Fluid Property Functions in Polar and Parabolic Coordinates

Scott A. Bortoff, Christopher R. Laughman, Vedang Deshpande and Hongtao Qiao [1]

[1]Mitsubishi Electric Research Laboratories, Cambridge, MA, USA {bortoff, laughman, deshpande, qiao}@merl.com

Abstract

This paper presents two methods for reallizing fluid property functions in Modelica simulation models. Each makes use of a coordinate transformation that aligns one coordinate with the saturation curve. This provides for a precise representation of the fluid property function at the saturation curve, and for connected domains of interest including the liquid, vapor, supercritical and two-phase regions. Both approaches make use of spline function approximation in the aligned coordinates, and are numerically efficient, well conditioned, and allow for efficient calculation of derivatives up to any desired order that are precise up to processor numerical tolerance.

Keywords: thermofluid models, fluid properties

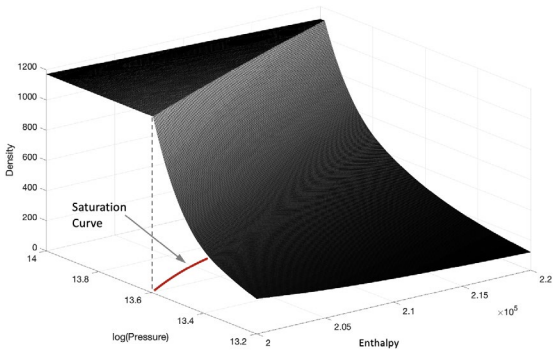

Figure 1. Density of R410A as function of h and $\log(P)$, showing the saturation curve (red).

1 Introduction

Fluid property functions relate fluid property variables such as temperature, pressure, density, etc., to one another. For a fluid of fixed composition in thermodynamic equilibrium, all fluid property variables can be calculated as a function of two independent variables: a mixture variable and a second variable. Pressure P and specific enthalpy h are often chosen for vapor compression models, but other combinations are also possible (Bejan, 1993). Fluid property functions are critical for thermofluid model simulation, and must be implemented in an accurate, computationally efficient manner. For some applications, fluid property function evaluations consume more than 70% of simulation time (Aute and Radermacher, 2014).

Mathematically, the domain of a fluid property function is the span of the two independent fluid property variables. For many thermofluid systems, such as vapor compression cycles, the domain includes values of the two independent fluid property variables that correspond to more than one of the fluid's states, such as the vapor state, the supercritical state, or the two-phase state. The boundary between the liquid region and two-phase region in the domain is the liquid saturation curve, and the boundary between the vapor region and two-phase region in the domain is the vapor saturation curve. These curves intersect smoothly at the critical point of the fluid, and their union is referred to as the saturation curve.

The saturation curve is distinguished because its image under a fluid property function is not smooth. The fluid property function is continuous (C_0), but not continuously differentiable (C_1), for all points on the saturation curve. For all other points in the domain, the fluid property function is a smooth (C_n) function of the two fluid property variables, for some $n \geq 1$, as shown in Figure 1, which plots density ρ as a function of specific enthalpy h and pressure P for R410A. Modelica models of thermofluid systems often make use of derivatives of a fluid property function with respect to the two fluid property variables, and it is important to compute these accurately, especially near the saturation curve.

Several approaches for computing fluid properties may be found in the literature. Some are based upon using the Helmholtz (or Gibbs) free energy equation (Span, 2000). Any fluid property of interest may be numerically computed by solving these equations using iterative methods, typically Newton's method of root finding. These methods are realized in available software such as REFPROP (Lemmon et al., 2018) and CoolProp (Bell et al., 2014), and also realized in the HelmholtzMedia Modelica library (Thorade and Saadat, 2012, 2013). While these methods are general and accurate, they tend to be computationally expensive for use in simulation models, especially for large models with long simulation times. Furthermore, iterative algorithms include a stopping criteria, and therefore small errors are introduced into the computed fluid property values. If these values are numerically differentiated, which may be done by a simulation tool to compute a Jacobian, for example, then small errors can be amplified to the point of being unacceptably large, especially in the region near the saturation curve. Moreover, these iterative methods can fail to converge for certain values of the two independent fluid property variables.

Other approaches for calculating fluid property functions for use in simulation include Taylor's series approximations or splines. Aute, et al describe a method using Chebyshev polynomials that is built from data obtained from REFPROP (Aute and Radermacher, 2014). This method demonstrates a significant speedup over standard iterative methods, but does not enforce consistency between the properties and their derivatives and cannot represent the behavior of the fluid close to the critical point. Kunick et al. describe a method using quadratic splines to represent the fluid properties of water and steam for the International Association for the Properties of Water and Steam (IAPWS) (Kunick, 2018). This method divides a domain of interest into three distinct regions of fluid state. But by splitting up the domain into non-overlapping sets, the method introduces inconsistencies at the saturation curve between these sets, resulting in errors in the property derivatives near the saturation curve.

US Patent Application 2020/0050158 (Xu, 2020) describes a thermodynamic property calculation method using a linear approximation of the properties, but this does not capture the nonlinearities that are pominant near the saturation curve. US Patent 7,676,352 B1 (Van Peursem and Xu, 2010) describes a method for calculating thermodynamic properties and their derivatives using local approximations of fluid property functions, but it is an iterative algorithm and fails to describe nonlinear fluid behavior on a large domain of interest, and does not provide accurate derivatives near the saturation curve.

Generally, previously published methods that use function approximations such as splines, or commonly used iterative methods based on the Helmholtz free energy function, for example, suffer from two fundamental problems. First, the coordinates used for numerical calculation are not aligned with the saturation curve. In other words, the saturation curve is not represented as a contour of one of the two independent coordinates. Therefore, the discontinuity in derivative across the saturation curve is not accurately represented. Secondly, the coordinates typically used can result in an ill-posed numerical calculation at or near the critical point. This is because one of the coordinates achieves a maximum when expressed as an explicit function of the other coordinate at this point. Iterative methods especially fail near the critical point, and may employ special curve-fit approximations near it. As such, many available fluid property libraries are limited to the sub-critical region. However, both of these problems are purely a consequence of poorly chosen coordinates: The saturation curve itself is smooth everywhere, and the property function itself is smooth everywhere except across the saturation curve.

