Robust Modeling of Directed Thermofluid Flows in Complex Networks

Dirk Zimmer   Daniel Bender   Alexander Pollok

Institute of System Dynamics and Control, DLR German Aerospace Center
`{Dirk.Zimmer,Daniel.Bender,Alexander.Pollok}@dlr.de`

Abstract

A new concept is presented on how to set up the equations of complex fluid networks. This concept avoids the creation of large non-linear equation systems and hence leads to a very robust modeling approach while still being competitive in levels of performance. The concept has been implemented in Modelica and tested for the rapid pre-design of aircraft environmental control systems.

Keywords: fluid systems, thermal systems

1   Motivation

Modelica has established itself as a valuable tool for the modeling of thermal fluid systems. Typical applications are the modeling of power plants (Casella 2005), building simulation (Wetter, 2016), or, as in our case, the pre-design of environmental control systems (ECS) for future aircraft (Schlabe, 2014; Sielemann, 2011).

To support these activities, several quasi-standards have been developed: a stream connector (Franke *et al*, 2009B) has been included in the Modelica language standard and a corresponding standard library supports the modeling of fluids. (Franke *et al*, 2009A) Furthermore, the Modelica.Media library (Casella, 2006) provides models for a multitude of different fluid media, so that the same fluid models can be applied to different media.

Yet despite these advances, there still remain reoccurring problems that make the application for the end user challenging. Most of them involve the solvability of (larger) non-linear 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 development time of new fluid architectures.

Also here, several attempts such as homotopy (Casella 2011) have been undertaken to address this problem but so far with mixed or limited success.

This paper presents a new approach to model fluid systems that avoids the creation of large non-linear equation systems in the first place. This leads to a very robust fluid library, and also high performing and scalable models.

2   The inertial pressure

In order to understand this approach, let us examine the root of the problem. What leads to the creation of large non-linear equation systems?

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. Especially critical are 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 fulfill this equivalence, the corresponding mass-flows become part of a non-linear equation system.
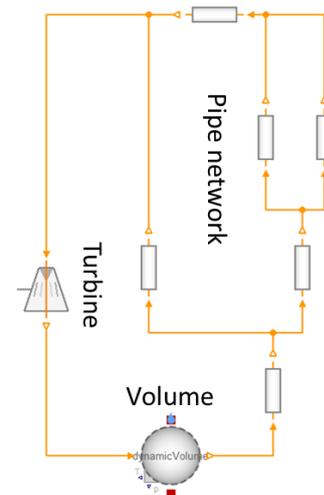

Figure 1: Simple fluid network

Figure 1 provides a simple example of a loop with two nested side-branches, leading to a non-linear equation system for the mass-flows. If modeled using conventional components and the stream connector, a system of 63 non-linear equations results that can be reduced to 8 iteration variables. More complex networks such as in (Zimmer, 2013) require more than 40 iteration variables. It is then a priori unclear how many solutions there are and whether a generic non-linear equation solver will find any of them (and which one), especially at (re-)initialization.

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 very favorable way to formulate this. To this end, let us review the fundamental equations of motion. Newton's second law can be formulated in terms of pressure gradient and density. This is the most basic part and form of the Euler equations for incompressible fluid flows, disregarding any external forces (such as gravity, friction, etc.):

$$\frac{D\mathbf{u}}{Dt} = -\frac{1}{\rho}\,\nabla p$$

with $\mathbf{u}$ being the velocity, p the pressure and ρ the density. Since we work in an Eulerian framework where particles move through a parcel, we shall expand the material derivative:

$$\frac{\partial \mathbf{u}}{\partial t} + \mathbf{u} \cdot \nabla \mathbf{u} = -\frac{1}{\rho}\,\nabla p$$

Since we concern ourselves here only with one-dimensional flows, we can form a scalar PDE where v expresses the flow velocity $\mathbf{u}$ in direction of s:

$$\frac{\partial v}{\partial t} + v\frac{\partial v}{\partial s} = -\frac{1}{\rho}\frac{\partial p}{\partial s}$$

Using a spatial element (such as a pipe section) of the length Δs and cross-section area A, this PDE is transformed into an ODE:

$$\frac{dv}{dt} + v\frac{\Delta v}{\Delta s} = -\frac{1}{\rho}\frac{\Delta p}{\Delta s}$$

Multiplying with $\rho\Delta s$ and substituting v in the first term with $\dot{m}/\rho A$ yields finally

$$\underbrace{\frac{d\dot{m}}{dt}\frac{\Delta s}{A}}_{\Delta r} + \underbrace{\rho\,v\Delta v}_{\Delta q} = \Delta p$$

In this way, the pressure difference Δp is decomposing into Δr and Δq. Δq is expressing the change in dynamic pressure for a given mass-flow, a well-established and frequently used quantity. The term expressed by Δr is all too often neglected. It expresses the inertial pressure to change the mass flow rate.

