Modeling of Thermo-Fluid Systems with Modelica

Hubertus Tummescheit, Modelon AB

Jonas Eborn, Modelon AB

With material from Hilding Elmqvist, Dynasim Martin Otter, DLR

Content

- Introduction
- Separation of Component and Medium
- Property models in Media: main concepts
- Components, control volumes and ports
- Balance equations
- Index reduction and state selection
- Numerical regularization
- Exercises

Separate Medium from Component

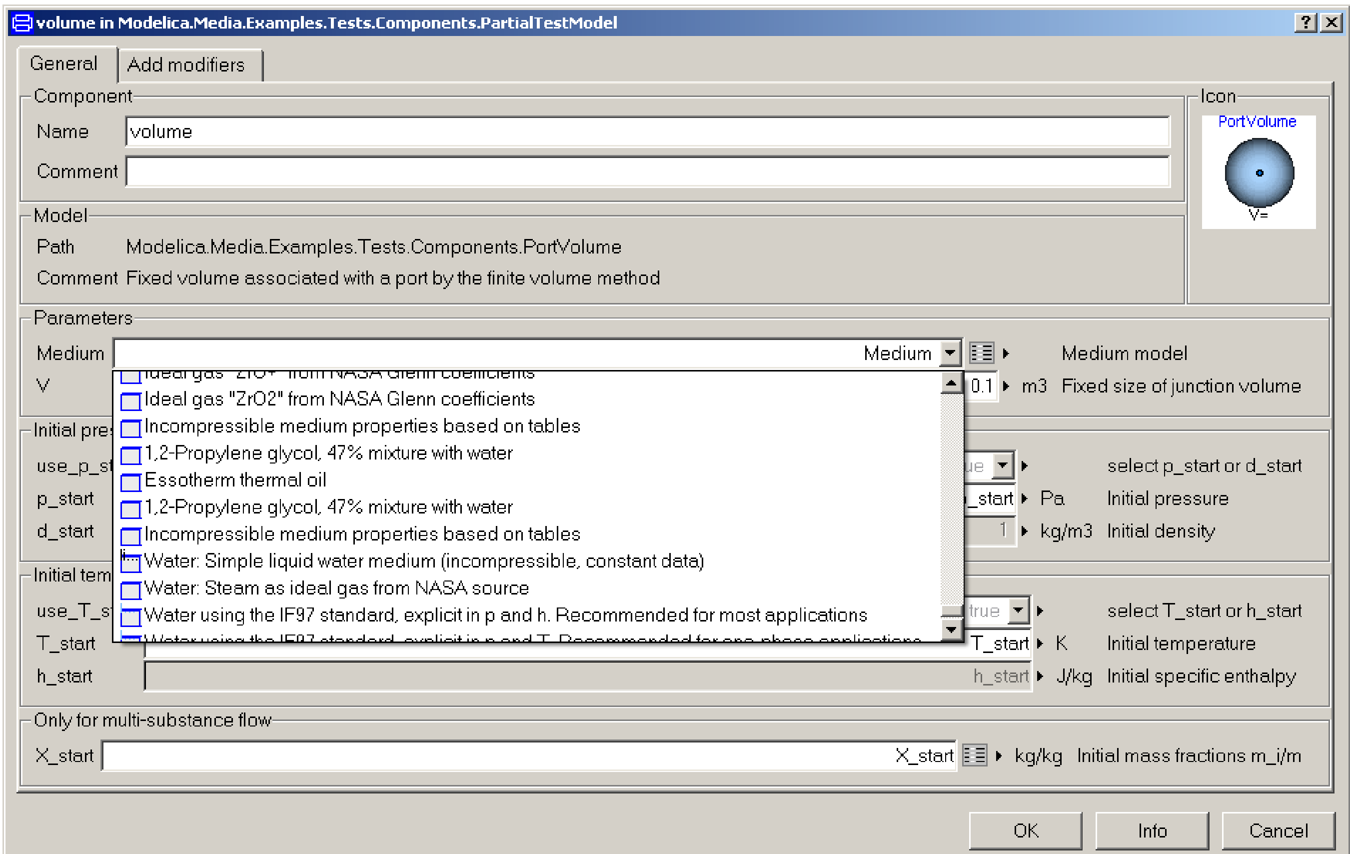
- Independent components for very different types of media (only constants or based on Helmholtz function)
- Introduce "Thermodynamic State" concept: minimal and *replaceable* set of variables needed to compute all properties
- Calls to functions take a state-record as input and are therefore identical for e.g. Ideal gas mixtures and Water

```
redeclare record extends ThermodynamicState "thermo state variables"
      AbsolutePressure p "Absolute pressure of medium";
      Temperature T "Temperature of medium";
      MassFraction X[nX] "Mass fractions (= (comp. mass)/total mass)";
end ThermodynamicState;
```

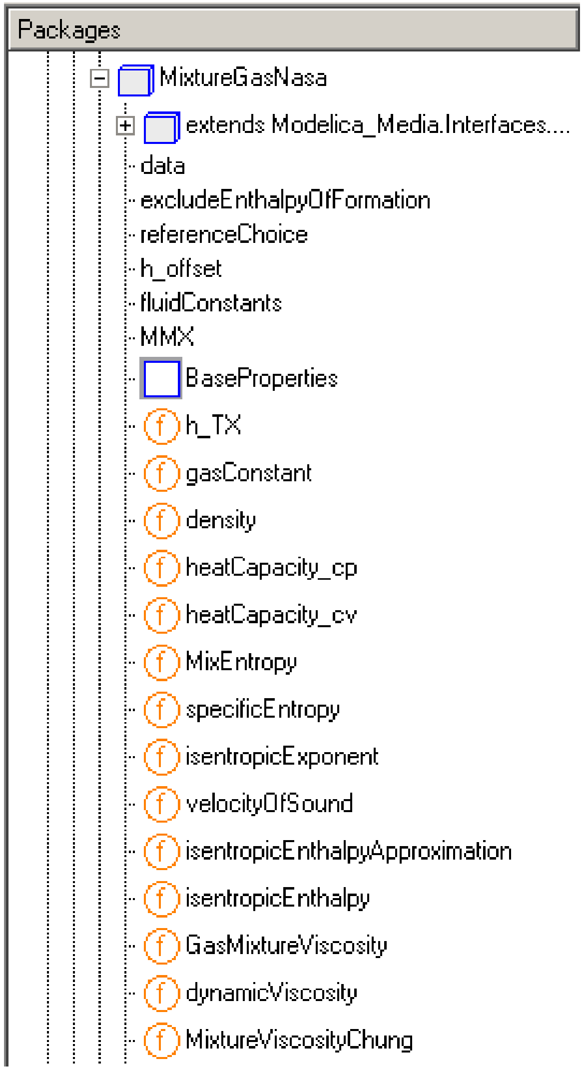
```
redeclare record extends ThermodynamicState "thermo state variables"
      AbsolutePressure p "Absolute pressure of medium";
      SpecificEnthalpy h "specific enthalpy of medium";
end ThermodynamicState;
```

Separate Medium from Component

- Medium models are “replaceable packages” in Modelica, selected via drop-down menus in Dymola



Media Models



Medium property definitions for

- Single and **multiple substances**
- Single and **multiple phases** (all phases have same speed)
- **free** selection of **independent variables** (p,T or p,h, T,ρ, or p, T, X_i)

Media mostly adapted from ThermoFluid:

- 1241 ideal gases (NASA source) and their mixtures
- IF97 water (high precision)
- Moist air
- Incompressible media defined by tabular data
- simple air, and water models (for machine cooling)

- Every medium model provides **3 equations** for **5 + nX variables**

Variable	Unit	Description
T	K	temperature
p	Pa	absolute pressure
d	kg/m^3	density
u	J/kg	specific internal energy
h	J/kg	specific enthalpy (h = u + p/d)
X_i[nX_i]	kg/kg	independent mass fractions m_i/m
X[nX]	kg/kg	All mass fractions m_i/m. X is defined in BaseProperties by: X = **if** reducedX **then** vector([X_i; 1-**sum**(X_i)]) **else** X_i

Two variables out of p, d, h, or u, as well as the mass fractions X_i are the **independent** variables and the medium model basically provides equations to compute the remaining variables, including the full mass fraction vector X