In this paper, we introduce two coordinate systems for representation of fluid property functions that are well-defined for all regions of practical interest, including the two-phase, vapor, liquid, and super-critical regions. Both coordinate systems are defined to be *aligned* with the saturation curve, so that the discontinuity in derivative is represented in terms of only one of the coordinates, which is defined to be constant along the saturation curve (Laughman et al., 2023). In both coordinate systems, the critical point is no different from any other on the saturation curve, so that super-critical problems are no different than purely subcritical ones.

First, we show how a normalized polar coordinate system may be used to define fluid property functions. In these coordinates, the interior of the unit disk represents the two-phase region, and the liquid, supercritical and vapor regions are represented in the exterior of the unit disk. The saturation curve is an arc of the unit circle. Fluid property functions are represented as B-spline functions (de Boor, 1978; Piegl and Tiller, 1995), arranged such that the transition across the saturation curve is C_0 but not C_1. B-spline coefficients are computed off-line by solving a constrained least squares problem using data generated by a reference calculator such as REFPROP. The implementation is computationally and memory efficient, accurate, numerically well-conditioned and allows for evaluation of derivatives of the fluid property function of any desired order.

We derive a second implementation using normalized parabolic coordinates, which may be less familiar to the reader but for this application have a certain elegance. In normalized parabolic coordinates, the saturation curve is represented as a unit parabola in one of the two coordinates, which is naturally similar in shape to the saturation curve expressed in conventional (h, P) variables. The resulting fluid property functions are computationally efficient and well-conditioned, but some of the peculiarities of parabolic coordinates require additional attention.

Both are realized as a set of C language functions, with interface to Modelica though the external function interface. This makes the coordinates entirely invisible to the user. However, the result begs a question: Is there a benefit to expressing the fluid dynamics equations explicitly in these variables, instead of using conventional physical variables? This might be possible if the coordinate transformations were defined natively in the Modelica language. Addressing this question is left to future research.

This paper is organized as follows. In Sections 2 and 3 we derive the polar and parabolic coordinate transformations and property functions realizations, respectively. We discuss some of the implications in Section 4, and draw some conclusions in Section 5.

2 Polar Coordinates

Consider density ρ (kg/m^3) as a representative fluid property, to be computed as a function of independent fluid property variables pressure P_e (Pa) and specific enthalpy h_e (J/kg), where the subscript e denotes that the variables are in engineering units. Consider a domain of interest Ω in the $h_e - P_e$ plane, on which an approximation $\hat{\rho}$ of ρ is defined. Ω may include the two-phase region, the liquid and vapor regions, and the super-critical region, and

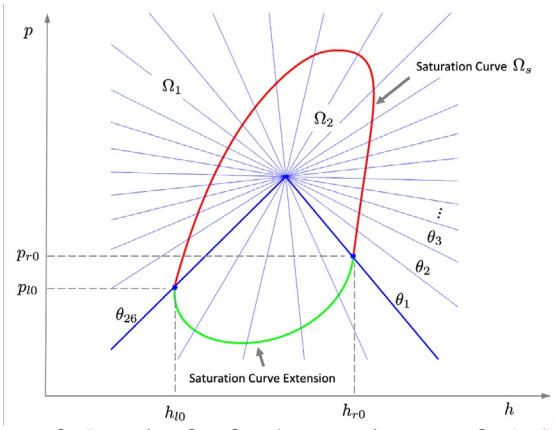

Figure 2. Domains Ω_1, Ω_2, the saturation curve Ω_s (red) and the saturation curve extension (green).

will be defined below. $\hat{\rho}$ is computed in a normalized polar coordinate system defined by the composition of three coordinate transformations $T = T_3 \circ T_2 \circ T_1$.

Scaling Coordinate Transformation T_1

Choose an origin $(h_{e0}, P_{e0}) \in \Omega$, inside the two-phase region, and values for two scaling factors, p_s (dimensionless) and h_s (J/kg), to define the scaling coordinate transformation $T_1 : \mathbb{R}^2 \to \mathbb{R}^2 : (h_e, P_e) \mapsto (h, p)$ as

$$h = (h_e - h_{e0})/h_s \tag{1a}$$
$$p = p_s \cdot \log(P_e/P_{e0}). \tag{1b}$$

The scaling factors are chosen such that the dimensionless p and h are $O(1)$ over Ω. The inverse scaling coordinate transformation, $T_1^{-1} : \mathbb{R}^2 \to \mathbb{R}^2 : (h, p) \mapsto (h_e, P_e)$, is

$$h_e = h_s \cdot h + h_{e0} \tag{2a}$$
$$P_e = P_{e0} \cdot \exp(p/p_s). \tag{2b}$$

Polar Coordinate Transformation T_2

In the scaled (h, p) coordinates, define the polar coordinate transformation $T_2 : \mathbb{R}^2 \to \mathbb{R}^2 : (p, h) \mapsto (r, \theta)$ as

$$r = \sqrt{h^2 + p^2} \tag{3a}$$
$$\theta = \text{atan}(p, h), \tag{3b}$$

where $\text{atan}(\cdot, \cdot)$ is the two-argument, four quadrant inverse tangent function. The inverse polar coordinate transformation $T_2^{-1} : \mathbb{R}^2 \to \mathbb{R}^2 : (r, \theta) \mapsto (h, p)$ is

$$h = r \cdot \cos(\theta) \tag{4a}$$
$$p = r \cdot \sin(\theta). \tag{4b}$$

Saturation Curve Radial Distance Normalization T_3

Figure 2 shows a domain Ω on the (h, p) - plane, divided into three regions: Ω_2 is the two-phase region; Ω_1 is outside the two phase region, and may include the liquid, vapor and super-critical regions; and Ω_s is the saturation curve, which is the boundary between Ω_1 and Ω_2. Define p_{r0} to be a small value of p on the vapor side of the saturation curve Ω_s at or near the lower boundary of Ω. Consider a small value p_{l0} of p on the liquid side of the saturation curve, at or near the lower boundary of Ω. A precise value for p_{l0} will be computed from p_{r0} and the choice of spline knots in the θ-direction below. Define h_{r0} and h_{l0} to be the scaled enthalpies corresponding to p_{r0} and p_{l0}, respectively, on Ω_s. This defines the points (h_{r0}, p_{r0}) and (h_{l0}, p_{l0}) on Ω_s. Express these points in polar coordinates as

