# Development of a Refrigerant Mixture Package for Dynamic Simulation of Auto-Cascade Refrigeration: A Case Study with R23/R134a

Rohit Dhumane[1] Dan Gorman[1] Rajkumar K S[2] Dongping Huang[3]

[1]Trane Technologies, USA, {Rohit.Dhumane,Daniel.Gorman}@tranetechnologies.com
[2]Trane Technologies, India, rajkumar.kidaramshanmugam@tranetechnologies.com
[3]Trane Technologies, China, Dongping.Huang@tranetechnologies.com

## Abstract

The auto-cascade refrigeration cycle offers higher reliability and lower manufacturing costs compared to the cascade refrigeration cycle, making it particularly attractive for applications involving significant temperature differences between the heat source and sink. However, this advantage comes with increased complexity in its operation. This article aims to develop a media package in Modelica to simulate auto-cascade refrigeration cycles, with the goal of enhancing understanding of their operation and control. The method is demonstrated using an R23/R134a mixture, a commonly used refrigerant combination in such applications. The media package is compatible with the `Modelica Standard Library` and is constructed using curve fits based on the REFPROP dataset. Unlike typical refrigeration systems, the media package for auto-cascade components includes the independent mixture composition (Xi) as an additional variable in function calls. A combination of polynomial and Chebyshev polynomial curve-fits for the refrigerant properties has been shown to provide an optimal balance between accuracy and computational efficiency. The article presents example simulations performed to demonstrate use of the media package with component models from the `Modelica.Fluid` package. A lumped model for the phase separator is developed and simulated with the R23/R134a media to demonstrate ideal phase separation.

*Keywords: refrigeration, auto-cascade, two-phase*

## 1 Introduction

Modern and next generation vapor compression systems are increasingly using refrigerant mixtures to address multiple competing design constraints like low global warming potential, improved system performance and costs. Accurate and fast refrigerant property models are essential for model-based design of vapor compression systems, where simulation speed is often bottlenecked by iterative property calculations.

### 1.1 Auto-Cascade Refrigeration Cycle

The auto-cascade refrigeration cycle (ARC) offers the advantage of achieving a higher temperature difference between the source and sink using a single compressor. This is made possible by employing a zeotropic refrigerant mixture and adjusting its composition to meet operational requirements. However, calculating the thermodynamic properties of zeotropic mixtures is more complex than for pure refrigerants, making the analysis of ARC systems more complicated (He et al. 2022).

### 1.2 Refrigerant Mixture Properties

Modeling and simulation is an indispensable tool to improve our understanding about complex systems like the ARC. Modelica provides a mature and robust environment for simulating refrigeration systems, while REFPROP (Lemmon et al. 2018) is the industry standard for calculating refrigerant properties. However, the complexity of mixture models in REFPROP can significantly increase the computational cost of simulations. To address this, curve fits derived from REFPROP-generated data are used to reduce computational demands during simulation.

REFPROP is a powerful and widely accepted tool for thermodynamic property calculations, but it presents several limitations when used for dynamic simulations. REFPROP's reliance on nested iterative root-finding algorithms often leads to non-convergence or local numerical anomalies, especially near phase boundaries and critical points. These anomalies manifest as spikes in density derivatives, which can destabilize dynamic simulations or yield inaccurate results (Laughman et al. 2024). Dynamic simulations in Modelica involve frequent property evaluations which can run into millions of calls in a short time window. REFPROP's iterative nature makes these calls computationally expensive. Simulations using REFPROP can be 100–5000 times slower than those using optimized lookup or interpolatory methods (Aute and Radermacher 2014). Additionally, computing thermodynamic properties for these mixtures using first principles is numerically complex and often yields non-physical anomalies, especially in standard tools like REFPROP (Laughman et al. 2024). These anomalies, particularly in the two-phase region near the critical point can disrupt dynamic simulations by introducing discontinuities in property derivatives, leading to inaccurate or failed predictions. To overcome these limitations, researchers have developed specialized refrigerant property models using spline-based table look-up methods, polynomial fits, and interpolatory techniques (Aute and Radermacher 2014; Laughman et al. 2024; Li et al. 2018).

### 1.3 Chebyshev Polynomials

Chebyshev polynomials possess several attractive properties when it comes to function approximation. It has been proven that for any continuous function on a given interval, there exists a Chebyshev polynomial that can approximate the said function. These polynomials have been found particularly effective for approximating refrigerant properties, offering a favorable balance between accuracy and computational efficiency (Aute and Radermacher 2014). In this article, we will use Chebyshev polynomials of the first kind $T_n$ which are defined by Equation 1.

$$T_n(cos(\theta)) = cos(n\theta) \tag{1}$$

These are typically calculated with a recurrence equation as shown by Equation 2 - Equation 4.

$$T_0(x) = 1 \tag{2}$$
$$T_1(x) = x \tag{3}$$
$$T_{n+1}(x) = 2xT_n(x) - T_{n-1}(x) \tag{4}$$

A polynomial in a single variable, referred to as 1D, can now be calculated with a coefficient vector $c_i$, which can be calculated from a least squares fit over the data.

$$T_{n+1}(x) = \sum_{i=0}^{n} c_i T_i(x) \tag{5}$$

For the case of polynomials with two variables, separate series for each of the input is created. The orthogonality property of the Chebyshev polynomials is then used to create the 2D equation. The methodology and terminology from the Python's numpy package (Harris et al. 2020) is followed in this article. The reader may refer to its documentation for details.

$$T_{n+1,m+1}(x,y) = \sum_{i=0}^{n}\sum_{j=0}^{m} c_{i,j} T_i(x) T_j(y) \tag{6}$$

Similarly for an equation with three variables, the Equation 7 can be used.

$$T_{n+1,m+1,l+1}(x,y,z) = \sum_{i=0}^{n}\sum_{j=0}^{m}\sum_{k=0}^{l} c_{i,j,k} T_i(x) T_j(y) T_k(z) \tag{7}$$

The ability to create polynomial functions with different maximum degree for the inputs helps to reduce computational cost.

