperatures a reheater is installed before the turbine.

This model contains 25 time-dependent variables as internal states (as $\mathbf{x}$ in Equation 29): 3 times 2 for the 3 moist-air heat exchangers, and 2 times 8 for the two moving boundary heat exchangers in the vapour cycle model. Two for the refrigerant volume. One further state is implemented for the water extractor. Without the approach of HEXHEX, many large non-linear equation systems would occur after BLT transformation (typically at least 3 that require more than 10 iteration variables). This makes the solution of the overall system a delicate issue, especially at initialization.

Applying the proposed structural format and the equation for the inertial pressure adds 9 more states to the system. One for each separate mass flow (2 for the ram-air, 1 for the refrigerant flow, 1 for the water flow and 5 for the outside air flow with its bypasses). The important gain is that no large non-linear equation system occurs anymore. Using stiff-system solvers such as DASSL [23] or ESDIRK23 [24], the system robustly simulates through many different environmental conditions and internal configurations.

This is just one example of many possible architectures and configurations. It is therefore important that we provided a general argument in Section 3 and that the practical implementation and our daily work at many other examples confirm these general claims.

5. Validity of the proposed structural format

The proposed structural format for the equation system says very little about the validity of the actual equations being used. Using it, the modeller is free to create components of various degrees of validity and compose arbitrary systems with them. However, the proposed structure is only enabled by neglecting the impact of the inertial pressure on the thermodynamic state. We shall hence investigate in this chapter how this may impact the validity of potential system models.

Since we have used $\hat{p}$ for the computation of the thermodynamic state (and not p), this means that the impact of the pressure gradient imposed by accelerating mass flows on thermodynamic properties such as viscosity or density is not reflected in components that are between boundary or volume elements.

It is clear that this is irrelevant for the mass-flow static case (with all $r = 0$). Although this is a trivial statement, it is important because many practical industrial applications focus on a static or quasi-static analysis of thermodynamic processes. The term quasi-static means here that there is dynamic behaviour of interest but at frequencies significantly lower than mass-flow dynamics. This statement holds true also for our application field.

Yet, what about the transient behaviour? Even when the interest is rather on the mass-flow static case, a large deviation in transient behaviour may lead to a wrong static solution (in case they are multiple static solutions). To estimate any error, we are interested in the fraction of $\Delta r/p$. For an ideal gas flowing through a parcel of length Δs, we can do a simple analysis. Formulating the law for the inertial pressure in terms of velocity v and not in terms of mass flow $\dot{m}$ yields:

$$\Delta s \rho \frac{dv}{dt} = \Delta r \tag{44}$$

Using the speed of sound $c^2 = \gamma p/\rho$, with γ being the adiabatic index, we can express the quotient $\Delta r/p$ by:

$$\frac{\Delta r}{p} = \gamma \frac{d\frac{v}{c}}{dt}\frac{\Delta s}{c} \tag{45}$$

For $\Delta r/p$ to become significant, the acceleration must be expressed in Mach per second and/or the pipe length in sound-seconds. For instance, for the quotient to become roughly 10%, one must accelerate air in a (frictionless) pipe of 100 metres with 10 times the gravitational acceleration. For many typical applications, such values represent very high numbers and the error can be tolerated.

For incompressible fluids the speed of sound is determined by $c^2 = K/\rho$ with K being the bulk modulus and the statement from above does not hold. Strong (mass-flow) accelerations can indeed lead to a shift of evaporation regions or even cavitation. Nevertheless, in many cases, this method still remains applicable (although with care)

since for many liquids, important characteristics such as viscosity or density are rather insensitive to a change in pressure.

To better understand the limits of validity and how to alleviate them when needed, let us take another perspective. We can also interpret the proposed method as applying different spatial discretization schemes to the terms $\hat{p}$ and r. As with most discretization schemes, this introduces an error that shrinks with higher spatial resolution. While $\hat{p}$ is discretized on the resolution of a component (or its subcomponents), r is only discretized on a lower resolution between those components that represent boundaries or volumes. This is illustrated in Figure 11. Our method is advantageous as long it is meaningful to discretize these pressure gradients differently. This viewpoint also suggest an easy remedy for local cases where the impact of the inertial pressure gradient becomes relevant: include (more) volume elements (at a higher spatial resolution). Of course, this increases the computational cost.

