Modeling and Numerical Solution of Thermo-Hydraulic Networks

Hauke Hirsch, Andreas Nicolai and the SIM-VICUS Developers[*]

October 10, 2020

Contents

[*] Institut für Bauklimatik, Technische Universität Dresden, `andreas.nicolai@tu-dresden.de`

1 Configuration of Thermo-Hydraulic Networks

- Network is composed of nodes and elements
- Pressure drop is formulated only along elements
- elements can be pipes, pumps, heat exchanges, valves etc.
- Each *element* has exactly one inlet (left) and one outlet (right) in the schematics
- Nodes connect elements and hold a pressure and temperature state (no flow, no pressure drop)
- Mass flux through an element is conserved, i.e. the same at either side.

1.1 Model Generation Process / Transformation of GIS based Topologies

The transformation process of GIS-based network data to model data is a three-step process:

1. Determine geometry of all active components ("nodes" in GIS terminology), e.g. houses, pumps, storage systems
2. Semi-automatic routing of pipes ("edges" in GIS terminology); Based on some predefined pipe layout the remaining pipe connections are determined using a "shortest path" algorithm. In the result all active component nodes are connected with the network. Afterwards all redundant edges are merged, when they have the same property, yet keeping their total length.
3. Idealization and creation of the Thermo-Hydraulic network. Each of the "nodes" and "edges" from the GIS import process are transformed into "nodes" and "elements" in terms of the thermo-hydraulic network model. Hereby all elements of the network are expressed in terms of elements with exactly two flow connections and a finite (though maybe extremely small) pressure drop across the element. All elements are connected in nodes.

TODO : Hauke, add figs from presentation and add specific example of step 3

1.2 Abstract Governing Equations

1.2.1 Elements

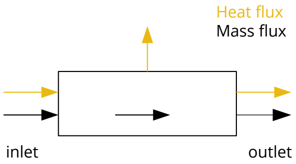

Figure 1: Schematics of an abstract flow element and sign definition of fluxes across model boundary

For each network element the following equations are defined. The system equation for each element (1) defines the relationship between mass flow across the element and the pressures at inlet (in) and outlet (out). The mass flux direction and sign definition is shown in Figure 1. For a typical pipe, the pressure drop would be formulated as $\Delta p = p_{in} - p_{out}$.

$$0 = f\left(\dot{m}, p_{in}, p_{out}\right) \tag{1}$$

$$0 = \dot{Q}_{in} - \dot{Q}_{out} - \dot{Q}_{loss} \tag{2}$$

Equation (2) is the energy balance across the flow element, which typically involves the energy quantities/enthalpies transported with the mass flow across the inlet and outlet. Optionally, there may be other energy fluxes between the flow element and its environment, for example through heat conduction over its surface, denoted by the $\dot{Q}_{loss}$ term in the equation.

Note that the actual pressure drop equation may be a function of temperature as well. Since a flow element is connected to two nodes, it is hence also in contact with two node temperatures. With respect to numerical stability of the solution procedure, the temperature at the inflow node should be used when evaluating the media properties in the element's system equation, a method called *upwinding*.

1.2.2 Nodes

For each node the following conservation equations are defined:

$$\sum \dot{m}_i = 0 \tag{3}$$

$$\sum \dot{Q}_i = 0 \tag{4}$$

Also, the pressure p_i at some node i is equal to all inlet/outlet pressures of the connected flow elements (no pressure drop/change across a node). This equation is not explicitely formulation but will be used to eliminate variables in the solution process.

1.2.3 Closing/complementory equations

The system itself needs additional constraints to be solved, usually a reference pressure defined at *exactly one node*, for example the node connected to an inlet of a pump. In the solution process it might then possible to eliminate this pressure from the connected flow element's equations. However, a more pragmatic approach from point of view of implementing such a network solver is to define a pressure "boundary condition" as discussed below.

1.3 Temperature/Pressure Controlled Flow Network Elements

Some flow network elements may be controlled, for example pumps and valves. The control equation may use pressures or temperatures of the flow network itself, thus creating a tightly coupled problem. With respect to engineering requirements, often these direct feedback formulations can be softened by introducing typical time scales of reaction. For example, if a signal for pump activation is received (e.g. temperature trigger point passed), it may be possible to introduce a delay of some minutes until the pump has reached the required speed. The time delay must be selected such that the network itself is solved sufficiently correctly.

