Approximation of Refrigerant Properties for Dynamic Vapor Compression Cycle Models

Laughman, C.R.; Van, C.

TR2018-027 February 2018

Abstract

Thermophysical refrigerant property models play an essential role in dynamic models of vapor compression cycles, due to their highly nonlinear behavior, their coupling with many phenomena of interest, and the number of property computations required to simulate a cycle. As conventional iterative calculation methods are often too slow to be practically useful for such simulations, we compare two different approximation approaches, including one method that incorporates approximations of the liquid and vapor saturation lines, and another which approximates the entire property surface over a given domain. These methods are implemented in Modelica, and are demonstrated to successfully describe the nonlinear behavior of the refrigerant R-32 in a computationally and memory-efficient manner.

MATHMOD

Approximation of Refrigerant Properties for Dynamic Vapor Compression Cycle Models

Cong Tuan Son Van [*] **Christopher R. Laughman** [**]

[*] *Kansas State University (e-mail: congvan@ksu.edu)*
[**] *Mitsubishi Electric Research Laboratories, Cambridge, MA, 02139*
USA (e-mail: laughman@merl.com)

Abstract: Thermophysical refrigerant property models play an essential role in dynamic models of vapor compression cycles, due to their highly nonlinear behavior, their coupling with many phenomena of interest, and the number of property computations required to simulate a cycle. As conventional iterative calculation methods are often too slow to be practically useful for such simulations, we compare two different approximation approaches, including one method that incorporates approximations of the liquid and vapor saturation lines, and another which approximates the entire property surface over a given domain. These methods are implemented in Modelica, and are demonstrated to successfully describe the nonlinear behavior of the refrigerant R-32 in a computationally and memory-efficient manner.

Keywords: Thermodynamics, Iterative methods, Splines, Interpolation approximation, Modelica

1. INTRODUCTION

The use of dynamic models to design variable capacity vapor compression cycles for comfort conditioning, refrigeration, and energy recovery applications is becoming increasingly important to understand performance limits and analytically design controls, because these systems' behavior and their dynamic interactions with the enclosing environment are multivariable and occur over many timescales. While these overall cycle models have many components, the thermophysical refrigerant property models used are a vital part of their structure, as these properties are both highly nonlinear and tightly coupled to many phenomena of interest. Beyond these physical considerations, a significant portion of the time required for cycle simulations is spent performing these property computations. The successful implementation of a model-based design process is thus deeply intertwined with the accuracy and execution speed of the computational property models.

While there is no reference implementation of computational methods pertaining to thermophysical refrigerant properties, the de facto standard appearing in the literature is a set of Fortran codes created by researchers at NIST called REFPROP (Lemmon et al., 2013), which implements reference equations of state (EoS) for many refrigerants that can be used to calculate all fluid properties. There are distinct differences between the objectives of REFPROP and the requirements of property models used in dynamic cycle simulation, however; where REFPROP is primarily concerned with the accuracy of the fluid simulations and the characterization of fluid mixture properties for different applications, dynamic cycle simulation usually assumes a given composition for the working fluid and seeks to conduct cycle simulations with tradeoffs between accuracy and computational speed that are appropriate for dynamic cycle simulations.

The computational needs of refrigerant property models have motivated a wealth of previous investigations in approximating thermophysical properties that improve the speed of calculation while maintaining high accuracy. Due to the importance of water and steam for power generation, a great deal of technical work has been done in constructing both polynomial (Åberg et al., 2017) and spline (IAPWS, 2015) approximations for this working fluid properties which are significantly faster than iterative methods. Application-oriented perspectives are also presented by Aute and Radermacher (2014) and Pini et al. (2015), in which the authors construct fast rational polynomial and look-up table descriptions to approximate the properties for the study of vapor compression cycles.

This prior work thus demonstrates the feasibility of approximating these thermophysical properties for dynamic simulation, but a range of opportunities remain for developing improved modeling approaches. While there are multiple methods by which the refrigerant properties may be approximated, tradeoffs may be implicitly present in these methods so that different applications need different approximate models. In this work, we compare two different property models that reduce the number of iterations needed for flash calculations: one in which we approximate only the saturation curves, as well as an alternate method of directly approximating all of the thermodynamic variables over the entire domain of interest in a given set of coordinates. Moreover, we also study the implementation of these methods in Modelica (Modelica Association, 2017), an equation-based and object-oriented physical modeling language, that has emerged as a promising technology for expressing and working with models that enables the distributed and scalable development of large-scale cycles.

