International Journal of Refrigeration 189 (2026) 106990

Contents lists available at ScienceDirect

International Journal of Refrigeration

journal homepage: www.elsevier.com/locate/ijrefrig

INSTITUT INTERNATIONAL DU FROID
INTERNATIONAL INSTITUTE OF REFRIGERATION

Approximation of refrigerant thermophysical properties using neural networks to speed up transient thermofluid simulations ⋆

Jiacheng Ma [a,*], Matthis Thorade [b], Donghun Kim [c]

[a] Carrier Global Corporation, Plam Beach Gardens, FL, USA
[b] Modelon Deutschland GmbH, Hamburg, Germany
[c] Building Technology & Urban Systems Division, Lawrence Berkeley National Laboratory, Berkeley, CA, USA

ARTICLE INFO

Keywords:
Thermophysical properties
Derivative-informed neural networks
Modelica
Transient simulations
Analytic Jacobian
Two-phase flow

ABSTRACT

Accurate and efficient evaluations of refrigerant thermophysical properties and their partial derivatives are essential for transient simulations of thermofluid systems, where several computations need to be executed at each integration time step. Since the utilization of an Equation of State for retrieving properties based on a pair of independent inputs typically involves numerical iterations in solution procedures, when the input variables differ from the refrigerant state variables employed in dynamic models, a variety of approaches including lookup table interpolation and curve fitting have been developed to explicitly approximate these properties based on the state variables, and consequently eliminate internal iterations. This paper presents an alternative method that exploits derivative-informed neural networks to model refrigerant properties explicitly from inputs of pressure and enthalpy, while ensuring consistent partial derivatives generated by differentiating the neural networks. Computational speed and accuracy of the proposed approach are demonstrated via transient simulations of a discretized heat exchanger model in Modelica, and comparisons against other property evaluation routines. Simulation results indicate that the proposed approach can realize a significant speedup with negligible discrepancies in predicted transients. The method is implemented in an open-source Modelica library.

1. Introduction

High-fidelity physics-based transient models play a critical role in control design workflows for a wide range of two-phase thermofluid applications. These models yield extensive capabilities in predicting complicated two-phase flow behaviors of vapor compression and expansion systems with a discretization of governing conservation equations, however, often at the expense of substantial computational complexities and slow simulations due to model descriptions by a large set of nonlinear differential algebraic equations (Li et al., 2014). In particular, computations associated with evaluating refrigerant thermophysical properties can consume a significant portion of the simulation time, as a number of thermodynamic and transport properties as well as their derivatives with respect to state variables need to be retrieved for solving conservation equations (i.e., mass, energy and momentum) associated with each discretized control volume at every integration time step.

As a consequence, the choice of fluid property models constitutes a prominent factor in the overall simulation performance regarding speed and robustness. Methods based on the fundamental Equation of State (EOS), that solves for properties from fittings of Helmholtz or Gibbs free energy, are widely employed in modeling refrigerant cycles for their high accuracy and comprehensive coverage of refrigerants (Thorade and Saadat, 2013). The EOS based property models and solution algorithms for retrieving thermophysical properties from various pairs of independent states have been implemented in databases such as REFPROP (Huber et al., 2022) and CoolProp (Bell et al., 2014), and are available through wrappers for a number of software tools. Nevertheless, the utilization of EOS based methods in transient simulations typically involves numerical iterations realized in a Newton's method manner during solution procedures. This is due to a mismatch of state variables (e.g., pressure and enthalpy) that describe dynamics of the refrigerant flow, with input variables to the EOS (e.g., density and temperature). As a result, the computational complexities associated with iterative solution procedures for retrieving thermophysical properties can lead to simulation speed slowdown as well as deterioration of the overall model robustness in case it fails to converge for EOS solutions.

To ease the barrier for practical deployment of transient models in design and control applications, a variety of approaches have thus been

⋆ This article is an extended version of a conference paper presented at the 7th IIR Conference on Thermophysical Properties and Transfer Processes of Refrigerants: Fast evaluation of refrigerant thermophysical properties using neural networks for transient simulations. http://dx.doi.org/10.18462/iir.tptpr2025.1121

* Corresponding author.
E-mail address: ma516@purdue.edu (J. Ma).

https://doi.org/10.1016/j.ijrefrig.2026.106990
Received 6 January 2026; Received in revised form 26 May 2026; Accepted 27 May 2026
Available online 5 June 2026

Nomenclature

Symbols

$\dot{m}$	Mass flow rate [$kg\,s^{-1}$]
$\dot{Q}$	Heat transfer rate [W]
A	Area [m^2]
c_p	Specific heat capacity at constant pressure [$J\,kg^{-1}$]
c_v	Specific heat capacity at constant volume [$J\,kg^{-1}$]
h	Specific enthalpy [$J\,kg^{-1}$]
p	Pressure [Pa]
T	Temperature [K]
V	Volume [m^3]

Greek letters

α	Heat transfer coefficient [$W\,m^{-2}K^{-1}$]
η	Dynamic viscosity [Pa s]
λ	Thermal conductivity [$W\,m^{-1}K^{-1}$]
ρ	Density [$kg\,m^{-3}$]

Subscript

in	Inlet
out	Outlet
w	Heat exchanger wall

proposed to explicitly approximate thermophysical properties based on a pair of independent properties. For example, pressure and enthalpy (p, h) are commonly selected as state variables for two-phase flow modeling, and are used to determine other properties. In general, these approximation methods can be classified into two categories: precomputed lookup table interpolation and curve fitting.

The former class of methods perform interpolation over the coordinates of state variables using precomputed lookup tables generated from property databases that rely on the EOS backend. Namely, the CoolProp software implemented two interpolation schemes for fast property evaluations based on Taylor series and bicubic functions from inputs of pressure and enthalpy. Van and Laughman (2018) applied B-splines to interpolate refrigerant thermophysical properties over a (p, h) domain in Modelica, and suggested a speedup about 100 times over a REFPROP-like implementation. In terms of interpolation based methods utilized in transient simulations, an early attempt was found in Bendapudi (2004) where spline interpolation was applied with lookup tables generated from EES® (Engineering Equation Solver) to compute refrigerant and water thermophysical properties for dynamic modeling of a centrifugal chiller. Similarly, Li et al. (2018) implemented spline interpolation over lookup tables in Modelica. Computational benefits of the implementation were demonstrated by comparing accuracy and speed of test benches ranging from function calls to complete vapor compression system models, against the Helmholtz energy EOS approach. It was shown that half of CPU time can be saved using the interpolation-based property models for an air conditioning system simulation. Bortoff et al. (2024) proposed a modified spline-based interpolation scheme by using polar and parabolic coordinate transformations to align one coordinate with the saturation curve, which led to a precise representation of fluid properties and derivatives at the saturation curve for improved numerical efficiency.

