

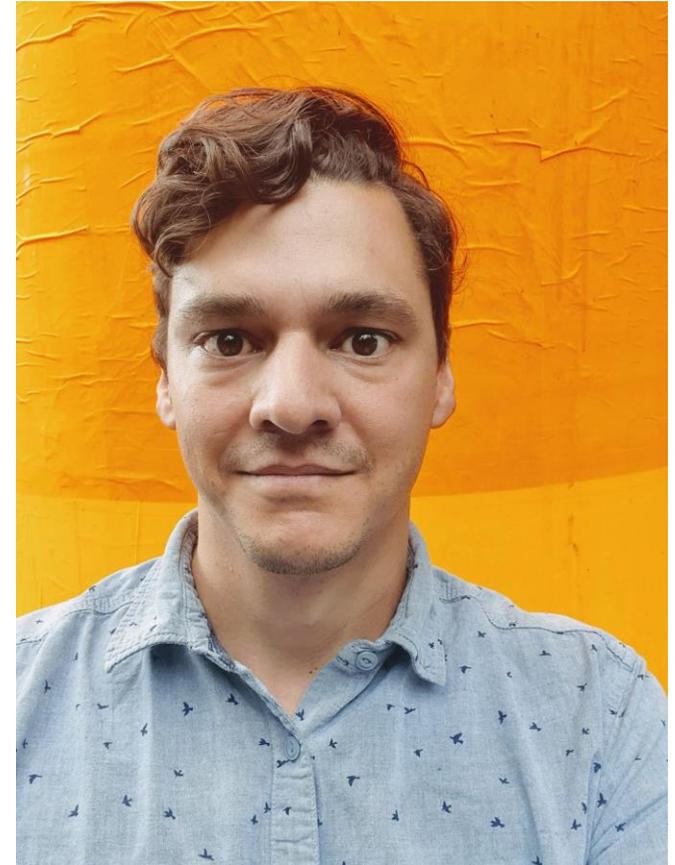
# Initialization

Initialization of closed-loop thermal management cycles



# About Me

- Matthis Thorade
- Senior Simulation Engineer
- Library Developer for Thermal Management Libraries
  
- Studied mechanical engineering
- PhD in thermodynamics
- Using Modelica since 2009



# Initialization basics

States, initial equations, start values

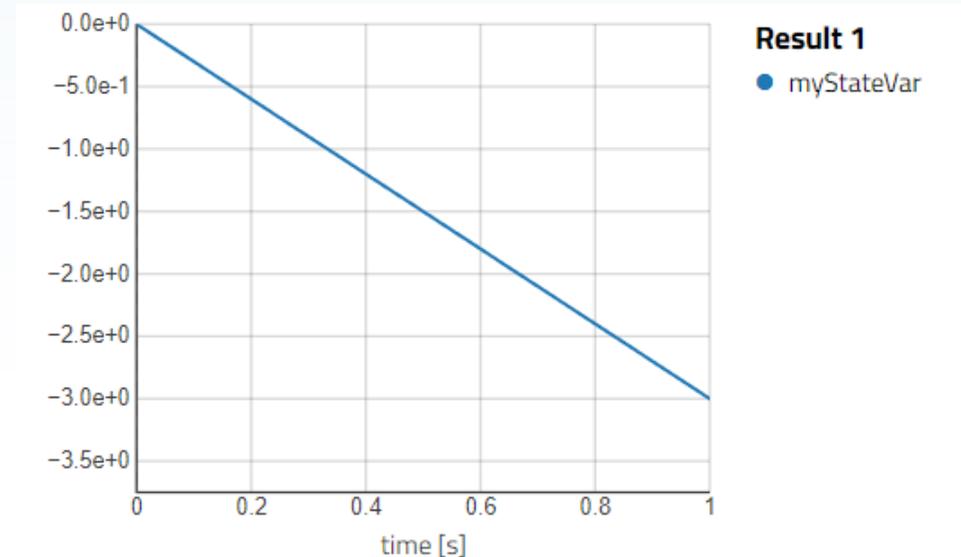
# Where it all begins

- Dynamic variables: time derivatives of dynamic variables appear in equations
- Algebraic variables: equations do not contain time derivative
  
- State variables: Minimal set of variables that uniquely define the state of the system
- Values for all other variables are calculated from the states
- State variables are integrated over time (using the time derivative)
- Value for state variables at time=0 ??