The structure of this paper is as follows: in Section 2, we describe three methods for calculating refrigerant properties: the iterative calculation approach used in REFPROP (referred to as the "standard method"), the method of approximating only the liquid and vapor saturation curves, and the method of approximating the entire property surface. We then compare the efficacy of Modelica implementations of all three methods in Section 3 by comparing their computational speed, accuracy, and consistency of these approximation methods, and also present salient implementation details. Finally, we briefly offer some conclusions and suggestions for future work in Section 4.

2. PROPERTY CALCULATION METHODS

In this section, we first review the fundamental thermodynamic principles that govern these calculations and the approach taken in the standard method as a means of motivation, and then describe the three approximation methods that were explored. While these methods can be adapted to many different thermodynamic coordinates, we focus on the use of pressure p and mixture specific enthalpy h due to their common use in modeling vapor compression cycles. We also limit the study of these methods to the refrigerant R-32 and its technical EoS, described in Span and Wagner (2003), due to its increasing use in the air-conditioning industry because of its favorable environmental characteristics.

2.1 Standard Method

Modern equations of state (EoS) used to describe the thermophysical properties of refrigerants largely characterize a given fluid in terms of its Helmholtz energy $A(T, \rho)$, as its caloric properties can be calculated without integrating this EoS, and all thermodynamic properties can be calculated via linear combinations of derivatives of A with respect to the temperature T and the mixture density ρ (Span, 2000). One challenge of using such a Helmholtz EoS for cycle models is that p and h are often used as thermodynamic coordinates, while the Helmholtz energy is parameterized in terms of T and ρ. Moreover, the phase region of a given state point is often not known a priori, requiring the use of flash calculations to determine the phase and perform the appropriate equilibrium calculations. This is a particular challenge for dynamic Modelica cycle simulations, as there are few guarantees on the initial values of the state variables chosen by a Modelica tool to ensure consistency among the differential algebraic equations defining the system. As a result, iterative methods are often used to determine the values of (T, ρ) that are consistent with the input values of (p, h) before calculating other thermophysical property outputs. The basic structure of the iterative method used by REFPROP for a pure fluid is illustrated in Figure 1 for a representative flash calculation; this program is often called PHFLASH.

This method essentially has three distinct levels. At the highest level, illustrated in Figure 1, the algorithm sequentially searches through the possible phase regions in which

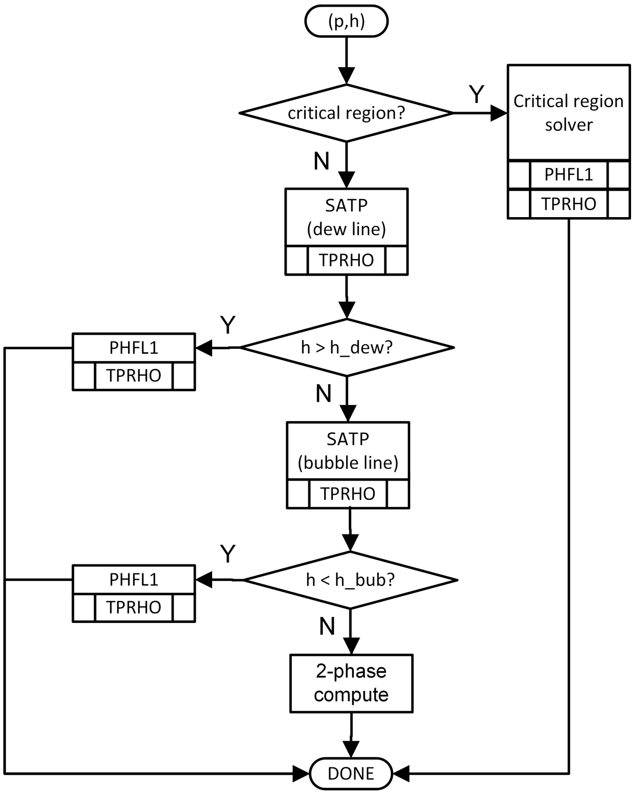

Fig. 1. Flow chart describing iterative calculations used in PHFLASH by REFPROP.

the given coordinates may lie: the supercritical region, the vapor region, the liquid region, or the two-phase region. While the location of the state point in the supercritical region can be identified via the proximity of the specified pressure to the established critical pressure of the fluid, location in the remainder of the regions is determined by calculating T and ρ on the saturation curves from p using the function SATP, given a specified fluid phase.