- A medium model might provide **optional functions**

Function call	Unit	Description
Medium.dynamicViscosity(medium.state)	Pa.s	dynamic viscosity
Medium.thermalConductivity(medium.state)	W/(m.K)	thermal conductivity
Medium.prandtlNumber(medium.state)	1	Prandtl number
Medium.specificEntropy(medium.state)	J/(kg.K)	specific entropy
Medium.heatCapacity_cp(medium.state)	J/(kg.K)	specific heat capacity at constant pressure
Medium.heatCapacity_cv(medium.state)	J/(kg.K)	specific heat capacity at constant density
Medium.isentropicExponent(medium.state)	1	isentropic exponent
Medium.isentropicEnthatlpy(pressure, medium.state)	J/kg	isentropic enthalpy
Medium.velocityOfSound(medium.state)	m/s	velocity of sound
Medium.isobaricExpansionCoefficient(medium.state)	1/K	isobaric expansion coefficient
Medium.isothermalCompressibility(medium.state)	1/Pa	isothermal compressibility
Medium.density_derp_h(medium.state)	kg/(m3.Pa)	derivative of density by pressure at constant enthalpy
Medium.density_derh_p(medium.state)	kg2/(m3.J)	derivative of density by enthalpy at constant pressure
Medium.density_derp_T(medium.state)	kg/(m3.Pa)	derivative of density by pressure at constant temperature
Medium.density_derT_p(medium.state)	kg/(m3.K)	derivative of density by temperature at constant pressure
Medium.density_derX(medium.state)	kg/m3	derivative of density by mass fraction
Medium.molarMass(medium.state)	kg/mol	molar mass

Medium package

```
package SimpleAir
  ...
  constant Integer nX = 0;
  model BaseProperties
    constant SpecificHeatCapacity cp_air=1005.45
    "Specific heat capacity of dry air";
    AbsolutePressure       p;
    Temperature            T;
    Density                d;
    SpecificInternalEnergy u;
    SpecificEnthalpy       h;
    MassFraction           X[nX];
    constant MolarMass MM_air=0.0289651159 "Molar mass";
    constant SpecificHeatCapacity R_air=Constants.R/MM_air
  equation
    d = p/(R_air*T);
    h = cp_air*T + h0;
    u = h – p/d;
    state.T = T;
    state.p = p;
  end BaseProperties;
  ...
end SimpleAir;
```

Separate Medium from Component, II

How should components be written that are independent of the medium (and its independent variables)?

......

```
package Medium = Modelica.Media.Interfaces.PartialMedium;
    Medium.BaseProperties medium;
  equation
  // mass balances
       der(M)  = port_a.m_flow + port_b.m_flow;
       der(MX) = port_a_mX_flow + port_b_mX_flow;
             M = V*medium.d;
            MX = M*medium.X;
  // Energy balance
             U = M*medium.u;
        der(U) = port_a.H_flow+port_b.H_flow;
```

Balance Equations and Media Models are decoupled

```
// Balance equations in volume for single substance:
     m = V*d;           // mass of fluid in volume
     U = m*u;           // internal energy in volume
der(m) = port.m_flow;    // mass balance
der(U) = port.H_flow;    // energy balance

// Equations in medium (independent of balance equations)
   d = f_d(p,T);
   h = f_h(p,T);
   u = h – p/d;
```

Assume m, U are selected as states, i.e., m, U are assumed to be known:

```
   u := U/m;
   d := m/V;
   res1 := d – f_d(p,T)       }
   res2 := u + p/d – f_h(p,T) }
```

As a result, non-linear equations have to be solved for p and T:

Use preferred states

the independent variables in media models are declared as preferred states:

AbsolutePressure p(stateSelect = StateSelect.prefer)
Tool will select p as state, if this is possible

```
d := f_d(p,T);
h := f_h(p,T);
u := h - p/d;
m := V*d;
U := m*u;
```

→

```
der(U) = der(m)*u + m*der(u)
der(m) = V*der(d)
der(u) = der(h) - der(p)/d + p/d^2*der(d)
der(d) = der(f_d,p)*der(p) + der(f_d,T)*der(T)
der(h) = der(f_h,p)*der(p) + der(f_h,T)*der(T)
```

der (f_d, p) is the partial derivative of f_d w.r.t. p

- index reduction is automatically applied by tool to rewrite the equations using p, T as states (linear system in **der**(p) and **der**(T))
- no non-linear systems of equations anymore
- different independent variables are possible (tool just performs different index reductions)

Incompressible Media

Same balance equations + special medium model:

- Equation stating that density is constant (d = d_const) or that density is a function of T, (d = d(T))
- Provided initial value for p or d is used as guess value (i.e. 1 initial equation and not 2 initial equations)

Automatic index reduction transforms differential equation for mass balance into algebraic equation:

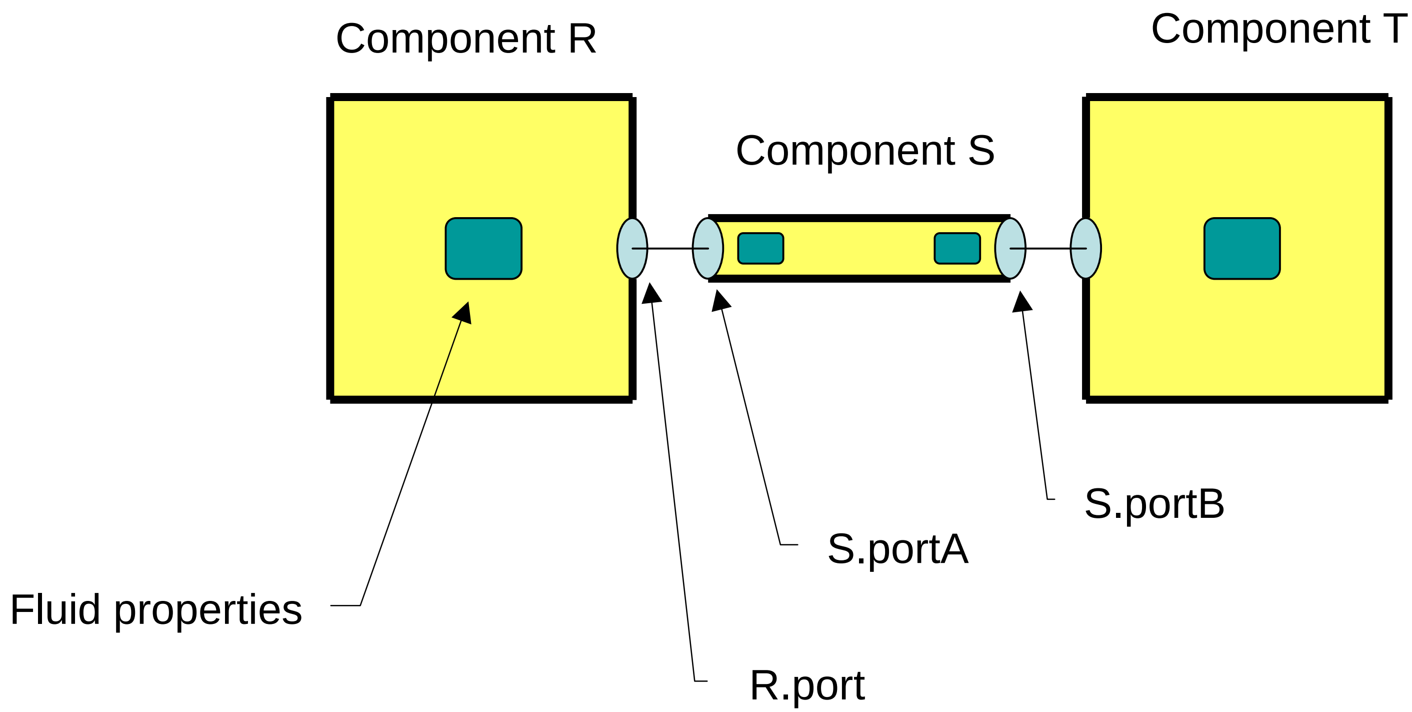
```
     m = V*d;
der(m) = port.m_flow;
```

```
der(m) = V*der(d);
```