Suppose we name a control variable $r\left(p,T\right)$ and denote by the parameters p and T some dependence of flow network pressures/temperatures. Then the system equation for a pump might read:

$$0 = f\left(\dot{m}, p_{in}, p_{out}, r\left(p,T\right)\right)$$

and the resulting network will be fully coupled. If, however, we introduce a time variation on the control variable, we obtain a differential equation:

$$\frac{dr}{dt} = \tau\left(r_{set}\left(p,T\right) - r\right)$$

with τ being some time constant parameter. $r_{set}\left(p,T\right)$ will be the target control signal obtained from the control algorithm equations that use the current network's state. $r\left(t\right)$ will be a transient quantity that more or less slowly approaches this set point value. With respect to the steady-state solution of the flow network, the control variable r is now again a constant, which simplifies solution of the network significantly.

2 Solution procedure

We will derive the general solution procedure for all kinds of thermo-hydraulic networks by starting with a simple example.

2.1 Example 1: Pump with Pipes

The first example consists of three nodes and three flow elements (Fig. 2). This is an adiabatic model without energy balances. Already with this simple model we can identify several problems associated with hydraulic networks.

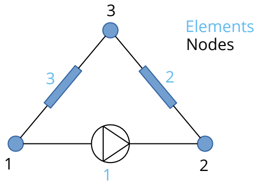

Figure 2: Example 1: Simple flow network with a pump and two pipes

For each of the flow elements we have the system equation. When formulating the equations, we already substitute p_{in} and p_{out} with the pressure at the connected node.

$$
\begin{aligned}
0 &= f_1\left(\dot{m}_1, p_1, p_2\right) \\
0 &= f_2\left(\dot{m}_2, p_2, p_3\right) \\
0 &= f_3\left(\dot{m}_3, p_3, p_1\right)
\end{aligned}
$$

In addition, we have the nodal equations:

$$
\begin{aligned}
0 &= n_1\left(\dot{m}_1, \dot{m}_3\right) = \dot{m}_3 - \dot{m}_1 \\
0 &= n_2\left(\dot{m}_1, \dot{m}_2\right) = \dot{m}_1 - \dot{m}_2 \\
0 &= n_3\left(\dot{m}_2, \dot{m}_3\right) = \dot{m}_2 - \dot{m}_3
\end{aligned}
$$

We could now formulate the equation system in a generic way as $\boldsymbol{F}\left(\boldsymbol{p}, \dot{\boldsymbol{m}}\right)$ where $\boldsymbol{p}$ and $\dot{\boldsymbol{m}}$ are vectors of the nodal pressures and flows across flow elements. Thus, we have 6 unknowns and the 6 equations above and might assume that this is an easily solvable problem (though non-linear).

However, in this small example we have formulated a linear combination of the same related. You can consider this by substituting the flow equations: $\dot{m}_3 = \dot{m}_1$ and $\dot{m}_2 = \dot{m}_1$ from the first two equations, inserted into the last equation we get $\dot{m}_1 = \dot{m}_1$ which obviously doesn't give any new information.

After eliminating the mass fluxes $\dot{m}_3$ and $\dot{m}_2$ this leaves us with 4 unknowns: $p_1, p_2, p_3, \dot{m}$ (we use $\dot{m} = \dot{m}_1$ for simplicity here), but only 3 flow element system equations.

Now the same process of variable elimination can be done with pressures as well. Suppose all flow elements have a rather trivial system equation, that relates the inlet and outlet pressures in a convenient way. Say, we have $p_1 = p_2 + g_1$, $p_2 = p_3 + g_2$ and $p_3 = p_1 + g_3$. We might then eliminate pressures by inserting equations into each other and obtain: $p_1 = p_1 + g_1 + g_2 + g_3$ and consequently $0 = g_1 + g_2 + g_3$. The parameters g are likely functions of the mass flux $\dot{m}$, so solving the equation will give us a solution for $\dot{m}$. However, the corresponding pressures will be arbitrary.

Interestingly, if one were to solve the equation system above using some generic solution method, one *might* even get a result, but that would be ambigous (and depend on some initial guess value of one of the pressures). For example, the result could be $p_1 = 0\,\text{Pa}$, $p_2 = 100\,\text{Pa}$, $p_3 = 70\,\text{Pa}$ or $p_1 = 1000\,\text{Pa}$, $p_2 = 1100\,\text{Pa}$, $p_3 = 1070\,\text{Pa}$ or any other pressure level.