$$(r_1, \theta_1) = T_2(h_{r0}, p_{r0}) \tag{5}$$

and

$$(r_{j^*}, \theta_{j^*}) = T_2(h_{l0}, p_{l0}), \tag{6}$$

where j^* is defined below. Then the saturation curve between (h_{r0}, p_{r0}) and (h_{l0}, p_{l0}), including the critical point (h_c, p_c), may be represented on the (h, p) plane in polar coordinates as the image of $(h_{\text{sat}}, p_{\text{sat}}) = T_2^{-1}(r_{\text{sat}}, \theta)$, where $f_{\text{sat}} : \mathbb{R} \to \mathbb{R} : \theta \mapsto r$, is

$$r_{\text{sat}} = f_{\text{sat}}(\theta) \quad \text{for} \quad \theta \in [\theta_1, \theta_{j^*}]. \tag{7}$$

As shown in Figure 2, choose an extension of f_{sat} on the open interval $(\theta_{j^*}, \theta_1 + 2\pi)$ so that the extended f_{sat} is periodic in θ and C^{n-1} (continuous up to $(n-1)^{th}$ derivative) for all $\theta \in \mathbb{R}$, for some value of $n > 0$. (A value for n is defined below as the degree of a spline.) Essentially, this defines a closed curve (a loop) to be the image of the extended f_{sat} that is the saturation curve for scaled pressures larger than p_{r0} and p_{l0}, and connects (h_{l0}, p_{l0}) and (h_{r0}, p_{r0}) through the two-phase region.

The extended function $f_{\text{sat}}(\theta)$ is approximated with a periodic B-spline denoted $\hat{f}_{\text{sat}}(\theta)$, which is fit to data on Ω_s that is generated by a thermofluid property calculator such as REFPROP. Other functional representations, such as NURBS, Fourier series or Chebychev polynomials might also be used. Define

$$\Theta_s = \{\theta_1, \ldots, \theta_{j^*}, \ldots, \theta_N\} \tag{8}$$

to be a set of (periodic) knots in the θ-direction, and denote the corresponding i^{th} degree-n periodic B-spline basis function as $B_{i,n}(\theta)$, $1 \le i \le N$ (de Boor, 1978; Piegl and Tiller, 1995). Then

$$\hat{f}_{\text{sat}}(\theta) = \sum_{i=1}^{N} c_{si} B_{i,n}(\theta). \tag{9}$$

The coefficients c_{si} $1 \le i \le N$ are computed by solving a least squares curve fit to data, as follows. First compute a number N_{ds} of pairs of values of (h, p) along the liquid and vapor sides of the saturation curve from (h_{r0}, p_{r0}) and (h_{l0}, p_{l0}), respectively, up to but not including the critical point (h_c, p_c), using a thermofluid property calculator and the transformations T_1. For many fluids the values of P_e

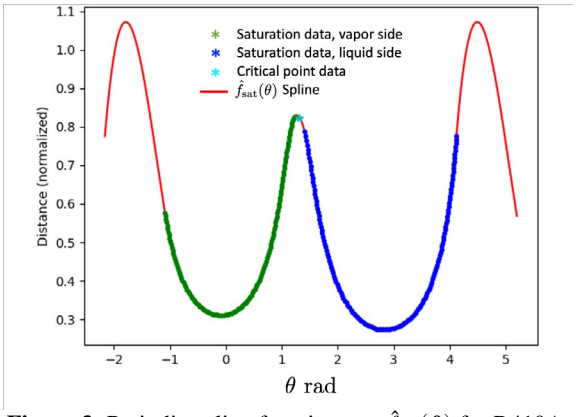

Figure 3. Periodic spline function $r = \hat{f}_{sat}(\theta)$ for R410A.

and h_e on the saturation curve near the critical point is difficult to compute and may be inaccurate, but the value of P_e and h_e at the critical point may be computed accurately. Add the calculated value of (h_c, p_c) to the set of data from the vapor and liquid saturation curves, giving $N_{ds}+1$ data pairs. This set is transformed into polar coordinates using T_2 giving a set of data points (r_k, θ_k) for $1 \le k \le N_{ds}+1$, and this set is used to solve a least squares curve fit problem to compute c_{si}, $1 \le i \le N$.

Note that the set Θ_s need not be uniform, and we may set $j^* = N$, so that the saturation curve extension is defined by a single spline interval. If the data set is accurate, then Θ_s may be defined by the values of θ_k in the data, so that the spline function is interpolating between the data points.

The third coordinate transformation $T_3 : \mathbb{R}^2 \to \mathbb{R}^2 : (r,\theta) \mapsto (\bar{r},\bar{\theta})$, which normalizes the distance between the origin and Ω_s to a constant value of one, is defined as

$$\bar{r} = r / \hat{f}_{\text{sat}}(\theta) \tag{10a}$$
$$\bar{\theta} = \theta, \tag{10b}$$

with inverse $T_3^{-1} : \mathbb{R}^2 \to \mathbb{R}^2 : (\bar{r},\bar{\theta}) \mapsto (r,\theta)$

$$r = \bar{r} \cdot \hat{f}_{\text{sat}}(\bar{\theta}) \tag{11a}$$
$$\theta = \bar{\theta}. \tag{11b}$$

Polar Splines

The fluid property function ρ is approximated by a two-dimensional spline function $\hat{\rho}$ of degree n defined in the $(\bar{r},\bar{\theta})$-coordinates. Spline functions in dimensions higher than one are conventionally constructed for Cartesian coordinates, and the presence of the origin, where conventional polar coordinates exhibit a singularity, requires some care.