On the other hand, curve-fitting approaches generate functions with specific formulas to approximate curves or surfaces that are originally represented by the EOS models. Coefficients of those functions are fitted using property data. Ding et al. (2005) described a fitting procedure for implicit polynomials of properties sourced from REFPROP. Explicit formulations were subsequently obtained as analytical solutions of the implicit polynomials. It was revealed that the calculation speed could be up to 1000 times than REFPROP with minor discrepancies. Al Khoury et al. (2024) implemented the implicit fitting approach in Modelica and examined simulation speedup by comparing against the CoolProp EOS backend. It was found that utilizing the implicit fitting method to approximate refrigerant properties accelerated simulations of a cascade refrigeration cycle by 30 times. Aute and Radermacher (2014) proposed using Chebyshev rational polynomials to retrieve each property as a function of pressure and enthalpy. Comparisons for property calls as well as heat exchanger simulations indicated that the proposed method was more than two orders of magnitude faster than REFPROP. Unlike the lookup table interpolation approaches that rely on precomputed data tables, it is straightforward to implement curve fitting models regardless of modeling languages or tools with less memory storage requirements. However, a critical oversight associated with the curve fitting approaches reported in the literature was that consistency between the predicted properties and their derivatives with respect to inputs was not enforced or concerned.

This paper presents an alternative method that exploits neural networks (NN) to model pure and pseudo-pure refrigerant thermophysical properties explicitly from inputs of pressure and enthalpy. In particular, derivative-informed neural networks that can enforce consistency between predicted properties and their partial derivatives with respect to inputs are considered. NN models are trained for single-phase (i.e., liquid and vapor) and saturated (i.e., bubble and dew) properties. After that, consistent derivatives can be generated by differentiating the NN models. Two-phase properties and their partial derivatives are subsequently determined using the predicted saturated properties/derivatives with a homogeneous assumption. With properties and derivatives approximated for each phase region, a Fuzzy modeling approach is presented to enable smooth phase transitions. The proposed method was implemented in an open-source Modelica library (Ma, 2025), which includes examples of the refrigerant R410A and R134A as well as test benches of function calls and heat exchanger simulations for performance demonstration. Computational speed and accuracy are benchmarked against CoolProp EOS (Helmholtz energy) as well as bicubic interpolation backends.

The remainder of this paper is structured as follows: Section 2 describes the proposed methodology; Section 3 presents implementation for the near-azeotropic refrigerant mixture R410A as an illustration; Section 4 reports simulation results that demonstrate computational performance of the proposed approach; conclusions of the present work are summarized in Section 5.

2. Methodologies

2.1. Properties evaluations in dynamic thermofluid models

In general, a discretized two-phase dynamic pipe model can be described by a set of mass and energy balances for fixed control volumes (Ma et al., 2023). Pressure and enthalpy of the working fluids are often selected as state variables, which lead to formulations for each control volume shown in Eqs. (1)-(2)

$$V\left[\frac{\partial\rho}{\partial p}\bigg|_h\frac{dp}{dt}+\frac{\partial\rho}{\partial h}\bigg|_p\frac{dh}{dt}\right]=\dot{m}_{in}-\dot{m}_{out}, \tag{1}$$

$$V\left[\left(h\frac{\partial\rho}{\partial p}\bigg|_h-1\right)\frac{dp}{dt}+\left(h\frac{\partial\rho}{\partial h}\bigg|_p+\rho\right)\frac{dh}{dt}\right]=\dot{m}_{in}h_{in}-\dot{m}_{out}h_{out}+\alpha A_w(T_w-T) \tag{2}$$

where the heat transfer coefficient α is obtained by empirical correlations typically dependent on properties of specific heat capacity c_p, thermal conductivity λ and dynamic viscosity η. The governing equations indicate that a set of thermodynamic properties and their partial derivatives, as well as transport properties, are necessary for evaluating transient behaviors. Furthermore, saturated properties and some other (e.g., entropy) absent in the equations need to be evaluated as well to complete simulation models.

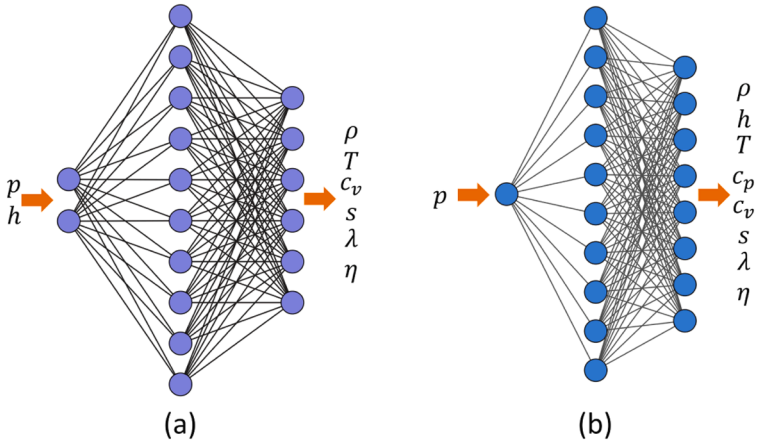

Fig. 1. Neural networks for predicting thermophysical properties: (a) single-phase (b) saturated.

Consequently, the objective of property model development is to fit $\boldsymbol{y} = \boldsymbol{f}(\boldsymbol{x})$ where

$$\boldsymbol{x} = \begin{bmatrix} p & h \end{bmatrix}^T, \tag{3}$$

$$\boldsymbol{y} = \begin{bmatrix} \rho & T & c_v & s & \lambda & \eta \end{bmatrix}^T. \tag{4}$$

that accounts for all eligible phases of a fluid (e.g., liquid, two-phase, and vapor for subcritical properties) as well as saturation lines.

Based on this setup, the partial derivatives of density appeared in mass and energy balances can be readily retrieved from derivatives of the fitted model. Similarly, c_p can be obtained from the temperature partial derivative as

$$\frac{1}{c_p} = \left.\frac{\partial T}{\partial h}\right|_p \tag{5}$$

The remainder of this section aims to lay out algorithms to construct $\boldsymbol{f}(\cdot)$ utilizing derivative-informed neural networks and Fuzzy modeling.