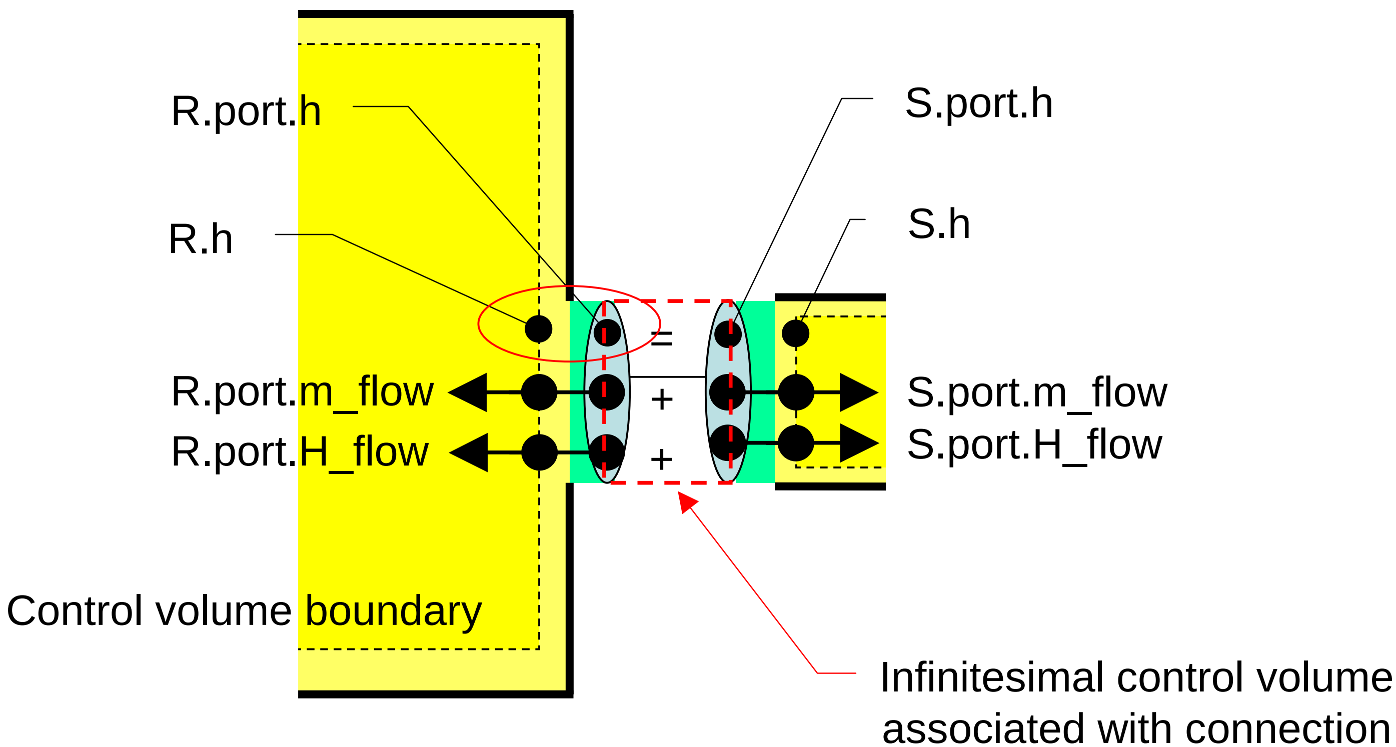
```
0 = port.m_flow;
```

Connectors and Reversible flow

- Compressible and non-compressible fluids
- Reversing flows
- Ideal mixing



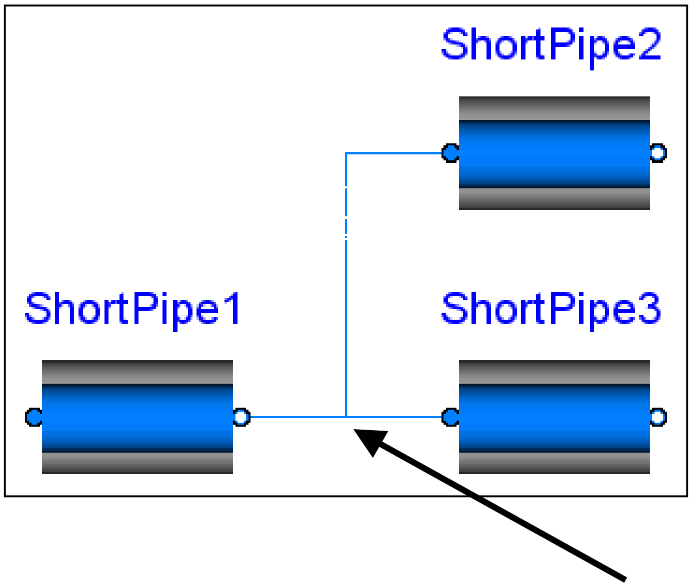
The details – boundary conditions



$$\dot{H} = \begin{cases} \dot{m} h_{port} & \dot{m} > 0 \\ \dot{m} h & otherwise \end{cases}$$

2. Modelica.Fluid Connector Definition

The interfaces (connector **FluidPort**) are defined, so that arbitrary components can be connected together



- correct for **all media** of Modelica_Media (incompressible/compressible, one/multiple substance, one/multiple phases)
- **Diffusion not included**

- Infinitesimal small volume in connection point.
- **Mass**- and **energy balance** are always **fulfilled** (= ideal mixing).
- If "ideal mixing is not sufficient", a special component to define the mixing must be introduced

Connector

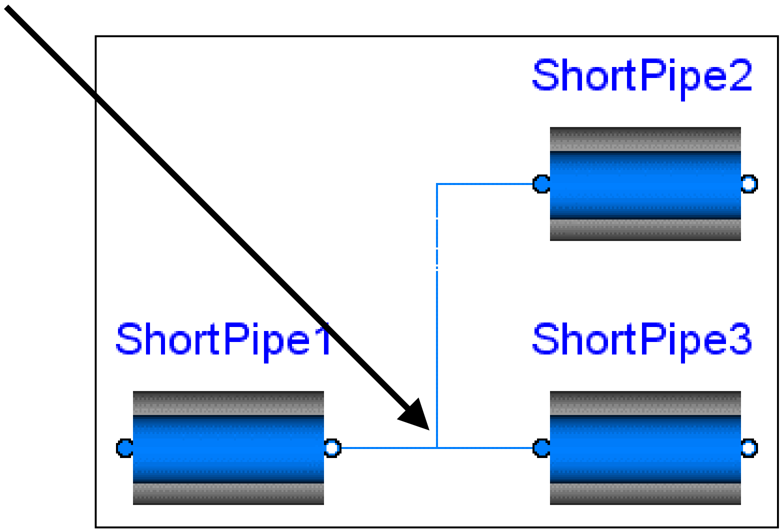
Infinitesimal control volume associated with connection

- **flow** variables give mass- and energy-balances
- Momentum balance not considered – forces on junction gives balance

```
connector FluidPort
 replaceable package Medium =
 Modelica_Media.Interfaces.PartialMedium;

 Medium.AbsolutePressure  p;
 flow Medium.MassFlowRate m_flow;

 Medium.SpecificEnthalpy      h;
 flow Medium.EnthalpyFlowRate H_flow;

 Medium.MassFraction      X     [Medium.nX-1]
 flow Medium.MassFlowRate mX_flow[Medium.nX-1]
end FluidPort;
```

Medium in connector allows to check that only valid connections can be made!

Balance equations for infinitesimale balance volume without mass/energy/momentum storage:



Intensive variables (since ideal mixing):

p1=p2=p3; h1=h2=h3; ....

Mass balance:

0 = m_flow1 + m_flow2 + m_flow3

Energy balance 2:

0 = H_flow1 + H_flow2 + H_flow3

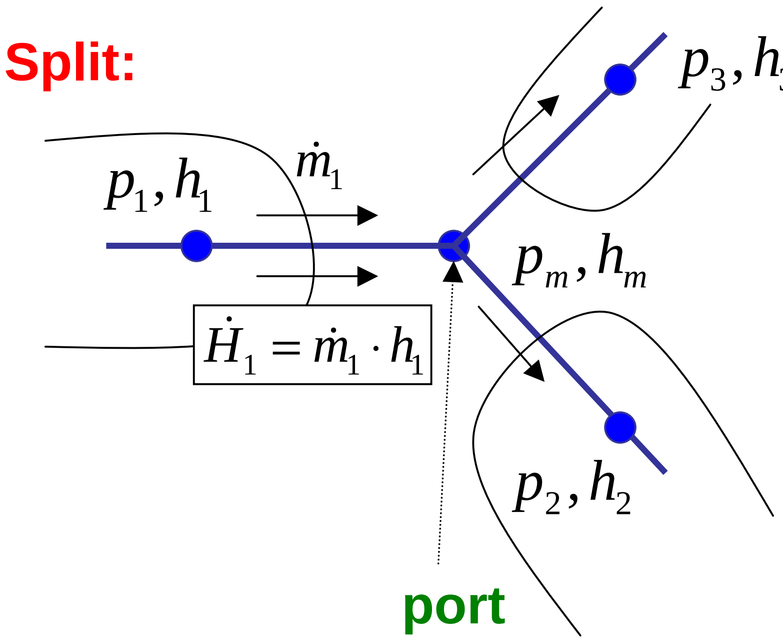
{Momentum balance (v=v1=v2=v3; i.e., velocity vectors are parallel)