The same remedy can also be applied to model the mixing of fluids in a junction, where we have chosen to ignore the last term of Equation (18). The mixing process here is modelled in an idealized fashion without an explicit volume for mixing. In case the neglected term becomes relevant for some reason, it is also appropriate to model the mixing volume explicitly. This enables to take all terms into account without corrupting the overall structure of the equation system. Hence, there is no hard limit to the validity of the proposed method. With additional volume elements, any desired precision can be reached at the cost of additional computational expense. However, if one has to add many volume elements, other modelling approaches likely become computationally advantageous.

Please also note that the underlying equations support the formulation of mass and energy conservation. Whereas the proposed approximation may cause individual mass and energy flows to deviate, the total balance can still be upheld, given correct component models. Finally, it also shall be noted that other existing fluid libraries [7] choose to ignore the inertial pressure gradient Δr in many cases. This represents just a (well-accepted) error in the different direction.

6. Conclusions

6.1. Positioning of the approach

How does the approach of HEXHEX compare to the most common other approaches? Figure 12 qualitatively positions the method in the dimension of robustness and performance and compares it with two main other approaches we find in the Modelica community. The first approach is a purely algebraic (AE) quasi-static modelling of fluid systems. This approach may avoid any states but yields large non-linear equation systems. Most volume-free components in the Modelica Standard Fluid library [7] are modelled in this style. The second approach is a realization of the finite volume method [25] (or similar) in one dimension where any flow represents the flow between two volume models. This leads to an ODE-style approach that avoids any larger non-linear equation system but at the price of creating many state-variables because of its many volumes. One possible object-oriented implementation is described in [26].

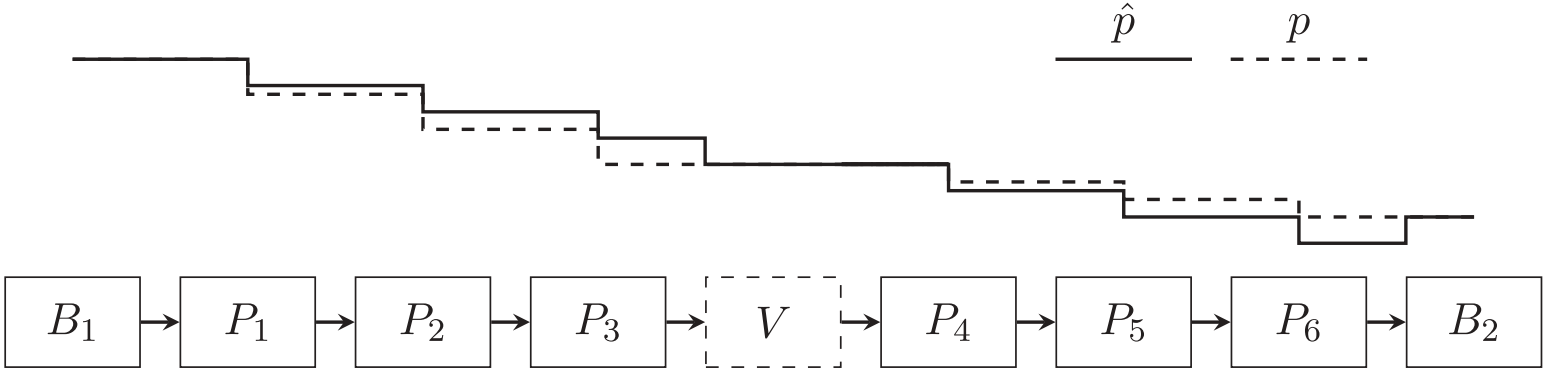

Figure 11. Qualitative illustration of the difference between p and $\hat{p}$. This absolute difference may be a source of error when computing the thermodynamic state of a medium. It is growing in downstream direction in between volumes and boundaries. Explanation: the flow from boundary B_1 to B_2 is going through a volume element V and six pipe components $P_1 - P_6$. Each pipe component models a step change in the steady mass flow pressure $\hat{p}$. The mass flow between B_1 and V is accelerating and hence $\hat{p}$ stays above p until the volume element. The mass flow between V and B_2 is decelerating and hence $\hat{p}$ stays below p until the boundary element.