The objective of the article is to present a method to develop a two-phase refrigerant mixture package for zeotropic mixture combinations which are applicable for auto-cascade refrigeration systems. Compatibility with `Modelica Standard Library` Media package is to be maintained to enable reuse of components especially from the `Fluids` package. The method is applied to a binary refrigerant mixture and the required property functions are created in Modelica. The applicability of the autocascade media is demonstrated by simulating a few component level simulations involving change in refrigerant mixture concentration.

## 2 Fitting Property Data

The entire p-h domain of the zeotropic refrigerant properties domain can be divided into four regions: 1. subcooled, 2. two phase, 3. super-heated and 4. super-critical (Aute and Radermacher 2014). These will be referred to as Regions 1 - 4 henceforth. The state of a binary refrigerant mixture can be defined by three independent properties. In this article pressure (p), specific enthalpy (h) and mixture independent composition (Xi) are used as the independent properties. The properties like density, specific heat capacity, thermal conductivity which cover all the four regions are functions of three variables. The properties on the saturation line like bubble enthalpy, dew enthalpy, bubble density, dew density and so on, require only two independent variables. Finally, critical properties like critical pressure and critical temperature will be fitted to a 1D Chebyshev polynomial.

The method will be demonstrated with a mixture combination of R23-R134a which is commonly used in auto-cascade refrigeration systems as seen in several publications(Du et al. 2009; Bai, Yan, and Yu 2019; He et al. 2022). The mixture combinations for auto-cascade are such that the boiling points of the constituents are significantly different.

Python is used for the purposes of fitting. Numpy library provides fitting functions for 1D Chebyshev. Sampling the points to generate dataset from REFPROP is performed using Python's Scipy library (Virtanen et al. 2020). Finally, metrics like RMSE are obtained from functions in Scikit-learn (Pedregosa et al. 2011).

### 2.1 Data Generation

The datasets obtained from REFPROP are sanitized by removing anomalies and replacing them with smooth, physically consistent estimates. This approach enables fast, accurate, and robust simulations in Modelica.

#### 2.1.1 Critical Properties

Critical properties of the zeotropic refrigerant mixture are functions of its mixture composition. These are hence a function of a single independent variable and referred to as 1D functions. For generating 1D dataset for critical properties, 1000 points are sampled within the range 0-1 by latin hypercube sampling (LHS). The critical properties are calculated by taking this 1D LHS as mass composition of R23.

To fit the data using Chebyshev polynomials, it must first be scaled to the range [-1,1]. This scaling is performed using the `MinMaxScaler` from the Scikit-learn library. The optimal degree of the polynomial is not known a priori, so a parametric study is conducted by varying the degree from 1 to 25. A tolerance value of $10^{-8}$ is selected for the fitting process.

The results of the parametric study for critical pressure are shown in Figure 1, while results for all properties are summarized in Table 1. A single calculation of the critical pressure is not sufficient to represent the computational expense accurately. Therefore, a random set of 1000 values of Xi is used for benchmarking.

The median computation time to calculate the critical pressure using REFPROP is calculated to be 357.335 ms. This is the median of the same calculation repeated 14 times. The mean was 357.799 ms $\pm$ 4.306 ms. For each polynomial degree between 1 and 25 in the Chebyshev fit, the computational expense for the same evaluation is approximately 30 $\mu$s. For these cases, 10000 calculations were used. The standard deviation is of the order 30 $\mu$s which contributes to the jagged nature of the plot of computational expense. Nevertheless, it is evident that using Chebyshev polynomials yields a speedup of approximately four orders of magnitude. Similar acceleration is observed for the computation of critical temperature and critical density.

**Table 1.** Polynomial degree to meet tolerance.

| *Property* | *Degree* |
|---|---|
| Critical Pressure | 20 |
| Critical Temperature | 12 |
| Critical Density | 24 |