SATP itself contains two levels of iteration. At the outer layer of iteration, SATP uses an iterative method to refine the temperature T according to the rule

$$\hat{T}_k(p,h) = \hat{T}_{k-1} + \Delta T,$$

until $\Delta T = |(\hat{g}_l - \hat{g}_v) \,/\, (\hat{s}_l - \hat{s}_v)|$ is less than a specified tolerance, where $\hat{g}_l$ and $\hat{g}_v$ are the saturated liquid and vapor Gibbs energies, $\hat{s}_l$ and $\hat{s}_v$ are the specific entropy of saturated liquid and vapor, and k signifies the number of the iteration. Because this Helmholtz energy-based formulation represents a fundamental EoS, the $\hat{g}$ and $\hat{s}$ at each step can be computed from $\hat{T}_k$ and $\hat{\rho}_k$; however, an estimate for $\hat{\rho}_k$ must itself be obtained at each iteration. This is accomplished by using $\hat{T}_k$ and p to determine the value of $\hat{\rho}_k$ in an inner set of iterations which call TPRHO to locate the root of

$$\hat{p}_j(\hat{T}_k, \hat{\rho}_j) - p = 0$$

via a second order Newton root-finding method for the given value of $\hat{T}_k$ in a given single-phase region until $|\hat{p}_j - p|$ is less than a specified tolerance.

After obtaining $\hat{T}$ and $\hat{\rho}$ on the given saturation line for the specified p, REFPROP calculates $\hat{h}$ on this saturation line and can determine if the desired state point is in the

Table 1. REFPROP/Python timing profile

Functions	Total time (in minutes)
PHFLASH	26.71
TPRHO	22.68
PHFL1	17.18
criticalRegionHelper	13.84
SATP	6.65

adjoining single-phase region. If this is the case, PHFL1 calculates the thermodynamic properties by also using the secant root-finding method to iteratively find the root of

$$\hat{h}_k(p, \hat{T}_k) - h = 0.$$

This iteration stops when $|\hat{h}_k - h|/h$ is less than some tolerance. The estimates for $\hat{\rho}_k$ needed to determine $\hat{h}_k$ are also calculated via TPRHO. In the case that the desired state point is not in either the vapor or the liquid regions, REFPROP determines that the state point is in the two-phase region and calculates the static quality via

$$x = \frac{h - \hat{h}_{bub}}{\hat{h}_{dew} - \hat{h}_{bub}},$$

where h_{bub} and h_{dew} refer to the specific enthalpy on the saturation boundaries between the liquid and two-phase regions, and between the two-phase and vapor regions, respectively. This quality x is then applied as the weight between the set of properties calculated on both saturation lines (Bejan, 2006).

While this structure of nested iterative loops is employed in REFPROP because of its connection to the underlying principles of thermodynamics and fluid equilibria, the sheer number of iterations needed for every property calculation represents a significant barrier in the construction of fast cycle models. To better study the structure of these algorithms and their impact on the run-time operation of property calculations, we created a test implementation of the REFPROP equilibrium algorithms in Python and profiled its performance over a domain of interest for cycle dynamics, e.g., $(p, h) \in \Omega := [0.3 \text{ MPa}, 12 \text{ MPa}] \times [100 \text{ kJ/kg}, 700 \text{ kJ/kg}]$ with step sizes $\Delta p = 20$ kPa, $\Delta h = 10$ kJ/kg, as illustrated in Table 1. This profile was generated by calculating the density with PHFLASH for each point in the domain and the amount of time in the specified subroutines was measured.

Both these profiling results and the analysis of PHFLASH motivated our study of two potential approaches to speed up the flash calculations:

- The approximation of T and p along the saturation curves eliminates the iterations of TPRHO in SATP, which significantly increases the speed of SATP. The inner iterations are maintained, however, to ensure the correct calculation of the density everywhere over the domain of interest. This approximation method also has limited memory requirements.
- We also directly approximated T and ρ from the thermodynamic input coordinates of p and h over the domain of interest. This second approach eliminates all the iterations within PHFLASH and drastically improves its speed, at a cost of higher memory requirements.

In addition to these two approximation methods, we also implemented the original REFPROP methods in Modelica. This facilitated the direct comparison of the speed and accuracy of the three different methods directly in Modelica; this Modelica implementation of the REFPROP algorithms will be referred to as REFPROP/Modelica, whereas the original Fortran implementation will be referred to as REFPROP/Fortran.