Knot Points

Referring to Figure 2, a set of knots Θ_ρ is defined in the $\bar{\theta}$-direction around the full circle, such that the first knot $\bar{\theta}_1$ is coincident with the point (h_{ro}, p_{ro}) on the vapor side of Ω_s, knots are spaced in a counter-clockwise (positive) direction, and the set includes θ_{j^*}. Note that Θ_ρ need not be the same as Θ_s (8), used to represent $\hat{f}_{\text{sat}}$. Computations are simplified if an even number of knots is used such that that both $\bar{\theta}_i$ and $\bar{\theta}_i + \pi$ are in Θ_ρ, simplifying consideration of negative $\bar{r}$. The multiplicity of the knots depends on the region and is discussed in the next section.

In the $\bar{r}$-direction, a set of knots

$$\Phi_\rho = \{-r_n, -r_{n-1}, \ldots, -r_1, 0, r_1, r_2, \ldots, r_{\max}\} \tag{12}$$

is defined such that 0 and 1 are elements, and n negative values are included to create some overlap at the origin. The multiplicity of the knots at $\bar{r} = 1$, corresponding Ω_s, is n so along Ω_s in the $\bar{r}$-direction, the spline function is C_0 but not C_1. All other knots have multiplicity 1 so that the spline function is C^n between any of the knots, C^{n-1} at any of the knots not on Ω_s.

Indexing

Indexing of B-spline functions in polar coordinates is more complex than for Cartesian coordinates. For the $\bar{r}$-direction, denote the set of integers that index the spline basis as

$$\mathscr{I} = \{i \in \mathbb{I} : 1 \le i \le i_{\max}\}, \tag{13}$$

where $i_{\max}$ is the number of spline basis functions. Let $i_{sp} \in \mathscr{I}$ denote the index for $\bar{r} = 1$, and decompose $\mathscr{I}$ into

$$\mathscr{I}_s = \{i_{\text{sp}}\} \tag{14a}$$
$$\mathscr{I}_1 = \{i \in \mathscr{I} : i > i_{\text{sp}}\} \tag{14b}$$
$$\mathscr{I}_2 = \{i \in \mathscr{I} : i < i_{\text{sp}}\}, \tag{14c}$$

so that $\mathscr{I}_s$ contains the basis indices in the $\bar{r}$-direction on Ω_s, $\mathscr{I}_1$ contains the basis indices in the $\bar{r}$-direction outside of Ω_s (region Ω_1), and $\mathscr{I}_2$ contains the basis indices in the $\bar{r}$-direction inside of Ω_s (region Ω_2).

In the $\bar{\theta}$-direction, the number of basis functions depends on the fluid state region, shown in Figure 4, making the B-spline indexing dependent on the region. In the two-phase region Ω_2, the B-spline basis functions in the $\bar{\theta}$ direction are periodic, defined for all values of $\bar{\theta}$, and all of the knots are multiplicity one. Then the set of integers that index the spline basis in the $\bar{\theta}$-direction in region Ω_2 is

$$\mathscr{J} = \{j \in \mathbb{I} : 1 \le i \le j_{\max}\}. \tag{15}$$

where $j_{\max}$ is the number of elements of Θ_ρ.

On the saturation curve, the density $\hat{\rho}$ is smooth as a function of $\bar{\theta}$ except at the points $\bar{\theta}_1$ and $\bar{\theta}_{j^*}$ where there is a transition between the actual saturation curve and the extended saturation curve, $\hat{\rho}$ is C_0 but not C_1 in the $\bar{\theta}$ direction, so the multiplicity of knots at $\hat{\theta}_1$ and $\hat{\theta}_{j^*}$ is n. This leads to a different number of B-spline basis functions in the $\bar{\theta}$ direction for Ω_s compared to Ω_2, requiring a different indexing. The set of integers that index the B-spline basis in the $\bar{\theta}$-direction in region Ω_s is

$$\mathscr{J}_s = \{j \in \mathbb{I} : 1 \le i \le j_{\max} + 2(n-1)\}. \tag{16}$$

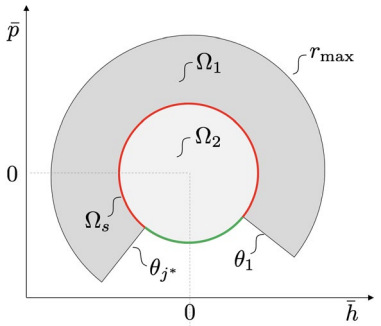

Figure 4. Domain Ω in normalized polar coordinates, showing the saturation curve Ω_s (red) the saturation curve extension (green), the two-phase region Ω_2, and single-phase region Ω_1.

As illustrated in Figure 4, $\hat{\rho}$ in the region Ω_1 is the partial annular set $(1, r_{\max}] \times [\bar{\theta}_1, \bar{\theta}_{j^*}]$. For many thermofluid systems of interest, the fluid property ρ for values of P_e and h_e corresponding to the region below the extended saturation curve, between the limits $\bar{\theta}_{j^*}$ and $\bar{\theta}_1$, is outside the region of interest and is therefore excluded from Ω.

Since the region Ω_1 is only partially annular, the B-spline functions in the $\bar{\theta}$ direction are Cartesian and not periodic in $\bar{\theta}$. The set of integers that index the spline basis functions in the $\bar{\theta}$-direction in region Ω_1 is

$$\mathscr{J}_1 = \{1 \leq j \leq j^* - 1 + n\} \tag{17}$$

Normalized Polar Spline Functions

The normalized polar spline function $\hat{\rho}$ is

$$\begin{aligned}\hat{\rho}(\bar{r},\bar{\theta}) = &\underbrace{\sum_{i\in\mathscr{I}_2}\sum_{j\in\mathscr{J}} c_{ij}B_{i,n}(\bar{r})B_{j,n}(\bar{\theta})}_{\Omega_2} \\ &+ \underbrace{\sum_{i\in\mathscr{I}_s}\sum_{j\in\mathscr{J}_s} c_{ij}B_{i,n}(\bar{r})B_{j,n}(\bar{\theta})}_{\Omega_s} \\ &+ \underbrace{\sum_{i\in\mathscr{I}_1}\sum_{j\in\mathscr{J}_1} c_{ij}B_{i,n}(\bar{r})B_{j,n}(\bar{\theta})}_{\Omega_1}\end{aligned} \tag{18}$$

where $B_{i,n}(\bar{r})$ and $B_{j,n}(\bar{\theta})$ are n-degree B-spline basis functions defined by knot sets Φ_ρ and Θ_ρ, respectively, and c_{ij} are spline coefficients that are computed by solving a constrained least squares or equivalent curve fitting algorithm. Note that although the knot sets are identical for each region, the multiplicities differ, so the $B_{jn}(\cdot)$ are different in each region.