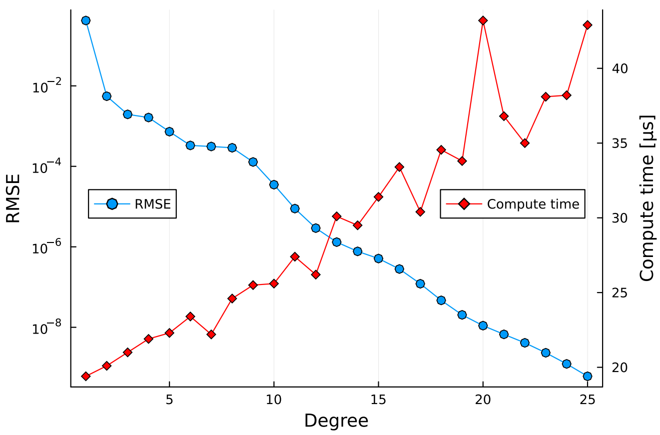


**Figure 1.** RMSE and compute time for Chebyshev polynomial fit as a function of its degree.

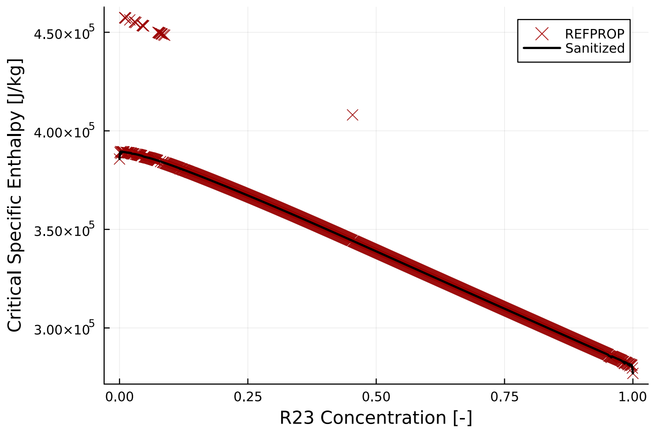


**Figure 2.** Sanitized values of specific enthalpy at critical points as function of R23 concentration

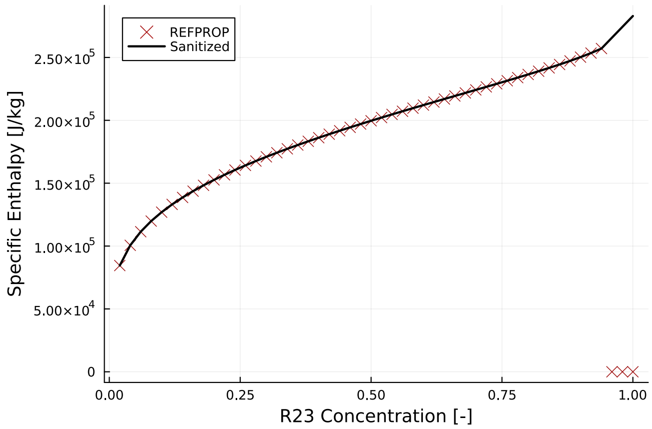


**Figure 3.** Sanitized values of bubble enthalpy as function of R23 concentration

### 2.1.2 Saturated Properties

The saturated properties, i.e. the properties which lie on the saturation line, require two independent variables as inputs. These are referred to as 2D properties in this article. Pressure and specific enthalpy are the most commonly used input variables for the properties of this type. To generate the dataset, a Latin Hypercube Sampling (LHS) of two variables of size 100 with values between 0 and 1 is created. Using these two columns of LHS, a total of 10000 is created as a tabular grid. The first variable of the LHS is assigned as the mixture concentration by mass. For pressure, the upper and lower ranges are not constant. This helps with filtering noisy data by observing property trends.

The upper limit is the critical pressure of the refrigerant mixture at that concentration. Lower limit ($p_{min}$) determination is arbitrary. For the purpose of this article, the bubble pressure corresponding to a temperature of $-80°C$ for the given mixture concentration is taken as the lower limit. The second variable of the LHS is then used to scale the point in between these two limits. This is achieved by applying Equation 8 to each row. $x_i$ is $i^{th}$ sample of the second variable of the LHS.

$$p_i = p_{min,i} + x_i(p_{crit,i} - p_{min,i}) \tag{8}$$

Now that the pressure and mixture concentration is available, the following refrigerant properties on the vapor dome are calculated: bubble enthalpy, bubble temperature, bubble specific volume, bubble specific heat capacity at constant pressure, bubble thermal conductivity, bubble dynamic viscosity, surface tension, dew enthalpy, dew temperature, dew specific volume, dew specific heat capacity at constant pressure, dew thermal conductivity and dew dynamic viscosity.

As mentioned earlier, data obtained from REFPROP needs to be sanitized. The method used for the current article is described with an example of bubble enthalpy. The bubble enthalpy function depends on two independent variables - reduced pressure and Xi. A latin hypercube design of 100 rows and 2 columns can be used to obtain two vectors varying between 0 and 1. Specific enthalpy of critical points (Hcrit) is a function of Xi. For Xi values starting from 0 to 1 with a step size of 0.001, Hcrit is calculated using REFPROP and shown in Figure 2. It can be observed that there are a few points (22 points out of 1001) which are not following the trend and need to be removed. A simple filter (value > 4e5) is used to eliminate these anomalous values. Linear interpolation is then used to estimate the values for these points to create the sanitized set for critical specific enthalpy.

The linear interpolation of critical specific enthalpy can be used to fill in data by interpolation for some points which fail to compute near the critical point. For the evaluation of 10000 point dataset, a for loop can be constructed which calculates the bubble enthalpy for all the values of reduced pressure at a given value of R23 concentration. For example, for Xi = 0.98 the data is shown in Figure 3. A try-catch block is used and for the failed simulation cases near critical point a value of 0 is provided. These error points can be then easily filtered out (value < 1000). The final point in the interpolation data is for reduced pressure of 1 which corresponds to the critical specific enthalpy for the given Xi. This can be obtained from the interpolation of Figure 2. The removed points can now be added by interpolation since the end point is now available. This is shown in Figure 3. The method is repeated for all the values of Xi to generate the dataset.

Polynomials of degree 16 are used to create curve-fits of the dataset. It is observed that using natural logarithm of the pressure as a feature instead of absolute pressure provides a better fit in some cases. Standard scaling is also used since the two inputs have different orders of magnitude. Several statistics along with visual comparison are used to observe the goodness of the fit. An example plot showing the quality of the fit is shown in Figure 4. In this plot, the values obtained from REFPROP are on the x-axis while curve fits obtained are on the y-axis. Table showing

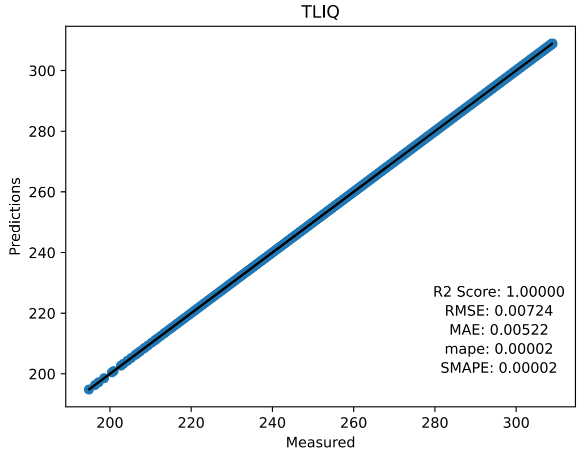


**Figure 4.** Checking goodness of polynomial fit for bubble temperature

the list of all the 2D properties and their inputs is shown in Table 2.

### 2.1.3 Regular Properties

The properties of the refrigerant mixture like specific entropy, density, etc. cover all the regions. These properties are referred to as 3D properties in this article. A dataset is created covering 10000 samples in 3 variables in each of the regions.