2.2 Saturation Curve Approximation

While many possible approaches can be used to approximate the 1-D saturation curves which represent (T, ρ_l, ρ_v) as $f(p)$ by decomposing functions $f(x)$ (in L^2) provided a basis $\{\phi_n(x)\}$ as

$$f(x) = \sum_{n=0}^{\infty} \langle f, \phi_n \rangle \, \phi_n(x),$$

we used B-splines (Piegl and Tiller, 1995) as the particular choice of basis to approximate these functions because of the ability to control their smoothness and the simplicity of their representation. Adopting the notation of Piegl and Tiller (1995), we describe B-splines for a given set of non-decreasing collocation points $\mathcal{Q} := \{\bar{u}_k\}, 0 \le k \le n$ to generate an interpolated curve $\mathcal{C}(u), u \in \mathbb{R}^1$, going through $\mathcal{Q}$, which can be decomposed as

$$\mathcal{C}(u) = \sum_{i=0}^{n} \mathcal{P}_i \mathcal{N}_{i,p}(u), \qquad (1)$$

where

- $p \ge 1$, the polynomial degree, is an integer,
- $\mathcal{P}_i$ are the control points, which are unknown,
- $m := n + p + 1, m + 1$ is the number of knots,
- $\mathcal{U} := \{u_j\}, 0 \le j \le m$ is the knot vector, where u_j (called knots) are calculated by

  $$\begin{aligned} &u_0 = \cdots = u_p = \bar{u}_0, \\ &u_{m-p} = \cdots = u_m = \bar{u}_n, \\ &u_{j+p} = \frac{1}{p} \sum_{i=j}^{j+p-1} \bar{u}_i, \quad 1 \le j \le n - p, \text{and} \end{aligned}$$

  the knot vector $\mathcal{U}$ is used to calculate the basis $\{\mathcal{N}_{i,p}\}$, and
- the basis $\{\mathcal{N}_{i,p}\}$, which is defined as

  $$\mathcal{N}_{i,0}(u) = \begin{cases} 1, & \text{if} \quad u_i \le u < u_{i+1}, \\ 0, & \text{otherwise} \end{cases}$$

  $$\begin{aligned} \mathcal{N}_{i,p}(u) = &\frac{u - u_i}{u_{i+p} - u_i} \mathcal{N}_{i,p-1}(u) \\ &+ \frac{u_{i+p+1} - u}{u_{i+p+1} - u_{i+1}} \mathcal{N}_{i+1,p-1}(u), \end{aligned}$$

  where $0 \le i \le m - 1$.

The unknown control points $\mathcal{P}_i, 0 \le i \le n$, can be found by solving a $(n+1) \times (n+1)$ linear system that is derived from substituting $\bar{u}_k, 0 \le k \le n$ into (1).

The saturation curves were thus approximated by dividing the input pressure p into n uniform sub-intervals and constructing B-splines to interpolate the output variables ρ_l and ρ_v. The Modelica implementation consisted of a distinct model to generate the control points $\mathcal{P}_i, 0 \le i \le$

$n = 100$ for each of the variables using the REFPROP/-Modelica implementation, which were then allocated as Modelica constants and used to do the interpolation by reconstructing the saturation lines via Equation (1). The resulting 6 vectors $\mathcal{P}$ consume little memory and do not significantly increase the simulation and initialization time of the overall model.

2.3 Surface Approximation

As many vapor compression cycle models use p and h as the state variables from which all other thermophysical properties are computed, we also investigated the direct approximation of the properties over the whole (p, h) domain, rather than just approximating only the saturation curves. Accordingly, we divide the surface $[p_{\min}, p_{\max}] \times [h_{\min}, h_{\max}]$ into $n \times m$ cells, and use two dimensional B-splines interpolate the surface using the grid of collocation points $\mathcal{Q} = \{\bar{u}_0 = p_{\min}, \ldots, \bar{u}_n = p_{\max}\} \times \{\bar{v}_0 = h_{\min}, \ldots, \bar{v}_m = h_{\max}\}$. Given the general collocation grid $\mathcal{Q}_{k,l} = \{(\bar{u}_k, \bar{v}_l)\}$, $0 \le k \le n, 0 \le l \le m$, we construct the B-spline $\mathcal{S}(u, v)$ of this grid by taking a bidirectional net of control points $\mathcal{P}_{i,j}$, two knot vectors $\mathcal{U}, \mathcal{V}$, and the tensor products $\mathcal{N}_{i,p}(u)\mathcal{N}_{j,q}(v)$ of the 1D B-spline basis

$$\mathcal{S}(u, v) = \sum_{i=0}^{n} \sum_{j=0}^{m} \mathcal{N}_{i,p}(u)\mathcal{N}_{j,q}(v) P_{i,j}. \tag{2}$$