If we go back to the incomplete equation system problem, we just need to provide the network with additional information to add another constraint, i.e. by specifying a nodal pressure somewhere. As we've seen above in the elimination of pressures example, *we must not specify a mass flux* as additional constraint, since the mass flux itself is already sufficiently determined.

Specification of a pressure can be done easiest by adding another equation, for example:

$$p_1 = 0\,\text{Pa}$$

With that additional constraint the equation system can be solved together.

2.2 Example 2: Splitter and Mixer

Consider the adjusted example in Figure 3. Here, the nodes 2 and 3 are connected by two parallel pipes with potentially different diameters, length and friction coefficients and thus effictively different $\Delta p\left(\dot{m}\right)$ relations.

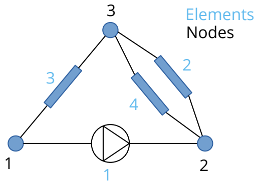

Figure 3: Example 2: Flow network with a mixer and splitter

Following the procedure from example 1, we can formulate again our 4 flow element equations. Also, we have the nodal equations, where node 2 is now a *splitter node* and node 3 becomes a *mixer node*.

$$\begin{aligned}
0 &= n_1\left(\dot{m}_1, \dot{m}_3\right) = \dot{m}_3 - \dot{m}_1 \\
0 &= n_2\left(\dot{m}_1, \dot{m}_2, \dot{m}_4\right) = \dot{m}_1 - \dot{m}_2 - \dot{m}_4 \\
0 &= n_3\left(\dot{m}_2, \dot{m}_3, \dot{m}_4\right) = \dot{m}_2 + \dot{m}_4 - \dot{m}_3
\end{aligned}$$

If we proceed to elimate mass fluxes as in example 1, we can simply sum up $\dot{m}_2$ and $\dot{m}_3$ and end up with a formulation just as in example 1. In terms of solving the problem there is nothing different to example 1.

2.3 Trivial and Non-Existing Solutions

Consider the problem of having a pipe instead of the pump as flow element 1, or when the pump is turned off. In the latter case the pump will then become a simple friction-type flow element just like a heat exchanger or pipe. Then, there is exactly one trivial solution to the problem: zero mass flux. The pressures in the system are all the same, fixed to the given pressure at node 1. This problem is solvable.

For systems with active pumps there can only be a solution if there is at least one flow element in the network that has a positive and *monotonically increasing* $\Delta p\left(\dot{m}\right)$ system equation (i.e. a pipe or other element with flow resistance that increases with flow rate). Otherwise the problem cannot be solved.

2.4 Generic Solution and Equation System Setup

Manual elimination of variables to obtain a reduced set of solution variables is normally not an option. However, we could be tempted just to formulate the nodal mass balances, insert expressions for the mass fluxes and hereby replace mass fluxes with functions of nodal pressures and simply solve for the pressures. In the example 1 this would lead to the following equations:

$$\boldsymbol{G}\left(\boldsymbol{p}\right) = 0 = \begin{cases} 0 & = \dot{m}_3\left(p_1, p_3\right) - \dot{m}_1\left(p_1, p_2\right) \quad + p_1 - p_{ref} \\ 0 & = \dot{m}_1\left(p_1, p_2\right) - \dot{m}_2\left(p_2, p_3\right) \\ 0 & = \dot{m}_2\left(p_2, p_3\right) - \dot{m}_3\left(p_1, p_3\right) \end{cases}$$

Note that we simply added the additional constraint with the pressure at node 1 to the first equation.

2.4.1 Solving for nodal pressures alone

The composition of such an equation system like in example 1 is rather straight-forward. The number of nodes in the network determine the number of unknowns and equations. For each node the mass fluxes through the connected flow elements are summed up using the inlet/outlet sign definition shown in Figure 1.