Alternative derivations or explanations of this inertial component in the pressure gradient can be found in (Truckenbrodt, 1983) and in (Longwell, 1966) for line flows of fluids and particles. These build upon a stream of fluid or particles through a pipe. For illustration let us look at a straight pipe.
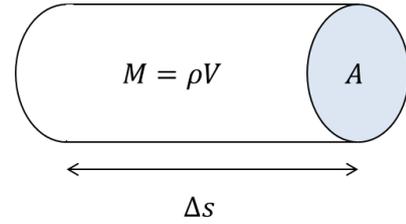

Figure 2: mass flow through a pipe

Figure 2 displays a section through a straight pipe. Its content corresponds to a mass M and its acceleration a will require a force f

$$f = \Delta r A = Ma$$

Assuming that the flow through such a pipe fulfills the mass balance so that inflow equals outflow, we can express the acceleration in terms of volume flow:

$$\Delta r A = V\rho\frac{1}{A}\frac{d\dot{V}}{dt}$$

Dividing the volume by A yields Δs and multiplying the volume flow by the density yields the mass flow.

$$\Delta r A = \Delta s\frac{dm}{dt}$$

or

$$\Delta r = \frac{d\dot{m}}{dt}\frac{\Delta s}{A}$$

Again, we retrieve the formula for Δr. Let us note that it is a simple and linear equation that is independent of the thermodynamic state of the fluid and whose parameters remain constant for pipes with non-changing geometry. This will become very useful.

The reason Δr is so often neglected is that it is often very small and does not influence the thermodynamic state significantly (see Section 5). However, although mostly negligible, it is still a very useful quantity to compute changes in the mass-flow from occurring pressure differences. Using it, we can describe a dynamic that leads to pressure balance even in complex fluid networks. To this end, we shall revisit the equations for fluid junctions.

Prerequisite of this approach is that we work with fluid flows that uphold the mass flow balance $\dot{m}_{in} = \dot{m}_{out}$, meaning that the dynamics of compression are regarded as irrelevant for the fluid transport and the medium moves like being incompressible. However, the medium is allowed to change its thermodynamic state, even its density, when being subject to thermodynamic manipulation such as in a compressor, turbine, heat-exchanger, etc. Hence, we call this approach quasi-incompressible because the assumption of incompressibility (actually, mass flow balance) only affects the transport dynamics. In case transport phenomena of compressible flows are relevant, other approaches such as (Sielemann, 2012) have to be used.
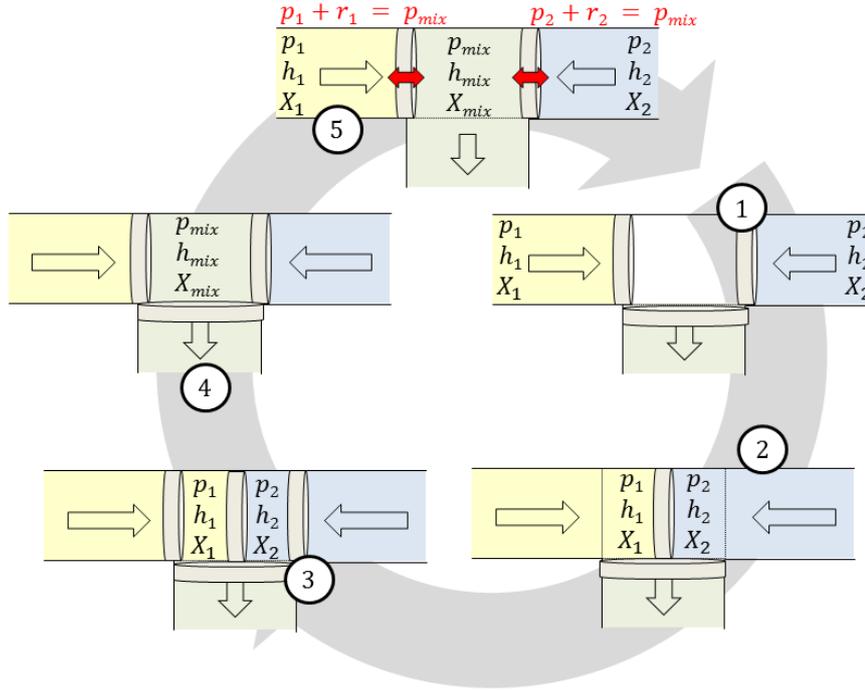
Figure 3: Illustration of the thought experiment for the mixing of two fluids

3 Pressure balance

By using this decomposition of the pressure gradient, we can revisit the model for junctions of a fluid flow. Let the pressure $\hat{p}$ be composed of the inertial pressure r and the pressure p for the case of static mass-flow:

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

Pressure in general represents a sum of forces per area. These forces may result from macroscopic motion such as acceleration or from the microscope state of the medium to make place for its volume. Since we attribute the macroscopic acceleration of the mass flow already to r, p is now interpreted as resulting from the microscopic state that is bound to its specific volume and transported with the fluid. Hence p can here be treated similar to a specific quantity and is consequently also subject to mixing. We can hence formulate a mixing law for p:

$$p_{out} = f(p_{in,1}, h_{in,1}, \dot{m}_{in,1}, \dots, p_{in,2}, h_{in,2}, \dot{m}_{in,2}, \dots)$$

To compute this specific mixing pressure of two fluid flows, we shall conduct the following thought experiment. We fill a chamber with an amount of each fluid that is proportional to its mass-flow. The chamber is sealed on both ends and the two fluids a separated by a virtual disc. As soon as we remove this disc, mixing of the two fluids takes place and a new pressure establishes. Figure 3 illustrates this thought experiment in its stages 1 to 4.