2.2. NN saturated and single-phase models

Neural networks are universal function approximators that can be employed to generate maps of a certain set of outputs corresponding to specified inputs in a continuous domain. Calculations of thermophysical properties using neural networks fall under the category of curve-fitting methods. However, neural networks offer favorable numerical characteristics over the polynomial models mentioned in Section 1. Firstly, neural networks can efficiently approximate multi-input multi-output functions. This enables using a neural network model to predict all properties of interest given a pair of (p, h) inputs (or p input for saturated properties), which can reduce computational load as well as overhead associated with property calls. Secondly, neural networks are differentiable (given that most commonly used activation functions are infinitely differentiable) and allow equation-based simulation tools (e.g., Modelica-based) to compute system Jacobians analytically. To ensure accuracy and consistency of derivatives associated with predicted outputs, derivative-informed neural networks (O'Leary-Roseberry et al., 2024) that can incorporate derivatives data into loss functions as constraints during training are adopted in the present work. It is important to note that the derivative-informed neural networks constructed in this work focus on consistency of function derivatives with respect to inputs, not to be confused with temporal or spatial derivatives considered in training physics-informed neural networks that are often applied for learning differential equation solutions.

In particular, NN models are constructed for retrieving liquid or vapor properties with inputs of pressure and enthalpy and for bubble or dew properties with an input of pressure, as shown in Fig. 1.

It was found through a trial-and-error process that feedforward neural networks with a $\tanh(\cdot)$ activation function and up to two hidden layers were capable of accurately predicting single-phase and saturated thermophysical properties as well as derivatives for two widely adopted refrigerants R410A and R134A. Similarly, it was reported in Ma et al. (2018) that feedforward neural networks with one hidden layer were able to predict single-phase and saturated properties of R134A in good agreement. Therefore, it is reasonable to assume that the basic neural network structure (consisting of one or two hidden layers) applied in this work will be valid for modeling most pure and pseudo-pure refrigerants. For simplification, the configuration with a single hidden layer is presented below, while the case of two hidden layers can be derived similarly. In case of both single-phase and saturated, the NN model exploits the same form

$$\boldsymbol{y} = \boldsymbol{W}_2 \tanh\left(\boldsymbol{W}_1 \boldsymbol{x} + \boldsymbol{b}_1\right) + \boldsymbol{b}_2 \tag{6}$$

where $\boldsymbol{W}_1$ denotes the hidden layer weight matrix, $\boldsymbol{b}_1$ denotes the bias vector, $\tanh(\cdot)$ is element-wise activation applied to a vector. $\boldsymbol{W}_2$ and $\boldsymbol{b}_2$ denote the output layer weight matrix and bias vector, respectively. Since the first and second order derivatives of the properties are often of interest in transient simulations, the neural network model in Eq. (6) can be differentiated to obtain analytical forms of those derivatives. The Jacobian matrix that contains all first-order derivatives is computed by

$$\frac{d\boldsymbol{y}}{d\boldsymbol{x}} = \boldsymbol{W}_2 \mathrm{diag}\left(\boldsymbol{1} - \tanh^2(\boldsymbol{z})\right)\boldsymbol{W}_1 \tag{7}$$

where $\boldsymbol{z} = \boldsymbol{W}_1 \boldsymbol{x} + \boldsymbol{b}_1$ represent the linear transformation associated with the hidden layer, and $\mathrm{diag}\left(\boldsymbol{1} - \tanh^2(\boldsymbol{z})\right)$ is a diagonal matrix whose diagonal entries are specified by the vector $\boldsymbol{1} - \tanh^2(\boldsymbol{z})$. Differentiation of the first derivatives shown in Eq. (7) yields the second derivative of the ith output with respect to the jth input and the kth input as

$$\frac{\partial^2 y_i}{\partial x_j x_k} = -2\boldsymbol{W}_2^{(i)} \mathrm{diag}\left(\tanh(\boldsymbol{z}) \odot \left(\boldsymbol{1} - \tanh^2(\boldsymbol{z})\right)\right)\left(\boldsymbol{W}_1^{(:,j)} \odot \boldsymbol{W}_1^{(:,k)}\right) \tag{8}$$

where $\boldsymbol{W}_2^{(i)}$ denotes the ith row of the output layer weights matrix, $\odot$ denotes the Hadamard product (element-wise multiplication), $\boldsymbol{W}_1^{(:,j)}$ and $\boldsymbol{W}_1^{(:,k)}$ denote the jth and kth column of the hidden layer weights matrix, respectively.

Ensuring accuracy and consistency is critical for the development of a property modeling approach. As shown above, neural network models inherently guarantee consistency between the predicted properties and their derivatives with respect to inputs, since derivatives to any order can be generated by differentiating the neural networks. However, in order to enforce the prediction accuracy of the first-order and second-order derivatives, training losses associated with those derivatives can be taken into consideration during training processes. Derivative-informed neural networks that incorporate physical constraints as a regularizer is adopted in the present work. This is realized by aggregating losses associated with derivative predictions made by Eqs. (7)-(8) into a total

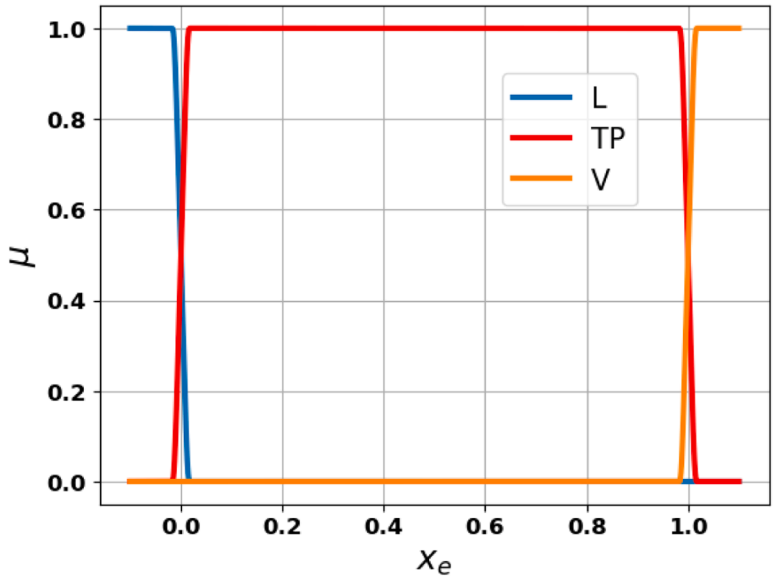

Fig. 2. Membership functions associated with Fuzzy numbers.