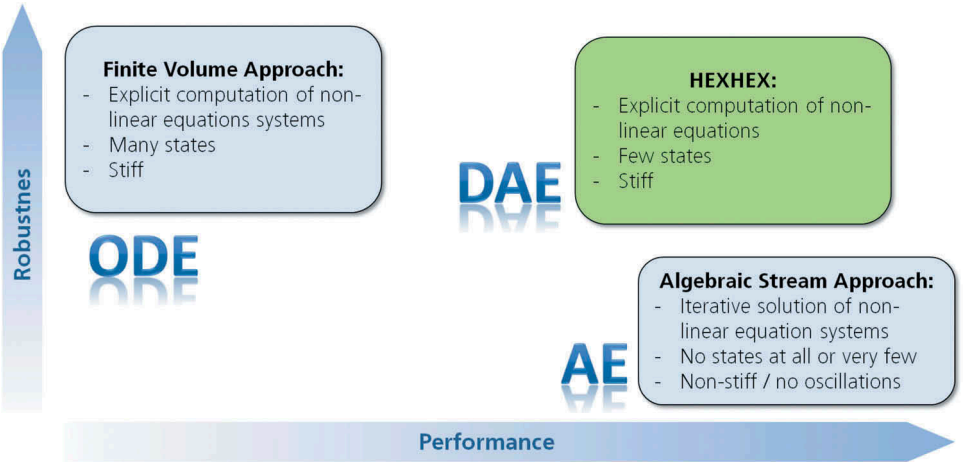

Figure 12. Qualitative positioning of the presented approach. HEXHEX is more robust than a pure algebraic approach at the cost of adding dynamic states to the system. Yet the number of added states is far less than with other approaches based on staggered grids and finite volumes.

HEXHEX is right in the middle, looking for the sweet spot. It avoids completely any large non-linear equation system but creates a DAE system that after applying the dummy derivative method can be handled by an ODE solver. Hence, robustness on the component level leads to robustness on the total system level. Yet, it creates only a small set of states. The set is much smaller than for a finite volume approach, since only the mass-flow rates are used as state variables and not the full thermodynamic state of a volume. Furthermore, the mass-flow rates are typically shared for serially connected components. This enables the dummy derivative method to further reduce the set of state variables.

To have more concrete figures and better understanding in the differences of dynamic behaviour, let us model the academic example of Figure 2 (or 5, respectively) using these different approaches. For the sake of the example, moist-air is taken as a medium and the black boxes represent pipes of $1m$ length and $0.001m^2$ cross-section area with non-linear flow resistance. The inertance for each pipe is consequently $L = 1000m^{-1}$. The finite volume methods are applied here simplistically by just modelling the junctions A and B as volumes (of corresponding $0.002m^3$) so that there is always a volume between the algebraic flow components. The method is applied with (denoted as ODE dynamic) and without (denoted as ODE) taking into account mass-flow dynamics.

The system is at steady state when a discrete step of input pressure is applied. The steady-state solution is the same for all methods. Differences occur only in transient behaviour. Figure 13 displays the dynamic response of the outlet mass-flows, for all 4 methods. The algebraic approach does not add any further dynamics and hence its response directly represents the step change at the inlet. The DAE approach of this paper shows an immediate response of first-order dynamics where the mass flow adapts to the new inlet pressure. It is similar for the ODE-based solution using finite volumes. Here the pressure drop has to propagate through two volumes. Adding mass flow dynamics in the pipes adds a damped oscillatory behaviour of high frequency. In this