# States at time=0

- Time derivative of myStateVar appears in equation
- Dynamic variable selected as state
- No initial value specified, starts at 0

```
1 model Init_Example
2
3   Real myStateVar;
4
5   parameter Real rateofchange = -3;
6   Real someOtherVar;
7
8 equation
9   // all equations are fulfilled at all times
10  // number of equations must equal the number of variables
11  der(myStateVar) = rateofchange;
12  someOtherVar = 100 - myStateVar;
13
14 end Init_Example;
```



# Initial equation

- Most basic way to specify initial value of states
- Normal equations hold at all times,  
Initial equations hold at time=0
- Number of initial equations cannot be larger  
than number of states

```
1 model Init_Example_ineq
2   Real myStateVar;
3   parameter Real rateofchange = -3;
4   Real someOtherVar;
5
6   initial equation
7     myStateVar = 4;
8     // someOtherVar = 100;
9
10  equation
11    der(myStateVar) = rateofchange;
12    someOtherVar = 100 - myStateVar;
13
14 end Init_Example_ineq;
```

# Start value

- Start value for a state variable will generate an initial equation
- Start value for other variables is only used as initial guess during iterations
- Number of start values specified can be higher than number of states (e.g. types for units have a start value defined)

```
1 model Init_Example_nonlinear
2   Real myStateVar(start=4);
3
4   parameter Real rateofchange = -3;
5   Real someOtherVar(start=100);
6   Real nonlin(start=+1);
7
8 equation
9   der(myStateVar) = rateofchange;
10  someOtherVar = 100 - myStateVar;
11  myStateVar = nonlin^2;
12
13 end Init_Example_nonlinear;
```

# Start value

- Start value for a state variable will generate an initial equation

```
1 model Init_Example_start
2   Real myStateVar(start=4);
3   parameter Real rateofchange = -3;
4   Real someOtherVar(start=100);
5
6 equation
7   der(myStateVar) = rateofchange;
8   someOtherVar = 100 - myStateVar;
9
10 end Init_Example_start;
```



```
1 model Init_Example_ineq
2   Real myStateVar;
3   parameter Real rateofchange = -3;
4   Real someOtherVar;
5
6 initial equation
7   myStateVar = 4;
8   // someOtherVar = 100;
9
10 equation
11   der(myStateVar) = rateofchange;
12   someOtherVar = 100 - myStateVar;
13
14 end Init_Example_ineq;
```

# Start value

- Use attribute `fixed=true` to enforce generation of initial equation for non-state

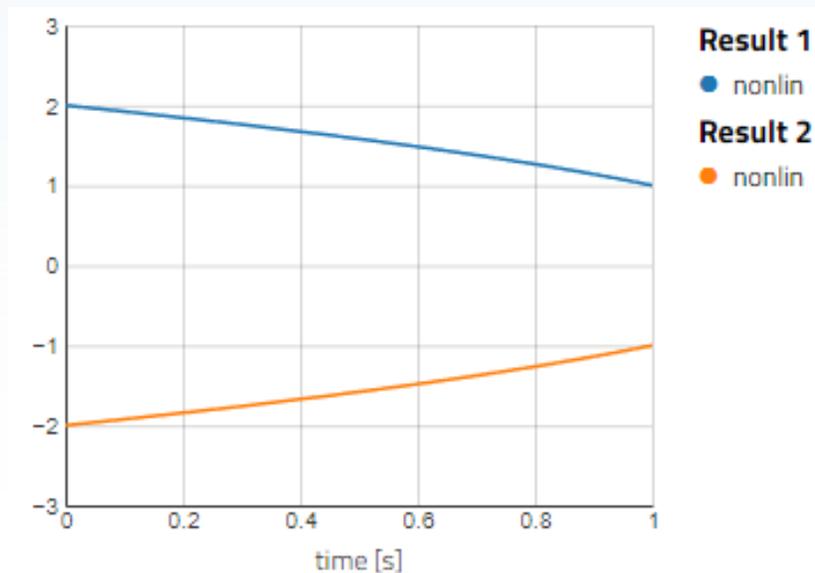
```
1 model Init_Example_start
2   Real myStateVar(start=4, fixed=false);
3   parameter Real rateofchange = -3;
4   Real someOtherVar(start=100, fixed=true);
5
6 equation
7   der(myStateVar) = rateofchange;
8   someOtherVar = 100 - myStateVar;
9
10 end Init_Example_start;
```



```
1 model Init_Example_ineq
2   Real myStateVar;
3   parameter Real rateofchange = -3;
4   Real someOtherVar;
5
6 initial equation
7   // myStateVar = 4;
8   someOtherVar = 100;
9
10 equation
11   der(myStateVar) = rateofchange;
12   someOtherVar = 100 - myStateVar;
13
14 end Init_Example_ineq;
```