Note that the small cells can be non-uniform; more cells should be distributed on the non-smooth part of the surface and less cells can be distributed on the smooth part of the surface. While the large size of matrix $\mathcal{P}$ can present challenges in some Modelica tools due to memory limitations, the `ExternalObject` interface was found to reduce both the memory storage and model initialization time by retrieving the points grid from external files.

3. RESULTS

The basic comparison between methods over the sub- and supercritical regions was performed in Modelica over $(p, h) \in \Omega := [0.3 \text{ MPa}, 12 \text{ MPa}] \times [100 \text{ kJ/kg}, 700 \text{ kJ/kg}]$ with step sizes $\Delta p = 20$ kPa, $\Delta h = 10$ kJ/kg in which we compute the functions $T(p, h)$ and $\rho(p, h)$ over Ω ten times to reduce the possible variation between experiments. Similarly, the methods were compared only in the subcritical region over $(p, h) \in \Omega_{sc} := [0.3 \text{ MPa}, 5.6 \text{ MPa}] \times [100 \text{ kJ/kg}, 700 \text{ kJ/kg}]$. We used the default tolerances from REFPROP, which in `PHFL1` equal 10^{-8} and in `TPRHO` equal 10^{-3}. The CPU running time was also measured during the experiment and averaged over all ten runs. These experiments were done on a Windows 10 machine running Dymola 2017 FD01 using one core of i7-7700K @ 4.20 GHz and 20 Gb of RAM.

These approximation methods also necessitated the selection of a few important numerical parameters. The saturation curve approximation was constructed by dividing the interval $[p_{\min}, p_c]$ into 200 uniform sub-intervals to approximate T, ρ_l, and ρ_v along the saturation lines. To implement the surface approximation method, we divided the surface $[p_{\min}, p_{\max}] \times [h_{\min}, h_{\max}]$ into a non-uniform set of 2000 × 2000 small cells. We divided the interval $[p_{\min}, 4300]$ on the p-axis into 1000 uniform sub-intervals with step size $\Delta p = 4$, and the interval $[4300, p_{\max}]$ into 1000 uniform sub-intervals with step size $\Delta p = 7.7$. Similarly, we divided the interval $[h_{\min}, 200]$ on the h-axis into 1000 uniform sub-intervals with step size $\Delta h = 0.1$, and the interval $[200, h_{\max}]$ into 1000 uniform sub-intervals with step size $\Delta h = 0.5$. Such non-uniform cell sizes were required because additional cells were needed to compensate for the large derivatives of ρ with respect to p and h along the saturated liquid line, especially in the region $[p_{\min}, 4300] \times [h_{\min}, 200]$.

Table 2. Accuracy comparison over Ω

Type	RP/Fortran vs. RP/Modelica	RP/Modelica vs. Sat. Approx.	RP/Modelica vs. 2-D Approx.
$\bar{\rho}_{\text{err}}$ [kg/m^3]	2.65e-5	1.51e-3	5.39e-3
$\tilde{\rho}_{\text{err}}$ [kg/m^3]	2.50e-5	2.90e-5	1.30e-5
ρ_{eVar}	0.999999999997	0.999999901093	0.999988997021
$\max \rho_{\text{err}}$	0.05	6.47	109.67
$\bar{T}_{\text{err}}$ [K]	2.51e-5	5.53e-5	2.08e-5
$\tilde{T}_{\text{err}}$ [K]	2.50e-5	2.90e-5	1.20e-5
T_{eVar}	0.999999999834	0.999999999279	0.999999999816
$\max T_{\text{err}}$	1.94e-4	3.44e-2	6.20e-3

Results comparing REFPROP/Fortran (i.e., RP/Fortran) to REFPROP/Modelica (i.e., RP/Modelica) for the domain Ω are listed in Table 2, while the results for the domain Ω_{sc} are listed in Table 3. In both of these tables, $\tilde{x}$ denotes the median value, $\bar{x}$ denotes the mean value, and 'eVar' and 'err' stand for explained variation and absolute error.