Coefficient Calculation

Values for the coefficients c_{ij} in (18) are computed by solving a constrained least squares problem using a reference property calculator such as REFPROP. First, note that for values of $(\bar{r},\bar{\theta}) \in \Omega_s$,

$$\hat{\rho}(\bar{r},\bar{\theta}) = \sum_{j\in\mathscr{J}_s} c_{i_{\text{sp}}j}B_{i,n}(\bar{\theta}). \tag{19}$$

This is because all of the B-spline basis functions in the $\bar{r}$-direction vanish on Ω_s, except for those corresponding to index i_{sp}, which is identically 1 for $\bar{r} = 1$. This makes the contributions from the Ω_2 and Ω_1 terms in (18) to be zero for $(\bar{r},\bar{\theta}) \in \Omega_s$.

The coefficients c_{ij} for the Ω_s term in equation (18) are computed first using equation (19). For each value of $\bar{\theta}_j$ from a data set $\mathscr{D}_s = \{\bar{\theta}_j : 1 \leq j \leq N_s\}$, where N_s is any integer greater or equal to the number of coefficients in (19) and $\bar{\theta}_j$ suitable sample Ω_s, ρ_j is computed on the extended saturation curve using a thermofluid property calculator such as REFPROP. Then equation (19) may be solved for $c_{i_{\text{sp}}}$ by solving a least squares or similar curve fit problem.

Once the coefficients c_{ij} are computed for the saturation curve Ω_s, then equation (18) decomposes into two decoupled equations

$$\hat{\rho}(\bar{r},\bar{\theta}) - \underbrace{\sum_{i\in\mathscr{I}_s}\sum_{j\in\mathscr{J}_s} c_{ij}b_i^n(\bar{r})b_j^n(\bar{\theta})}_{\Omega_s} = \underbrace{\sum_{i\in\mathscr{I}_2}\sum_{j\in\mathscr{J}} c_{ij}b_i^n(\bar{r})b_j^n(\bar{\theta})}_{\Omega_2} \tag{20}$$

for the two-phase region Ω_2, and

$$\hat{\rho}(\bar{r},\bar{\theta}) - \underbrace{\sum_{i\in\mathscr{I}_s}\sum_{j\in\mathscr{J}_s} c_{ij}b_i^n(\bar{r})b_j^n(\bar{\theta})}_{\Omega_s} = \underbrace{\sum_{i\in\mathscr{I}_1}\sum_{j\in\mathscr{J}_1} c_{ij}b_i^n(\bar{r})b_j^n(\bar{\theta})}_{\Omega_1} \tag{21}$$

for the region Ω_1. Note that the terms on the left-hand sides of (20) and (21) labeled Ω_s may be assigned a numerical value given a value for $(\bar{r},\bar{\theta})$. For each element of a set of data $\mathscr{D}_2 = \{(\bar{r}_i,\bar{\theta}_j) : 1 \leq i \leq N_2,\ 1 \leq j \leq M_2\}$ over the region Ω_2, where $\bar{r}_i$ and $\bar{\theta}_j$ suitably sample Ω_2, and N_2 and M_2 are sufficiently large, values of ρ_{ij} are computed using a thermofluid property calculator such as REFPROP. These values are substituted for $\hat{\rho}$ in equations (20), defining an constrained least squares problem, which is solved for the unknown coefficients c_{ij}. The constraint arises because for coefficients near the origin, c_{ij} for positive $\bar{r}_i$ and $\bar{\theta}_j$ is identical to the coefficient c_{ij} for negative $\bar{r}_i$ and $\bar{\theta}_i + \pi$. This is repeated for a set of data $\mathscr{D}_1 = \{(\bar{r}_i,\bar{\theta}_j) : 1 \leq i \leq N_1\ 1 \leq j \leq M_1\}$ over the region Ω_1, where $\bar{r}_i$ and $\bar{\theta}_j$ suitably sample Ω_1.

Derivative Evaluation

Derivatives of $\hat{\rho}$ with respect to the $(\bar{r},\bar{\theta})$ variables may computed using efficient algorithms (de Boor, 1978; Piegl and Tiller, 1995), and add marginal overhead to the computational cost of evaluation of the B-spline function $\hat{\rho}$ at a given $(\bar{r},\bar{\theta})$. These derivative calculations are exact; there is no numerical differentiation. The derivatives of $\hat{\rho}$ with respect to the two input fluid property variables h_e and P_e are computed with the Jacobian of T, denoted DT,

$$\begin{bmatrix} \frac{d\hat{\rho}}{dh_e} \\ \frac{d\hat{\rho}}{dP_e} \end{bmatrix} = DT \cdot \begin{bmatrix} \frac{d\hat{\rho}}{d\bar{r}} \\ \frac{d\hat{\rho}}{d\bar{\theta}} \end{bmatrix} = DT_3 \cdot DT_2 \cdot DT_1 \cdot \begin{bmatrix} \frac{d\hat{\rho}}{d\bar{r}} \\ \frac{d\hat{\rho}}{d\bar{\theta}} \end{bmatrix}, \tag{22}$$

where DT_1, DT_2 and DT_3 are the Jacobians of T_1, T_2 and T_3, respectively.