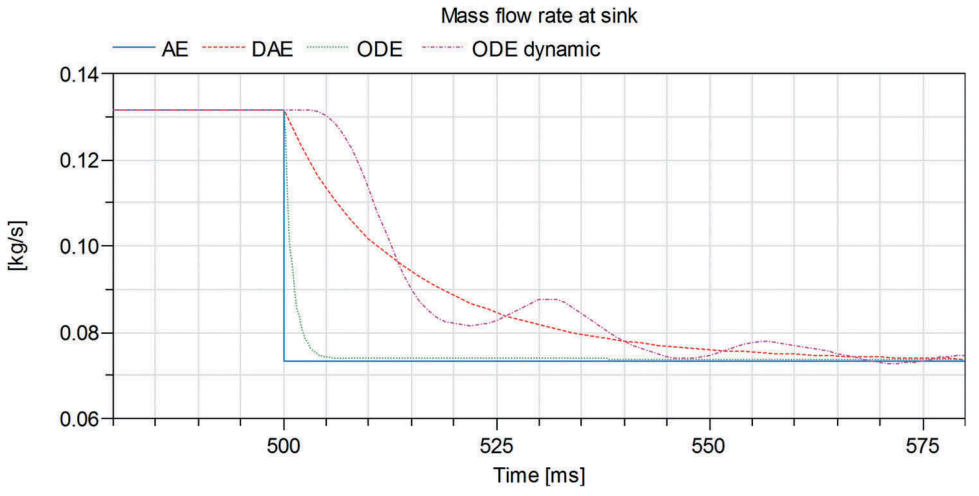

Figure 13. Transient behaviour of different methods as reaction to discrete drop in inlet pressure.

Table 1. This table lists the states and the occurrence of non-linear systems in implicit form for the given example. For the algebraic approach, a system of 19 non-linear equations occur for this example. A Modelica tool may reduce the number of iteration variables for the numerical solution. Here, the tool Dymola reduces the system to 2 iteration variables.

Method	Set of state variables	# states	Non-linear
AE	$\{\}$	0	19
DAE	$\{\dot{m}_1, \dot{m}_2\}$	2	0
ODE	$\{p_A, u_A, X_A, p_B, u_B, X_B\}$	6	0
ODE dynamic	$\{p_A, u_A, X_A, p_B, u_B, X_B, \dot{m}_0, \dot{m}_1, \dot{m}_2, \dot{m}_3\}$	10	0

example, the time-constant for the gas compression is faster than for the gas inertia but in general that depends on the geometry and the medium.

Table 1 lists all states for each method and the size of the resulting non-linear equation system. The DAE-based method of this paper avoids the iterative solution of non-linear equation system by using the smallest number of states. This was also the original intent when designing this method. However, one has to be careful when extrapolating the findings of this table to larger systems. The results depend on the topology. The method described here, benefits from topologies where components are predominantly connected in series with few cycles and mass flow balance can be assumed for large parts of the system. Topologies featuring many parallel, potentially circular flows are rather unsuited.

6.2. Initialization and boundary conditions

A robust method for initialization is also important for the end-user. Ideally, only the boundary conditions are defined and the components of the system do not require additional information for initialization. For many cases, an initialization at rest with all mass-flow rates being zero is a good starting condition. The system will then ramp-up

according to the boundary condition. It is actually the same as plugging in (or switching on) the actual device.

In some cases, this ramping up may lead to a different state than expected. This typically indicates flaws in the actual system model. In any case, there will be a trajectory leading up to the issue, which allows for better diagnostics than just a failed initialization with a non-linear solver.

So far, we presented a fixed schematic for the boundary conditions of the system: prescribing the full thermodynamic state at the inlet and the pressure at the outlet. It is however possible to work with other boundary conditions as well.

One can release boundary conditions for the pressure and replace them with boundary conditions for the mass flow. As long as these are one-time differentiable with respect to time and non-redundant, the resulting system can be solved using linear equations.

As popular with many object-oriented modelling approaches, it is also possible to prescribe the outlet state and compute upstream. This will however involve non-linear equation systems again and all the problems regarding the absence and multitude of solutions.

6.3. Outlook on future work

Our daily practical work shows the usefulness of the proposed structural format. The time needed for modelling and simulation of aircraft ECS systems has been drastically reduced (by more than 80% given our personal experience) and we are convinced that the usefulness of this modelling approach extends beyond applications for aviation. The end-user can apply this method without being aware of any of the analysis undertaken in Section 3. Airbus is planning an intensive use of this approach to conduct extensive studies for an early design phase.