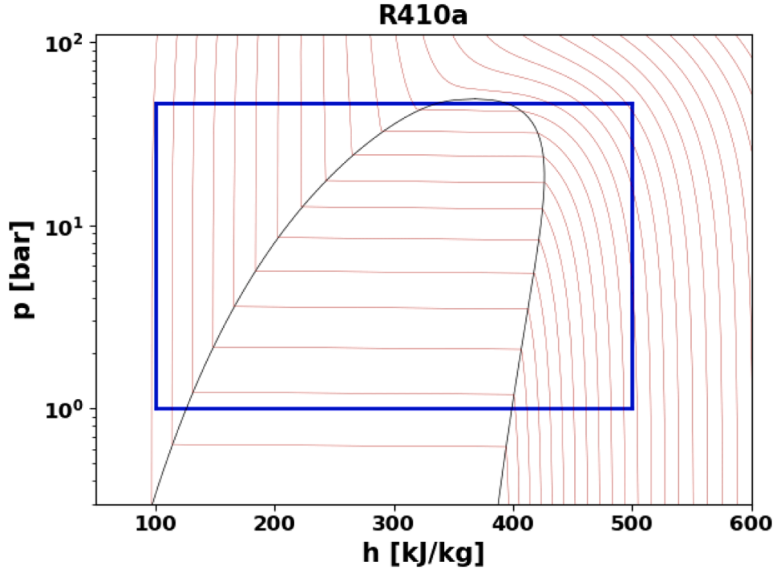

Fig. 3. Domain of interest on p-h diagram for the refrigerant R410A.

training loss at every training epoch. The mean-squared-error is employed as a criterion to measure the training loss

$$l = \frac{1}{N}\sum_{k=1}^{N}\left(\left\|\boldsymbol{y}^{(k)} - \hat{\boldsymbol{y}}^{(k)}\right\|_2^2 + \omega_1\left\|\boldsymbol{J}^{(k)} - \hat{\boldsymbol{J}}^{(k)}\right\|_2^2 + \omega_2\left\|\boldsymbol{H}^{(k)} - \hat{\boldsymbol{H}}^{(k)}\right\|_2^2\right) \quad (9)$$

where the first term represents the property prediction loss for N training samples, $\boldsymbol{y}^{(k)}$ denotes a sample vector of the training data, $\hat{\boldsymbol{y}}^{(k)}$ denotes the corresponding predicted values by the neural network, $\|\cdot\|_2$ represents L2 norm of a vector or matrix. Similarly, the second and third terms represent losses associated with the first-order and second-order derivatives, respectively. ω_1 and ω_2 are weights that parametrize contribution of these losses to the total training loss. It should be noted that the Jacobian matrix $\boldsymbol{J}^{(k)}$ and Hessian matrix (tensor in case of single-phase models with 2 inputs) $\boldsymbol{H}^{(k)}$ which contain all the first-order and second-order derivatives are shown here for notational convenience. In practice, solely a subset of derivatives that are relevant to transient thermofluid models (e.g., first-order and second-order derivatives of density and temperature) is considered.

To train a NN model with the input-output setup specified in Fig. 1, property data is randomly generated for each phase (or saturated line) within a predefined $p-h$ domain from the property database CoolProp. After that, the generated data is standardized by removing the mean and scaling to a unit variance to improve training efficiency. As a result, the NN model shown in Eq. (6) is trained using the scaled property data. For instance, an input pair to the NN model consists of scaled pressure and enthalpy

$$(p^*, h^*) = \left(\frac{p-\bar{p}}{\sigma_p}, \frac{h-\bar{h}}{\sigma_h}\right) \quad (10)$$

where $\bar{p}$ and $\bar{h}$ are mean values, σ_p and σ_h are standard deviations of the data. The output data is scaled to approximate a normally distributed dataset in the same way. Note that, correspondingly, the derivative data used for evaluating the training loss should be scaled as well. Take density partial derivatives as an example, the scaled first-order and

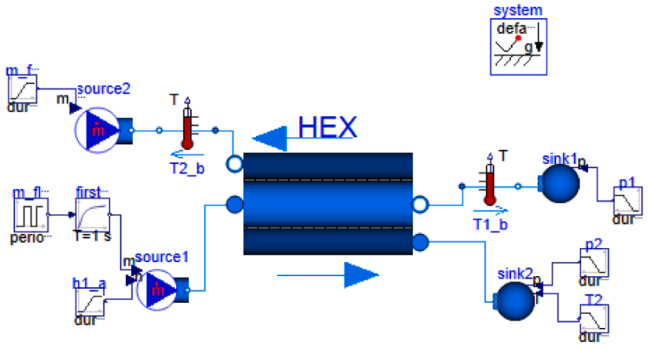

Fig. 4. Heat exchanger model test bench.

second-order derivatives with respect to pressure are

$$\left.\frac{\partial \rho^*}{\partial p^*}\right|_{h^*} = \frac{\sigma_p}{\sigma_\rho}\left.\frac{\partial \rho}{\partial p}\right|_h, \quad \left.\frac{\partial^2 \rho^*}{\partial p^{*2}}\right|_{h^*} = \frac{\sigma_p^2}{\sigma_\rho}\left.\frac{\partial^2 \rho}{\partial p^2}\right|_h \tag{11}$$

where $\partial\rho/\partial p|_h$ and $\partial^2\rho/\partial p^2|h$ are original derivatives obtained from CoolProp. Derivatives of saturated properties with respect to pressure are scaled in the same manner.

2.3. Two-phase properties and partial derivatives

With an assumption of homogeneous flow, two-phase properties can be determined by saturated properties and a vapor quality. Given an input pair of (p, h), saturated properties listed in Fig. 1 and their derivatives with respect to pressure can be obtained by the saturated NN models using the input p. The thermodynamic equilibrium quality can subsequently be determined using the predicted saturated enthalpies

$$x_e = \frac{h - h_f}{h_g - h_f} \tag{12}$$

where h_g and h_f represent dew and bubble enthalpies, respectively. It is important to note that x_e is defined beyond the two-phase zone, which is utilized for phase transition calculations in the next section.

