

# Superancillary Equations for Nonpolar Pure Fluids Modeled with the PC-SAFT Equation of State

Ian H. Bell<sup>\*,†</sup> and Ulrich K. Deiters<sup>‡</sup>

<sup>†</sup>*Applied Chemicals and Materials Division, National Institute of Standards and Technology, Boulder, CO 80305, USA*

<sup>‡</sup>*Institute of Physical Chemistry, University of Cologne, 50939 Köln, Germany*

E-mail: ian.bell@nist.gov

## Abstract

Superancillary equations are presented for the PC-SAFT equation of state of Gross and Sadowski for nonpolar pure fluids. The equations cover the range of number of segments from 1 to 64. These equations, formed as nested sets of Chebyshev expansions, represent the densities for vapor-liquid equilibria better than can be achieved by iterative calculations in double precision arithmetic. Furthermore, the expressions are more than 16 times faster to evaluate than the iterative calculations from the fastest thermodynamic property library. By their construction, the superancillary equations are guaranteed to be reliable over the entire temperature range. For user-friendliness, the functions have been packaged into a Python package available on PYPI.

## 1 Introduction

The PC-SAFT (Perturbed-chain statistical associating fluid theory) equation of state (EOS)<sup>1</sup> has found a place in the canon of empirical thermodynamic models due to its good representation of some – although not all – thermodynamic properties, even if this EOS is not without defects.<sup>2–4</sup> A challenge with this model (as is the case for nearly all thermodynamic models) is to computationally efficiently and reli-

ably carry out vapor-liquid equilibria (VLE) calculations. The most challenging part of the calculation is to obtain initial guesses for the co-existing densities that are sufficiently accurate, and determining what constitutes sufficiently accurate guesses is itself a non-trivial task. A further challenge with the PC-SAFT EOS is that it does not have a single van der Waals loop at all temperatures, which makes density rootfinding challenging.

In the publications associated with the recent multiparameter equations of state used in NIST REFPROP,<sup>5</sup> CoolProp,<sup>6</sup> simple mathematical functions are provided that can be evaluated to obtain reasonably accurate orthobaric densities used as guess values for the VLE calculation. A few examples would include those for water,<sup>7</sup> carbon dioxide,<sup>8</sup> or propane.<sup>9</sup> These curves usually, but not always, allow VLE calculations to converge properly.

Constructing a mathematical formulation of the ancillaries that is accurate enough to replace the VLE calculations entirely is the focus of the development of superancillary equations.<sup>10,11</sup> These superancillary equations use the well-accepted approach of the approximation of continuous smooth functions by Chebyshev polynomials.<sup>12</sup> Evaluation of the superancillary functions is significantly faster than full VLE calculations. By their non-iterative nature, evaluation of these functions can be guar-

anted to not fail; failure of iterative routines is relatively common in libraries that implement the PC-SAFT equation of state, even for pure fluids.

The structure of this work is as follows: the model is described, some advanced numerical analysis tools are described, critical points are calculated, superancillaries are built, and the speed and accuracy of the formulation is assessed.

## 2 Model

In this work we use the complete equation of state of Gross and Sadowski<sup>1</sup> for multi-component mixtures, although only pure fluids are considered in this work. There are a few errata in the original publication, and while an erratum has not been filed, a response to a comment enumerates some of the errors.<sup>13</sup> The association contribution is not included and therefore a fluid is characterized by its values of  $\varepsilon/k_B$ ,  $\sigma$ , and  $m$ . In the case of a pure fluid, PC-SAFT can be expressed in a simpler form with the non-dimensional quantities  $\tilde{T} = T/(\varepsilon/k_B)$  and  $\tilde{\rho} = \rho_N \sigma^3$  as the independent variables. The PC-SAFT model for non-polar pure fluids is summarized in Polishuk<sup>3</sup>. The units of temperature and number density  $\rho_N$  must match those of  $\varepsilon/k_B$  and  $\sigma^3$  for dimensional consistency. In the remainder of this paper, we use the tilde-scaled variables, and the parameters  $\varepsilon/k_B$  and  $\sigma$  are taken as arbitrary parameters that can be given any desired value; the superancillaries are developed in tilde-reduced form, and are therefore independent of the variables  $\varepsilon/k_B$  and  $\sigma^3$ . The implementation in `teqp` (a very new<sup>14</sup> EOS evaluation library that uses automatic differentiation rather than analytic derivatives) uses molar-like units by default, and internal conversions are needed, for which the value of Avogadro’s constant of  $6.02214076 \times 10^{23}$  was used.<sup>15</sup> The value of  $\varepsilon/k_B$  can be set to any desired value, and 100 K was used with no loss in generality.

## 3 Numerics

This work makes heavy use of advanced numerical analysis techniques, namely function approximation with Chebyshev expansions and arbitrary precision calculations, so those concepts are explained at a high-level here.

### 3.1 Chebyshev expansions

This section will cover the most salient parts of function approximation theory relevant to the problem at hand. Further exposition is available in Boyd<sup>12</sup>. The theory of Chebyshev approximation relevant to this problem is described in Bell and Alpert<sup>16</sup>. Further discussion related to superancillary construction is available in Bell and Alpert<sup>10</sup>.

Chebyshev expansions, or the more general approach of approximating smooth continuous functions with orthogonal polynomials in closed intervals, is by now a well accepted approach with practical implementations available for `ChebTools`<sup>17</sup> for C++/Python, `chebfun`<sup>18</sup> in MATLAB, and `ApproxFun`<sup>19</sup> for Julia. Orthogonal polynomials allow for functions to be approximated in a straightforward manner to numerical precision (further discussed in the next section). Approximation  $F(x)$  to the function  $f(x)$  in the domain  $[a, b]$  is obtained with the summation form

$$F(x) = \sum_i^n c_i T_i(\hat{x}) \quad (1)$$

where  $\hat{x} \in [-1, 1]$  is the value of  $x$  linearly mapped from  $[a, b]$  and  $T_i$  are the Chebyshev basis functions of the first kind, which are defined recursively from  $T_0 = 1$ ,  $T_1 = x$ , and  $T_n(x) = 2xT_{n-1}(x) - T_{n-2}(x)$ .