# Start value for iterations

- Start value for “nonlin” variable is used as initial guess during iterations



```
1 model Init_Example_nonlinear
2   Real myStateVar(start=4);
3
4   parameter Real rateofchange = -3;
5   Real someOtherVar(start=100);
6   Real nonlin(start=+1);
7
8 equation
9   der(myStateVar) = rateofchange;
10  someOtherVar = 100 - myStateVar;
11  myStateVar = nonlin^2;
12
13 end Init_Example_nonlinear;
```

# Parameter to set start value

- Add a parameter to specify start value for states from parameter dialogue
- Separate tab in parameter dialogue

```
model Init_Example_parameter
  Real myStateVar(start=myStateVar_start);
  parameter Real myStateVar_start = 4 "expose start value as parameter"
    annotation(Dialog(tab="Initialization"));
```

# Parameter to set start value

- Most components in Modelon libraries follow convention to have parameters to set start values of states, on a tab named “Initialization”
- Additionally, parameters to set start values of important iteration variables like mass flow or wall temperature

**Pipe**  
Distributed pipe model  
InitDemoLibraryPackage.Basics.Pipe

**INFORMATION**

**PROPERTIES**

General Initialization Advanced Balance Equations Variables

Initialization option

initOpt	:	Initialization using start values
Pressure		
p_in_start	:	1e5
p_out_start	:	1e5
p_start [ni+1]	:	linspace(p_in_start,p_out_start,ni+1)
Enthalpy		
initFromEnthalpy	:	<input type="checkbox"/> »
h_in_start	:	300e3 J
h_out_start	:	300e3 J
h_start [ni+1]	:	linspace(h_in_start,h_out_start,ni+1) J
Temperature		
T_in_start	:	298.15
T_out_start	:	298.15
T_start [ni+1]	:	linspace(T_in_start,T_out_start,ni+1)

# Steady-state initialization

- Instead of prescribing a value for state at time=0, add an initial equation  $\text{der}(\text{state})=0$
- System of equations for initialization has to be balanced
- $\text{der}(M)=0$  means  $m\_flow$  is equal in adjacent components, leads to large/coupled equation systems