Following the homogeneous assumption, any of the mass-based two-phase properties (e.g., specific volume instead of density) is obtained by

$$y = x_e y_g + (1 - x_e) y_f \tag{13}$$

where y_g and y_f represent dew and bubble properties, respectively, that are outputs of the saturated NN models as a function of pressure. A number of correlations are available for evaluating two-phase transport properties. In the present work, two-phase thermal conductivity is approximated using the homogeneous model, while two-phase dynamic viscosity is evaluated using the correlation (McAdams et al., 1942)

$$\eta = \frac{1}{x_e/\eta_g + (1 - x_e)/\eta_f} \tag{14}$$

to align with CoolProp.

Partial derivatives of two-phase properties which are functions of (p, h) are then derived by differentiating Eq. (13) which yields

$$\left.\frac{\partial y(p,h)}{\partial p}\right|_h = x_e \frac{dy_g}{dp} + (1 - x_e)\frac{dy_f}{dp} + (y_g - y_f)\frac{dx_e}{dp}, \tag{15}$$

$$\left.\frac{\partial y(p,h)}{\partial h}\right|_p = \frac{y_g - y_f}{h_g - h_f} \tag{16}$$

where the derivative of x_e is obtained by

$$\frac{dx_e}{dp} = \frac{1}{h_f - h_g}\left[x_e \frac{dh_g}{dp} + (1 - x_e)\frac{dh_f}{dp}\right]. \tag{17}$$

Correspondingly, second-order two-phase partial derivatives can be derived by differentiating Eqs. (15)-(16), which are not presented here for the sake of brevity.

2.4. Fuzzy modeling for smooth phase transition

As properties of each phase (i.e., liquid, two-phase, vapor) are evaluated by an individual model, a transition algorithm is necessary to determine the phase of the refrigerant state given an input pair of (p, h), and subsequently retrieve properties and their partial derivatives using the corresponding model. This could be realized by applying conditional statements and interpolations at phase boundaries. Nonetheless, concerning numerical robustness of thermofluid models it is essential that a transition algorithm ensures the overall property model being differentiable to facilitate the generation of analytical Jacobians for a Modelica simulation tool as well as avoids discrete events during simulations.

A Fuzzy modeling approach (Kim et al., 2021) is therefore employed to execute phase-dependent transitions, with x_e being naturally selected as a switching variable. Firstly, three Fuzzy numbers (L, TP, V) are constructed and membership functions $\mu(x_e)$ associated with them are depicted in Fig. 2. It is important to note that each membership function is smooth across the entire domain. After that, membership function values are evaluated to determine a weighted combination of predictions from property models of all three phases given an input of x_e

$$\boldsymbol{y} = \frac{\mu_l \boldsymbol{y}_l + \mu_{tp}\boldsymbol{y}_{tp} + \mu_v \boldsymbol{y}_v}{\mu_l + \mu_{tp} + \mu_v} \tag{18}$$

where μ_l, μ_{tp}, and μ_v are membership function values evaluated at x_e, $\boldsymbol{y}_l$ represents outputs of the liquid phase NN model, $\boldsymbol{y}_v$ represents those of the vapor NN model, and $\boldsymbol{y}_{tp}$ represents predictions of the two-phase model presented in Section 2.3. Similarly, partial derivative predictions are obtained using outputs from models of all phases and the weights as shown in Eq. (18).

3. Implementation

This section demonstrates implementation of the proposed approach to model thermophysical properties of the pseudo-pure refrigerant R410A, that is widely applied in existing vapor compression cycles. Though R410A will be phased out due to environmental concerns, the selection for methodology demonstration in this paper and implementation in the open-source Modelica library aim to facilitate numerical performance benchmarking for transient thermofluid simulations, as systems that utilized R410A have been extensively investigated in transient modeling studies with experimental validations (e.g., Qiao et al., 2015). Fig. 3 depicts a domain of interest for approximation with pressures ranging from 1 bar to 46.56 bar (95% of the critical pressure) and enthalpies ranging from 100 kJ/kg to 500 kJ/kg.

Training of the single-phase and saturated NN models are implemented using the deep learning library PyTorch (Steiner et al., 2019) in Python, while training data is generated from CoolProp. In regard to single-phase NN models, a training set of 10^4 samples is generated for each phase. 10^3 samples are generated for each saturation line. Neural networks with one hidden layer are found to be capable of accurately predicting properties and their derivatives (first-order and second-order), except that the NN model for vapor phase is constructed with two hidden layers. The dimension of hidden layers for models of liquid phase, vapor phase, bubble line, and dew line are 9, 9, 7, 8, respectively. Since the NN model configurations applied in the present work are fairly simple, and the training dataset size is small compared to other deep learning applications, a second-order optimization algorithm *LBFGS* is adopted in training over first-order methods (e.g., stochastic gradient descent) for faster convergence. It has been observed that training the NN models generally converged in 20 epochs.

Property models are implemented using the Modelica language as a reusable open-source fluid property library (Ma, 2025). Partial derivatives of the predicted properties listed in Eq. (4) with respect to pressure and enthalpy are explicitly specified for Modelica functions that retrieve those properties. Consequently, a Modelica compiler is enabled to differentiate these property call functions analytically instead of numerically

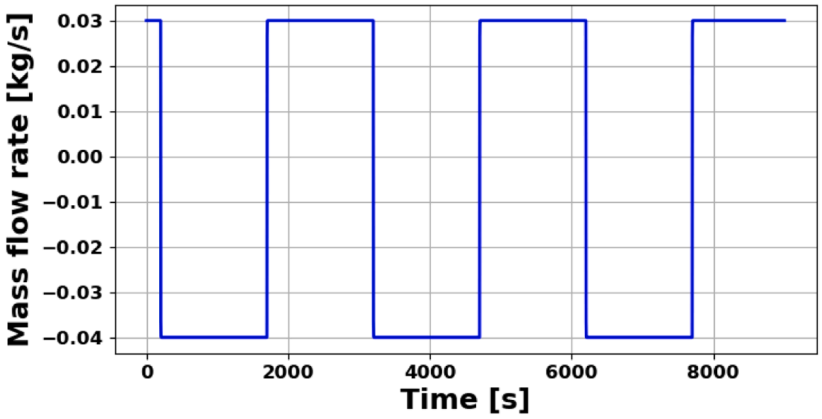

Fig. 5. Inlet refrigerant mass flow rate boundary condition.