These tables indicate that the errors are quite small and that the explained variations are nearly equal to unity, suggesting that the is very little scatter around the mean errors. Closer scrutiny of these errors indicates that they are concentrated around the critical area $\Omega_{crit} := [5.6 \text{ MPa}, 6 \text{ MPa}] \times [350 \text{ kJ/kg}, 550 \text{ kJ/kg}]$, whereas the errors of T and ρ over the subcritical area as illustrated in Table 3 are much smaller than the comparable errors over the whole region. This can be seen in Figure 2, which illustrates a checkerboard plot of the errors between REFPROP/Fortran and REFPROP/Modelica in the immediate vicinity of the critical region. Additional analysis indicates that the discrepancy between these model outputs can be attributed to the different numbers of decimal places used in Fortran (14 digits) and Modelica (16 digits). This quantity is important when the number of iterations in `PHFL1` and `TPRHO` become large; for example, at the seventh iteration of `PHFL1` at the coordinates $(p, h) = (5.8 \text{ MPa}, 410 \text{ kJ/kg})$ the output of the `TPRHO` iterations for both REFPROP/Fortran and REFPROP/Modelica start at nearly the same value (only differing at the 13th decimal place), but a deviation appears at the 12th decimal place after only 5 iterations, and reaches the 9th decimal place

Table 3. Accuracy comparison over Ω_{sc}

Type	RP/Fortran vs. RP/Modelica	RP/Modelica vs. Sat. Approx.	RP/Modelica vs. 2-D Approx.
$\bar{\rho}_{\text{err}}$ [kg/m^3]	2.50e-5	4.14e-4	2.30e-5
ρ_{eVar}	0.999999999995	0.999999999289	0.999999999973
$\max \rho_{\text{err}}$	5.00e-5	1.63e-1	6.10e-2
$\bar{T}_{\text{err}}$ [K]	2.51e-5	8.60e-5	7.85e-6
T_{eVar}	0.999999999356	0.999999997667	0.999999999743
$\max T_{\text{err}}$	5.50e-5	1.56e-3	6.19e-3

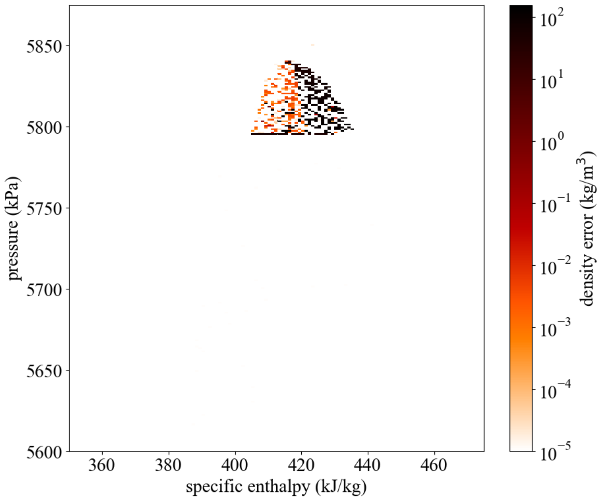

Fig. 2. Errors between RP/Fortran and RP/Modelica around the critical region.

when the number of iterations hits its limit of 40. This explains both the maximum absolute error of ρ between REFPROP/Fortran and REFPROP/Modelica, and also explains why the mean and median values are of order 10^{-5} in light of the large number of iterations used in both PHFL1 and TPRHO. Moreover, this suggests that it is difficult to evaluate the approximation methods in this neighborhood due to the fact that the data generated by REFPROP/Fortran is dominated by iteration limits, rather than convergent behavior.

Table 4 shows the CPU running time of three approaches. The saturation curve approximation improves the speed by about 25%, as expected from the profile obtained in Section 2.1, while the use of the full surface approximation improves the speed by about 100 times.

The consistency of the property calculations, or the fact that the same results should be obtained via different valid choices of thermodynamic coordinates, is another important consideration. We checked the consistency of the flash calculations by first using (p, h) to calculate $(\hat{T}, \hat{\rho})$, and then using these calculated values of temperature and density to re-calculate $(\hat{p}, \hat{h})$ for comparison with the initial values. Table 5 shows the mean, median and maximum absolute errors from this consistency check between REFPROP/Fortran and REFPROP/Modelica; these errors are also concentrated around the critical region.