For the sub-cooled region (Region 1), the mixture composition by mass is obtained directly from one of the three columns of the 3x10000 LHS. Pressure is again obtained using the second variable and the application of Equation 8. To calculate specific enthalpy in the region, the range of enthalpy values and the third variable of the sampling are used. The maximum value of enthalpy possible for the region is the bubble enthalpy $h_{bub}$. The lower limit for enthalpy $h_{min}$ is the value for which the subcooling is $-50°$C for the given value of pressure and the mixture concentration. The enthalpy values can now be calculated using an equation similar to Equation 8 for these limits.

For the two phase region (Region 2), the mixture composition and pressure are taken from region 1. The third variable of sampling is used as the refrigerant quality to calculate the enthalpy in the two phase region using the bubble and dew enthalpy.

For the superheated region (Region 3), the mixture composition and pressure are taken from region 1. The minimum value of enthalpy is the dew enthalpy. The maximum value of enthalpy $h_{max}$ is obtained by taking a superheat value of $+50°$C at the given pressure and mixture composition. An equation of the form Equation 8 for these enthalpy limits is then used to calculate the specific enthalpy in Region 3.

For the supercritical region (Region 4), the mixture composition is obtained from the first variable of LHS. The pressure is obtained by using Equation 9. The maximum pressure is 1.5 times the critical pressure. The en-


DOI 10.3384/ecp218303

thalpy is sampled by calculating temperature in the range of ±50°C of the critical temperature. The enthalpy is calculated using the pressure and this sampled temperature.

$$p_i = p_{crit,i} + x_i(p_{max,i} - p_{crit,i}) \tag{9}$$

Now with pressure, specific enthalpy and mixture composition available in each of the regions, temperature, specific entropy, specific volume, specific heat capacity, thermal conductivity, dynamic viscosity and speed of sound are calculated using REFPROP. A regular polynomial of degree 8 is used to create curve-fits for the dataset. Several regions are combined to create a curve-fit if the accuracy is good with combined datasets. However, only specific entropy is observed to have good accuracy (less than 1% mean absolute percentage error) for all four regions with a single 8 degree polynomial. In more typical cases, multiple polynomial fits of degree 8 are created and then a spline interpolation is performed at the border of the regions.

# 3 Media Class

The `Media` package in `Modelica Standard Library` provides examples for creating media. The refrigerant mixture media is creating by extending the class `Modelica.Media.Interfaces.PartialMedium`. In the extended class the `mediumName` and `substanceName` parameters are defined with string *R23|R134a* and two element string array *"R23","R134a"* respectively.

Since the mixture comprises of multiple components, `singleState` is false. It is preferable to work with only the independent mass fractions of the mixture since they sum to one. For this reason, `reducedX` is set as true. Finally, since mixture composition is expected to change during simulation, `fixedX` is set to false. The `ThermoStates` parameter is set to `phX` to define pressure, specific enthalpy and mixture composition by mass as the three independent variables defining the thermodynamic state of the media. The `BaseProperties` class is defined as shown in Listing 1.

**Listing 1.** Definition of the base class BaseProperties

```
redeclare replaceable model extends
    BaseProperties(
  h(stateSelect=if preferredMediumStates
      then StateSelect.prefer else
        StateSelect.default),
  p(stateSelect=if preferredMediumStates
      then StateSelect.prefer else
        StateSelect.default),
  Xi(each stateSelect=if
      preferredMediumStates then
      StateSelect.prefer
          else StateSelect.default),
  final standardOrderComponents=true)
  "Base properties of medium"
equation
  d =density_phX(
      p,
      h,
      X);
  T =temperature_phX(
      p,
      h,
      X);
  u = h - p/d;
  MM = 1/(Xi[R23]/MMX[R23] + (1.0 - Xi[
      R23])/MMX[R134a]);
  R_s = Modelica.Constants.R/MM;
  state = setState_phX(p,h,X);
```

Now that the basic structure of the media is setup, the property functions obtained from curve-fits can be added to it.

## 3.1 1D Functions

The critical pressure, critical temperature and critical density are not constant for the refrigerant mixture. These properties are functions of the mixture composition. The independent composition of the first component (R23 in this case) is referred to as Xi. The composition array of both the components is referred to by the variable X. The functions are defined as a function of X.

The recurrence relationship to calculate the Chebyshev (Equation 5), is coded as a function. This is used to create a Vandermonde matrix. A dot product of the coefficient vector with the Vandermonde matrix gives the value of the output. The implementation is similar to the numpy library (Harris et al. 2020). The inputs and outputs for Chebyshev functions are scaled to the range [-1,1] and then scaled back to output using `MinMaxScaler` (Pedregosa et al. 2011).

## 3.2 2D Functions

Recall that the 2D functions represent variables on the vapor dome. These are represented by two independent variables, typically pressure and composition. Standard scaling (Pedregosa et al. 2011) is applied to the inputs and then a dot product between the polynomial terms and the coefficients obtained from curve-fits is done to obtain the desired property. The functions created with this method as listed in Table 2.

The bubble point is the temperature at which a liquid creates the first bubble of vapor and begins the process of boiling. The dew point is the temperature at which the first drop of dew is formed from vapor commencing the condensation of the liquid. The bubble enthalpy is also referred to as liquid point enthalpy and dew enthalpy is also referred to as vapor point enthalpy in literature. The authors have preferred to use bubble and dew points since liquid and vapor enthalpy may also refer to those of subcooled or superheated phases causing confusion to readers.