At the origin, derivatives of $\hat{\rho}$ with respect to the engineering coordinates h_e and P_e are well defined and are evaluated by computing derivatives of $\hat{\rho}$ with respect to $\bar{r}$ evaluated at $(\bar{r},\bar{\theta}) = (0,0)$ and $(\bar{r},\bar{\theta}) = (0,\pi/2)$, respectively, and then by using elements of DT_1, DT_2 and DT_3 to transform back to engineering units:

$$\frac{\partial\hat{\rho}}{\partial h_e}\Big|_{\substack{P_e=P_{e0}\\ h_e=h_{e0}}} = \frac{\partial\hat{\rho}}{\partial\bar{r}}\Big|_{\substack{\bar{r}=0\\ \bar{\theta}=0}} \cdot \frac{1}{\hat{f}(0)} \cdot \frac{1}{h_s} \tag{23a}$$

$$\frac{\partial\hat{\rho}}{\partial P_e}\Big|_{\substack{P_e=P_{e0}\\ h_e=h_{e0}}} = \frac{\partial\hat{\rho}}{\partial\bar{r}}\Big|_{\substack{\bar{r}=0\\ \bar{\theta}=\pi/2}} \cdot \frac{1}{\hat{f}(\pi/2)} \cdot \frac{p_s}{P_{e0}}. \tag{23b}$$

This calculation is well-defined because the domain of the spline in the $\bar{r}$ direction was extended to include negative values of $\bar{r}$, and also because of the structures of T_1, T_2 and T_3 make some of the off-diagonal terms in the Jacobians zero. Higher derivatives with respect to h_e and P_e are computed similarly.

3 Parabolic Coordinates

Normalized parabolic coordinates are similar to normalized polar coordinates, defined by the composition of three coordinate transformations, $T = T_3 \circ T_2 \circ T_1$, reusing notation from Section 2. T_1 is the same, but here T_2 defines parabolic instead of polar coordinates.

Parabolic Coordinate Transformation T_2

In the scaled (h,p) coordinates, define the parabolic coordinate transformation $T_2 : \mathbb{R}^2 \to \mathbb{R}^2 : (p,h) \mapsto (\sigma,\tau)$ as

$$\sigma = \text{sign}(h) \cdot \sqrt{\sqrt{h^2+p^2} - p} \tag{24a}$$

$$\tau = \sqrt{\sqrt{h^2+p^2} + p}, \tag{24b}$$

with inverse $T_2^{-1} : \mathbb{R}^2 \to \mathbb{R}^2 : (\sigma,\tau) \mapsto (h,p)$

$$h = \sigma \cdot \tau \tag{25a}$$

$$p = (\tau^2 - \sigma^2)/2. \tag{25b}$$

Figure 5 shows constant contours of σ and τ on the (h,p)-plane, along with the saturation curve for R410A for reference.

Saturation Curve τ-Distance Normalization T_3

The third coordinate transformation $T_3 : \mathbb{R}^2 \to \mathbb{R}^2 : (\sigma,\tau) \mapsto (\bar{\sigma},\bar{\tau})$ normalizes the saturation curve to be the locus $\bar{\tau} = 1$, and is defined as

$$\bar{\sigma} = \sigma \tag{26a}$$

$$\bar{\tau} = \tau / \hat{f}_{\text{sat}}(\sigma), \tag{26b}$$

with inverse $T_3^{-1} : \mathbb{R}^2 \to \mathbb{R}^2 : (\bar{\sigma},\bar{\tau}) \mapsto (\sigma,\tau)$

$$\sigma = \bar{\sigma} \tag{27a}$$

$$\tau = \bar{\tau} \cdot \hat{f}_{\text{sat}}(\bar{\sigma}), \tag{27b}$$

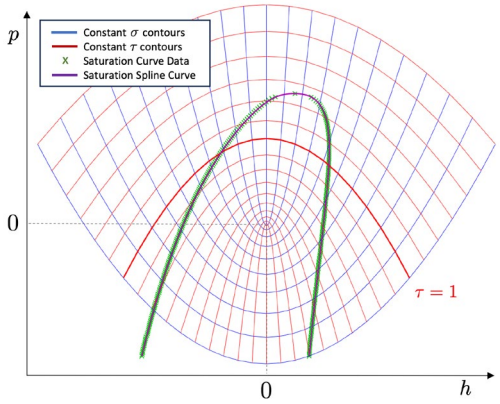

Figure 5. Constant contours of σ (blue) and τ (red), on the (h,p)-plane. Also shown is the saturation curve represented as a spline function (purple), fit to saturation curve data (*).

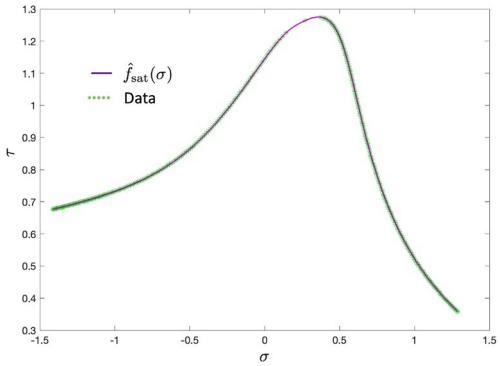

Figure 6. Saturation curve defined as the spline function $\tau = \hat{f}_{\text{sat}}(\sigma)$ for R410A. Note the gap in data around the critical point.

where $\tau = \hat{f}_{\text{sat}}(\sigma)$ denotes an approximation to $\tau = f_{\text{sat}}(\sigma)$, which defines the saturation curve in τ as a function of σ. Just as for polar coordinates, we use a property calculator to compute pairs of values for τ and σ on the saturation curve, using T_1 and T_2, and then fit a spline to the data to give the approximation $\tau = \hat{f}_{\text{sat}}(\sigma)$, as shown in Figure 6 for R410A.

Normalized Parabolic Spline Functions

The density function $\hat{\rho}$ is realized as a 2-dimensional n-degree B-spline function in $(\bar{\sigma},\bar{\tau})$ coordinates,

$$\hat{\rho}(\bar{\sigma},\bar{\tau}) = \sum_{i=1}^{M}\sum_{j=1}^{N} c_{ij} B_{i,n}(\bar{\sigma}) B_{j,n}(\bar{\tau}), \tag{28}$$

defined on the rectangular domain $[-\bar{\sigma}_{\max},\bar{\sigma}_{\max}] \times [0,\bar{\tau}_{\max}]$, which defines $\Omega = \Omega_1 \bigcup \Omega_2 \bigcup \Omega_s$. Figure 7 shows the domain in the $(\bar{\sigma},\bar{\tau})$ coordinates, and its pull back into the (h,p) coordinates, for R410A.