The Chebyshev-Lobatto nodes of an expansion of degree  $n$  are at the values (in  $[-1, 1]$ )

$$x_j = \cos\left(\frac{j\pi}{n}\right) \quad (2)$$

for  $j$  between 0 and  $n$ , where the first and last nodes are the values 1 and  $-1$ , respectively.

The transformation from nodal functional values to coefficients uses the  $(n + 1) \times (n + 1)$

matrix  $\mathbf{V}$  multiplied by the vector of functional values at the Chebyshev-Lobatto nodes

$$\mathbf{c} = \mathbf{V}\mathbf{f}_n \quad (3)$$

with elements  $v_{jk}$

$$v_{jk} = \frac{2}{np_n(j)p_n(k)} \cos\left(\frac{\pi jk}{n}\right) \quad (4)$$

where  $p_n(i)$  is 2 if  $i = 0$  or  $i = n$  and  $p_n(i) = 1$  otherwise. The matrix  $\mathbf{V}$  does not depend on anything other than the degree of the expansion and can therefore be constructed once and cached for further re-use.

The approximation of smooth continuous functions in a closed domain with Chebyshev expansions follows a straightforward path. The approximation will in general improve as the degree of the expansion is increased, until the last coefficients are approximately the epsilon of the numerical precision relative to the first coefficients. In double precision, that would mean that they are  $10^{16}$  times smaller than those of the leading coefficients.

An alternative approach to increasing the degree of the expansion is to fix the degree of the expansion but decrease the interval width, recursively subdividing the overall interval into smaller and smaller segments. This approach has the advantage that bisection to find the right interval and evaluation of a lower-degree expansion is in general much faster than the evaluation of a single higher-degree expansion. This timing comparison becomes even more evident when using the expansions for rootfinding, where eigenvalue solving is needed, for which the number of operations scales like  $\mathcal{O}(n^3)$ , with  $n$  the degree of the expansion in general (with the Jacobi eigenvalue algorithm).

### 3.2 Numerical precision

Computing the addition  $0.1 + 0.1$  does not (in general) give exactly  $0.2$  in a computer. This is because  $0.1$  cannot be exactly expressed in double precision arithmetic, and there is a small error introduced by this approximation. When the magnitudes of the numbers added together become greatly different in magnitude,

catastrophic cancellation can occur. In double precision, the sum  $1.0 + 1.0\text{e-}17$  is equal to  $1.0$  because the so-called epsilon of double precision arithmetic is approximately equal to  $2.2\text{e-}16$ . The epsilon is the smallest number that can be added to  $1.0$  and obtain a number greater than  $1.0$ .

In the VLE calculations below, there are cases where the densities of the co-existing phases differ by more than a ratio of  $10^{16}$  in magnitude. As a consequence, the double precision numerical type (default for basically all computational environments due to its low-level support in processors) is not sufficient.

In software-emulated arbitrary precision libraries, rather than using the processor's extended capabilities, software libraries implement all the necessary mathematical functions while allowing the user to decide how many digits of working precision they need. The more digits of working precision, the slower the code will be, and there is already a very significant computational penalty to the calculations as soon as software emulation is enabled. Slowdowns of factors of hundreds or thousands (or more!) are not uncommon when switching to software-emulated arbitrary precision arithmetic. Extended precision arithmetic should only be used when double precision arithmetic has been fully exhausted.

## 4 Critical Points

The critical points have a number of uses in this work. For one, the limiting values of the critical curves for infinite values of  $m$  are interesting for polymer physics, and the critical point represents the end of vapor-liquid-equilibria for a given value of  $m$ . The particular nature of the PC-SAFT EOS also allows the VLE to be traced by extrapolation from the critical point, so the critical curve is used as a starting point of sorts for the VLE calculation. Finally, the normalized VLE curves use the critical point as a portion of the scaling.

The conditions for a critical point for a pure fluid are  $(\partial p/\partial \rho)_T = (\partial^2 p/\partial \rho^2)_T = 0$ . The `solve_pure_critical` method in `teqp`<sup>14</sup>

is used to solve these two conditions given a starting value for temperature and density with Newton iterations based upon an analytical Jacobian matrix. If the initial guess is good enough, the solver reliably converges to the critical point. Numerical danger-zones are avoided in this calculation, so it is acceptable to carry out the calculation in double precision arithmetic.

The set of critical points is obtained by starting at  $m = 1$  and increasing  $m$  in very small increments. The initial values at  $m = 1$  are  $\tilde{T} = 1.2757\dots$  and  $\tilde{\rho} = 0.28239\dots$ , obtained manually (the values are close to those of the Lennard-Jones fluid<sup>20</sup>). At every step, the critical point solver is initiated with the temperature and density from the previous value of  $m$ . As a result, we are well within the radius of convergence of the solver, and the critical point is reliably obtained. The values of the temperature and density are shown in Fig. 1. Although the PC-SAFT equation demonstrates spurious critical points,<sup>2,4</sup> that concern is avoided in this case because the approach taken for tracing the critical points safely follows the main branch of critical points. At  $m = 10^8$  (not shown) the critical temperature is  $\tilde{T} = 5.3376$  and the density approaches a power-law relationship with  $m$  for large values of  $m$  (linear in double logarithmic coordinates).

The critical points are represented as a collection of 1-dimensional Chebyshev expansions for each of  $\tilde{T}$  and  $\tilde{\rho}$  as a function of  $m$  in the closed domain  $[1, 100]$ . The `dyadic_splitting` method of `ChebTools` is used to do the interval subdivision. At each functional evaluation in expansion construction, linearly interpolated values of  $\tilde{T}$  and  $\tilde{\rho}$  are obtained from the stepping method, which ensures a reliable expansion construction. The obtained set of expansions is available in the supporting information. Conceptually, the Chebyshev collection curves in this work are like those of the polynomial curves of Moine et al.<sup>21</sup>, though the curves in this work are many orders of magnitude more accurate.