In general, this leads to a non-linear mixing law. In case, the specific gas constant for both sections of the chamber is equal, a linear law can be used:

$$p_{out} = p_{mix} = \frac{\sum \dot{m}_i p_i}{\sum \dot{m}_i + \epsilon}$$

This linear law can also be used as approximation for the non-linear law when the exact pressure dynamics are not regarded as important during mass-flow transients or the pressure differences are low. Although we prefer to use a linear law for the sake of simplicity, it does not harm the overall equation structure when a non-linear law is used.

To get from a single stage mixing to a continuous process, let us look at stage 5 in Figure 3 that completes the circular process. We recognize that the mixing chamber has now a pressure p_{mix} that is different from both p_1 and p_2. This pressure difference occurs across the sealing virtual discs at both ends. If we now unhinge these discs, we shall attribute this pressure difference to the inertial pressure r.

Hence the pressure balance at a junction can be formulated for any inflowing stream i or outflowing stream j in terms of $\hat{p}$:

$$\hat{p}_i = \hat{p}_{mix}$$

or

$$p_i + r_i = p_j + r_j$$

4 Computing an unidirectional network

The decomposition of $\hat{p}$ into p and r does not only affect the pressure balance at a junction but it restructures the overall equation system of the network into a very favorable form. This is best explained by means of an example. Figure 4 represents a simple fluid network structure where a split (A) fluid stream of given total mass flow rate rejoins (B). 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:

$$\frac{d\dot{m}}{dt}\frac{\Delta s}{A} = \Delta r$$

This law is applied to all thermofluid components and causes 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 straight forward manner. All required variables represent knowns. This forward computation is symbolized by the black variables in Figure 4. 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, $\hat{p}$ equals the desired outlet pressure. The pressure balance law is applied for junction B whereas the split at junction A defines equality of its inertial pressures. The black box components contain the law for the inertial pressure.

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

$$\frac{d\dot{m}_1}{dt}\frac{\Delta s_1}{A} = r_1 - r_A$$

$$\frac{d\dot{m}_2}{dt}\frac{\Delta s_2}{A} = r_2 - r_A$$

$$p_1 + r_1 = p_2 + r_2$$

$$\frac{d\dot{m}_1}{dt} = -\frac{d\dot{m}_2}{dt}$$

The last equation thereby results from the index reduction of the system: it represents the time-derivative of the mass-flow constraint $\dot{m}_1 + \dot{m}_2 = \dot{m}_0$ with $\dot{m}_0$ being given. We can rewrite the equations above as linear equation system:

$$\begin{bmatrix} -1 & & \Delta s_1/A & \\ & -1 & & \Delta s_2/A \\ 1 & -1 & & \\ & & 1 & -1 \end{bmatrix}\begin{bmatrix} r_1 \\ r_2 \\ d\dot{m}_1/dt \\ d\dot{m}_2/dt \end{bmatrix} = \begin{bmatrix} 0 - r_A \\ 0 - r_A \\ p_2 - p_1 \\ 0 \end{bmatrix}$$

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 only occurs as simple matrix-vector multiplication during simulation time. When implementing this in Modelica, a tool like Dymola (Brück, 2002) is smart enough to do exactly this.

We can generalize the lessons from the example in Figure 4 and derive general statements for the resulting structure of the equation system of any network of fluid systems. To this end, we have to postulate two underlying modeling assumptions that have to be fulfilled by the modeler:
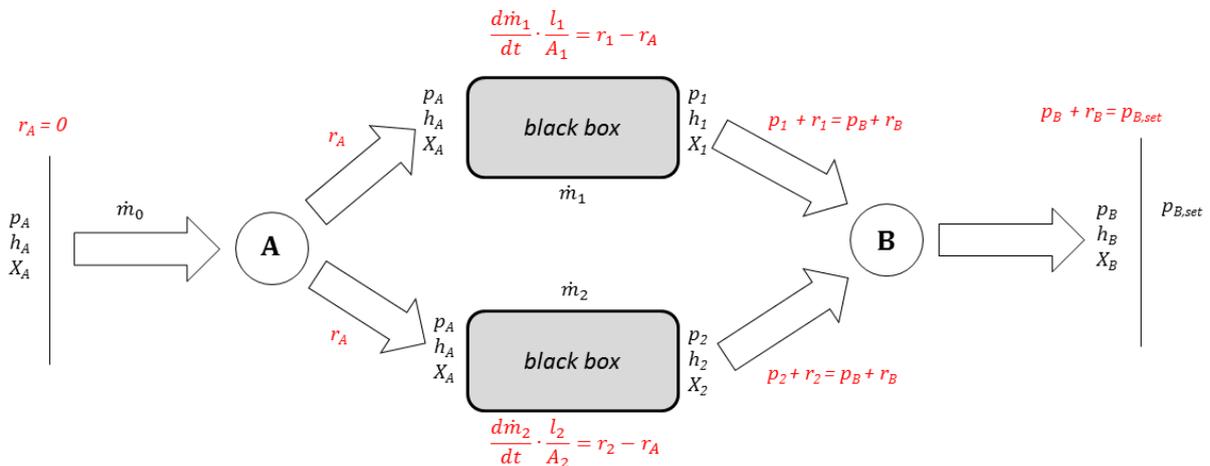

Figure 4: Computation of a simple directed fluid flow. The black terms represent a straight forward computation from source to sink. The red equations describe the mass-flow dynamics and form a linear system of equations.