# Initialization, what's the problem?

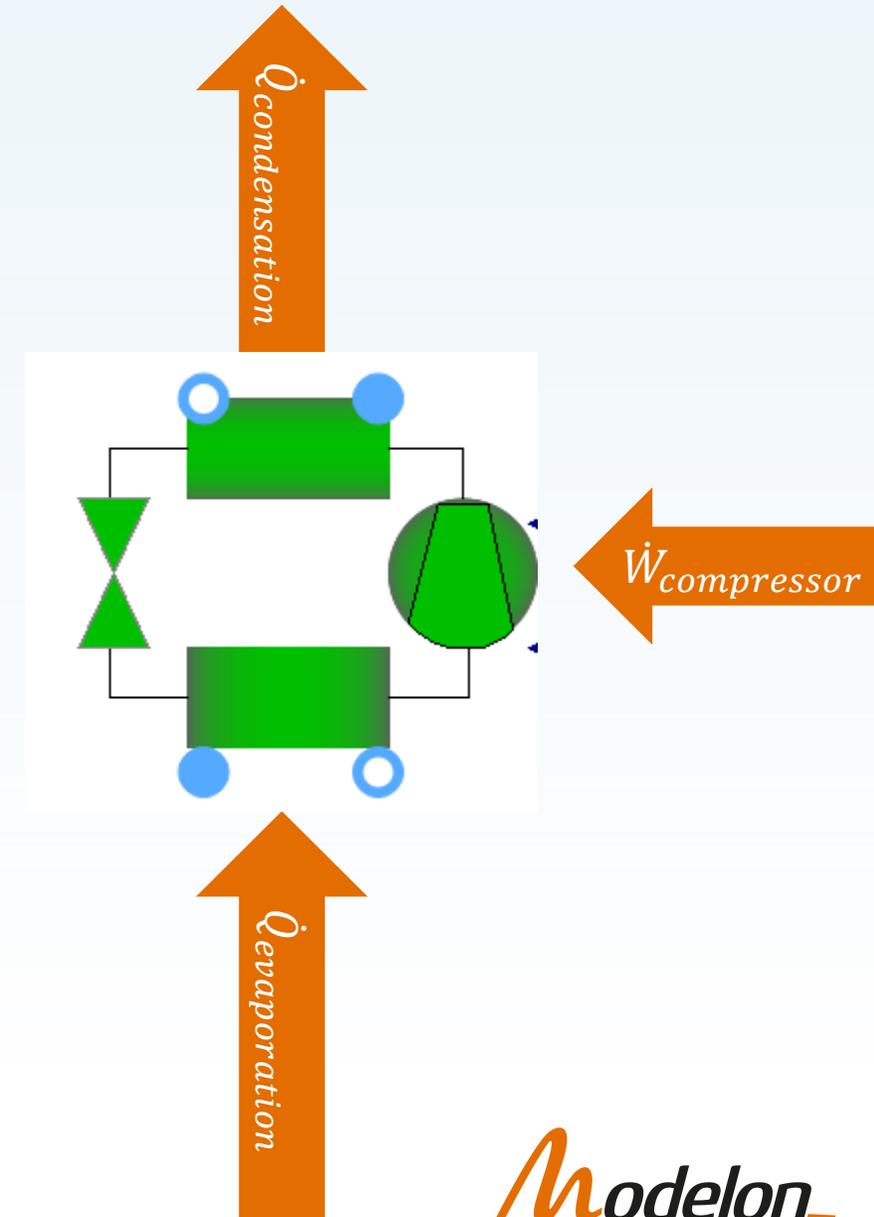
- Bad start values
  - Enthalpy or pressure equal to zero
- Contradicting start values
- Large equation systems
- Start values: initialization and iterations mixed
  - Other attributes: nominal, min, max