**Table 2.** Saturation properties and their inputs.

| *Property* | *Input 1* | *Input 2* |
|---|---|---|
| bubbleTemperature | p | X |
| dewTemperature | p | X |
| bubbleEnthalpy | p | X |
| dewEnthalpy | p | X |
| bubbleDensity | p | X |
| dewDensity | p | X |
| bubbleCp | p | X |
| dewCp | p | X |
| bubbleThermalConductivity | p | X |
| dewThermalConductivity | p | X |
| bubbleViscosity | p | X |
| dewViscosity | p | X |
| bubblePressure | T | X |
| dewPressure | T | X |
| bubbleXi | T | X |
| dewXi | T | X |

### 3.3 3D Functions

Finally, the property functions which work in all the regions of the refrigerant mixture are added to the media package. The complete list of these functions is seen is Table 3. These are the functions available in the `PartialMedium` class and defined in the media with the 8 order polynomial.

**Table 3.** Saturation properties and their inputs.

| *Property* | *Input 1* | *Input 2* | *Input 3* |
|---|---|---|---|
| temperature | p | h | X |
| density | p | h | X |
| dynamicViscosity | p | h | X |
| thermalConductivity | p | h | X |
| specificEntropy | p | h | X |
| specificHeatCapacityCp | p | h | X |
| specificHeatCapacityCv | p | h | X |
| velocityOfSound | p | h | X |
| specificEnthalpy | p | s | X |
| temperature | p | s | X |
| density-derh-p | p | h | X |
| density-derp-h | p | h | X |
| density-derX | p | h | X |

Except for specific entropy, it is not possible to obtain a curve-fit with accuracy of 1% mean absolute percentage error with a single polynomial fitting all the regions. In these cases, polynomial fits are performed over single or two or three regions which meet the accuracy target. Then spline interpolation is done at the border of these regions. The spline function used is similar to the `SpliceFunction` in `Modelica.Media.Air.MoistAir.Utilities`. Separation of the transcritical region from the remaining is done with reduced pressure ($p/pcrit$). Then in an inner nested loop, the refrigerant quality is used to separate the remaining regions.

The density calculation is critical because it directly influences the conservation equations and can lead to non-physical scenarios if inaccurate. Additionally, density derivatives are required for time-stepping the state variables. To address this, a 3D lookup table is employed for density computation. The implementation is achieved using the `DataFiles` library by using the functions from `TableND` package. The grid variables for the table are reduced pressure, pseudo vapor quality, and Xi. Pseudo vapor quality is computed using the vapor quality formula but is not constrained between 0 and 1. Consequently, liquid enthalpy values may be negative, and vapor enthalpy values may exceed 1.0. The pseudo vapor quality range used spans from -0.5 to 1.5. Table of size 50 $\times$ 41 $\times$ 21 is used for the purpose with dimensions for the three independent variables respectively.

The partial derivatives of density are obtained by finite difference method with symmetric difference quotient. Equation 10 shows the implementation for the density derivative with specific enthalpy with pressure and mixture composition constant.

$$\left(\frac{\partial \rho}{\partial h}\right)_{p,Xi} \approx \frac{\rho(h+\delta h)-\rho(h-\delta h)}{2\delta h} \tag{10}$$

## 4 Model Development

With the media package available it is now possible to run simulation. The `Fluid` subpackage in `Modelica Standard Library` contains several component models which can be used to run simulations.

### 4.1 Simulating Concentration Change

Figure 6 shows a model to simulate change in concentration of the mixture with time. The `MassFlowSource-h` and `Boundary-ph` blocks are taken from `Modelica.Fluid.Sources`. Inlet enthalpy of 300e3 kJ/kg and mass flow rate of 0.1 kg/s are provided to the mass flow source block. The mass flow source block exposes the concentration as an input. The ramp block sets up the variation of R23 composition in mixture from 0.2 to 0.7 during the simulation. Together with the constant block with input 1 and add block, the composition array X of {Xi, 1-Xi} is provided. A pressure of 10 bar is set in the boundary.

### 4.2 Simulating Fixed Temperature

An important benefit of Modelica is designing of controllers. For the purpose of demonstration, a simple simulation including a PID controller is created. As in the previous example from 4.1, let us say we want to maintain the temperature to a fixed value. This can be achieved by changing the specific enthalpy.


DOI 10.3384/ecp218303

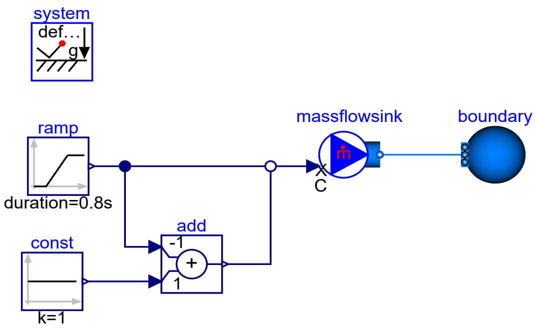


**Figure 5.** Simulating changing concentration of refrigerant mixture

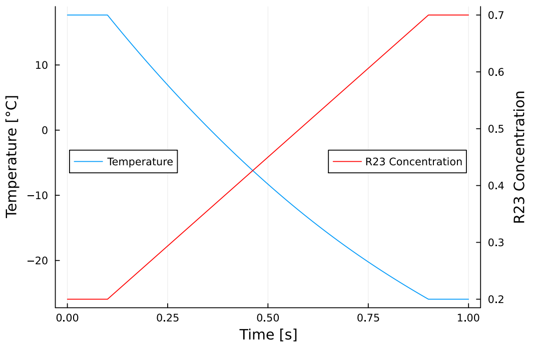


**Figure 6.** Temperature change of mixture with change of mixture composition

This model is setup as shown in Figure 7. A temperature sensor model is used to obtain the measured value for the PID. Setpoint value for the PID block is provided by a constant block. The value is set to 300 K.

The results can be seen in Figure 8. It can be observed that the PID is changing the enthalpy to obtain the setpoint temperature. As seen in Figure 6, the temperature of refrigerant mixture would have changed with the change in concentration. But by adjusting the specific enthalpy, it is maintained to the setpoint.

## 4.3 Flow Models

Thermal systems have models which determine refrigerant mass flow rate. A valve is an example of this type of model. There is no change required for these kind of models because the mass conservation, conservation of mixture constituents and energy conservation are trivial in these components. Valve models from the `Modelica.Fluid.Valves` work as is with the new media.