- The directed fluid flow gives rise to a partial order of its fluid connectors along the flow. (This means that no circular flows must occur unless these are cut by a volume(-like) element)
- There is no direct algebraic coupling between two flows within a component. (This means in practice that on the component level differential equations may be employed instead of pure algebraic equations)

If these two assumptions are upheld by the modeler, then any fluid system will be represented by a structure incidence matrix (Cellier, 2006) of the following block-lower-triangular (BLT) form in Figure 5:
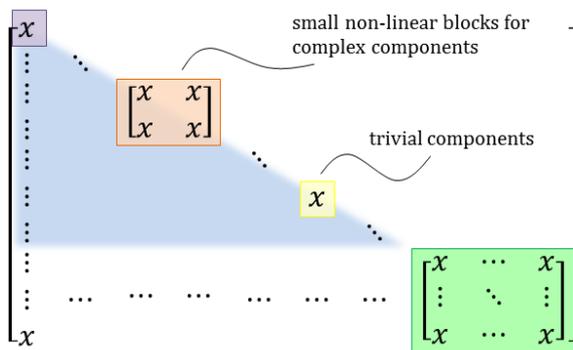

Figure 5: Structure incidence matrix in BLT Form

The blue part of this incidence matrix may represent a straight-forward computation of non-linear equations. There might be small non-linear equations systems stemming from the components but they do not grow in size when connecting these components. This means that if the solvability has been proven on the component level, it will remain solvable on the system level.

The only large equation system is marked by the green part. This is however a linear equation system with constant coefficients that can be inverted upfront. The size of this system is thereby proportional to the number of different branches in the fluid network. It is, thanks to index-reduction, invariant to the length or complexity of each branch.

Given this BLT form, it is now clear how high robustness for the end user is achieved. No large non-linear equation system will be created by building a complex network out of its components. If the components are very robust then the total system will be as well.

5 Validity of the approach

Although the above structure is very favorable it also contains a little shortcut since we have used p for the computation of the thermodynamic state and not $\hat{p}$. This means that the gradient in the thermodynamic state between junctions is not properly taken into account. It is clear that this is irrelevant for the mass-flow static case (with all $r = 0$) but what about the transient behavior?

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 and not in terms of mass flow yields:

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

Using the speed of sound $c^2 = \kappa\, p/\rho$, we can express the quotient $\Delta r/p$ by:

$$\frac{\Delta r}{p} = \kappa\, \frac{d\frac{v}{c}}{dt}\, \frac{\Delta s}{c}$$

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 meters 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 $c^2 = K/\rho$ and the simple statement from above does not hold. Strong (mass-flow) accelerations can indeed lead here to a shift of evaporation regions or even cavitation. Nevertheless, in many cases, this can be neglected as well. In some cases such as a water hammer, strong acceleration does occur but the thermodynamic state is rather insensitive to pressure and hence the approach remains applicable (although with care).

In cases where transient effects become relevant there is a multitude of potential work-arounds that shall be omitted here in order to be concise. For our application field, this is however irrelevant.

It also shall be noted that many existing fluid libraries choose to ignore the inertial pressure completely. This is just doing a (well-accepted) error in the different direction. Important is just that the modeler understands the underlying assumptions.

6 Implementation in Modelica

The design of a suitable connector is here the most fundamental decision. The inertial pressure `r` and the mass-flow `m_flow` hereby constitute a classic pair of potential and flow variable. All the remaining variables, such as mass-flow static pressure `p`, specific enthalpy `h`, etc., are then regarded as specific quantities and transmitted via input and output connectors:

```
connector Inlet
 replaceable package Medium = Modelica.
    Media.Interfaces.PartialMedium;
 Modelica.SIunits.Pressure r;
 flow Medium.MassFlowRate m_flow;
 input Medium.AbsolutePressure p;
 input Medium.SpecificEnthalpy h;
 input Medium.MassFraction Xi[Medium.nXi];
end Inlet;
```

respectively:

```
connector Outlet
 replaceable package Medium = Modelica.
    Media.Interfaces.PartialMedium;
 Modelica.SIunits.Pressure r;
 flow Medium.MassFlowRate m_flow;
 output Medium.AbsolutePressure p;
 output Medium.SpecificEnthalpy h;
 output Medium.MassFraction Xi[Medium.nXi];
end Outlet;
```

An alternative design is to use the thermodynamic state record of the Modelica.Media Library directly in the connector instead of the variables $(p, h, ...)$. This can lead to a better performing solution but maybe slightly more bulky equations.

Also one can artificially enrich the connector by additional signals in order to ensure that there are no open ends or unwanted loops. For instance, an integer input/output signal in reverse flow direction counting upwards from sink to source, would lead to errors in case of open endings or unwanted loops. So far, such mechanism have however shown to be unnecessary.