# Initialization of cycles

Helper calculations to estimate start values

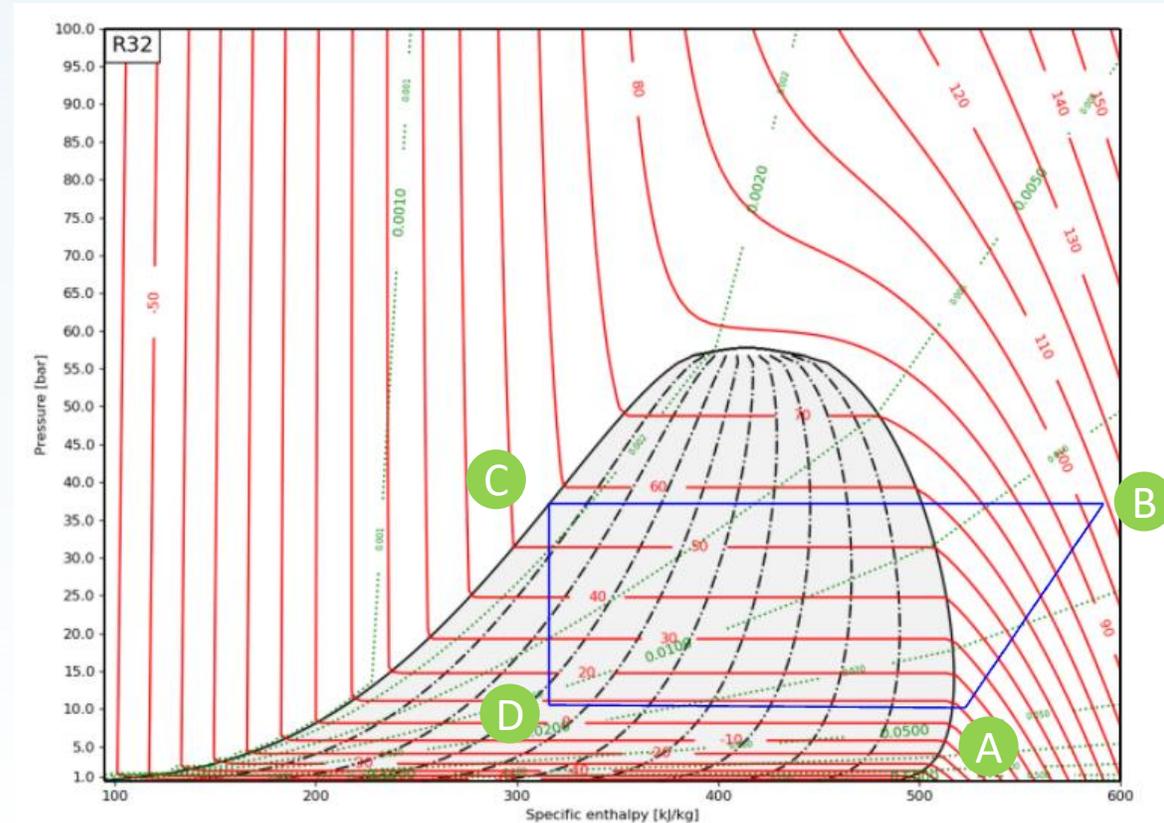
# Vapor compression cycle balance

- Heat pump: use condensation heat for heating
  - $T_{\text{evaporation}} < T_{\text{ambient}}$
  - $T_{\text{condensation}} > T_{\text{room}}$
  - $\text{COP}_{\text{heating}} = Q_{\text{cond}}/W_{\text{comp}}$
- Refrigerator: use evaporation heat for cooling
  - $T_{\text{evaporation}} < T_{\text{cabinet}}$
  - $T_{\text{condensation}} > T_{\text{ambient}}$
  - $\text{COP}_{\text{cooling}} = Q_{\text{evap}}/W_{\text{comp}}$
- $Q_{\text{cond}} = Q_{\text{evap}} + W_{\text{comp}}$
- $T_{\text{cond}} > T_{\text{evap}}$



# Vapor compression cycle ph-diagram

- A-B: compression
- B-C: condensation ~ isobaric
- C-D: expansion ~ isenthalpic
- D-A: evaporation ~ isobaric