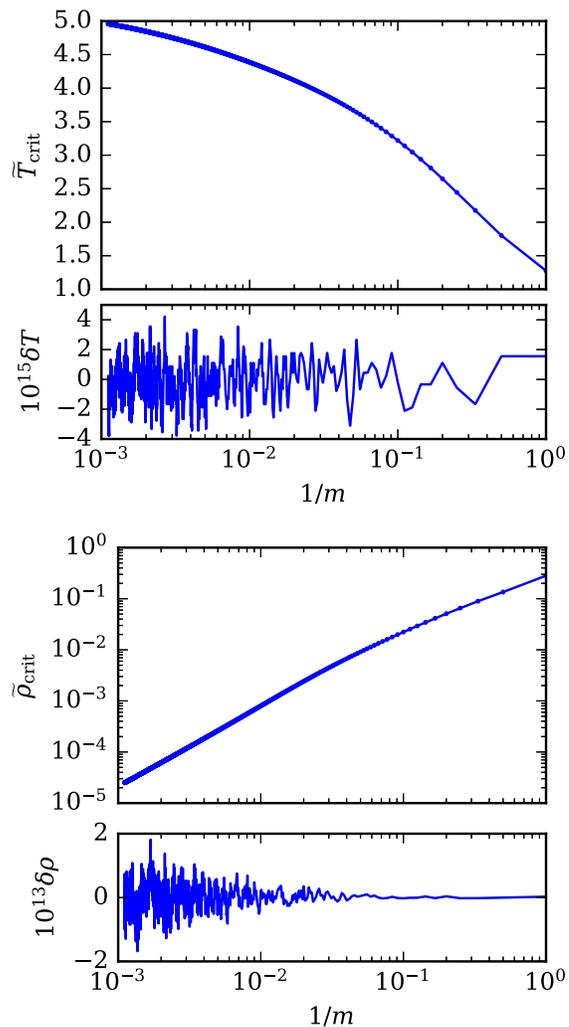


Figure 1: Critical point temperature and density and deviations  $\delta Y = (\tilde{Y}_{\text{crit,tab}}/\tilde{Y}_{\text{crit,fit}} - 1)$  for a range of values of  $1/m$

## 5 VLE Superancillary Equations

The definition of VLE for a pure fluid is having the same pressure and Gibbs energy in both phases, and the densities of the co-existing phases are the independent variables of the iteration. As above, there is a method in `teqp` for carrying out this calculation given guess values for the densities: `pure_VLE_T`. In the case of VLE, the calculations are carried out in extended precision, with 200 digits of working precision (to be contrasted with the 16 digits of working precision in double precision arithmetic). This extended precision is needed to be able to carry out vapor-liquid equilibria at

low reduced temperatures at which the density ratio between phases is well above  $10^{40}$  (quantified later on). The use of software-emulated variable precision introduces an exceptionally severe computational speed penalty, but that price is only paid at the time of expansion construction, which happens once.

The problem with the VLE calculations is that they often represent a computational bottleneck in process simulation codes and that they can fail. Failures are usually caused by insufficient working precision in the iterations themselves as well as insufficiently accurate initial guesses for the orthobaric densities. The superancillary equation solves both problems. It is a mathematical formulation (one could think of it as a supercharged polynomial function because Chebyshev functions themselves can be expressed in terms of the monomial bases  $x^0$ ,  $x^1$ , etc.). First a method is obtained for getting very reliable guess values based on extrapolation from the critical point, followed by the use of these very accurate guess values to obtain orthobaric densities in VLE calculations in extended precision.

## 5.1 Tabulation of Initial Guesses

In order to obtain initial guesses for the more accurate calculations to follow, for a given value of  $m$ , the VLE curve is traced as far down in temperature as possible. The tracing is initiated at the critical point obtained by the method described above, at an accuracy equal to numerical precision in double precision arithmetic. A small step away from the critical point is taken towards lower temperature equal to  $\tilde{T}_{\text{crit}}/1000$  with the critical extrapolation method `extrapolate_from_critical` in `teqp`. This method applies the extrapolation formula of

$$\rho^\alpha = \rho_{\text{crit}} \pm \sqrt{6T_{\text{crit}} \frac{\left(\frac{\partial^2 p}{\partial \rho \partial T}\right)}{\left(\frac{\partial^3 p}{\partial \rho^3}\right)_T} \left(\frac{T_{\text{crit}} - T}{T_{\text{crit}}}\right)^{1/2}} \quad (5)$$

to yield the orthobaric (liquid and vapor) densities for the phases  $\alpha$  for a given temperature  $T$ . The derivation of this approach is in the

supporting information. This scheme yields estimated values for the co-existing phase densities for near-critical temperatures which are subsequently polished in extended precision.

Once the first good step has been obtained at a subcritical temperature and the densities stored in their extended precision values, the temperature is decreased by a small decrement and the process repeats; the starting densities for the iteration are those of the previous temperature. By the use of such small steps in temperature, the densities can be assured to be “good enough” starting points for the iteration such that convergence to the appropriate solution in extended precision arithmetic can be practically guaranteed. The VLE stepping process is terminated when an invalid solution for the densities is obtained or when the ratio of liquid to vapor density exceeds  $10^{40}$ , or when the liquid density decreases upon further reducing in temperature. Failure usually occurs because of the limitations of finite precision arithmetic rather than non-convergence of the VLE rootfinding. The final result is a pair of tabulated values of the orthobaric densities,  $\tilde{\rho}'(\tilde{T})$  and  $\tilde{\rho}''(\tilde{T})$ , for discrete values of  $m$ . Examples of such curves are shown in Fig. 2. To re-iterate, these tabulated VLE data points are used only as starting points for the VLE calculations in extended precision.