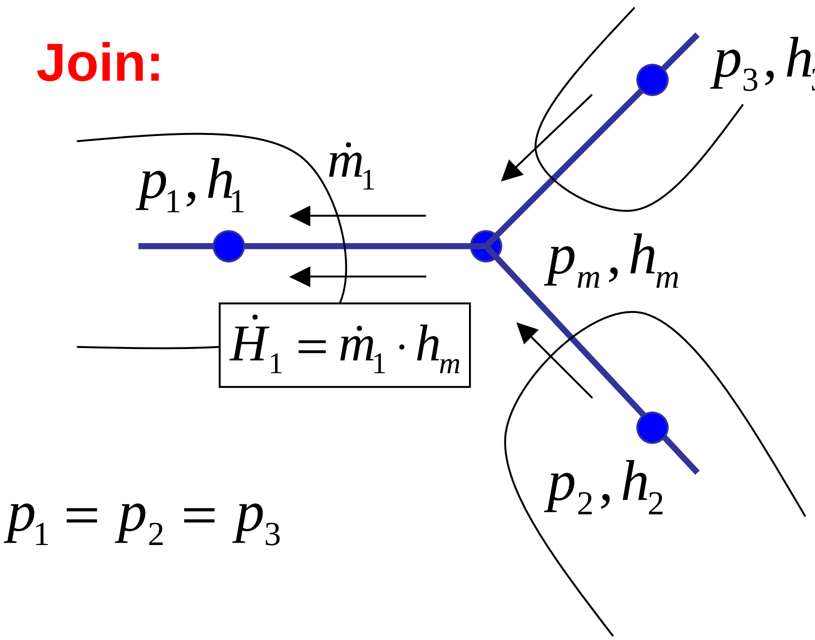
0 = m_flow1*v1 + m_flow2*v2 + m_flow3*v3
= v*(m_flow1 + m_flow2 + m_flow3)}

Conclusion:

Connectors must have "**m_flow**" and "**H_flow**" and define them as "**flow**" variable since the default connection equations generate the mass/energy/momentum balance!

2.1 "Upstream" discretisation + flow direction unknown





$$p_m = p_1 = p_2 = p_3$$

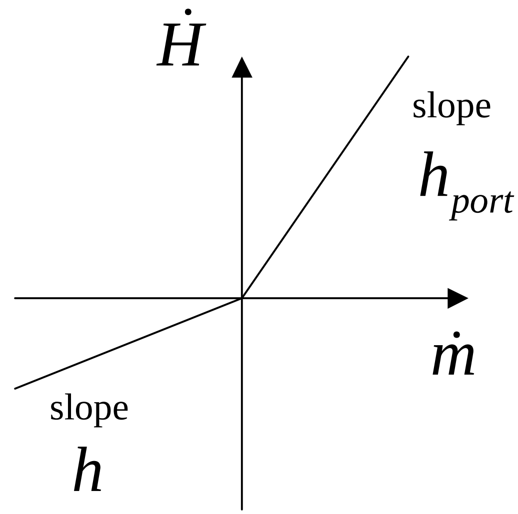
Variables in connector **port** of component 1:

$$\boxed{p_m, h_m, \dot{m}_1, \dot{H}_1}$$

```
connector FluidPort
  SI.Pressure                  p; //pm
  SI.SpecificEnthalpy          h; //hm
  flow SI.MassFlowRate         m_flow;
  flow SI.EnthalpyFlowRate     H_flow;
end FluidPort;
```

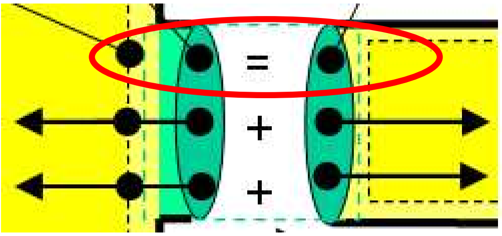
Energy flow rate and port specific enthalpy

$$\dot{H} = \begin{cases} \dot{m} h_{port} & \dot{m} > 0 \\ \dot{m} h & otherwise \end{cases}$$




```
H_flow=semiLinear(m_flow, h_port, h)
```

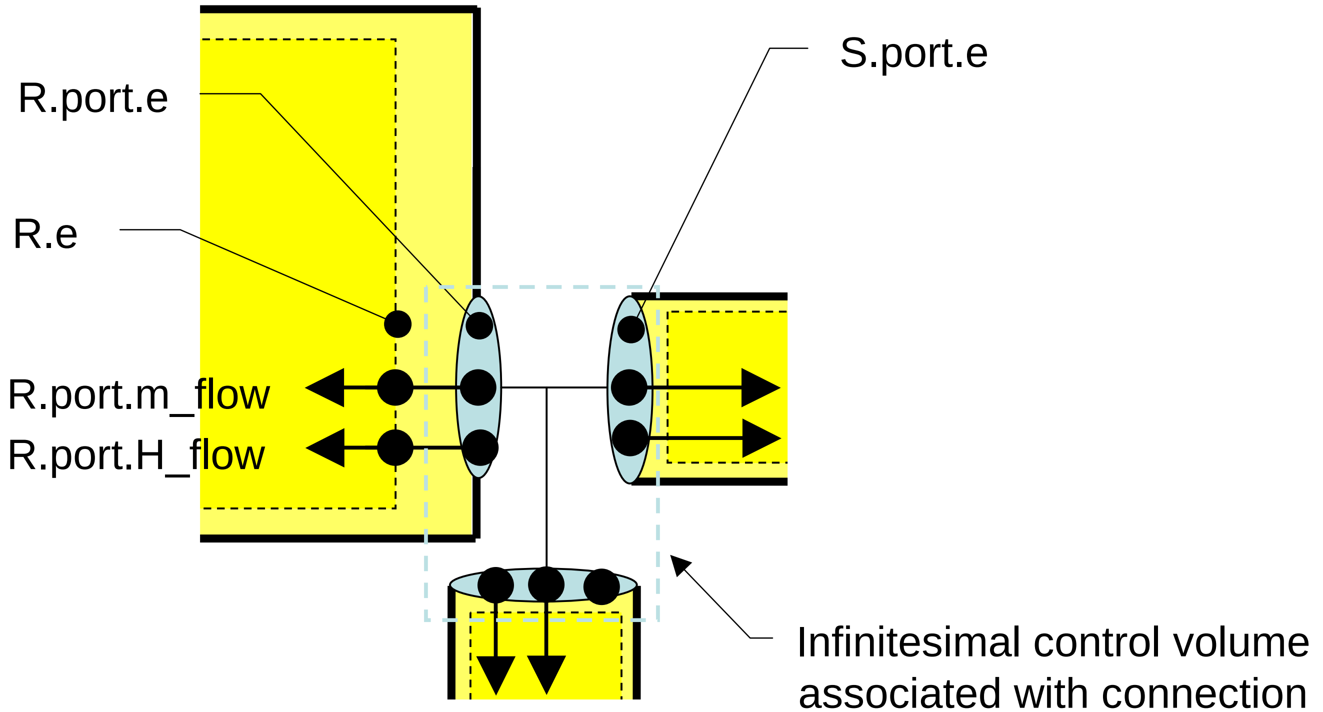
Solving semiLinear equations




```
R.H_flow=semiLinear(R.m_flow, R.h_port, R.h)
S.H_flow=semiLinear(S.m_flow, S.h_port, S.h)
// Connection equations
R.e_port = S.e_port
R.m_flow + S.m_flow = 0
R.H_flow + S.H_flow = 0
```

$$e_{port} = \begin{cases} e_S & \dot{m}_R > 0 \\ e_R & \dot{m}_R < 0 \\ undefined & \dot{m}_R = 0 \end{cases}$$

Three connected components



- Use of semiLinear() results in systems of equations with many if-statements.
- In many situations, these equations can be solved symbolically

Splitting flow

- For a *splitting* flow
  from R to S and T
  (R.port.m_flow < 0, S.port.m_flow > 0 and T.port.m_flow > 0)

- h = -R.port.m_flow*R.h /
  (S.port.m_flow + T.port.m_flow)

- h = R.h

Mixing flow

- For a *mixing* flow
  from R and T into S
  (R.port.m_flow < 0, S.port.m_flow < 0 and T.port.m_flow > 0)

  h = -(R.port.m_flow*R.h + S.port.m_flow*S.h) /
  T.port.m_flow
- or

  h = (R.port.m_flow*R.h+S.port.m_flow*S.h) /
  (R.port.m_flow + S.port.m_flow)