3.1 Saturation line comparison

As the property values along the saturation line are often used in dynamic cycle models, we studied the accuracy of these approximation methods along both the liquid and vapor saturation lines. Figure 3 illustrates these saturation lines for the standard method and for the two approximation methods, as well as the errors between the approximated curves and the baseline curves. The upper plot of this figure demonstrates the reasonable accuracy with which both methods can describe the saturation lines, though some errors along the liquid saturation line are evident for the surface approximation method. Further examination of the errors in the lower plot of this figure provides confirmation of this fact; while the errors for both methods are higher in the critical region, the average errors along the saturation lines are higher for the surface approximation method than they are for the saturation line approximation method. This difference is caused by the fact that the saturation line crosses through many cells in the surface approximation method, so that the average value of the density in a given cell close to a saturation line may not accurately represent the density in a corner of that cell.

Table 4. Speed comparison over Ω

Approach	Average time (s)
Standard	7.6
Saturation Curve	5.8
2-D Surface	0.074

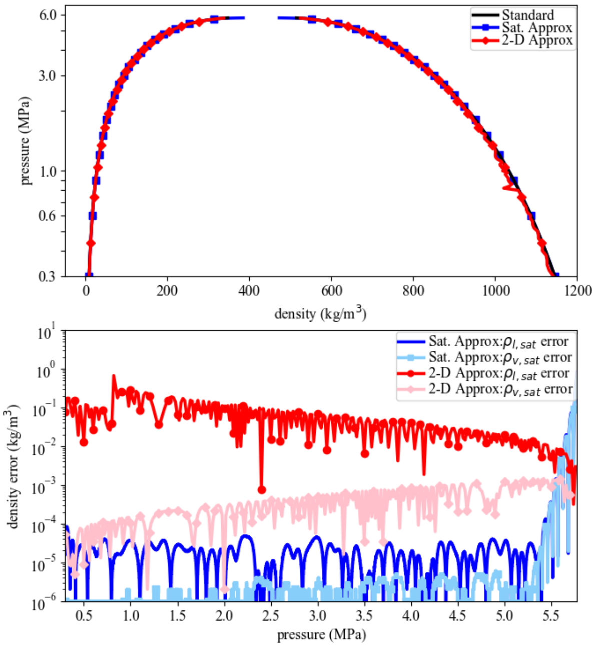

Fig. 3. Pressure saturation lines from standard and approximated methods.

3.2 Density surface comparison

While the global characterization of errors discussed in the beginning of this section is useful to obtain bounds on the accuracy of the approximation methods, it is also useful to study the errors between the standard method and each approximation method in a more local manner. Figures 4 and 5 illustrate checkerboard plots of the errors for the saturation curve approximation and the surface approximation methods, respectively. These figures demonstrate in general that the errors represent less than 1% discrepancy over the domain of interest except in the

Table 5. Consistency check of (p, h) over Ω

Error type	RP/Modelica	Sat. Approx.	2-D Approx.
$\bar{p}_{\rm err}$ [Pa]	1.66e-3	2.17e-3	1.09e-2
$\tilde{p}_{\rm err}$ [Pa]	0.0	0.0	3.69e-3
max $p_{\rm err}$	0.92	1.02	3.18
$\bar{h}_{\rm err}$ [J/kg]	1.71e-3	1.73e-3	1.76e-3
$\tilde{h}_{\rm err}$ [J/kg]	0.0	0.0	3.6e-5
max $h_{\rm err}$	25.33	25.33	19.92

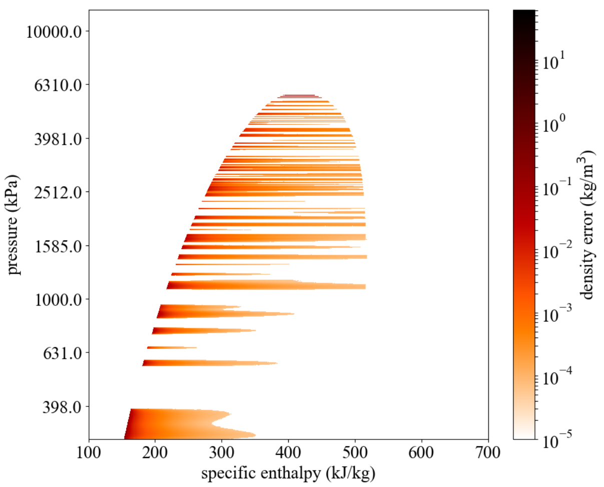

Fig. 4. Errors between REFPROP/Modelica and saturation approximation method.

vicinity of the critical point; however, further study of these figures indicates some important differences between these two methods.