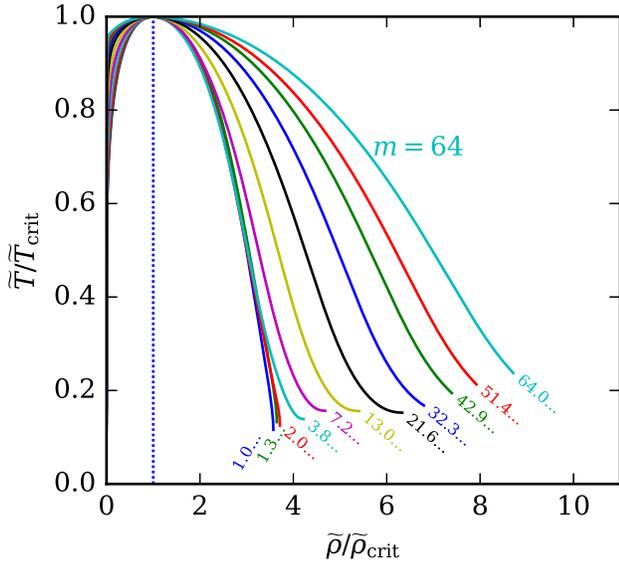


Figure 2: VLE curves reduced by the critical point values for values of  $m$  at the interval edges of the final fit. The interval edges are labeled at the bottom at their value of  $m$  with color matching the respective curve.

## 5.2 Minimum Temperature

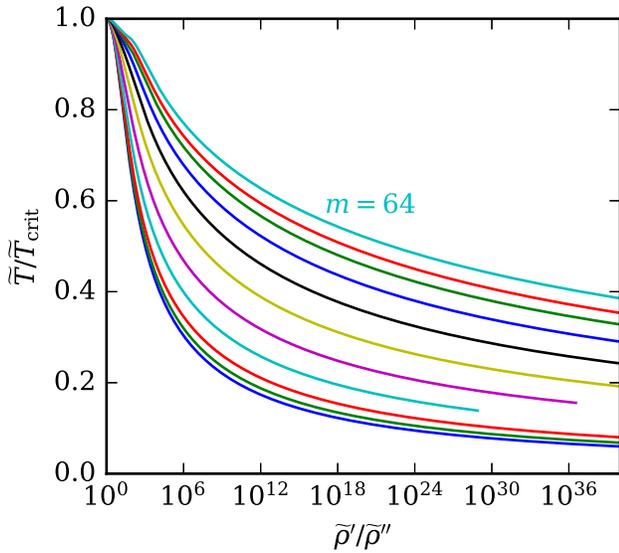


Figure 3: Reduced temperature as a function of density ratio (liquid/vapor) for values of  $m$  at the interval edges of the final fit. Color scheme matches Fig. 2.

In order to make the functional form for fitting better behaved, it was necessary to determine a simple functional form to scale the VLE data.

This was done by interpolating to find the temperature which yielded the density ratio of  $10^{20}$ . The values are plotted in Fig. 3. This density ratio is extremely far beyond what is possible to achieve with iterative calculations in double precision arithmetic so this bound is very conservative and will cover all possible VLE calculations in double precision arithmetic. The PC-SAFT EOS is simply no longer meaningful below this temperature in double precision arithmetic. The shape of the VLE curves for longer chains is no longer smooth in the critical region, there is a kink in the orthobaric density curves for  $T/T_{\text{crit}} > 0.9$  for  $m > 32$ . This is a non-physical artifact of the PC-SAFT EOS, and is practically the reason why the superancillary stopped at  $m = 64$  because it proved necessary to avoid this problematic region.

The interpolated values are shown in Fig. 4 as a function of  $m$ . The curve to fit the data was fit by a best-fit curve in the form

$$\ln(\tilde{T}_{\text{red,min}}) = c_0 \ln(m) + c_1 \quad (6)$$

with the polyfit function of numpy, for which the functional form reads:

$$\tilde{T}_{\text{red,min}} = \exp(c_1)m^{c_0} \quad (7)$$

with  $[c_0, c_1] = [0.37627892, -2.20078778]$ . To reiterate, the function in Eq. (7) is only used as a reasonable lower bound in temperature, and it is not accurate to numerical precision.

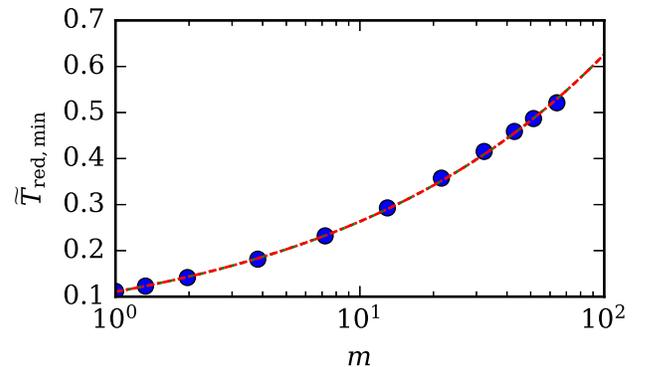


Figure 4: Minimum temperatures obtained for each value of  $m$  at an interval edge. The minimum temperature corresponds to a density ratio (liquid/vapor) of approximately  $10^{20}$

Combining the minimum temperature curve from Eq. (7) and the critical temperature obtained from the Chebyshev expansion, the normalized VLE curves are plotted in Fig. 5 for some values of  $m$ . Normalization by defining the normalized variable

$$\Theta = (\tilde{T} - \tilde{T}_{\min}) / (\tilde{T}_{\text{crit}} - \tilde{T}_{\min}) \quad (8)$$

has the effect of creating a regularized set of smooth curves as a function of  $m$ .