- Perfect mixing condition

Mass- momentum- and energy-balances

$$\frac{\partial(\rho A)}{\partial t}+\frac{\partial(\rho A v)}{\partial x}=0$$

$$\frac{\partial(\rho v A)}{\partial t}+\frac{\partial(\rho v^2 A)}{\partial x}=-A\frac{\partial p}{\partial x}-F_F-A\rho g\frac{\partial z}{\partial x}$$

$$\frac{\partial(\rho(u+\frac{v^2}{2})A)}{\partial t}+\frac{\partial(\rho v(u+\frac{p}{\rho}+\frac{v^2}{2})A)}{\partial x}=-A\rho v g\frac{\partial z}{\partial x}+\frac{\partial}{\partial x}(kA\frac{\partial T}{\partial x})$$

$$F_F=\frac{1}{2}\rho v\left|v\right| fS$$

Alternative energy equation

- Subtract v times momentum equation

$$\frac{\partial(\rho u A)}{\partial t} + \frac{\partial(\rho v(u + \frac{p}{\rho})A)}{\partial x} = vA\frac{\partial p}{\partial x} + vF_F + \frac{\partial}{\partial x}(kA\frac{\partial T}{\partial x})$$

Finite volume method

- Integrate equations over small segment
- Introduce appropriate mean values

$$\int_a^b \frac{\partial(\rho A)}{\partial t} dx + \rho A v\Big|_{x=b} - \rho A v\Big|_{x=a} = 0$$

$$\frac{dm}{dt} = \dot{m}_a + \dot{m}_b$$

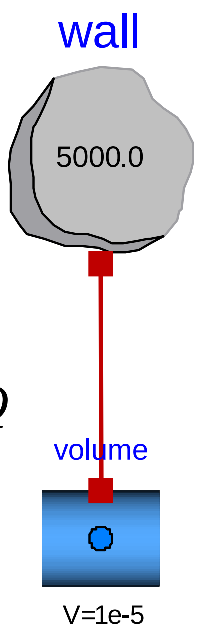
$$m = \rho_m A_m L$$

Index reduction and state selection

- Component oriented modeling needs to have maximum number of differential equations in components
- Constraints are introduced through connections
- Tool needs to figure out how many states are independent
- Common situation in mechanics, not common in fluid in thermodynamic modeling
- Very useful also in thermo-fluid systems

Index reduction and state selection

- Example: connect an incompressible medium to a metal body under the assumption of infinite heat conduction (Exercise 3-1).
- Realistic real-world example: model of slow dynamics in risers and drum in a drum boiler



unknown

heat flow $\dot{Q}$

Dynamic equations

$$m\,cp\frac{dT_{wall}}{dt} = \dot{Q}$$

$$\frac{dU}{dt} = \sum \dot{H} - \dot{Q}$$

constraint equation

$$T_{wall} = T_{fluid}$$

Index reduction and state selection

Differentiate the constraint equation:

$$\frac{dT_{wall}}{dt} = \frac{dT_{fluid}}{dt}$$

Definition of u for simple Incompressible fluid

$$U = u\,m$$

$$u(T) = h(T) - \frac{p_0}{\rho}$$

Expand fluid definition

$$\frac{dh(T)}{dT}\frac{dT_{fluid}}{dt} - \frac{p_0}{\rho} = cp(T)\frac{dT_{fluid}}{dt} - \frac{p_0}{\rho}$$

Re-write energy balance with T as state

$$\frac{dU}{dt} = m\left(cp(T)\frac{dT_{fluid}}{dt} - \frac{p_0}{\rho} \right)$$

Index reduction and state selection

Using $$\frac{dT_{wall}}{dt} = \frac{dT_{fluid}}{dt}$$

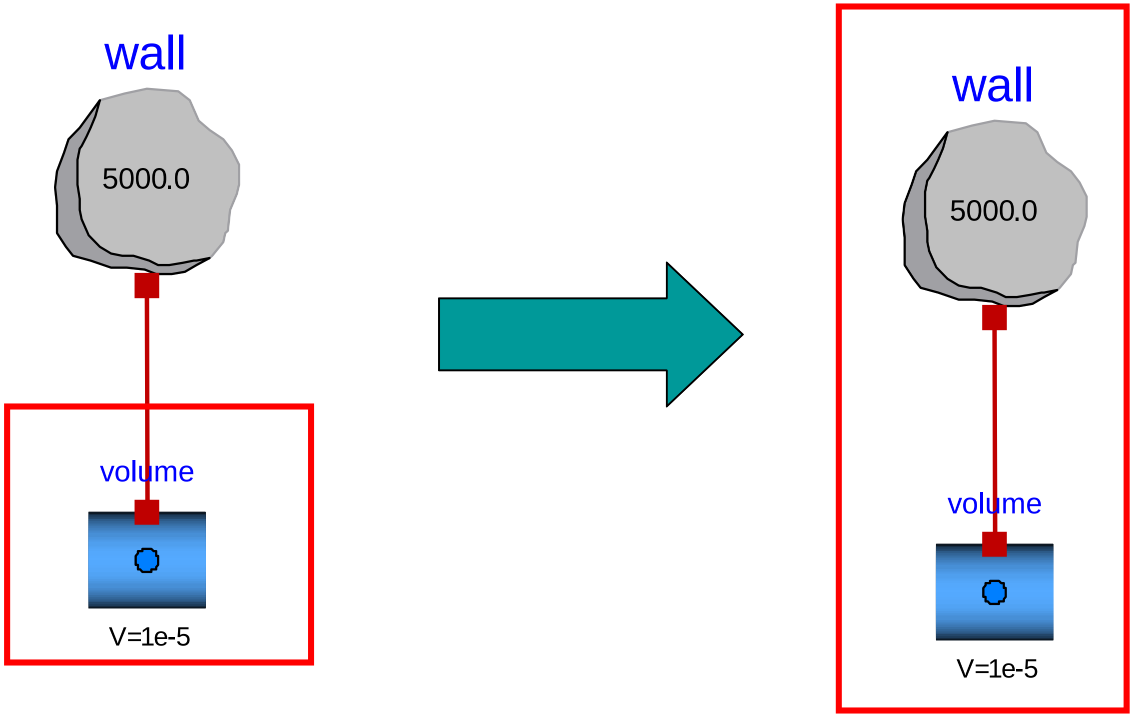
Some rearranging yields:

$$\left(m_{wall}cp_{wall} + m_{fluid}cp_{fluid}\right)\frac{dT}{dt} = \sum \dot{H} + \frac{p_0}{\rho}$$

→ Combined energy balance for metal and fluid

Index reduction and state selection

- Because temperatures are forced to be equal, we get to one energy balance instead of 2
- Side effect: the independent state variable is now T, not U any more
- Heavily used in Modelica.Media and Fluid to get efficient dynamic models



Index reduction and state selection

- Many other situations:
  - 2 volumes are connected
  - Tanks have equal pressure at bottom
- Huge advantage in most cases:
  - Independence of fluid and plant model
  - Highly efficient
- Potential drawbacks
  - All functions have to be differentiable
  - Complex manipulations
  - If manipulations not right, model can have unnecessary non-linear equations
  - See exercises

Regularizing Numerical Expressions

- Robustness: reliable solutions wanted
  in the complete operating range!
- Difference between static and dynamic models
- Empirical correlations not adapted to robust numerical solutions (only locally valid)
- Non-linear equation systems or functions
- Singularities
- Handling of discontinuities

Singularities

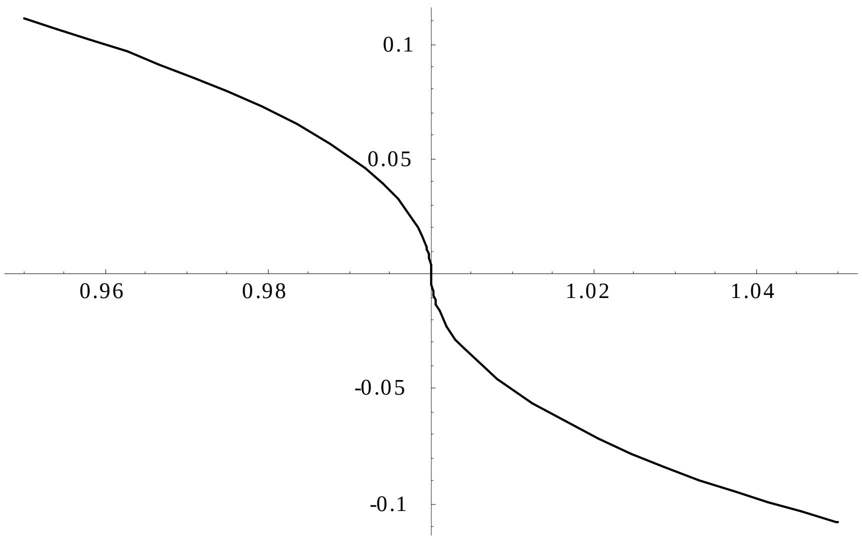
- *functions with singular points or singular derivatives should be regularized.*

  - Empirical functions are often used outside their region of validity to simplify models.
  - Most common problem: infinite derivative, causing *inflection*.