Parabolic coordinates exhibit two characteristics that at first seem to be obstacles but with some thought present no problems. First, there is an apparent singularity along the p-axis ($h = 0$), where $\tau = 0$ along the negative p axis, and where $\sigma = 0$ along the positive p axis. Along a constant σ contour, the sign of σ changes discontinuously from positive in the right half plane to negative in the left half plane

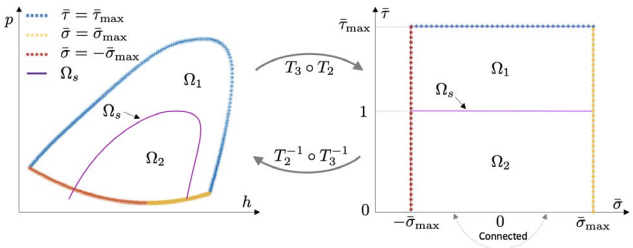

Figure 7. Ω_1, Ω_2, Ω_s and their boundaries in (h, p) coordinates (left) and $(\bar{\sigma}, \bar{\tau})$ coordinates (right) for R410A.

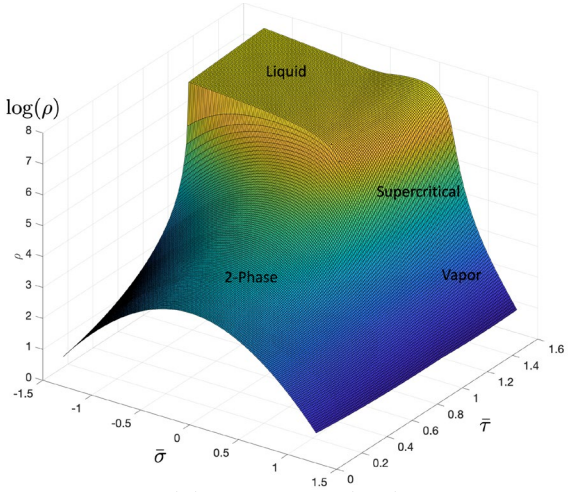

Figure 8. $\log(\hat{\rho})$ for R410A in $(\bar{\sigma}, \bar{\tau})$-coordinates.

at the p axis. This is caused by the sign(h) used to define σ in (24). Second, the boundary of Ω seems to have a different number of edges when represented in $(\bar{\sigma}, \bar{\tau})$ coordinates, compared to (h, p) coordinates. Indeed, the rectangular region in (σ, τ), with four boundary edges, maps to a region in (h, p) coordinates that is bounded by two parabolas, as shown in Figures 5 and 7.

Fortunately these characteristics do not present any obsticles. Figure 7 shows how the rectangular domain Ω in $(\bar{\sigma}, \bar{\tau})$ coordinates maps back to (h, p) coordinates, with the boundaries shown in color. The two vertical boundaries along $-\bar{\sigma}_{\max}$ and $\bar{\sigma}_{\max}$ map to the lower boundary of Ω in (h, p), while the lower boundary $\bar{\tau} = 0$ is in fact not a boundary at all. Points on the positive $\bar{\sigma}$-axis are connected to points on the negative $\bar{\sigma}$-axis, so that $\hat{\rho}(\bar{\sigma}, 0)$ is equivalent to $\hat{\rho}(-\bar{\sigma}, 0)$, for $0 < \bar{\sigma} \leq \sigma_{\max}$. The $\bar{\tau} = 0$ axis is equivalent to the negative p axis, which is inside Ω. Therefore, when defining a spline function $\hat{\rho}$ on Ω, we simply need to ensure that coefficients are constrained so that the spline function is connected across the $\tau = 0$ axis. This ensures that the spline $\hat{\rho}$ and its first $n-1$ derivatives are continuous across $\tau = 0$, and are well defined for all points in $\Omega_1 \bigcup \Omega_2$. This is precisely how the spline coefficients were computed for polar coordinates around the origin (by extending $\bar{r}$ to be negative, and then constraining coefficients for positive and negative $\bar{r}$ to ensure continuity at 0) except for parabolic coordinates, it must be done across the entire $\bar{\tau}$ axis.

Knot indexing is simplified compared to polar coordinates. In the $\bar{\tau}$-direction, knots are spaced from 0 to $\tau_{\max}$, with a knot of multiplicity n placed at 1, which is the saturation curve in $(\bar{\sigma}, \bar{\tau})$ coordinates. All other knots are multiplicity 1. In the $\bar{\sigma}$-direction, knots are spaced from $-\bar{\sigma}_{\max}$ to $\bar{\sigma}_{\max}$, all of multiplicity 1. This defines the degree-n B-spline basis functions $B_{i,n}$ and $B_{j,n}$ in the $\bar{\sigma}$ and $\bar{\tau}$ directions, respectively.

4 Discussion

Both methods are computationally efficient. The calculation of T in polar coordinates requires 11 floating point operations, compared to 14 for the equivalent calculation in parabolic coordinates. Only one floating point division is needed, but if $\hat{f}_{\text{sat}}^{-1}$ is approximated by a spline instead of $\hat{f}_{\text{sat}}$, then that division becomes a multiplication. Additionally, both methods require one 1-d spline function evaluation of $\hat{f}_{\text{sat}}$, plus evaluation of the 2-d spline function $\hat{\rho}$. Spline derivatives are computed essentially for free and pulled back into the (P_e, h_e) coordinates using DT.

Polar coordinates have the advantage of familiarity and simplicity in terms of domain boundaries. Computing derivatives at the origin is not ill-posed because the spline is defined for some range of negative $\bar{r}$, and the derivatives are computed in the (P_e, h_e) coordinates using elements of DT that are all well defined at the origin.