We could now solve the equation system for the unknowns $\boldsymbol{p}$. Since the entire system is non-linear, a Newton iteration seems in order, which leads to the following Newton iteration scheme (with m being the iteration level):

$$\left.\frac{\partial G}{\partial p}\right|_m \Delta p_{m+1} = -G\left(p_m\right)$$

However, we face a minor problem because of our implicit formulation of the flow element equations, that we cannot easily re-arranged to be expressions for mass fluxes $\dot{m}$. This makes the evaluation of the non-linear equation system function $\boldsymbol{G}\left(\boldsymbol{p_m}\right)$ expensive (where $\boldsymbol{p_m}$ is the vector with the current estimate of the nodal pressures), since we cannot just take the estimated pressures and evaluate each mass flux to obtain the residuals. Instead, we would need to iteratively determine each flow element system function until we find the $\dot{m}$ for the given inlet and outlet pressures. In the example 1 this would mean that we take estimates for p_1 and p_3 and solve $0 = f_1\left(\dot{m}, p_1, p_3\right)$ with a Newton method

$$\left.\frac{df_1}{d\dot{m}}\right|_n \Delta \dot{m}_{n+1} = -f_1\left(\dot{m}_n\right)$$

until we have found the matching mass flux $\dot{m}$. In this Newton method n is the iteration level and p_1 and p_3 are given constants.

Having an iterative calculation procedure inside another iteration method is generally not a good idea, even though this is probably not as bad as it looks, since:

- it is a scalar function and thus a Newton method (or even Newton method stabilized by some secand method) is fairly fast and robust (i.e. no equation system solving involved)
- since we solve this function repeatedly for small changing $\dot{m}$ and p the initial guess will very often be very close to the solution
- selecting convergence tolerances sufficiently tight is not a problem
- implementation can be well encapsulated so that an evaluation of $\dot{m}\left(p_1, p_2\right)$ is not problematic
- the same mass fluxes are used in several equations (at least twice), so that caching these variables speeds up calculation of $\boldsymbol{G}\left(\boldsymbol{p_m}\right)$ somewhat

However, there is the issue of computing the Jacobian matrix $\frac{\partial G}{\partial p}$ of the general solution method which ultimately requires computation of derivatives $d\dot{m}/dp$ for the various mass flux equations and pressures. Obtaining analytical expressions for these derivatives is difficult (or nearly impossible in some cases).

The most simply and straight-forward approach is to generate a lookup table of the relation $\dot{m}\left(p\right)$ and use linear (or any other form of) interpolation to approximate the relationship. This can be implemented in a very generic way, and the small errors from the approximation won't be critical, since only the Jacobian is affected. For some flow elements like the pump there is a parameterized $\Delta p\left(\dot{m}\right)$ curve which can be inverted and yields the required derivative right away.

For models that have a large temperature dependence (where $\dot{m}\left(p\right)$ might change drastically depending on temperature), a single lookup table might not be sufficient. Either, a correction factor is used to adjust the results of some reference curve $\dot{m}\left(p, T_{ref}\right)$ somewhat to account for the temperature effect. Or, if that still gives too much of an error, we can still approximate the derivative with a small trick.

Suppose we had the partial derivatives $\frac{\partial f}{\partial p}$ and $\frac{\partial f}{\partial \dot{m}}$ and we can somehow invert the second gradient to $\frac{\partial \dot{m}}{\partial f}$ we get our required derivative:

$$\frac{\partial \dot{m}}{\partial p} = \frac{\partial \dot{m}}{\partial f}\frac{\partial f}{\partial p}$$

Since both variables p and $\dot{m}$ are related, the inversion of the gradient is done with:

$$\frac{\partial \dot{m}}{\partial f} = -\frac{1}{\frac{\partial f}{\partial \dot{m}}}$$

@Anne: is there somewhere a mathematical proof why this works? For scalar functions it is $\frac{dy}{dx} = \frac{1}{\frac{dx}{dy}}$ but for partial derivatives we need the minus sign. This has something to do with the difference between gradients and niveau-line slopes, but why does inserting the minus gives correct results?

Lets illustrate this with an example. Suppose our flow element system function looks like

$$f\left(p_1, p_2, \dot{m}\right) = 0 = p_1 - p_2 - a\dot{m}^2$$

where a is some friction coefficient and $a\dot{m}$ corresponds to the pressure loss that increases quadratically with the flow rate. This function can easily be rearranged to:

$$\dot{m}\left(p_1\right) = \sqrt{\frac{p_1 - p_2}{a}}$$

and the derivative we look for is

$$\frac{\partial \dot{m}}{\partial p_1} = \frac{1}{2}\left(\frac{p_1 - p_2}{a}\right)^{-\frac{1}{2}}\frac{1}{a} = \frac{1}{2a\sqrt{\frac{p_1-p_2}{a}}} = \frac{1}{2a\dot{m}}$$