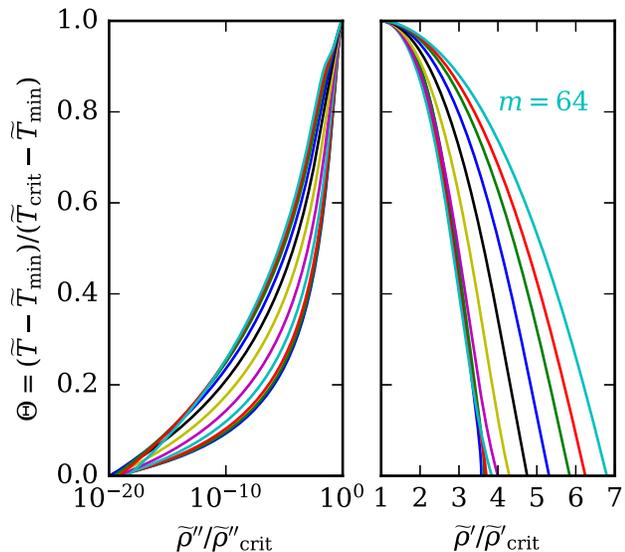


Figure 5: Normalized VLE curves, further reduced by the critical point values for values of  $m$  at the interval edges of the final fit. Minimum temperature for a given value of  $m$  is evaluated from Eq. (7). The density of the vapor curve is on a logarithmic scale. The color scheme matches Fig. 2

### 5.3 Superancillary

In general, construction of a Chebyshev expansion to approximate a function  $f$  follows the approach described above in Section 3.1. The function is evaluated at the  $n + 1$  Chebyshev-Lobatto nodes for a degree of  $n$ , and the coefficients of the Chebyshev expansion are obtained. This initial approximation function is not accurate enough to reproduce the function to numerical precision over the entire interval, so the degree of the expansion is fixed and interval bisection is used to recursively refine the Chebyshev expansion in each half.

The superancillary is the set of Chebyshev expansions required to span the range of temperature. For a given value of  $m$ , interval subdivision is used to construct a superancillary equation for the orthobaric densities as a function of  $\Theta$ . The superancillary equations are therefore of the form  $\tilde{\rho}'(\Theta)$  and  $\tilde{\rho}''(\Theta)$ . At each function evaluation, the guess values for the orthobaric densities are obtained and a complete VLE calculation is carried out in extended precision.

This process follows the same approach as in the `dyadic_splitting` function of `ChebTools`, except that interval subdivision happens if *either* the liquid or vapor density expansion has not reached its convergence tolerance. The specification of the approach is defined in Table 1. Thus the  $\Theta$  intervals are different for each value of  $m$  in general. As described above, the goal is to represent the orthobaric densities better than can be achieved in double precision VLE iterative calculations.

Table 1: Specification for interval subdivision in  $\Theta$  for a given value of  $m$

Parameter	Spec
Degree	16
Norm	3-element norm (ratio of norms of last three to first three coefficients)
Tolerance	Norm $< 10^{-12}$
Refine passes	13

The dyadic (interval bisection) splitting approach is applied, where the degree of the expansion in  $\Theta$  is fixed and the interval is split into sub-intervals as needed at its midpoint. This process is described in some detail in Bell and Alpert<sup>10</sup>. The initial densities for the iterative calculation are obtained by one of two options:

- If  $\tilde{T} > 0.999\tilde{T}_{\text{crit}}$ , then critical extrapolation is used (following Eq. (5))
- Otherwise, the VLE values obtained from the tabulated values of  $\tilde{T}$  (see Section 5) are linearly interpolated within to get the orthobaric density guess values

The `mix_VLE_Tx` method of `teqp` is used to do the polishing of the solution in extended precision with 100 digits of working precision.

## 6 Nested Superancillaries

To complete the full set of nested expansions, dyadic splitting is also applied to  $w = 1/m$ . The overall interval of  $[1/m_{\max}, 1/m_{\min}]$  is used initially, and a Chebyshev expansion of degree  $N_m$  is used to span the entire interval. This implies that superancillary curves are built at the  $N_m + 1$  Chebyshev-Lobatto nodes in  $w$  with values of  $m = 1/w$  in the set  $[1.0, 1.00954754, \dots, 39.86883981, 64.0]$  if  $N_m = 16$  and the range of  $m$  is 1 to 64. Note how the values of  $m$  are not linearly spaced. The specification used for interval splitting in  $m$  is defined in Table 2.

Table 2: Splitting specification for interval subdivision in  $w$

Parameter	Spec
Degree	16
Norm	3-element (ratio of norms of last three to first three coefficients)
Tolerance	Norm $< 10^{-12}$
Refine passes	10

Assessing convergence for the expansions in  $w$  is more challenging than for  $\Theta$ . Expansions are built for values of  $\Theta \in \{0.1, 0.5, 0.9\}$  for each phase. The convergence check is done for the expansions each phase and each value of  $\Theta$ . If any do not meet the convergence criterion, the interval in  $w$  is subdivided at its midpoint. The process of recursive subdivision repeats until either all the intervals in  $w$  are converged or the maximum number of refinement passes is reached. At completion, there are a number of intervals in  $w$ , and in each one, there is a superancillary equation at the Chebyshev-Lobatto nodes in  $w$ . This is illustrated in Fig. 6. The entire process from start to finish takes a few hours, a consequence of the overhead introduced by the use of extended precision arithmetic.

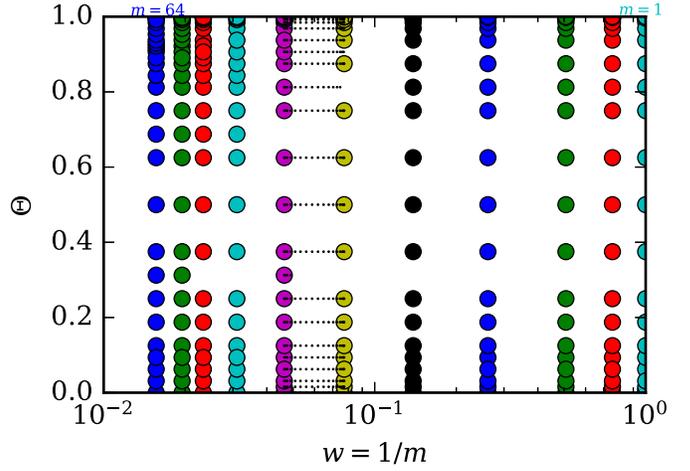


Figure 6: Locations of the edges of the intervals in  $\Theta$  and  $w$ . The small dots are the locations of the edges in the  $\Theta$  direction for each node in  $w$  for one selected interval.