Using this connector, we implemented the proprietary HEXHEX library for aircraft environmental control and cooling systems in an early design phase. It contains components for pumps, fans, compressors, turbines, heat-exchangers, etc. Boundary models enable to set the environmental conditions. A global world model enables to set common default values. Although similar in its look to the DENECS library (Sielemann, 2011) it represents a complete new implementation.

When implementing the components, robustness must be thoroughly tested. The components must not become singular at zero-mass flow and should continue to exhibit a plausible behavior even when being used outside their range of validity.

Different from the standard fluid library, junctions now demand for an extra model. There are classic T-Junctions, X-Junctions and 1-to-N Junctions. Although this could be regarded as additional burden, there were no complaints from the user base. The basic law:

$$\frac{d\dot{m}}{dt}\frac{\Delta s}{A} = \Delta r$$

for the inertial pressure of Section 2 is now present in every single component (that is not a boundary). Hence any classic two-port component extends this law from a partial base class, where this law has been implemented:

```
partial model TwoPort
  Inlet portA;
  Outlet portB;
  parameter SIunits.Area A = world.A
  parameter SIunits.Length L = world.L
  SIunits.MassFlowRate m_flow( start=0);
  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;
```

The pressure balance of Section 3 is then implemented for instance in a junction model where two inflows `portA`, `portB` meet and result in an outflow `portC`:

```
model Junction
  Inlet portA;
  Inlet portB;
  Outlet portC;
equation
 m_flowA = portA.m_flow;
 m_flowB = portB.m_flow;
 m_flowC = portC.m_flow;
 m_flowA + m_flowB + m_flowC =0;

 portC.Xi = (portA.Xi*(m_flowA + eps)
           + portB.Xi*(m_flowB + eps))
           /(m_flowA+m_flowB+2*eps);
 portC.h = (portA.h*(m_flowA + eps)
           + portB.h*(m_flowB + eps))
           /(m_flowA+m_flowB+2*eps);
 portC.p = (portA.p*(m_flowA + eps)
           + portB.p*(m_flowB + eps))
           /(m_flowA + m_flowB + 2*eps);

 portC.p + portC.r = portA.p + portA.r;
 portC.p + portC.r = portB.p + portB.r;

end Junction;
```

In this junction model, the simple linear approximation for the resulting mixing pressure is applied. Also a small regularization is applied to cover the case of zero mass flow rate. This is especially helpful for initialization when the system is ramping up from zero mass flow.

Aside from a few basic classes such as the TwoPort component from above, the library avoids the overuse of inheritance since multiple inheritance levels have shown to be detrimental to the readability of the code (Pollok, 2016).

Because also the stream connector is avoided, the thermodynamic equations of components like heat exchangers, turbines, fans, etc. can be written in a straight forward way. This is greatly enhancing the readability of each component code. In this way, the code becomes easy to understand even by the non-expert. Also the development time of individual components has been reduced by an approximate of 50%.

The library has undergone a significant testing effort by external users for the rapid pre-design of a large variety of aircraft environmental control systems. We estimate that the development time of such architectures has been reduced roughly by at least 80%. One particular architecture is presented in the next section in order to show the application of the HEXHEX library.

7 Exemplary Use Case

To demonstrate the feasibility of HEXHEX, a complex electric architecture for an aircraft environmental control system has been modeled.