Now let try to construct this without rearranging the equation. We first get the partial derivatives

$$\begin{aligned}\frac{\partial f}{\partial p_1} &= 1\\ \frac{\partial f}{\partial \dot{m}} &= -2a\dot{m}\end{aligned}$$

invert the second partial derivative

$$\frac{\partial \dot{m}}{\partial f} = \frac{1}{2a\dot{m}}$$

and apply the chain rule to get:

$$\frac{\partial \dot{m}}{\partial f}\frac{\partial f}{\partial p_1} = \frac{\partial \dot{m}}{\partial p_1} = \frac{1}{2a\dot{m}}$$

This even works with more complex functions, like $f\left(x, y\right) = y/\sin x + 2$:

$$\begin{aligned}y &= -2\sin x\\ -2 &= \frac{y}{\sin x}\\ \frac{dy}{dx} &= -2\cos x = \frac{y}{\sin x}\cos x\end{aligned}$$

and with partial derivatives:

$$\begin{aligned}\frac{\partial f}{\partial x} &= -\frac{y}{\sin^2 x}\cos x\\ \frac{\partial f}{\partial y} &= \frac{1}{\sin x}\\ \frac{\partial y}{\partial f} &= -\sin x\\ \frac{\partial y}{\partial x} &= \frac{\partial y}{\partial f}\frac{\partial f}{\partial x} = -\sin x \cdot -\frac{y}{\sin^2 x}\cos x\\ &= \frac{y}{\sin x}\cos x\end{aligned}$$

In a generic implementation, i.e. where there's no knowledge of the function $f\left(p_1, p_2, \dot{m}\right)$, the derivatives $\frac{\partial f}{\partial p}$ and $\frac{\partial f}{\partial \dot{m}}$ can be obtained using finite difference quotients.

2.4.2 Solving for mass fluxes together

Just as with the previously described variant, we could simply use the mass fluxes across the flow elements as solution variables and solve $\boldsymbol{G}(\dot{\boldsymbol{m}})$. However, this would lead to several difficulties:

- we have to evaluate the implicit flow element system functions with an estimate of the mass flux only, which leads to the same problems as described earlier for the pressures
- the evaluation of the flow element function would yield pressure differences (pressure drops) but no actual pressure values
- the equations for nodal mass balances do not account for computing the actual nodal pressure values and computing residuals for each equation

In fact, in order to solve the system directly for mass flows, which is possible, requires adding additional terms to the nodal mass balance equations which depend on the networks topography and make the solution less generic.

2.4.3 Solving for nodal pressures and element mass fluxes together

An alternative approach allows a much more generic way of evaluation the system function and does not require specific flow element based solution functions. The problems addressed above resulted from the fact that only pressures were known and mass fluxes needed to be computed and vice versa.

The flow element system equations would be best evaluated if *nodal pressures and mass fluxes* were available. In example 1 this would mean that we define a vector of solution variables $y = \{p_1, p_2, p_3, \dot{m}_1, \dot{m}_2, \dot{m}_3\}$ and the following 6 equations:

$$\boldsymbol{G}(\boldsymbol{y}) = 0 = \begin{cases} 0 & = & \dot{m}_3(p_1,p_3) - \dot{m}_1(p_1,p_2) & +p_1 - p_{ref} \\ 0 & = & \dot{m}_1(p_1,p_2) - \dot{m}_2(p_2,p_3) & \\ 0 & = & \dot{m}_2(p_2,p_3) - \dot{m}_3(p_1,p_3) & \\ 0 & = & f_1(\dot{m}_1,p_1,p_2) & \\ 0 & = & f_2(\dot{m}_2,p_2,p_3) & \\ 0 & = & f_3(\dot{m}_3,p_3,p_1) & \end{cases}$$

The system size (and most remarkable Jacobian matrix size) is doubled, yet now we have all variables available to evaluate the system function and its derivatives easily in a very generic way.

3 Flow Element Equations

The sections below detail the flow equations for the various elements in the network.