### 6.1 Evaluation

Once the set of expansions have been built, evaluation of the superancillary equation for a set of  $m$  and  $\tilde{T}$  proceeds as follows:

- The Chebyshev expansions for the critical curve and Eq. (7) are used in concert to obtain  $\Theta$
- Interval bisection in  $w$  is used to obtain the interval containing the value of  $w$ .
- Values of  $\tilde{\rho}^\alpha(\Theta)$  are calculated for each phase for each value of  $w$  at a Chebyshev-Lobatto node in the interval. This yields functional values at the Chebyshev-Lobatto nodes in  $w$  for each phase.
- With the given functional values at the nodes, discrete cosine transform (DCT) is used to build the Chebyshev expansions in the form  $\tilde{\rho}^\alpha(w)$ <sup>12</sup>
- The expansions are evaluated as a function of  $w$ , and the final values for the densities are obtained

This process, though it may seem convoluted, can be concisely applied with the use of the `ChebTools` library. A complete example of evaluating the nested superancillaries in the C++ programming language is presented

in Fig. 7. The coefficients and other data structures are in the referenced headers which are auto-populated based on the method described above. This code is precisely the implementation used in the Python package.

```

#pragma once

#include "PCSAFTSuperancillary_data.hpp"
#include "PCSAFTSuperancillary_crit.hpp"

using namespace ChebTools;
using namespace PCSAFTSuperAncillary;

const auto& get_interval(double w) {
    /// Return the index of the expansion that is desired
    auto get_index = [&](double w) {
        auto midpoint_Knuth = [](int x, int y) {
            return (x & y) + ((x ^ y) >> 1);
        };
        int iL = 0, iR = static_cast<int>(domain.Wedges.size() - 2, iM);
        while (iR - iL > 1) {
            iM = midpoint_Knuth(iL, iR);
            if (w >= domain.intervals[iM].wmin()) {
                iL = iM;
            }
            else {
                iR = iM;
            }
        }
        return (w < domain.intervals[iL].wmax()) ? iL : iR;
    };
    int i = get_index(w);
    return domain.intervals[i];
}

auto get_funcvals(double Theta, const WInterval& interval) {
    Eigen::Index Nm = interval.expsL.size()-1;
    Eigen::ArrayXd rhoLfvals(Nm+1), rhoVfvals(Nm+1);
    for (auto i = 0; i <= Nm; ++i) {
        rhoLfvals[i] = interval.expsL[i](Theta);
        rhoVfvals[i] = interval.expsV[i](Theta);
    }
    return std::make_tuple(rhoLfvals, rhoVfvals);
}

auto get_expansion(const Eigen::ArrayXd& vals, double wmin, double wmax) {
    return ChebyshevExpansion(
        std::move((V16 * vals.matrix()).eval().matrix()), wmin, wmax);
}

auto get_Ttilde_crit_min(double m){
    auto wmin = cc_Ttilde.get_exps()[0].xmin(), wmax = cc_Ttilde.get_exps().back().xmax();
    auto mmin = 1/wmax, mmax = 1/wmin;
    if (m < mmin){
        throw std::invalid_argument("Provided value of m of " + std::to_string(m)
            + " is less than min of " + std::to_string(mmin));
    }
    if (m > mmax){
        throw std::invalid_argument("Provided value of m of " + std::to_string(m)
            + " is greater than max of " + std::to_string(mmax));
    }
    auto Ttilde_crit = cc_Ttilde(1/m);
    auto Ttilde_min = exp(-2.20078778)*pow(m, 0.37627892)*Ttilde_crit;
    return std::make_tuple(Ttilde_crit, Ttilde_min);
}

auto PCSAFTSuperanc_rhoLV(double Ttilde, double m){
    auto [Ttilde_crit, Ttilde_min] = get_Ttilde_crit_min(m);
    auto Theta = (Ttilde - Ttilde_min) / (Ttilde_crit - Ttilde_min);
    if (Ttilde < Ttilde_min){
        throw std::invalid_argument("Provided value of Ttilde of " + std::to_string(Ttilde)
            + " is less than min of " + std::to_string(Ttilde_min));
    }
    if (Ttilde > Ttilde_crit){
        throw std::invalid_argument("Provided value of Ttilde of " + std::to_string(Ttilde)
            + " is greater than max of " + std::to_string(Ttilde_crit));
    }
}

/// Bisection to find the right interval in w=1/m
const auto& interval = get_interval(1/m);
double wmin = interval.wmin();
double wmax = interval.wmax();
/// Then evaluate the values at the nodes of w for each phase in this interval
auto [rhoLfvals, rhoVfvals] = get_funcvals(Theta, interval);
/// DCT to get the expansions in 1/m for each phase
auto expl = get_expansion(rhoLfvals, wmin, wmax);
auto expV = get_expansion(rhoVfvals, wmin, wmax);
/// Evaluate the 1D expansions
return std::make_tuple(expl.y(1/m), expV.y(1/m));
}