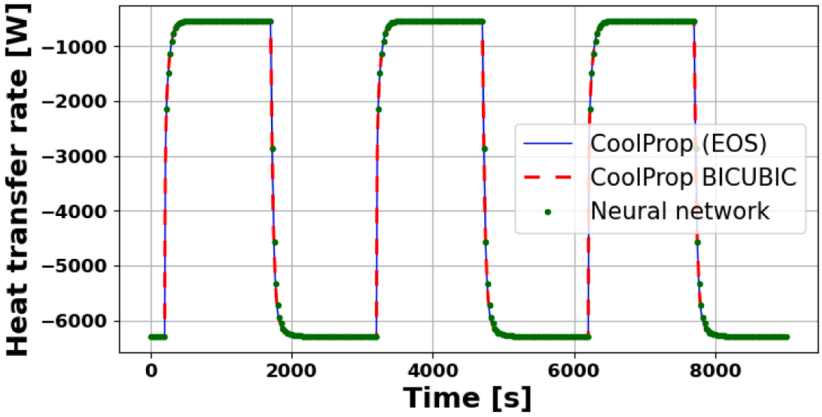

Fig. 6. Simulation results of the refrigerant capacity.

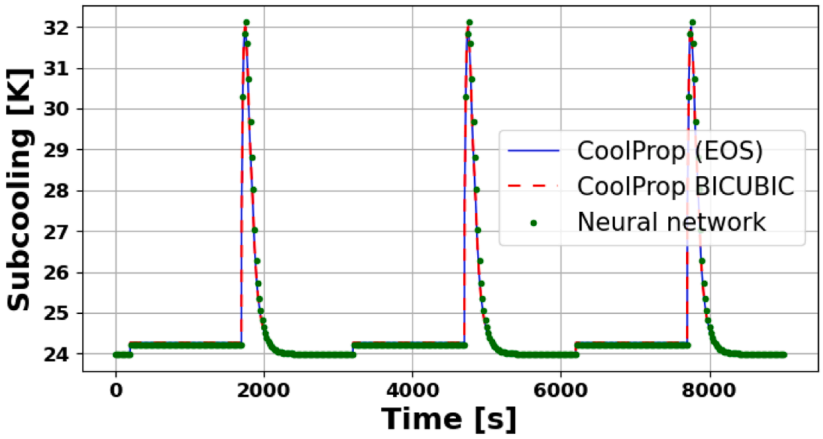

Fig. 7. Simulation results of the refrigerant subcooling.

by iterations. Since it is often the case that property functions, particularly in thermofluid application models without explicit derivatives, block a Modelica tool from generating system ODE Jacobians analytically, the proposed approach that addresses this issue has the potential to significantly improve simulation performance.

4. Simulation results and computational performance

Accuracy and computational speed of the proposed approach are demonstrated via property function calls and heat exchanger simulations. All evaluations are benchmarked against the standard EOS backend as well as a lookup table interpolation backend of CoolProp, which are accessed through the open-source Modelica library ExternalMedia 4.0 (Casella and Richter, 2008). Simulations are carried out using the Modelica tool Dymola R2025x on a standard office laptop.

4.1. Property function calls

To examine accuracy and computation speed of the developed property models, predictions within the approximation envelop shown in Fig. 3 are compared with CoolProp. Thermophysical properties are evaluated given 10^4 (p, h) input pairs that span all three phases, using the proposed neural network models, CoolProp EOS backend, and CoolProp bicubic interpolation scheme. Predictions are obtained by running

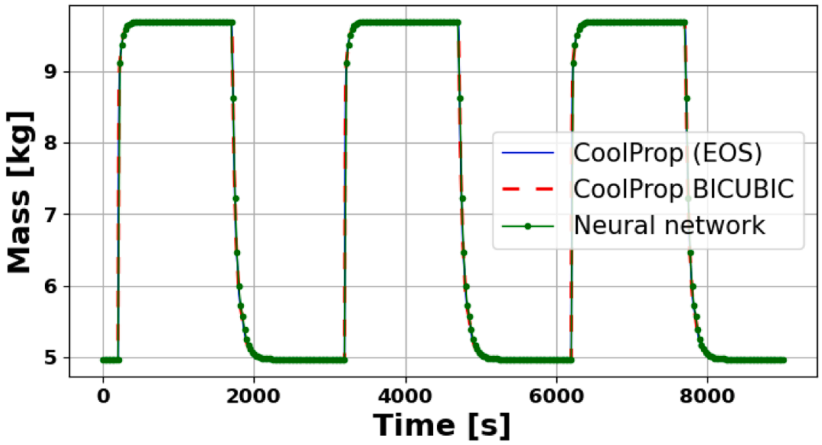

Fig. 8. Simulation results of the refrigerant mass.

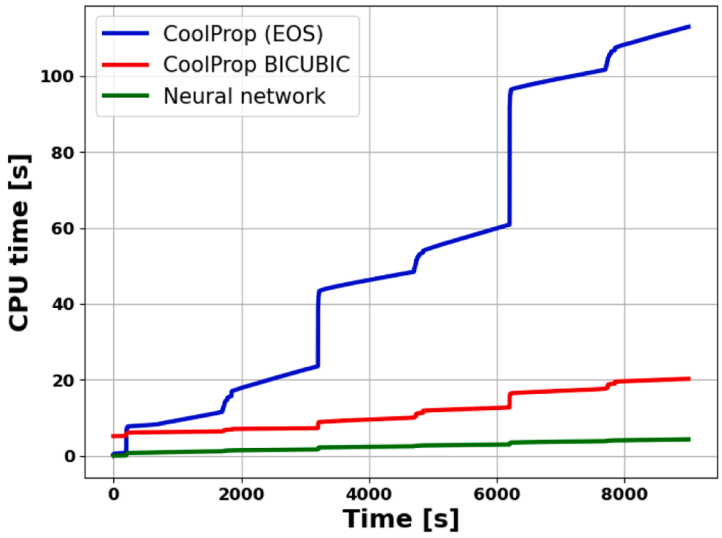

Fig. 9. CPU time associated with heat exchanger simulations.

Table 1
Accuracy of property predictions by the proposed method against CoolProp EOS backend.