## 4.4 Lumped Refrigerant Control Volume

The components which model the control volumes with mass and energy conservation need some changes for handling refrigerant mixture. Since the concentration of mixture is also an independent variable, the density is a function of three parameters - p, h, Xi. So the density derivatives used to transform the mass and energy conservation have to include a partial derivative with Xi. For the purpose of this derivation, we will assume that the control volume has one inlet and one outlet. This assumption does not affect the derivation in any way but helps to understand it better.

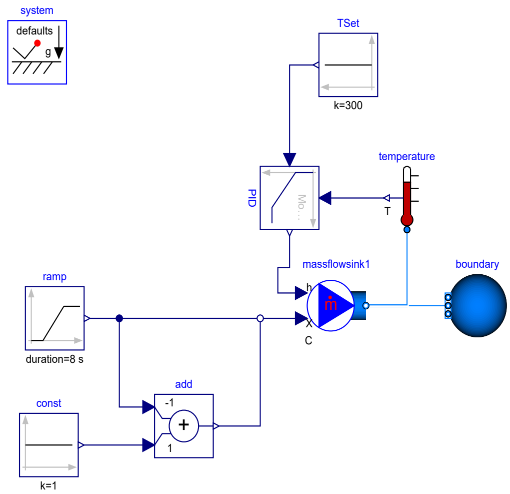


**Figure 7.** Simulating a simple controller to maintain temperature

### 4.4.1 Mass Conservation

The conservation of mass for this control volume with a volume $V$ is given by Equation 11.

$$V\frac{d\rho}{dt} = \dot{m}_{in} - \dot{m}_{out} \tag{11}$$

Since density is a function of three independent variables, the derivative of density with time can be written in terms of its partial derivatives.

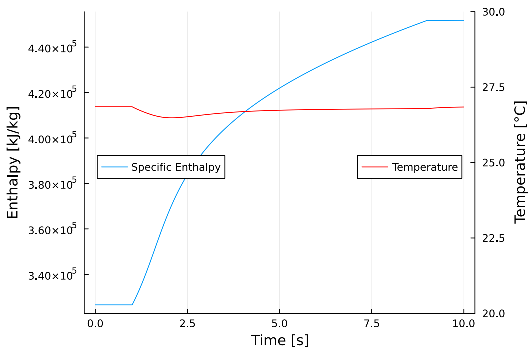


**Figure 8.** Controller to maintain a fixed temperature with change of mixture composition

$$V\left(\left.\frac{\partial\rho}{\partial p}\right|_{h,X}\frac{dp}{dt}+\left.\frac{\partial\rho}{\partial h}\right|_{p,X}\frac{dh}{dt}+\left.\frac{\partial\rho}{\partial X}\right|_{p,h}\frac{dX}{dt}\right)=\dot{m}_{in}-\dot{m}_{out} \tag{12}$$

This form can be evaluated with the availability of the partial derivatives calculated as shown in Equation 10.

### 4.4.2 Energy Conservation

Energy equation equation for the control volume can be written as below Equation 13.

$$\frac{d(\bar{m}\bar{u})}{dt}=\dot{m}_{in}h_{in}-\dot{m}_{out}h_{out} \tag{13}$$

Specific internal energy and specific enthalpy are related by:

$$\bar{u}=\bar{h}-\frac{p}{\rho} \tag{14}$$

Substituting for specific internal energy

$$\frac{d\left(\bar{m}\left(\bar{h}-\frac{p}{\bar{\rho}}\right)\right)}{dt}=\dot{m}_{in}h_{in}-\dot{m}_{out}h_{out} \tag{15}$$

For a control volume with a fixed volume.

$$V=\frac{\bar{m}}{\bar{\rho}} \tag{16}$$

Substituting Equation 16 into Equation 15. We obtain

$$\frac{d(\bar{m}\bar{h}-Vp)}{dt}=\dot{m}_{in}h_{in}-\dot{m}_{out}h_{out} \tag{17}$$

Since volume is fixed:

$$\frac{d(\bar{m}\bar{h})}{dt}-V\frac{p}{dt}=\dot{m}_{in}h_{in}-\dot{m}_{out}h_{out} \tag{18}$$

Using Chain rule of Calculus

$$\left(\bar{m}\frac{d\bar{h}}{dt}+\bar{h}\frac{d\bar{m}}{dt}\right)-V\frac{p}{dt}=\dot{m}_{in}h_{in}-\dot{m}_{out}h_{out} \tag{19}$$

We can use conservation of mass to substitute of $\frac{d\bar{m}}{dt}$

$$\bar{m}\frac{d\bar{h}}{dt}+\bar{h}(\dot{m}_{in}-\dot{m}_{out})-V\frac{p}{dt}=\dot{m}_{in}h_{in}-\dot{m}_{out}h_{out} \tag{20}$$

Simplifying

$$\bar{m}\frac{d\bar{h}}{dt}-V\frac{p}{dt}=\dot{m}_{in}(h_{in}-\bar{h})-\dot{m}_{out}(h_{out}-\bar{h}) \tag{21}$$

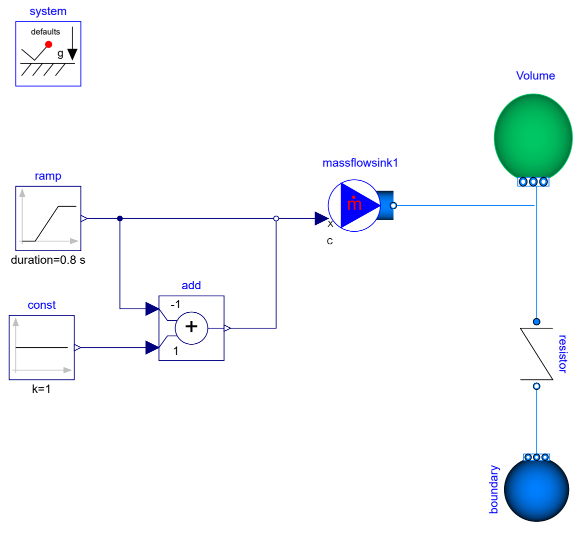


**Figure 9.** Control volume model simulation

### 4.4.3 Mixture Conservation

The conservation of the constituents of the mixture can be written as:

$$\frac{d(\bar{m}\bar{X}i)}{dt}=\dot{m}_{in}Xi_{in}-\dot{m}_{out}Xi_{out} \tag{22}$$