This paper had a strong focus on robustness because this was our main issue. Nevertheless, the structural format also seems very suitable for the purpose of thermo-fluid simulation under hard real-time conditions. First studies [27] on this topic revealed promising results.

As the title suggests, the approach presented here is restricted to directed fluid flows, meaning that the flow direction is known a priori. This is true for many technical systems but not for all of them. However, also new developments [28] demonstrate this restriction can be lifted and extensions of this format can be created that work for bi-directional fluid flows.

Meanwhile, research on future modelling languages like Modia [29] is adopting the solution of this paper and making it the default approach for their modellers. In this way, we think that this work has a long-lasting impact on the object-oriented modelling of thermofluid systems.

Acknowledgements

I would like to thank Airbus for providing incentive by their challenging problems, by offering support by intensive testing and by providing user feedback. Further acknowledgements go to Dr Daniel Bender and Dr Alexander Pollok who were main contributors to the implementation of the HEXHEX library. In addition, I would like to thank Prof. Martin Otter, Lâle Briese and Cristina Benéitez Ortega for

critical reviews and valuable feedback. Those parts of the research that relate to the EVCP example model received funding from European Union's Horizon 2020 program for the Clean Sky 2 Joint Technology Initiative under grant agreement H2020-CS2-CFP03-2016-01 GAN 737792.

Disclosure statement

No potential conflict of interest was reported by the author.

References

[1] M. Sielemann, T. Giese, B. Oehler, and M. Gräble, *Optimization of an unconventional environmental control system architecture*, SAE Int. J. Aerosp. 4 (2) (2011), pp. 1263–1275. Toulouse, France. doi:10.4271/2011-01-2691.

[2] S. Wischhusen, *Modelling and calibration of a thermal model for an automotive cabin using human comfort library*. Proceedings of the 9th International Modelica Conference, September 3-5, Munich, Germany, 2012

[3] F. Casella and A. Leva, *Object-oriented modelling & simulation of power plants with modelica*, Proceedings of the 44th IEEE Conference on Decision and Control, Seville, Spain, 2005.

[4] M. Wetter, M. Bonvini, and T.S. Nouidui, *Equation-based languages - A new paradigm for building energy modeling, simulation and optimization*, Energy Build. 117 (1) (2016), pp. 290–300. doi:10.1016/j.enbuild.2015.10.017.

[5] P. Fritzson, *Principles of Object-Oriented Modeling and Simulation with Modelica 3.3*, 2nd ed., IEEE Press, Piscataway, New Jersey, 2014, pp. 1256.

[6] F. Casella, M. Otter, K. Proelss, C. Richter, and H. Tummescheit, *The modelica fluid and media library for modeling of incompressible and compressible thermo-fluid pipe networks*, Proc. 5th International Modelica Conference, Vienna, Austria, Vol. 2, pp.559–568, 2006

[7] R. Franke, et. al., *Standardization of thermo-fluid modeling in Modelica.Fluid*, Proceedings 7th Modelica Conference, September 20-22, Como, Italy, 2009

[8] R. Franke, et. al., *Stream connectors – an extension of modelica for device-oriented modeling of convective transport phenomena*, Proceedings 7th Modelica Conference, September 20-22, Como, Italy, 2009

[9] D. Zimmer, *Using artificial states in modeling dynamic systems: Turning malpractice into good practice*, Proceedings of the 5th International Workshop on Equation-Based Object-Oriented Languages and Tools (EOOLT), Nottingham, United Kingdom, 2013.

[10] F. Jorissen, M. Wetter, and L. Helsen, *Simplifications for hydronic system models in Modelica*, J Build. Perform. Simul. 11 (6) (2018), pp. 639–654. doi:10.1080/19401493.2017.1421263.

[11] M. Sielemann, F. Casella, M. Otter, C. Clauß, J. Eborn, S.E. Mattsson, and H. Olsson, *Robust initialization of differential-algebraic equations using homotopy*. Proceedings of 8th International Modelica Conference. 8th International Modelica Conference, 20-22. Mrz 2011, Dresden, 2011.