There are some disadvantages to polar coordinates. Indexing is complex. The extended saturation curve is clumsy, and the spline function $\hat{\rho}$ near the point (h_{l0}, p_{l0}) can fit data poorly because of the large change in derivative near this point in the $\bar{\theta}$ direction. Despite their unfamiliarity, parabolic coordinates seem more natural once their peculiarities are mastered, as these issues are avoided entirely. One issue with parabolic coordinates is that the domain Ω is "warped" by T_3. In particular, the vapor region to the right of the saturation curve may be insufficiently covered using a rectangular domain in $(\bar{\sigma}, \bar{\tau})$ coordinates. This is apparent in Figure 7. One solution is to make use of a non-rectangular domain, extending $\bar{\tau}$ for positive values of $\bar{\sigma}$.

It should be emphasized that the spline functions $\hat{f}_{\text{sat}}$ and $\hat{\rho}$ (and its derivatives) should be used consistently and exclusively in any simulation model. These in effect become the definitions of the saturation curve and fluid property function, even though they are spline approximations of a data set, which is in turn was computed from a Helmholtz energy function, which itself is defined to be the reference standard, but is in reality a curve fit to experimental data. It is important not to mix different representations of the saturation curve or fluid property functions in the same model, because even slight differences, especially near the saturation curve, can result in significant differences (or even failures) in simulation results.

For some properties, especially density ρ, we have noticed that a 2-dimensional spline fit to $\log \rho$, (or, more rigorously, $\log(\rho/\rho_0)$ for some constant density ρ_0) instead

of ρ, reduces approximation error, especially near the saturation curve, at the expense of one additional exponential computation during evaluation. A rendering of $\log(\hat{\rho})$ on Ω in $(\bar{\sigma}, \bar{\tau})$-coordinates for R410A is shown in Figure 8. The log reduces the magnitude of the first and second derivatives near the saturation curve, reducing approximation error. We have observed similar behavior for polar coordinates. However we offer no formal proof of this statement.

In practice, significant attention must be paid to data cleaning. Values for ρ, P and h computed by REFPROP, for example, may exhibit small errors that can adversely affect the curve fitting computation. This is especially true for mixtures such as R454C. Errors are caused by finite, nonzero iteration termination conditions in REFPROP's code, or sometimes by failures to converge, and are especially apparent around the saturation curve at high pressures, although other regions can also exhibit errors. A full discussion may be found in (Laughman et al., 2024).

There is little performance gain to be had by implementing these functions directly in Modelica, since the Modelica compiler will translate it into C anyway, and that code is unlikely to be more efficient than the relatively simple, hand coded function. However, one potential performance improvement is largely unexploited: derivative evaluation can be done largely for free when evaluating $\hat{\rho}$. Unfortunately the Modelica compiler does not know that, so it may evaluate the function multiple times to compute $\hat{\rho}$ and its derivatives. Perhaps there is a way to prevent this behavior in Modelica?

Finally, we speculate that the fluid property coefficients, and the 2-dimensional spline function evaluations, could be implemented in integer arithmetic. Although this is not likely to improve numerical efficiency by a large margin in a modern superscalar architecture, it would reduce memory storage requirements. This in turn could reduce simulation time because of improved cache efficiency.

5 Conclusion

This paper presents two methods for computing fluid property functions in simulation models. Both make use of coordinate transformations that align one coordinate with the saturation curve. This provides for precise representation of the fluid property function at the saturation curve, and for connected domains of interest including the liquid, vapor, supercritical and two-phase regions. Both approaches make use of spline function approximation in these special coordinates, and are numerically efficient, well conditioned, and allow for efficient calculation of derivatives up to any desired order that are precise up to processor numerical tolerance.

References

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

Adrian Bejan. *Advanced Engineering Thermodynamics*. Wiley, 1993.

Ian H. Bell, Jorrit Wronski, Sylvain Quoilin, and Vincent Lemort. Pure and pseudo-pure fluid thermophysical property evaluation and the open-source thermophysical property library coolprop. *Industrial & Engineering Chemistry Research*, 53(6):2498–2508, 2014. doi:10.1021/ie4033999. URL `http://pubs.acs.org/doi/abs/10.1021/ie4033999`.

Carl de Boor. *A Practical Guide to Splines*. Springer, 1978.

M. Kunick. *Fast Calculation of Thermophysical Properties in Extensive Process Simulations with the Spline-Based Table Look-Up Method (SBTL)*. Number 618. Fortschritt - Berichte VDI, 2018. ISBN 978-3-18-361806-4.

C. Laughman, H. Qiao, and S. A. Bortoff. System and method for calculation of thermofluid properties using saturation curve-aligned coordinates. U.S. Patent 11,739,996, Aug. 29 2023.

Christopher R. Laughman, Vedang Deshpande, Ankush Chakrabarty, and Hongtao Qiao. Enhancing thermodynamic data quality for refrigerant mixtures: Domain-informed anomaly detection and removal. In *Procedings of the* 20th *International Refrigeration and Air Conditioning Conference at Purdue*, 2024.

E. W. Lemmon, I.H. Bell, M. L. Huber, and M. O. McLinden. NIST Standard Reference Database 23: Reference Fluid Thermodynamic and Transport Properties-REFPROP, Version 10.0, National Institute of Standards and Technology, 2018. URL `https://www.nist.gov/srd/refprop`.

L. Piegl and W. Tiller. *The NURBS Book*. Springer, 2 edition, 1995.

R. Span. *Multiparameter Equations of State*. Springer-Verlag, 2000.

Matthis Thorade and Ali Saadat. HelmholtzMedia - a fluids properties library. In *Proceedings of the 9th International Modelica Conference*, pages 63–70, 2012.

Matthis Thorade and Ali Saadat. Partial derivatives of thermodynamic state properties for dynamic simulation. *Environmental Earth Sciences*, 2013.

D.J. Van Peursem and G. Xu. System and method for efficient computation of simulated thermodynamic property and phase equilibrium characteristics using comprehensive local property models, 2010.

Gang Xu. Super-linear approximation of dynamic property values in a process control environment, 2020.