Root function Example

- Textbook form of turbulent flow resistance

$$\dot{m} - k\, sign\left(\Delta p\right)\sqrt{\rho\, abs(\Delta p)} = 0$$



→ Infinite derivative at origin

Singularities

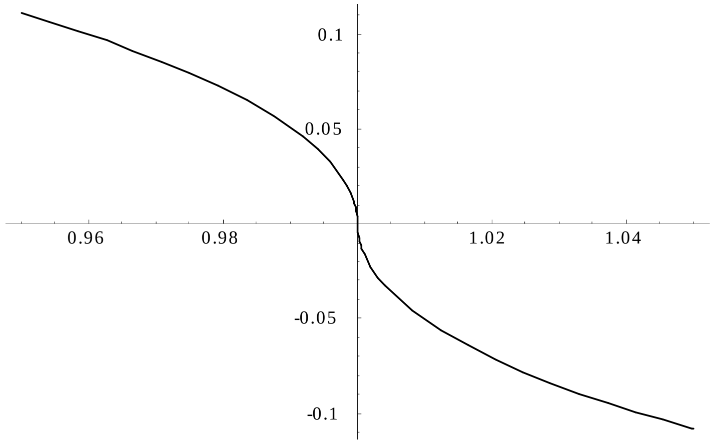
- Infinite derivative causes trouble with Newton-Raphson type solvers:
  Solutions are obtained from following iteration:

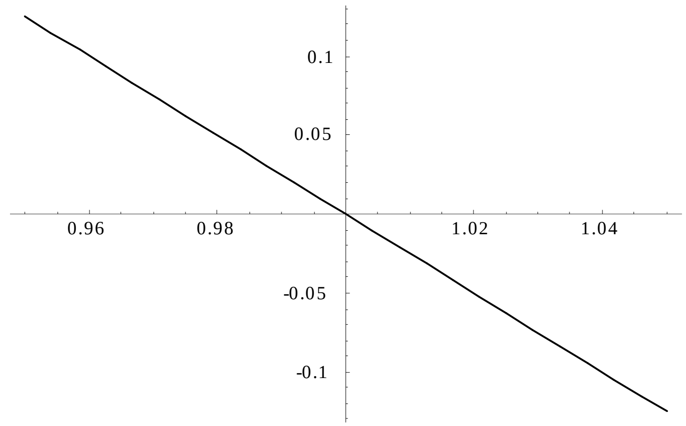
$$z^{j+1} = z^j + \frac{f(z^j)}{\frac{\partial f(z^j)}{\partial z^j}} \approx \Delta z^j + \frac{f(z^j)}{\frac{\Delta f(z^j)}{\Delta z^j}}$$

For $\frac{\partial f(z^j)}{\partial z^j} \to \infty$, the step size goes to 0.

This is called *inflection problem*

Root function remedy





- Replace singular part with local, non-singular substitute
  – result should be qualitatively correct
  – the overall function should be C^1 continuous
  – No singular derivatives should remain!

Log-mean Temperature

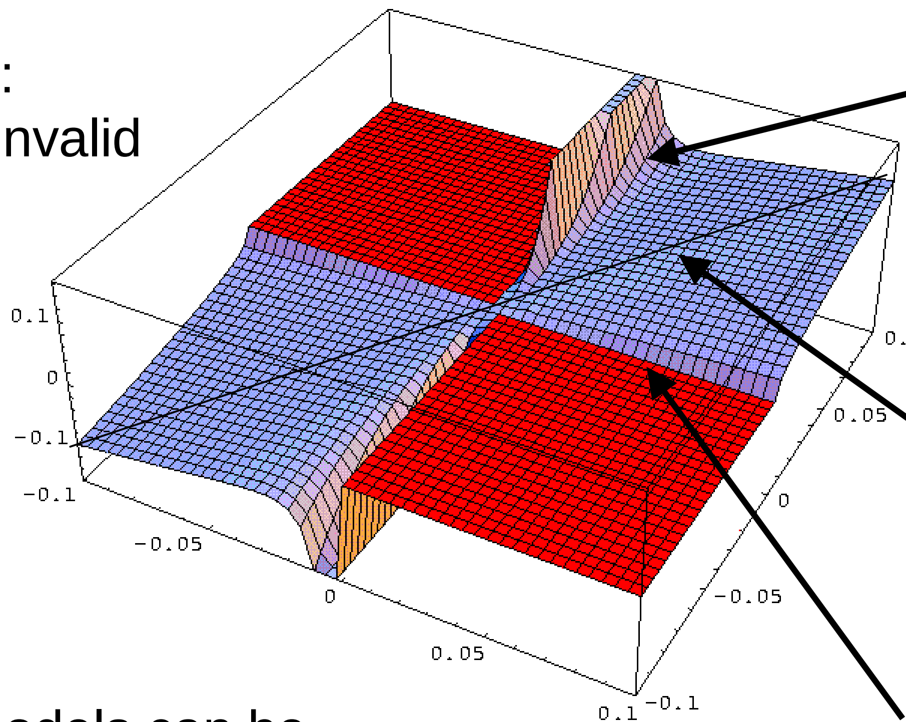
Log-mean Temperature $$\Delta T_{lm} = \frac{\Delta T_1 - \Delta T_2}{\ln(\Delta T_1 / \Delta T_2)}$$

- Invalid for all $\mathrm{sign}(\Delta T_1) \times \mathrm{sign}(\Delta T_2) < 0$
- numerical singularities for

$$\Delta T_1 = \Delta T_2,\ \Delta T_1 \to 0,\ \Delta T_2 \to 0$$

Log-mean Temperature Difference

Red areas:
log-mean invalid



Boundaries with ridges

$\Delta T1 = \Delta T2$ singularity

Boundaries with infinite derivative

Dynamic models can be in any quadrant!

Scaling

- *Scale extremely nonlinear functions to improve numerical behavior.*

- *Use min, max and nominal attributes so that the solver can scale.*

```
Real myVar(min = 1.0e-10, max =
  1.0, nominal =1e-3)
```

Scaling: chemical equilibrium of dissociation of Hydrogen

$$\frac{1}{2}\mathrm{H}_2 \longleftrightarrow \mathrm{H}$$

$$x_{\mathrm{H}} = \frac{\sqrt{x_{\mathrm{H}_2}}}{\sqrt{\mathrm{p}}}\exp(k)$$

$$k = 2.6727 - \frac{11.247}{T} - 0.0743T + 0.4317\log(T) + 0.002407T^2$$

Take the log of the equation and variables:

$$\log(x_{\mathrm{H}}) = \frac{1}{2}\left(\log(x_{\mathrm{H}_2}) - \log(p)\right) + k$$

Scaling: chemical equilibrium of dissociation of Hydrogen

$$\frac{1}{2}\mathrm{H}_2 \longleftrightarrow \mathrm{H}$$

Effect of log at 280K: ratio of $\dfrac{x_{\mathrm{H}_2}}{x_{\mathrm{H}}} = 1.3\times 10^{75}$

ratio of $\dfrac{\log(x_{\mathrm{H}_2})}{\log(x_{\mathrm{H}})} = 172$

Experiences tested with several non-linear solvers:

22 simultaneous, similar equilibrium reactions
exp form: solvable if T > 1200 K
log form: solvable if T > 250 K

Smoothing

- *Piece-wise and discontinuous function approximations which should be continuous for physical reasons shall be smoothened.*

Smoothing example: Heat transfer equations

- Convective heat transfer with flow perpendicular to a cylinder. Two Nusselt numbers for laminar and turbulent flow

$$Nu_{lam} = 0.664 Re^{1/2} Pr^{1/3}$$

$$Nu_{turb} = \frac{0.037 Re^{0.8} Pr}{1 + 2.443 Re^{-0.1} (Pr^{2/3} - 1)}$$

For *Re* < 10 we have to take care of the root function singularity as well!