Using Chain rule of calculus:

$$\bar{m}\frac{d(\bar{X}i)}{dt}+\bar{X}i\frac{d(\bar{m})}{dt}=\dot{m}_{in}Xi_{in}-\dot{m}_{out}Xi_{out} \tag{23}$$

Using conservation of mass Equation 12

$$\bar{m}\frac{d(\bar{X}i)}{dt}+\bar{X}i(\dot{m}_{in}-\dot{m}_{out})=\dot{m}_{in}Xi_{in}-\dot{m}_{out}Xi_{out} \tag{24}$$

Finally, we obtain

$$\bar{m}\frac{d(\bar{X}i)}{dt}=\dot{m}_{in}(Xi_{in}-\bar{X}i)-\dot{m}_{out}(Xi_{out}-\bar{X}i) \tag{25}$$

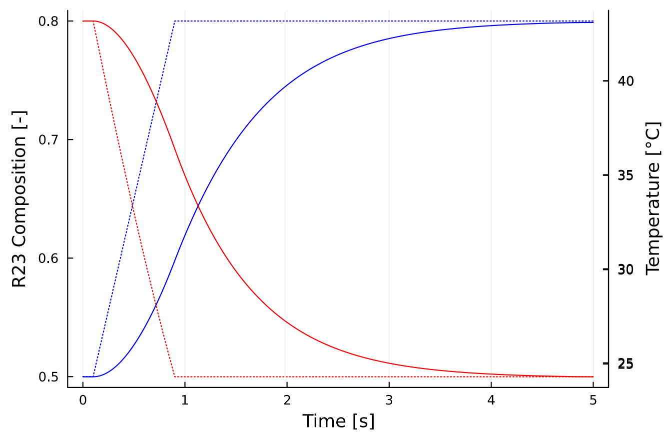


**Figure 10.** Control volume model simulation

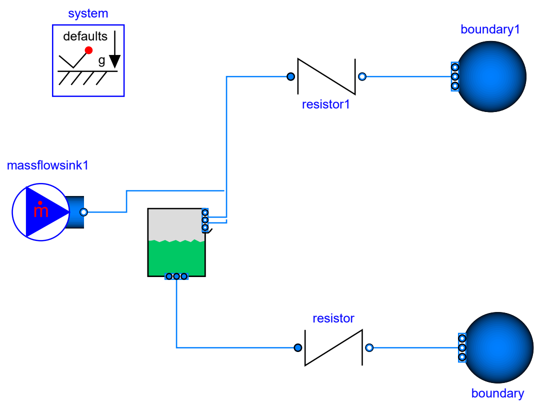


**Figure 11.** Separator tank model simulation

### 4.4.4 Model Simulation

The volume model is connected to `MassFlowSource-h` and `Boundary-ph`. The simulation model can be seen in Figure 9. A flow resistor is added between the volume and boundary. The model is simulated with varying refrigerant concentration. The refrigerant state of the lumped control volume is different from the inlet state of the refrigerant entering from `MassFlowSource-h`. The intention is to check that after the initial transients the two states become equal.

The results can be seen in Figure 10. The dotted lines are the refrigerant mixture state values from `MassFlowSource-h` while the solid lines are those from the `Volume`. It can be seen that the values converge after the initial transients.

## 4.5 Separator Tank

The separator tank is a key component of an auto-cascade refrigeration system. It is a specialized implementation of the lumped refrigerant control volume. A separator tank has an inlet port, a vapor outlet port and a liquid outlet port. When the liquid level is between 5% and 95% the vapor outlet port and liquid outlet port will provide pure vapor and pure liquid respectively. If the liquid level is at 0% or 100%, both outlets will provide vapor or liquid respectively. Between these values a mixed two phase flow is provided depending on a spline interpolation.

The liquid level (L) is calculated by the following equation:

$$L = \frac{1-q}{q \cdot \frac{\rho_{\text{liq}}}{\rho_{\text{vap}}} + (1-q)} \tag{26}$$

The liquid level is compared with the thresholds for dryout and flooded conditions (defaults 5% and 95%). Within the safe operating range, the outlet enthalpy at the vapor port corresponds to the dew enthalpy at the given pressure and mixture composition. For the liquid port, the bubble enthalpy at the same pressure and composition is used. For the mixture concentrations at the vapor port and liquid port, the functions `dew-Xi` and `bubble-Xi` which are functions of pressure and temperature are used. These were obtained from curve-fits of 2D REFPROP data. Since refrigerant vapor and liquid will be at thermodynamic equilibrium, they will have the same temperature and pressure. This will be equal to the temperature and pressure of the lumped control volume refrigerant mixture state.

Figure 11 shows the model with separator tank. The inlet port and vapor port can be connected into the connector array. There are two separate sinks and flow resistors for the vapor and liquid streams. The nominal flow rates in the flow resistors is adjusted to obtain points which do not have dryout or liquid overflow.

The separator tank model is initialized at 30 bar pressure, 250 kJ/kg specific enthalpy and a mixture composition of { 0.3, 0.7 }. The inlet refrigerant flow has 30 bar pressure, 350 kJ/kg specific enthalpy and a mixture composition of { 0.5, 0.5 }.