Figure 6 shows the diagram layer of the modelled electric driven vapour cycle pack (eVCP) architecture. The architecture is derived from a patent publication (Golle, 2016). Unlike the original 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 (Bender, 2017) 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 before entering the reheater. The main heat exchanger, mounted in the ram air channel, is passed before the evaporator cools the air. In case the saturation temperature is exceeded, a water separator extracts the condensate and leads it to the water injector located in the the ram air channel upstream the vapour cycle condenser. Before the conditioned fresh air is expanded in the turbine, it is reheated in the reheater. The discharged fresh air meets the recirculated cabin air in the mixing unit downstream the pack. The ram air channel functions as heat sink and is feed with air from the outside. During ground operations, the air flow is provided by a fan
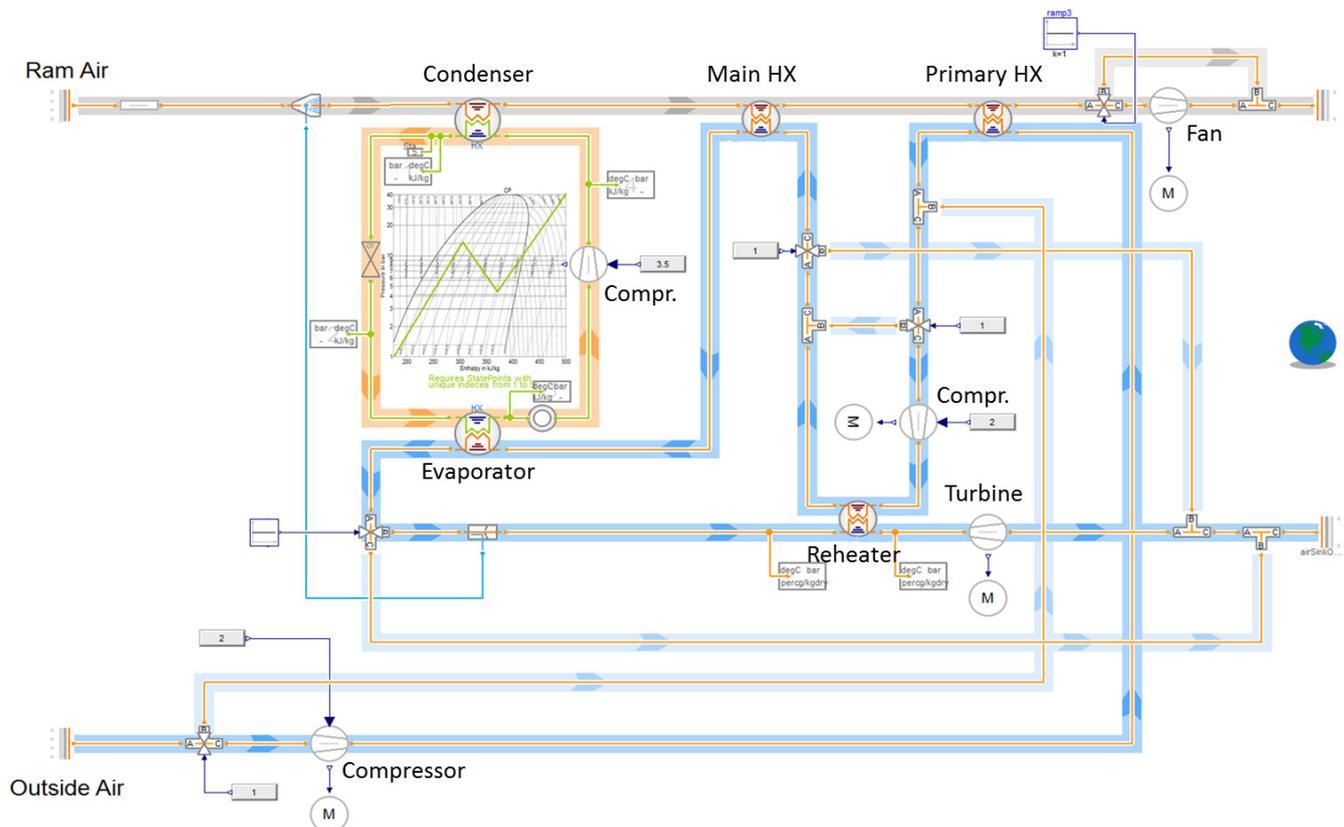

Figure 6: Modelica diagram of electric driven vapour cycle pack architecture using the HEXHEX library.

located near the ram air outlet. Contrary to a bleed air driven pack which is autonomously driven, all turbomachines within the eVCP are electrically powered.

The here presented modelling concept allows to build up such a system model as shown in Figure 6 within one day including first working simulations and a control concept for its many bypasses. Beside reasonable values for operational settings, i.e. compression/expansion ratios, flow resistances, and boundary conditions, no further settings concerning initialization have to be defined. Variants of such an architecture can then be generated much faster.

Looking at the statistics of the translated model, the advantage of the proposed concept becomes apparent: There is no large non-linear equation system at all. From the components there remain only 19 smaller non-linear equation systems with a single iteration variable.

In total the system contains 40 continuous time states. 9 of them represent mass-flows according to the proposed dynamics. 11 states originate from in-built controllers and the remaining states belong to the components. Most of them originate from quite detailed models of the evaporator and condenser within the vapour cycle.

The addition of the 9 state variables for the mass-flow by this concept do not cause any oscillation in the system. The impact on simulation performance has shown not to be detrimental.

The architecture of Figure 6 shows a number of junction valves that can open or close various different bypasses. Figure 7 displays the opening of a generic bypass valve with the resulting mass-flow dynamics and the corresponding inertial pressure. As outlined before, the inertial pressure is zero as long as the mass-flow rates remain constant. The opening of the valve then causes a pressure change and consequently the inertial pressure becomes non-zero. The affected mass flow rates then change until the new equilibrium point is reached. Evidently, a well-natured transient behavior can be observed that can be well-handled by numerical ODE solvers, especially by those suited for stiff-systems.

Although the valve is fully opened within a single second, the maximum peak for the absolute value of the inertial pressure remains around 100 Pascal. Being around one per mille of the fluid's static pressure, its effect on the thermodynamic state is negligible, which goes well in line with the analysis of Section 5.