- Combine as:

$$Nu = 0.3 + \sqrt{Nu_{lam}^2 + Nu_{turb}^2}$$

$$10 < Re < 10^7, \;\; 0.6 < Pr < 1000$$

Summary

Framework for object-oriented fluid modeling

- Media and component models decoupled
- Reversing flows
- Ideal models for mixing and separation
- Index reduction for
  - transformations of media equations
  - handling of incompressible media
  - Index reduction for combining volumes
- Some issues in numerical regularization

Exercises

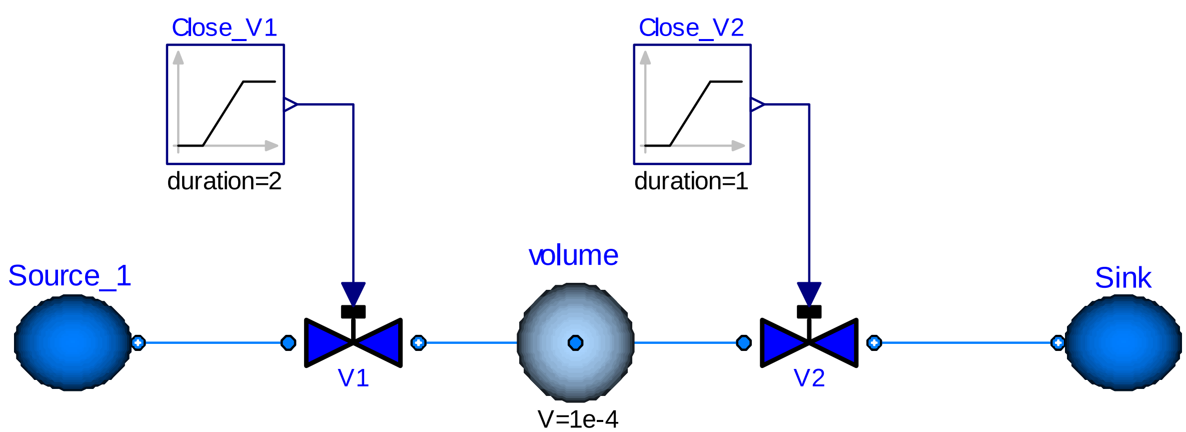
1. Build up small model and run with different media
   - Look at state selection
   - Check non-linear equation systems
   - Test different options for incompressible media
2. Reversing flow and singularity treatment
   - Build up model with potential backflow
   - Test with regularized and Text-book version of pressure drop
3. Index reduction and efficient state selection with fluid models, 3 different examples
   1. Index reduction through temperature constraint of solid body and fluid volume (used in power plant modeling)
   2. Index reduction between 2 well mixed volumes
   3. Index reduction between tanks without a pressure drop in between.
4. Non-linear equation systems in networks of simple pressure losses
5. Reversing and 0-flow with liquid valve models

Exercises

- All exercises are explained step by step in the info-layer in Dymola of the corresponding exercise models.
- All models are prepared and can be run directly, the the exercises modify and explore the models
- Run the script “setup.mos” to open the file and select all settings.