```

Figure 7: C++ implementation of the nested superancillary approach

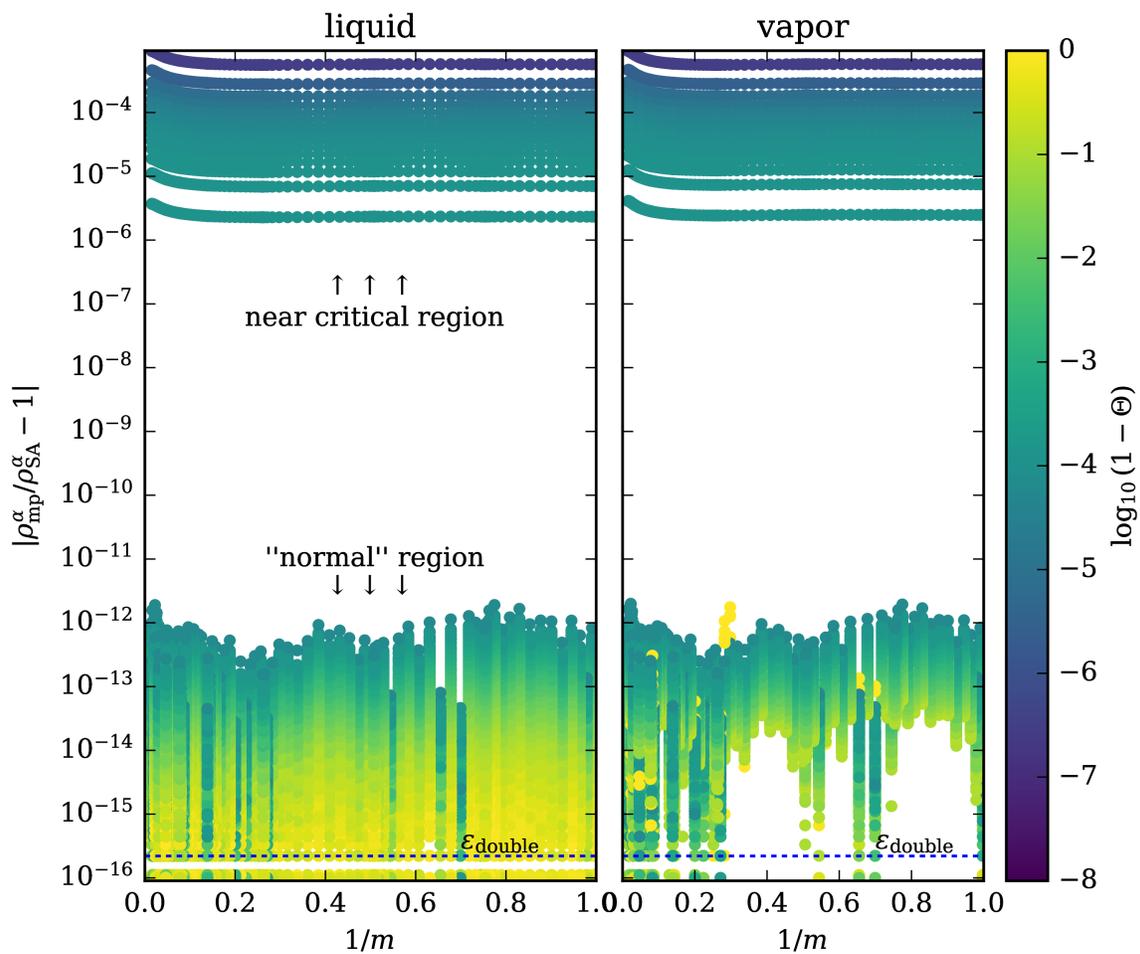


Figure 8: Worst-case absolute relative error in density (on a logarithmic scale) at the fitted  $w$  nodes at the double-sampled nodes for  $\Theta$ . The subscript of mp is for the multi-precision VLE calculations, the SA subscript corresponds to values from the superancillary function.

## 7 Results

The speed and accuracy of the superancillary equations are assessed in this section. We seek to answer the question: *Can the approximation functions completely replace the full phase equilibrium calculation?*

### 7.1 Accuracy

The use of C++ and standard mathematics libraries ensures that the same numerical results are obtained on all machines that follow IEEE754 arithmetic rules. The accuracy results in this section should be reproduced on all platforms.

For an expansion of degree  $N$ , if the degree is doubled to  $2N$ , the odd Chebyshev-Lobatto nodes of the degree  $N$  expansion are represented to numerical precision, while the even nodes are the locations of the worst-case error. Therefore, the odd nodes of the  $2N$  degree expansion are a straightforward and reliable test of the accuracy of the expansion.

In order to test the accuracy of the expansions in the  $\Theta$  direction, the error in density from the expansion sets at the values of  $m$  for which superancillary functions were developed are presented in Fig. 8. The absolute relative error is on a logarithmic scale, highlighting that with the exception of the points in the near critical region, all points are approximated to within a tolerance of approximately part in  $10^{12}$  which is only a few orders of magnitude worse than the epsilon of double precision arithmetic of  $\approx 2.2 \times 10^{-16}$ . As a demonstration of the critical region, Fig. 9 shows the error for the intervals nearest the critical point as a function of normalized temperature. Both figures demonstrate that the errors are very small, and worst nearest the critical point. The errors were calculated at each odd node for each interval in  $\Theta$  for doubled degree of expansion in each interval (the worst-case error). For values of  $\Theta$  below  $1 - 10^{-4}$  the error is less than a part in  $10^{12}$ . It is only in the very near critical region that serious problems can be found, this is a consequence of trying to resolve the singularity at  $\Theta = 1$  (the derivative  $d\rho^\alpha/d\Theta$  goes to infin-

ity at the critical point; see for instance Fig. 2). Further interval subdivision in the critical region would allow these points to be better represented, but the benefits are diminishing. In any case, most iterative calculations will fail for such near-critical inputs, and these curves are already in that sense infinitely better than the existing approaches.

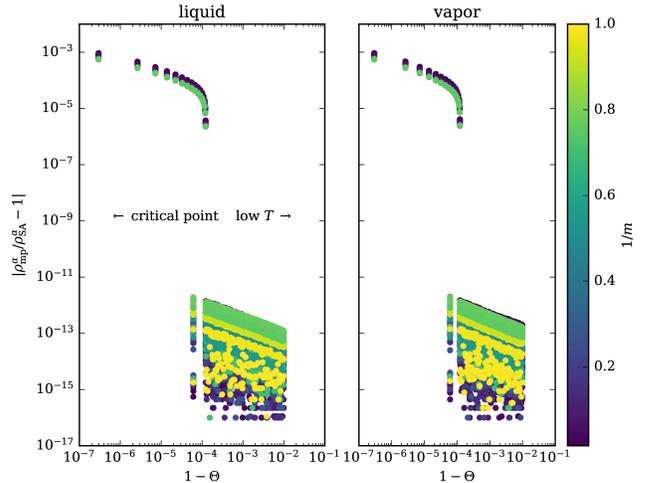


Figure 9: Worst-case absolute relative error in density at the fitted  $w$  nodes at the double-sampled nodes for  $\Theta$  in the near-critical region for  $\Theta > 0.99$ . The subscript of mp is for the multi-precision VLE calculations, the SA subscript corresponds to values from the superancillary function.