[12] F. Casella, M. Sielemann, and L. Savoldelli, *Steady-state initialization of object-oriented thermo-fluid models for homotopy methods*. Proceedings of 8th International Modelica Conference, Dresden, 2011.

[13] E. Truckenbrodt, *Lehrbuch der angewandten Fluidmechanik*,Springer-Verlag Berlin und Heidelberg GmbH & Co. KG, Germany, 1983.

[14] B.S. Massey, *Mechanics of Fluids*, Chapman & Hall. ISBN 0-412-34280-4, 1989.

[15] C.E. Brennen, *Internet Book on Fluid Dynamics*, Dankat Publishing, Pasadena, California, USA, 2015.

[16] S.E. Mattsson and G. Soederlind, *Index reduction in differential-algebraic equations using dummy derivatives*, SIAM J. Sci. Comput. 14 (3) (1993), pp. 677–692. doi:10.1137/0914043.

[17] D. Brück, H. Elmqvist, and H. Olsson. *Dymola for multi-engineering modeling and simulation*, Proc. of the 2nd International Modelica Conference, Oberpfaffenhofen, Germany, 2002.
[18] F.E. Cellier and E. Kofman, *Continuous System Simulation*, Springer Verlag, New Yorg, 2006, pp. 643.
[19] D.V. Steward, *Partitioning and tearing systems of equations*, J. Soc. Ind. Appl. Math. 2 (1965), pp. 345–365.
[20] C. Pantelides, *The consistent initialization of differential-algebraic systems*, SIAM J. Sci. Stat. Comput. 9 (2) (1988), pp. 213–231. doi:10.1137/0909014.
[21] S. Golle, et. al., *Betriebsphasenabhängig steuerbare Flugzeugklimaanlage und Verfahren zum Betreiben einer derartigen Flugzeugklimaanlage*, Patent, DE102015207447A1, October 27, 2016.
[22] D. Bender, *Integration of exergy analysis into model-based design and evaluation of aircraft environmental control systems*, Energy 137 (2017), pp. 739–751. doi:10.1016/j.energy.2017.05.182.
[23] L. Petzold, *Description of DASSL: A Differential/algebraic System Solver*, Sandia Natinoal Labs, Livermore, CA, USA, 1982.
[24] C.A. Kennedy and M.H. Carpenter, *Diagonally implicit runge-kutta methods for ordinary differential equations. A review*, NASA Report. TM-2016-219173, 2016.
[25] F. Moukalled, L. Mangani, and M. Darwish, *The Finite Volume Method in Computational Fluid Dynamics: An Advanced Introduction with OpenFOAM and Matlab*, Springer International Publishing, Switzerland, 2015, pp. 814.
[26] B. El Hefni and D. Bouskela. Dynamic modelling of a condenser with the thermo SysPro library, *Proceedings of the 10th International ModelicaConference*, Lund, Sweden, March 10-122014.
[27] D. Zimmer, *Towards hard real-time simulation of complex fluid networks*. Proceedings of the 13th International Modelica Conference, March 4-6, 2019, Regensburg, Germany, 2019.
[28] D. Zimmer, *Robust Simulation of Stream-Dominated Thermo-Fluid Systems: From Unidirectional to Bidirectional Applications. to be published as part of SNE/EUROSIM*, 2019.
[29] M. Otter, et al., *Thermodynamic property and fluid modeling with modern programming language constructs*, Proceedings of 8th International Modelica Conference. 13th International Modelica Conference, March 04-06, 2019, Regensburg, Germany, 2019.

Publication title	Robust object-oriented formulation of directed thermofluid stream networks
Authors	Dirk Zimmer
Issue Date	16.06.2020
Publisher	Taylor & Francis. Mathematical and Computer Modelling of Dynamical Sytems
Type of publication	Journal Publication. Gold Open Access
Acknowledgement	The underlying work of this paper has partly received funding from the Clean Sky 2 Joint Undertaking (JU) under grant agreement No . The JU receives support from the European Union's Horizon 2020 research and innovation programme and the Clean Sky 2 JU members other than the Union
Disclaimer	The content of this article reflects only the authors' view. The Clean Sky 2 Joint Undertaking is not responsible for any use that may be made of the information it contains.