Figure 4 provides evidence of the fact that while this method is accurate in the single-phase regions, bands of larger error are present that span portions of the two-phase region. These error bands propagate from small errors in the spline approximation of the saturated liquid density; discrepancies in the fourth or fifth decimal place are magnified in the computation of the static quality x and the corresponding computation of the mixture variables that result in the observed effects. These bands of error are strongly correlated to the number of control points used to create the splines, as larger numbers of control points result in narrower bands of higher errors. As a result, though the errors in the single phase regions are less than 10^{-5} kg/kJ, the errors for this method are somewhat higher in the two-phase region.

In comparison, the errors for the surface approximation method illustrated in Figure 5 are much different in character. While these errors are slightly larger along the saturated liquid line and near the critical point, they are more uniformly distributed across the entire domain than was the case for the saturation curve approximation. This results in lower errors for the density in the two-phase region, but higher errors for the single phase regions. Consequentially, it is difficult to heavily favor one approximation method over the other; while the saturation approximation method is quite accurate in the single phase regions and require little memory for implementation, the surface approximation method is much faster and more accurate in the two-phase region. The choice between these methods must therefore depend on preferred features of the property calculation method, or its accuracy for other variables, such as the density derivatives, which have not been explored in this work thus far.

4. CONCLUSION

This work suggests that thermophysical refrigerant properties can be successfully approximated using a variety of different methods, depending on the particular tradeoffs between accuracy and speed required by an application. Two alternate methods, the saturation curve approximation and surface approximation approaches, were implemented in Modelica and demonstrated to have relatively high accuracy while substantially reducing the computational time required by standard iterative methods. Future work in this area includes the implementation and comparison of these methods in full vapor compression cycles.

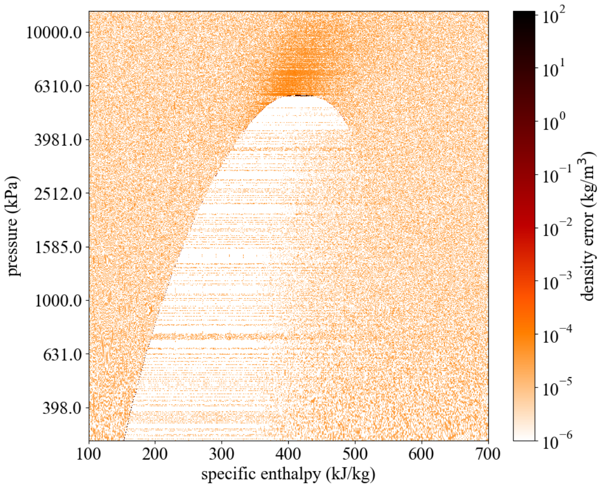

Fig. 5. Errors between REFPROP/Modelica and surface approximation method.

REFERENCES

Åberg, M., Windahl, J., Runvik, H., Magnusson, F., May 2017. Optimization-friendly thermodynamic properties of water and steam. In: Proceedings of the 12th International Modelica Conference. Prague, Czech Republic, pp. 449–458.

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

Bejan, A., 2006. Advanced Engineering Thermodynamics, 3rd Edition. Wiley.

IAPWS, 2015. Guideline on the fast calculation of steam and water properties with the spline-based table look-up method (STBL). Tech. rep., International Association for the Properties of Water and Steam.

Lemmon, E. W., Huber, M. L., McLinden, M. O., 2013. NIST Standard Reference Database 23: Reference Fluid Thermodynamic and Transport Properties-REFPROP, Version 9.1, National Institute of Standards and Technology. `http://www.nist.gov/srd/refprop`.

Modelica Association, 2017. Modelica specification, Version 3.4. `http://www.modelica.org`.

Piegl, L., Tiller, W., 1995. The NURBS Book. Springer-Verlag, London, UK.

Pini, M., Spinelli, A., Persico, G., Rebay, S., 2015. Consistent look-up table interpolation method for real-gas flow simulations. Computers & Fluids 107, 178–188.

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

Span, R., Wagner, W., 2003. Equations of state for technical applications. III. Results for polar fluids. International Journal of Thermophysics 24 (1), 111–162.