Property	ρ	T	s	c_v	c_p	λ	η
MAPE [%]	0.016	0.13	0.033	0.038	0.281	0.131	0.175

simulations in Dymola, with a fixed-step Euler solver and a relative error tolerance of 10^{-5}. Table 1 summarizes accuracy represented by the mean absolute percentage error for each of the predicted quantity, when comparing the proposed method against CoolProp

$$\text{MAPE} = \frac{1}{n}\sum_{i=1}^{n}\left|\frac{y_i - \hat{y}_i}{y_i}\right| \times 100\% \tag{19}$$

where n denotes the number of samples, $\hat{y}_i$ denotes a prediction by the proposed method, while y_i denotes a prediction by CoolProp. It reveals that the proposed method is capable of accurately predicting thermophysical properties with MAPE values less than 0.3%. In terms of computational speed, CPU times associated with the CoolProp EOS backend and bicubic interpolation backend are 5.83 and 0.92 seconds, respectively, while that of the proposed approach is 0.81 seconds, which yields a 86% speedup compared to the standard CoolProp backend and remarkably, even faster than its lookup table implementation.

4.2. Heat exchanger model test bench

A discretized heat exchanger model test bench driven by prescribed boundary conditions is set up to demonstrate computational benefits of the proposed thermophysical property approximation on transient simulations, as shown in Fig. 4.

The model is adapted from a heat exchanger simulation example in the Modelica Standard Library `Modelica.Fluid.Examples.HeatExchanger` where a counter-flow 1-D heat exchanger model is employed. The heat exchanger is divided into 10 control volumes, with liquid water as the secondary fluid. The water inlet (sink2 shown in the model diagram) pressure and temperature are fixed at 1 bar and 22 °C, while the water outlet (source2) mass flow rate is 0.1 kg/s. On the refrigerant side, inlet enthalpy and outlet pressure are specified as 460 kJ/kg and 35 bar associated with the nominal flow direction. In case a reversed refrigerant flow, the inlet enthalpy is determined at the opposite end (sink1), which is set to 250 kJ/kg. Transients of the heat exchanger are driven by a varying refrigerant inlet mass flow rate controlled by a pulse signal depicted in Fig. 5 to trigger repeated flow reversal, which has been acknowledged as a primary computational challenge associated with dynamic thermofluid modeling (Ma and Thorade, 2025).

Though the refrigerant pressure and enthalpy are selected as state variables that aligns with inputs to thermophysical property models,

Dymola fails to generate analytical Jacobian for the system ordinary differential equation (ODE) representations associated with CoolProp EOS and bicubic interpolation backends due to incomplete function derivative information. In contrast, the Dymola compiler is able to generate the system ODE Jacobian analytically when the proposed neural network property models are applied.

The heat exchanger model is simulated with all three thermophysical property backends (i.e., CoolProp EOS, CoolProp bicubic interpolation, and the proposed neural network models). Simulations are carried out using the Dassl solver with a relative error tolerance of 10^{-5}. To survey propagated errors associated with property evaluations in system models, simulation results of the refrigerant capacity, the refrigerant outlet subcooling corresponding to the nominal flow direction, and the total refrigerant mass inventory residing in the heat exchanger are revealed in Figs. 6–8. No evident discrepancy is observed among all three property routines. Quantitatively, prediction errors associated with the neural network models are obtained as 0.41%, 0.22%, and 0.1% for the refrigerant capacity, subcooling degree, and mass in terms of MAPE when compared with the CoolProp EOS backend, which indicates that the developed property approximation models yield negligible errors in transient thermofluid component/system simulations.

Since computational efficiency constitutes the primary concern in this work, the CPU time associated with each simulation is revealed in Fig. 9. As expected, the CoolProp EOS backend that involves numerical iterations to retrieve thermophysical properties yields the slowest simulation with a CPU time of 112.97 seconds corresponding to a simulated time of 9000 seconds, whereas the CoolProp bicubic interpolation scheme that avoids internal iterations leads to faster simulations with a CPU time of 20.31 seconds. The proposed neural network approach reveals a significant speedup compared to the CoolProp backends, as the simulation can complete in 4.35 seconds. It is important to note that the result associated with the neural network approach yields a 4.7 times speedup compared to that of the CoolProp bicubic interpolation backend, though the property function call results presented in Section 4.1 indicates nearly the same computation speeds. This can be attributed to the generation of analytical ODE Jacobian in system simulations when adopting the neural network property models. Clearly, it can be seen from Fig. 9 that flow reversals impose challenges to transient simulations where the CPU time associated with the CoolProp backends undergoes a rapid increase whenever the refrigerant flow rate approaches zero. This is because approximations of the ODE Jacobian numerically using finite-difference schemes are computationally intensive and expensive tasks when the equation system becomes stiff near zero flows. Furthermore, computing numerical Jacobians that have large magnitudes or are nearly singular during flow reversal, are unreliable and can deteriorate simulation robustness, that often leads to simulations get stuck or possibly fail.

5. Conclusions

This paper presents a modeling approach for efficiently retrieving refrigerant thermophysical properties and their consistent partial derivatives from pressure and enthalpy using derivative-informed neural networks. By explicitly mapping pressure and enthalpy to thermodynamic and transport properties across all phase regions, the proposed approach eliminates the need for iterative equation-of-state evaluations while preserving accuracy required for high-fidelity transient thermofluid simulations. A transition algorithm based on the Fuzzy modeling approach is also introduced to ensure smooth and numerically robust prediction of thermophysical properties in the vicinity of phase changes.

A key contribution of this modeling approach is the availability of analytically differentiable property models that enables a simulation tool to generate system ODE Jacobians analytically, rather than relying on numerically expensive finite-difference approximations.

Numerical performance of the proposed approach is demonstrated via an implementation for the refrigerant R410A using the Modelica modeling language, and comparisons against the property library CoolProp. Accuracy and computational speed are compared in numerical tests of property function calls as well as two-phase heat exchanger simulations under cyclic flow reversals. The simulation results indicate that the proposed approach yields a significant speedup compared to the standard CoolProp Equation of State backend as well as its bicubic interpolation backend, with negligible discrepancies for predicted quantities. Though the proposed models are smooth curves (saturated models) or surfaces (single-phase models) across the entire domain, predictions are valid solely within the approximation ranges. Therefore, it might be necessary to apply numerical treatments when state variables are initialized or the solver trails a guess state outside the valid domain. Examples implemented in the open-source Modelica library did not enforce input range validity, as boundary conditions in the numerical experiments were selected to drive state variables within the valid range. Future work can focus on efficacy and robustness of numerical techniques to overcome this issue. For instance, clamping strategies can be implemented to bound input variables while ensuring the overall property model is still smooth. Moreover, solver sensitivity to states outside the training bounds and the fallback mechanism to the valid domain or switch to other property evaluation routines (e.g., EOS) can be explored.