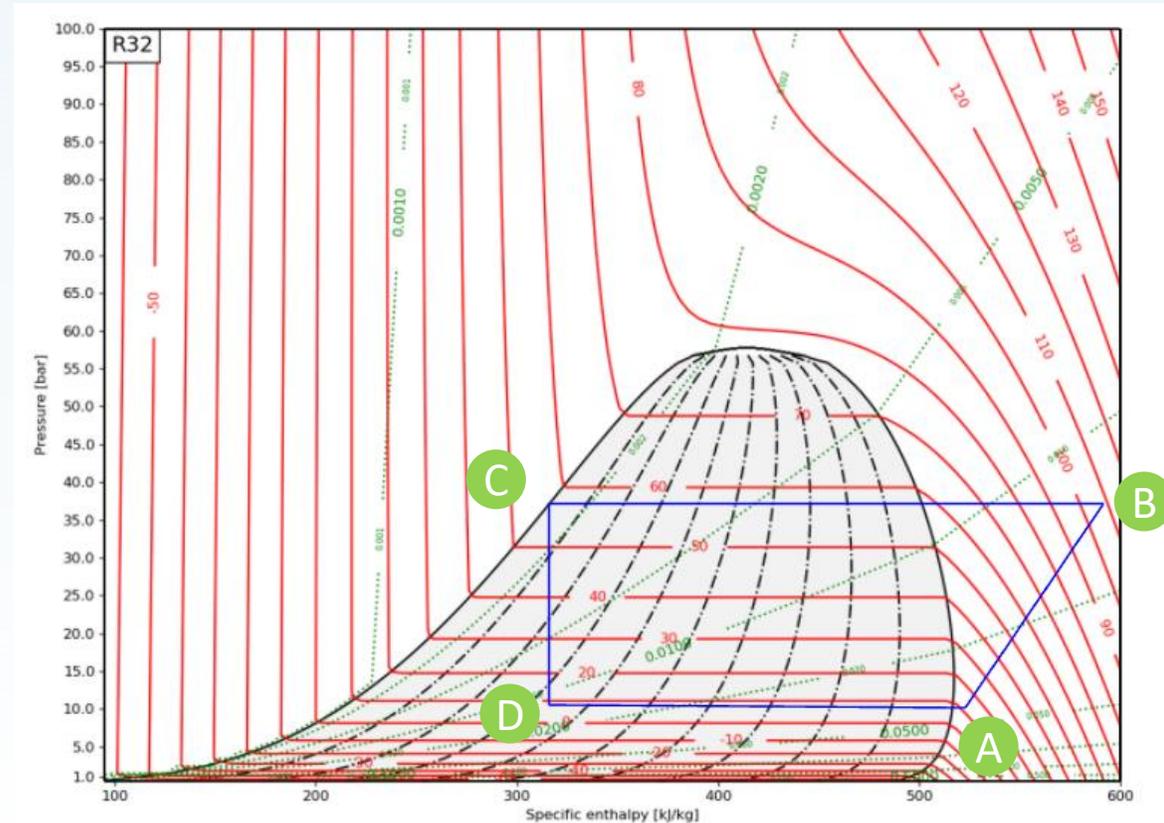


# Helper model, init record

Based on ph-diagram and balance,  
solve full load steady-state design point

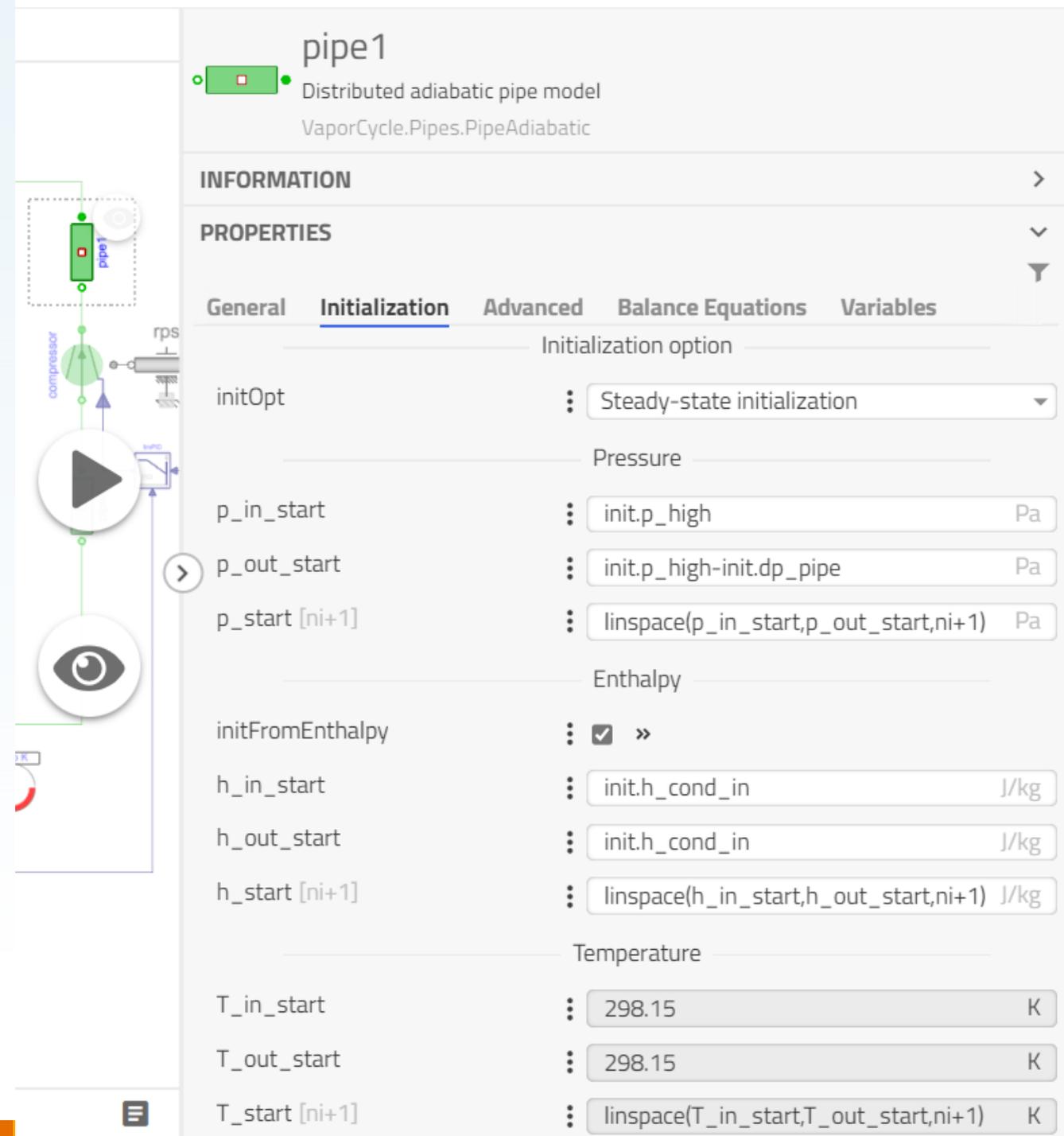
Step-by-step:

1. Evaporation and condensation temperatures and pressures above/below ambient and target
2. Subcooling and superheat  $\rightarrow$  C, A
3. Isenthalpic expansion  $\rightarrow$  D
4. Compressor isentropic efficiency  $\rightarrow$  B
5. Mass flow rate from capacity
6. Small friction losses for monotonous pressure gradient



# Reference

- Reference variables from helper model using dot-notation



**pipe1**  
Distributed adiabatic pipe model  
VaporCycle.Pipes.PipeAdiabatic

**INFORMATION**

**PROPERTIES**

General Initialization Advanced Balance Equations Variables

Initialization option

initOpt : Steady-state initialization

Pressure

p\_in\_start : init.p\_high Pa

p\_out\_start : init.p\_high-init.dp\_pipe Pa

p\_start [ni+1] : linspace(p\_in\_start,p\_out\_start,ni+1) Pa

Enthalpy

initFromEnthalpy :  »

h\_in\_start : init.h\_cond\_in J/kg

h\_out\_start : init.h\_cond\_in J/kg

h\_start [ni+1] : linspace(h\_in\_start,h\_out\_start,ni+1) J/kg

Temperature

T\_in\_start : 298.15 K

T\_out\_start : 298.15 K

T\_start [ni+1] : linspace(T\_in\_start,T\_out\_start,ni+1) K

# One size fits all?

- Assumptions in helper model
- Different topology:
  - Internal heat exchanger
  - Vapor injection compressor
  - Template per topology
- Sub- versus supercritical state
- Mixture with temperature glide
- Initialize to off-design, e.g. rest condition
- Prescribe temperature/pressure, or amount of charge

# Fixed charge

- During initialization
  - Calculate charge using AggregateTwoPhaseProperties block
  - Difference between calculated charge and desired charge
  - Specific enthalpy for receiver that would equalize the difference
- After initialization, at time=0
  - use reinit on system level to set receiver enthalpy and level

```
protected
  parameter Modelica.Units.SI.Time startTime(fixed=false);
initial equation
  startTime = time;
equation
  when sample(startTime, Modelica.Constants.inf) and init.initType == 1 then
    reinit(receiver.separator.volume.h, init.h_receiver);
  end when;
```

# Questions

Links to resources

*Modelon*

Accurate Simulations. Better Decisions.