The results are shown in Figure 12, which displays the liquid level, mass flow rates at different ports, vapor quality, and mixture composition. We can observe that the vapor outlet has output of vapor and liquid outlet has quality close to that of liquid. The R23 composition is higher in the vapor phase than in the liquid phase, indicating effective mixture separation. The R23 composition changes very slowly during the simulation and appears nearly constant in the plot. Note that the separator tank and selected mass flow rates may not be optimally sized for the simulation.

# 5 Conclusions and Future Work

The article presented a method for simulating thermodynamic systems in Modelica, specifically addressing scenarios where the refrigerant mixture composition changes dynamically during operation. A detailed description of the media package creation was provided, with particular attention to accuracy and computational efficiency. The approach enables the use of component models from the `Modelica Standard Library` to perform simulations with the developed media package. The value of the modeling is demonstrated by the implementation of controllers designed to regulate refrigerant concentration in the mixture to achieve desired performance. A key component of the system, the separator tank used in auto-cascade refrigeration, was modeled in detail. Simulation results showed qualitative agreement with expected qualitative behavior.

The current work is part of an ongoing effort to simulate a complete auto-cascade refrigeration cycle. This will require the development of additional component models like compressor and heat exchangers which can work with the newly created refrigerant mixture media package. There is a lot of scope for enhancing the speed

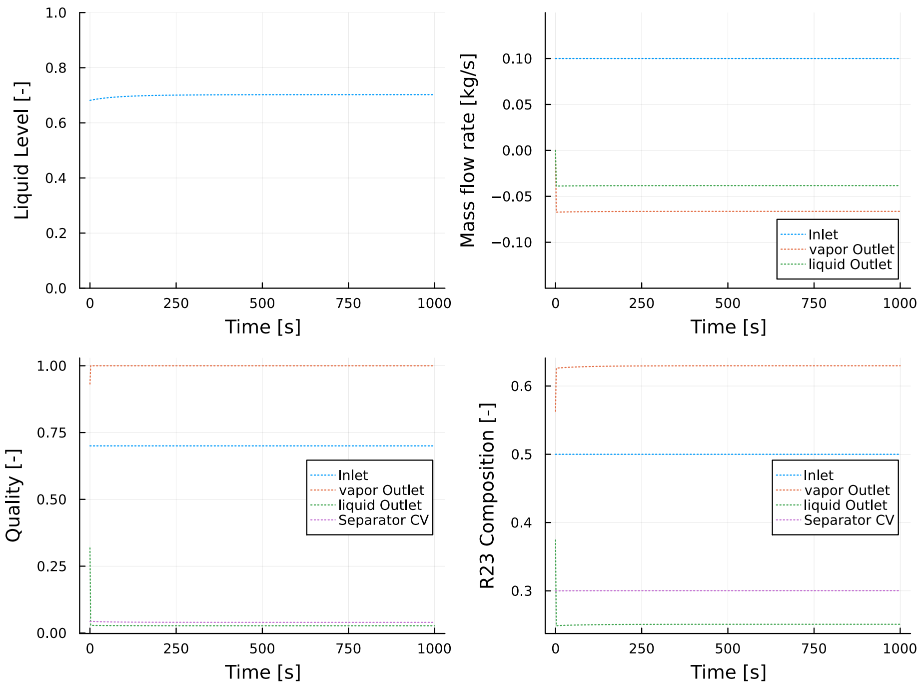


**Figure 12.** Results of separator tank model (a) Liquid Level (b) Mass Flow Rates (c) Quality (d) R23 Composition.

and accuracy of zeotropic refrigerant property evaluations. One promising direction is the application of multivariable Chebyshev polynomials to improve computational performance. To support applicability to other refrigerant mixture combinations, future developments will also address the automatic removal of erroneous REFPROP data.

## Acknowledgements

The authors would also like to thank Network of Excellence, Trane Commerical HVAC and ETC-B for the support of this work.

## References

Aute, Vikrant and Reinhard Radermacher (2014). "Standardized polynomials for fast evaluation of refrigerant thermophysical properties". In: *International Refrigeration and Air Conditioning Conference* Paper 1499.

Bai, Tao, Gang Yan, and Jianlin Yu (2019). "Experimental investigation on the concentration distribution behaviors of mixture in an ejector enhanced auto-cascade refrigeration system". In: *International Journal of Refrigeration* 99, pp. 145–152.

Du, Kai et al. (2009). "A study on the cycle characteristics of an auto-cascade refrigeration system". In: *Experimental Thermal and Fluid Science* 33.2, pp. 240–245.

Harris, Charles R et al. (2020). "Array programming with NumPy". In: *Nature* 585.7825, pp. 357–362.

He, Yongning et al. (2022). "Theoretical performance comparison for two-stage auto-cascade refrigeration system using hydrocarbon refrigerants". In: *International Journal of Refrigeration* 142, pp. 27–36.

Laughman, Christopher et al. (2024). "Enhancing Thermodynamic Data Quality for Refrigerant Mixtures: Domain-Informed Anomaly Detection and Removal". In: *International Refrigeration and Air Conditioning Conference*, Paper 2595. DOI: https://docs.lib.purdue.edu/iracc/2595.

Lemmon, E. W. et al. (2018). *NIST Standard Reference Database 23: Reference Fluid Thermodynamic and Transport Properties-REFPROP, Version 10.0, National Institute of Standards and Technology*. DOI: https://doi.org/10.18434/T4/1502528. URL: https://www.nist.gov/srd/refprop.

Li, Lixiang et al. (2018). "Fast calculation of refrigerant properties in vapor compression cycles using spline-based table look-up method (SBTL)". In: *The American Modelica Conference, Cambridge, MA, USA*, pp. 77–84.

Pedregosa, F. et al. (2011). "Scikit-learn: Machine Learning in Python". In: *Journal of Machine Learning Research* 12, pp. 2825–2830.

Virtanen, Pauli et al. (2020). "SciPy 1.0: fundamental algorithms for scientific computing in Python". In: *Nature methods* 17.3, pp. 261–272.