It is worth noting that despite the present work focused on subcritical properties, the proposed approach is applicable to supercritical applications (e.g., CO_2 cycles) as well by training a derivative-informed neural networks model that predicts supercritical properties and integrating that into the transition algorithm presented in Section 2.4. Potentially, pressure can be considered as an additional switching variable other than x_e. Additionally, though the developed approach addressed pure and pseudo-pure refrigerants with a homogeneous model employed for the two-phase region as illustrated in this paper, it can be extended to refrigerant mixtures using an appropriate two-phase model, and the same NN models for single-phase and saturated properties of all mixture components. Similarly, a slip flow assumption can be applied instead of the homogeneous flow illustrated in Section 2.3 to model two-phase behaviors. With appropriate void fraction correlations, two-phase properties can be obtained using the saturated properties predicted by NN models, though the overall performance of property models with a slip flow assumption needs to be examined in transient system simulations.

CRediT authorship contribution statement

Jiacheng Ma: Writing – original draft, Software, Methodology, Investigation, Formal analysis, Conceptualization; **Matthis Thorade:** Writing – review & editing, Methodology, Conceptualization; **Donghun Kim:** Writing – review & editing, Methodology, Conceptualization.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

Al Khoury, J., Al Haddad, R., Shakrina, G., Malham, C.B., Sayah, H., Bouallou, C., Nemer, M., 2024. A fast method for the calculation of refrigerant thermodynamic properties in a refrigeration cycle. Int. J. Refrig. 157, 98–108. https://doi.org/10.1016/j.ijrefrig.2023.10.010

Aute, V., Radermacher, R., 2014. Standardized polynomials for fast evaluation of refrigerant thermophysical properties. In: 15th International Refrigeration and Air Conditioning Conference at Purdue.

Bell, I.H., Wronski, J., Quoilin, S., Lemort, V., 2014. Pure and pseudo-pure fluid thermophysical property evaluation and the open-source thermophysical property library CoolProp. Ind. Eng. Chem. Res. 53 (6), 2498–2508.

Bendapudi, S., 2004. Development and Evaluation of Modeling Approaches for Transients in Centrifugal Chillers. Ph.D. thesis. Purdue University.

Bortoff, S.A., Laughman, C.R., Deshpande, V., Qiao, H., 2024. Fluid property functions in polar and parabolic coordinates. In: Proceedings of the American Modelica Conference, pp. 21–28.

Casella, F., Richter, C., 2008. ExternalMedia: a library for easy re-use of external fluid property code in Modelica. In: Proceedings of the 6th International Modelica Conference, pp. 157–161.

Ding, G., Wu, Z., Liu, J., Inagaki, T., Wang, K., Fukaya, M., 2005. An implicit curve-fitting method for fast calculation of thermal properties of pure and mixed refrigerants. Int. J. Refrig. 28 (6), 921–932.

Huber, M.L., Lemmon, E.W., Bell, I.H., McLinden, M.O., 2022. The NIST REFPROP database for highly accurate properties of industrially important fluids. Ind. Eng. Chem. Res. 61 (42), 15449–15472. https://doi.org/10.1021/acs.iecr.2c01427

Kim, D., Ma, J., Braun, J.E., Groll, E.A., 2021. Fuzzy modeling approach for transient vapor compression and expansion cycle simulation. Int. J. Refrig. 121, 114–125. https://doi.org/10.1016/j.ijrefrig.2020.10.025

Li, L., Gohl, J., Batteh, J., Greiner, C., Wang, K., 2018. Fast calculation of refrigerant properties in vapor compression cycles using spline-based table look-up method (SBTL). In: Proceedings of the American Modelica Conference, Cambridge, MA, USA, pp. 77–84.

Li, P., Li, Y., Seem, J.E., Qiao, H., Li, X., Winkler, J., 2014. Recent advances in dynamic modeling of HVAC equipment. part 2: Modelica-based modeling. HVAC&R Research 20 (1), 150–161.

Ma, J., 2025. https://github.com/jiachengma/NNMedia.

Ma, J., Kim, D., Braun, J.E., 2018. Development of a fast method for retrieving thermodynamic properties to accelerate transient vapor compression cycle simulation. 17th International Refrigeration and Air Conditioning Conference at Purdue .

Ma, J., Kim, D., Braun, J.E., Horton, W.T., 2023. Development and validation of a dynamic modeling framework for air-source heat pumps under cycling of frosting and reverse-cycle defrosting. Energy 272, 127030.

Ma, J., Thorade, M., 2025. Frost/defrost models for air-source heat pumps with retained-water refreezing considered. In: Proceedings of the 16th International Modelica&FMI Conference, Lucerne, Switzerland. https://doi.org/10.3384/ecp218335

McAdams, W.H., Woods, W.K., Heroman, Jr, L.C., 1942. Vaporization inside horizontal tubes–II benzene-oil mixtures. Trans. Am. Soc. Mech. Eng. 64 (3), 193–199.

O'Leary-Roseberry, T., Chen, P., Villa, U., Ghattas, O., 2024. Derivative-informed neural operator: an efficient framework for high-dimensional parametric derivative learning. J. Comput. Phys. 496, 112555.

Qiao, H., Xu, X., Aute, V., Radermacher, R., 2015. Transient modeling of a flash tank vapor injection heat pump system–Part II: simulation results and experimental validation. Int. J. Refrig. 49, 183–194.

Steiner, B., DeVito, Z., Chintala, S., Gross, S., Paske, A., Massa, F., Lerer, A., Chanan, G., Lin, Z., Yang, E., et al., 2019. PyTorch: an imperative style, high-performance deep learning library. In: NeurIPS.

Thorade, M., Saadat, A., 2013. Partial derivatives of thermodynamic state properties for dynamic simulation. Environ. Earth Sci. 70 (8), 3497–3503. https://doi.org/10.1007/s12665-013-2394-z

Van, C. T.S., Laughman, C.R., 2018. Approximation of refrigerant properties for dynamic vapor compression cycle models. IFAC-PapersOnLine 51 (2), 625–630.