3.1 Fluid Properties

We use the following constant fluid properties

thermal conductivity	λ	W/mK
density	ρ	kg/m^3
spec. heat capacity	c_p	$J/kg\,K$

and the temperature dependent properties

kinematic viscosity	$\nu = \nu(T)$	m^2/s

as well as the Prandtl number

$$Pr(T) = \frac{\nu(T)\rho c_p}{\lambda}$$

3.2 Pipes

3.2.1 General Hydraulic Pipe Flow Equations

The pressure drop along a pipe depends on the physical fluid properties and the parameters

length	l	m
inner diameter	d	m
roughness	k	m

The velocity is

$$v = \frac{\dot{m}}{\rho \frac{\pi}{4} d^2}$$

For laminar flows ($Re < 2300$)

$$\lambda = \frac{64}{Re}$$

For turbulent flows ($Re > 1e4$) we use the equation from Colebrook

$$\lambda = \left[-2 log_{10} \left(\frac{2.51}{Re\sqrt{\lambda}} + \frac{k}{3,71d} \right) \right]^{-2}$$

for $2300 \leq Re \leq 1e4$ we use linear interpolation between the two equations. (QUELLE VDI Wärmeatlas)

Note that the Reynolds number is temperature dependent

$$Re(T) = \frac{vd}{\nu(T)}$$

The pressure loss coefficient ζ can be expressed by the pipe friction factor λ, where

$$\zeta(T) = \lambda(T) \frac{l}{d}$$

The pressure drop is

$$\Delta p(T) = \zeta(T) \frac{\rho}{2} v^2$$

3.2.2 Convection coefficient inside pipe

The convection coefficient from fluid to pipe wall depends on

length	l	m
diameter	d	m

For laminar flows ($Re < 2300$)

$$Nu_{lam} = \left(49.37 + \left(1.615 * \left(Re\, Pr \frac{d}{l} \right)^{\frac{1}{3}} - 0.7 \right)^3 \right)^{\frac{1}{3}}$$

for turbulent flows ($Re > 1e4$)

$$Nu_{turb} = \frac{(\zeta/8)\, Re\, Pr}{1 + 12.7\sqrt{\zeta/8}\, \left(Pr^{2/3} - 1\right)} \left(1 + \left(\frac{d}{l} \right)^{2/3} \right)$$

for $2300 \leq Re \leq 1e4$ we use linear interpolation

$$Nu_{int} = (1-a)\,Nu_{lam,Re=2300} + a\,Nu_{turb,Re=1e4}$$

where

$$a = \frac{Re - 2300}{1e4 - 2300}$$

(QUELLE VDI Wärmeatlas)

Note that $Re(T)$ and $Pr(T)$ are temperature dependent.

The convection coefficient inside a pipe is

$$h_i(T) = Nu(T)\frac{\lambda}{d}$$

3.2.3 Static adiabatic pipe

This model considers pressure loss according to 3.2.1

$$p_{in} - p_{out} = \Delta p(\dot{m}, T_{in})$$

but has no heat exchange to the surrounding, thus

$$\dot{Q}_{loss} = 0$$

3.2.4 Static pipe with heat exchange with the surroundings

This model considers pressure loss according to 3.2.1

$$p_{in} - p_{out} = \Delta p(\dot{m}, T_{in})$$

The heat loss depends on the fluid properties and the parameters

length	l	m
inner diameter	d	m
outer diameter	d_o	m
outer heat exchange coefficient	h_o	m
thermal conductivity pipe wall	λ_{pipe}	W/mK

as well as the external inputs

ambient temperature	T_{amb}	C

$$\dot{Q}_{loss} = \dot{m}c_{fl}\,(T_{in} - T_{amb})\left(1 - e^{\left(-\frac{U_p}{\dot{m}c_p}l\right)}\right)$$

where $U_R$is the length-related heat transfer coefficient

$$U_p = \frac{\pi}{\frac{1}{h_i d} + \frac{1}{h_o d_o} + \frac{1}{2\lambda_{pipe}} ln\left(\frac{d_o}{d}\right)}$$

where $h_i(T_{in})$ is the convection coefficient inside the pipe, which can be calculated according to 3.2.2.

3.2.5 Dynamic adiabatic pipe

This model considers pressure loss according to 3.2.1

$$p_{in} - p_{out} = \Delta p(T_{in})$$

but has no heat exchange to the surrounding, thus

$$\dot{Q}_{loss} = 0$$

The parameter is

spatial discretization along the pipe	dx	m

... still a general flow element, but now exporting/formulating time state variables to be integrated outside the actual network solution algorithm, pretty much like the control variables.

3.3 Pumps

3.4 Heat Exchangers

3.5 Soil Heat Collectors

3.6 Valves