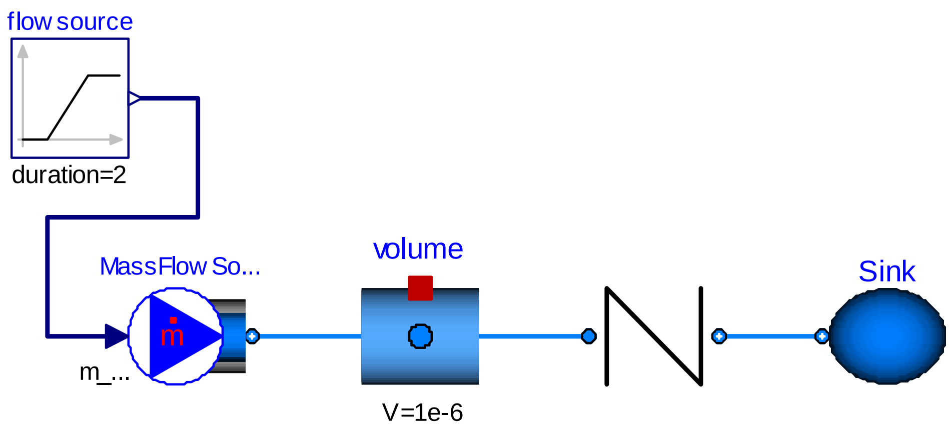
Exercise 1

- Using different Media with the same plant and test of 0-flow conditions



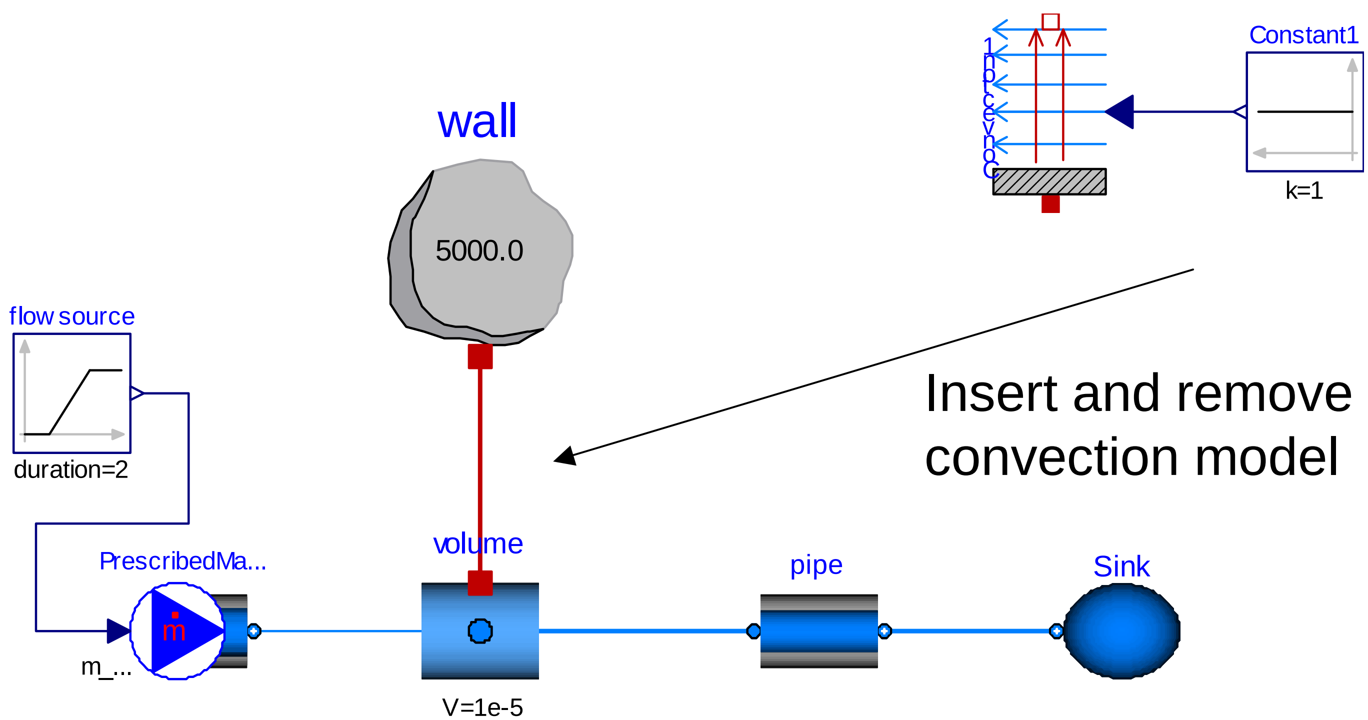
Exercise 2

- Effect of Square root singularity



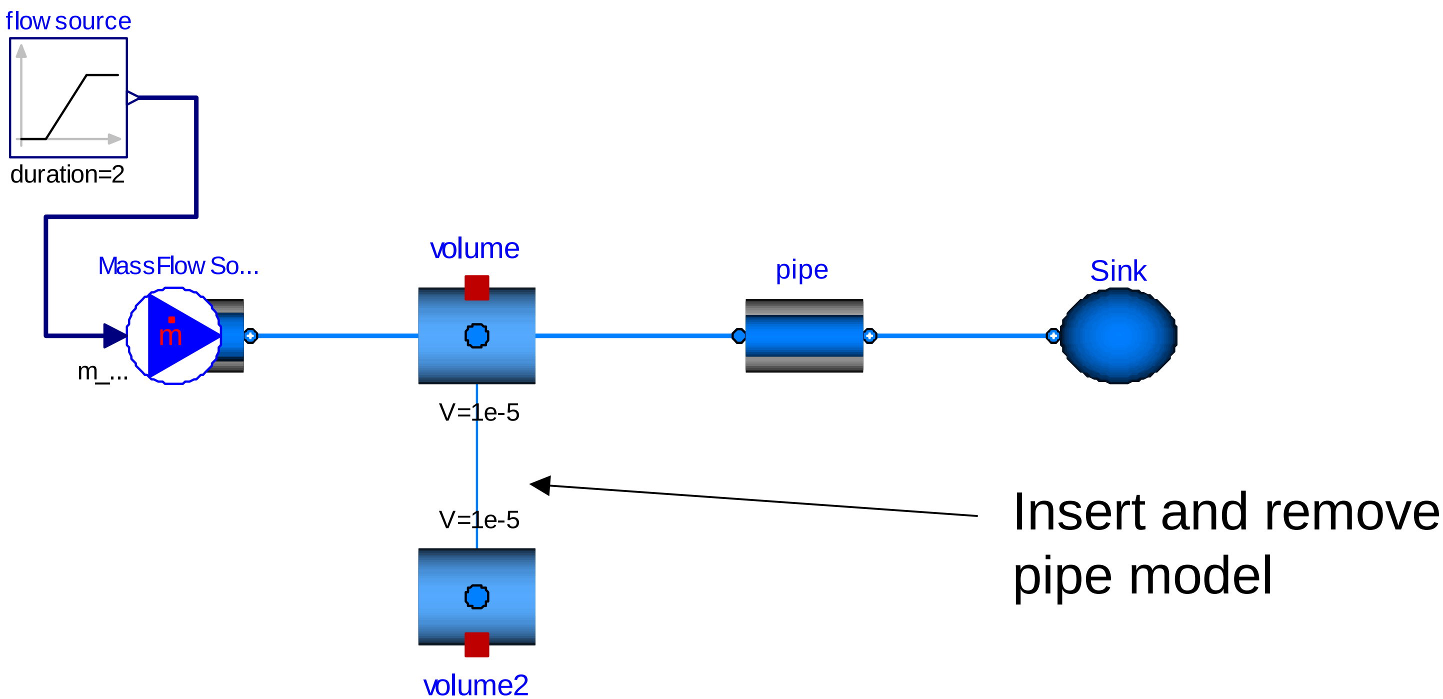
Exercise 3-1

- Index reduction between solid body and fluid volume



Exercise 3-2

- Index reduction between 2 well mixed fluid volumes

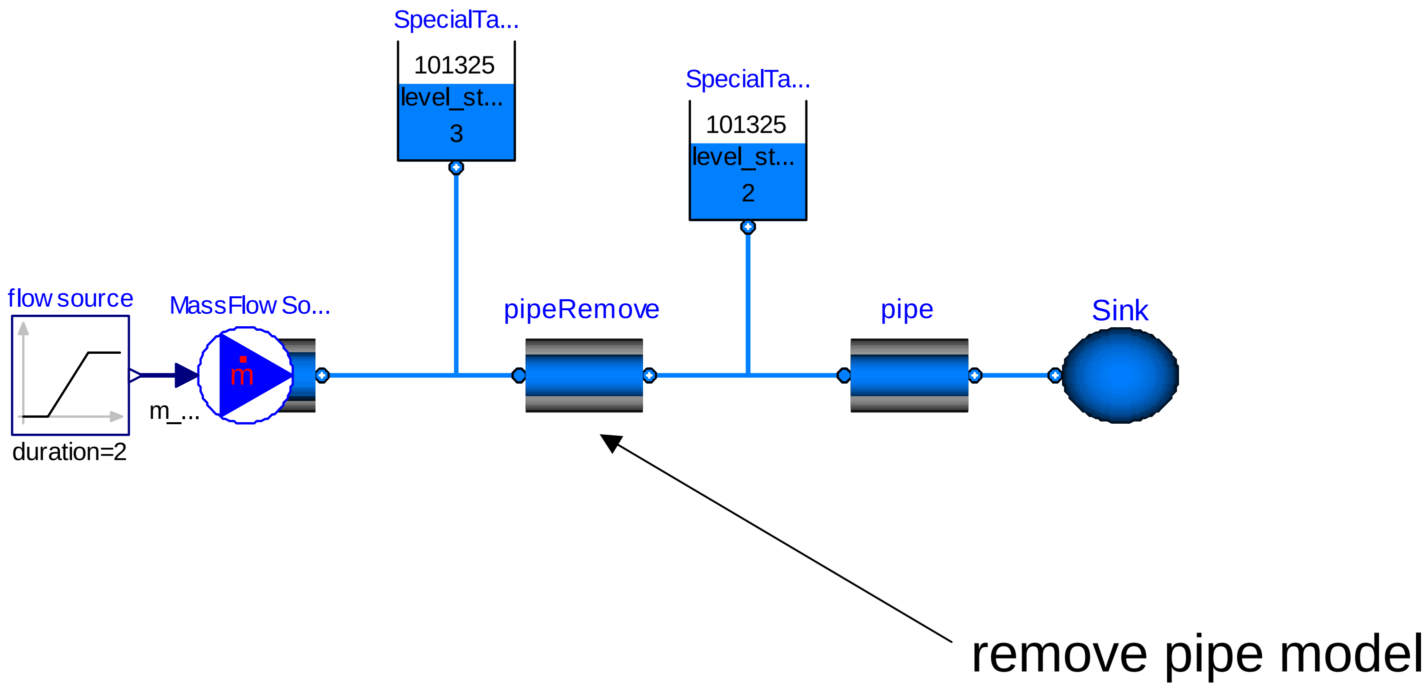


Index reduction with perfect mixing of the energy and identical pressure

4 states from 2 volumes are reduced to 2 states

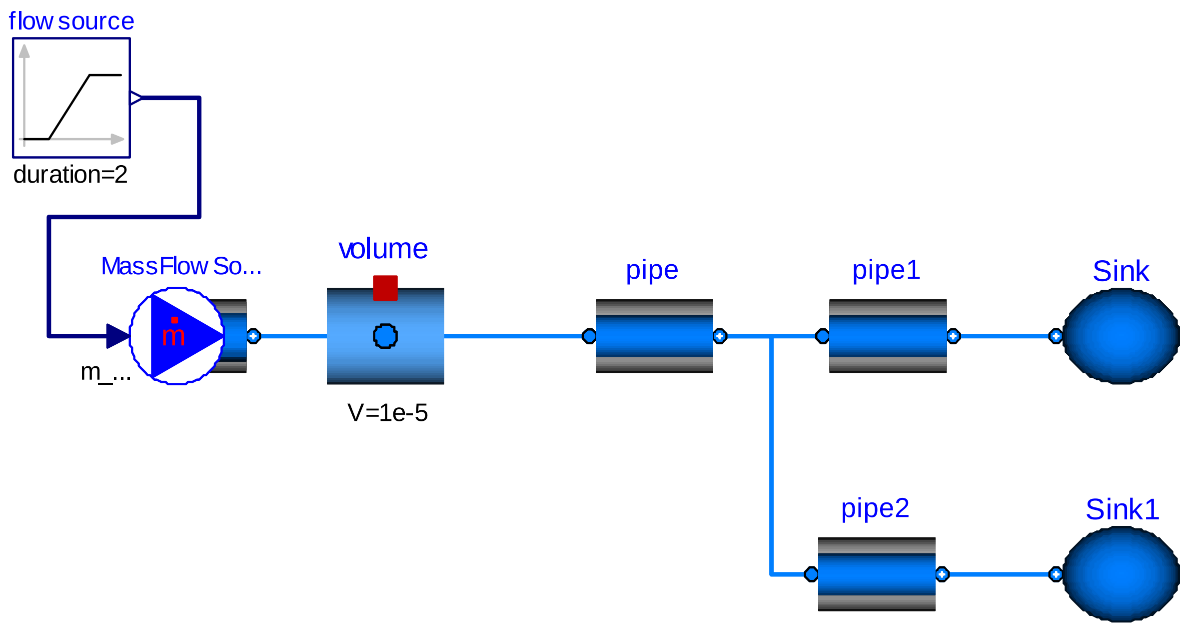
Exercise 3-3

- Index reduction of one state only between 2 tank models



Exercise 4

- Non-linear equations in pipe network models



Exercise 5

- Reversing flow for a liquid valve