If desired, the mass flow rate can also be stipulated by the addition of corresponding boundary conditions to the system. However, when doing so, these boundary conditions of mass flow rates must be differentiable with respect to time. Otherwise, the inertial pressure cannot be properly determined. Also one must take care that such boundary conditions do not conflict each other and cause the system to be overdetermined.
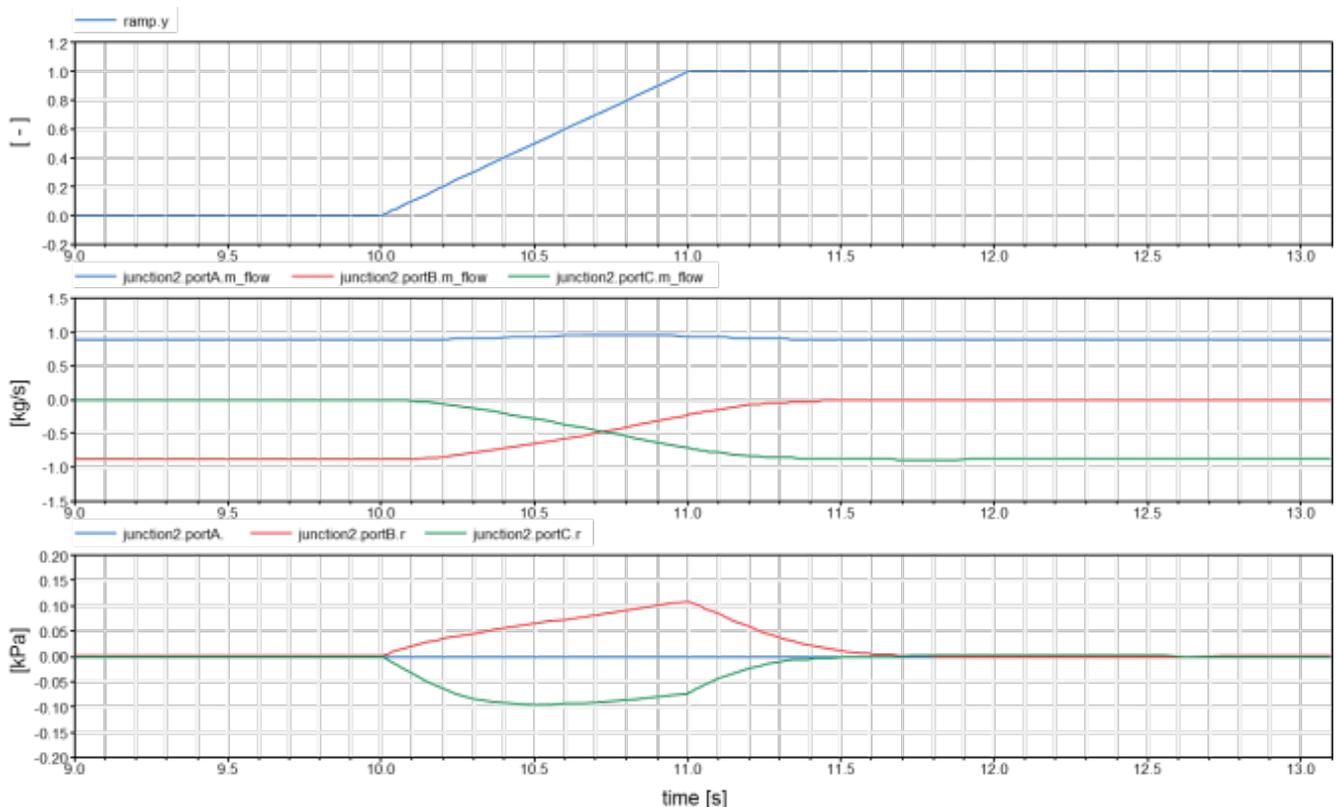

Figure 7: Mass flow dynamics for an opening valve

8 Final Remarks

8.1 Positioning of the approach

How does the approach of HEXHEX compare to the most common other approaches? Within the Modelica community, we find two main approaches.

The first approach is a purely algebraic quasi-static modeling 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 (Franke, 2009A) are modeled in this style.

The second approach is a realization of the finite volume method (or similar) where any flow represents the flow between two volume models. This avoids any larger non-linear equation system but at the price of creating many state-variables because of its many volumes. A classic example is (El Hefni, 2014).

HEXHEX is right in the middle, looking for the sweet spot. It avoids completely any large non-linear equation system and 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-flows are used as state variables and not the full thermodynamic state of a volume. Furthermore, the mass-flow can be shared for all serially connected flows. This enables index-reduction to further reduce the set of state variables.

A justified point of critique is that with the inertial pressure also fast dynamics enter the system asking for stiff-system solvers and that the mass-flow states for many modelers rather represent so-called artificial states (Zimmer, 2013). The answer to this critique is given in (Zimmer, 2013) and (Zimmer, 2014) that propose better ways how numerical ODE solvers shall deal with artificial states. Unfortunately, no Modelica tool currently supports such an approach. We hope that libraries like HEXHEX will further raise the value and importance of such a solution since after all a general support would also benefit the works of many others such as (Jorissen, 2018). Yet even without this support, the usability and performance of HEXHEX is not really impaired.

8.2 Initialization

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. With HEXHEX such an approach is absolutely feasible. For many cases, an initialization at rest with all mass-flows 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.

8.3 Adaption to Bidirectional Fluid flows

For our application field, a unidirectional solution is sufficient. Yet, can the approach be extended for bi-directional models? In principal yes, either one has to duplicate the equations for both flow directions or one takes use of the stream connector (Franke et al, 2009B). Both approaches will (in a different way) pollute the modeling equations to some degree but represent workable solutions.

In contrast to the standard fluid library, the built-in regularization scheme of the stream connector is not needed any longer since the mass-flow is a state variable.

A disadvantage of a generic bidirectional approach is that more loops will occur and hence more volume elements (or other means) will be needed to cut these loops. This is simply because there are statistically more loops in an undirected graph than in a directed graph of the same density. Hence while workable, the approach is losing some of its appeal for bidirectional flows.

8.4 Overall conclusion and future work

The dynamics of non-static fluid flows can very well be expressed using DAEs in Modelica, even for complex networks. If we regard the fluid streams as quasi-incompressible, then index-reduction enables to extract a small set of state variables for the total system.

Exploiting this fact enables a very robust modeling of complex fluid systems that frees the end-user from having to care about large non-linear equation systems and initialization.

The outlined approached has been implemented and thoroughly tested by a number of complex environmental control system architectures for aircraft and artificial testing examples. Development time of these architecture models could be drastically reduced compared to prior implementations.

Yet, this approach has not reached full maturity yet und the corresponding library is still subject to heavy development. Future work will hence more thoroughly compare this approach to others in terms of validity and performance. Furthermore, the release of an open interface standard shall be considered as soon as the development.

9 Acknowledgements

References

Daniel Bender, (2017) Integration of exergy analysis into model-based design and evaluation of aircraft environmental control systems, Energy, 137, p. 739-751, 2017

Dag Brück, Hilding Elmqvist, Hans Olsson, Dymola for Multi-engineering Modeling and Simulation. In: Proc. of the 2nd International Modelica Conference, Oberpfaffenhofen, Germany

Francesco Casella and Alberto Leva (2005), Object-Oriented Modelling & Simulation of Power Plants with Modelica, Proceedings of the 44th IEEE Conference on Decision and Control, andthe European Control Conference, Seville, Spain

Francesco Casella, Martin Otter, Katrin Proelss, Christoph Richter, Hubertus Tummescheit, (2006) 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.

Francesco Casella and Sielemann, Michael und Savoldelli, Luca (2011) Steady-state initialization of object-oriented thermo-fluid models for homotopy methods. In: Proceedings of 8th International Modelica Conference. 8th International Modelica Conference, 20.-22. Mrz 2011, Dresden.

Francois E. Cellier, Ernesto Kofman (2006). Continuous System Simulation, Springer Verlag, New Yorg, 643p.

Baligh El Hefni and Daniel Bouskela (2014), Dynamic modelling of a Condenser with the Thermo SysPro Library, In: Proceedings of the 10th International ModelicaConference, March 10-12, 2014, Lund, Sweden

Rüdiger Franke et. al. (2009A) Standardization of Thermo-Fluid Modeling in Modelica.Fluid. Proceedings 7th Modelica Conference, Como, Italy, Sep. 20-22, 2009

Rüdiger Franke et. al. (2009B) Stream Connectors – An Extension of Modelica for Device-Oriented Modeling of Convective Transport Phenomena. Proceedings 7th Modelica Conference, Como, Italy, Sep. 20-22, 2009

Steffen Golle et. Al., (2016), Betriebsphasenabhängig steuerbare Flugzeugklimaanlage und Verfahren zum Betreiben einer derartigen Flugzeugklimaanlage, Patent, DE102015207447A1, 27.10.2016

F. Jorissen, M. Wetter & L. Helsen (2018). Simplifications for hydronic system models in Modelica. In: Journal of Building Performance Simulation 12.01.2018

Longwell, P.A. (1966). Mechanics of Fluid Flow. McGraw-Hill.

Alexander Pollok, and Andreas Klöckner (2016) The use of Ockham's Razor in object-oriented modeling. In: Proceedings of the 7th International Workshop on Equation-Based Object-Oriented Modeling Languages and Tools, Milano, Italy. DOI: 10.1145/2904081.2904086

Daniel Schlabe and Lienig, Jens (2014) Model-Based Thermal Management Functions for Aircraft Systems. In: SAE 2014 Aerospace Systems and Technology Conference. SAE 2014 Aerospace Systems and Technology Conference, 23.-25. Sept 2014, Cincinnati, Ohio, USA. ISSN 01487191

Michael Sielemann, T. Giese, B. Oehler, M. Gräble (2011) Optimization of an Unconventional Environmental Control System Architecture. In: SAE International Journal of Aerospace, 4 (2), Seiten 1263-1275. SAE 2011 AeroTech Congress and Exhibition, 18.-21. Oct. 2011, Toulouse, France.

Michael Sielemann (2012) High-Speed Compressible Flow and Gas Dynamics, In: Proceedings of the 9th International Modelica Conference, 3.-5. Sept. 2012, Munich, Germany

Erich Truckenbrodt, Lehrbuch der angewandten Fluidmechanik (Kapitel zu instantionäre Fadenströmung dichtbeständiger Fluide) Springer Verlag, 1983

Michael Wetter, Marco Bonvini and Thierry S. Nouidui (2016) Equation-based languages - A new paradigm for building energy modeling, simulation and optimization. Energy and Buildings, 117(1), p. 290-300, 2016

Dirk Zimmer, (2014), Handling infinitely fast processes in continuous system modeling, Proceedings of the 6th International Workshop on Equation-Based Object-Oriented Modeling Languages and Tools, Berlin, Germany

Dirk Zimmer, (2013), 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