The worst-case mid-point nodes in  $w$  test the interpolation in the  $w$  (or  $1/m$ ) direction. The notion of doubled nodes in temperature no longer applies in between the expansions, but the results in Fig. 10 show that the additional interpolation in  $w$  does not dramatically alter the error, and the densities are still represented to close to numerical precision. The worst of the worst-case error is still less than a part in  $10^{12}$ . The critical region could again be expected to be somewhat worse, but not much worse than the results in Fig. 9

The deviation in density for iterative calculations in double precision arithmetic depends on the selected convergence tolerance in the software implementation. For instance in `Clapeyron.jl`<sup>22</sup> (an equation of state library analogous to `teqp`<sup>14</sup> written in Julia) the default convergence tolerance is a part in  $10^{12}$  in

pressure, but that convergence tolerance is not met for low temperature state points. Making a conclusive statement about the accuracy of the best deviations in density possible in iterative VLE calculations in double precision arithmetic is therefore difficult, if not impossible. Suffice to say that the superancillary equation approach is on par with, if not better than (particularly at low temperature and commensurately low pressure) the double precision iterations.

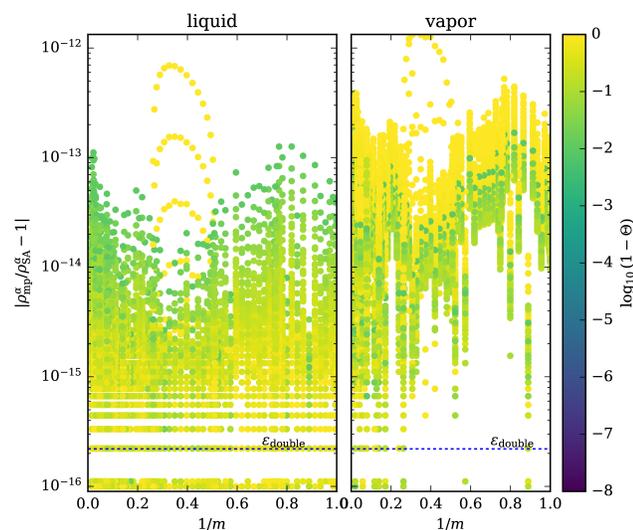


Figure 10: Worst-case absolute relative error in density (on a logarithmic scale) at double-sampled  $w$  nodes for  $\Theta$  linearly spaced in  $(0, 1)$  and therefore missing the near-critical region. The subscript of mp is for the multi-precision VLE calculations, the SA subscript corresponds to values from the superancillary function.

## 7.2 Python Implementation

The C++ code was wrapped into a Python module with `pybind11`<sup>23</sup> and uploaded to the Python package inventory (PYPI). The working code is available at <https://github.com/usnistgov/SAFTsuperanc>. The calling overhead from the Python  $\rightarrow$  C++  $\rightarrow$  Python round trip is small (roughly  $0.2 \mu\text{s}/\text{call}$ ). A high-level function `PCSAFTsuperanc_rhoLV` is exposed from the library which takes  $\tilde{T}$  and  $m$  as positional input arguments and the following timing results were obtained in an IPython instance:

```
In [1]: import PCSAFTsuperanc as P
```

```
In [2]: P.__version__
Out[2]: '0.0.7'
```

```
In [3]: %timeit P.PCSAFTsuperanc_rhoLV(1.1, 5.0)
1.62 μs ± 7.98 ns per loop
```

## 7.3 Speed

Neglecting the bisection to find the relevant expansion in  $\Theta$  and in  $w$ ,  $17 \times 2$  expansions need to be evaluated; if one evaluation of degree 16 Chebyshev expansion takes  $40 \text{ ns}$ , this should take  $1.4 \mu\text{s}$  in serial. Evaluation of a  $(17 \times 17) \times (17 \times 1)$  matrix-vector product takes  $1.2 \mu\text{s}$ . Construction of a Chebyshev expansion takes  $200 \text{ ns}$  per expansion. So in total the cost is roughly  $2 \mu\text{s}$ . In total, the computational cost is roughly 100 times higher than the evaluation of a single expansion of degree 16. This highlights the curse of dimensionality – the more dimensions one wants to cover with adaptive subdivision, the more computational cost will have to be paid.

Timing results were carried out on an Apple MacBook Air with an M1 processor. Among the open-source computational libraries supporting calculation of phase equilibria for PC-SAFT, `Clapeyron.jl`, is the fastest. With `Clapeyron.jl`, a single evaluation of a saturation pressure takes on the order of  $16 \mu\text{s}$  and no explicit guess values for the orthobaric densities are required. The evaluation of a complete superancillary call for orthobaric densities in C++ takes  $1.5 \mu\text{s}/\text{call}$ . So the speedup is on the order of  $10\times$ , without sacrificing accuracy and while also improving the reliability of the calculations (making them practically guaranteed to succeed).

## 7.4 Validation

Three PC-SAFT implementations were tested for their consistency:

1. `CoolProp`, version 6.4.2.dev0<sup>1</sup>
2. `Clapeyron.jl`, version 0.3.6
3. `teqp`, version 0.9.2

<sup>1</sup>at the commit with SHA1 a2a4040912dd12a134e9f987867b55c31fe85057

For all three implementations, the calculated value of the pressure for methane ( $m = 1.0$ ,  $\sigma = 3.7039 \text{ \AA}$ ,  $\varepsilon/k_B = 150.03 \text{ K}$ , values for methane taken from Gross and Sadowski<sup>1</sup>) at a temperature and density of 200 K and 300 mol/m<sup>3</sup> is 482824.756294445 Pa.

In order to ensure that the calculated values are properly being evaluated, selected results were obtained in extended precision iterative calculations, and the calculated values from the superancillaries. The values are shown in Table 3.

## 8 Conclusions

In this work a nested set of data structures were developed in the Chebyshev formulation for the PC-SAFT equation of state<sup>1</sup> covering the temperature range from the critical point to the lowest temperature that can be achieved in double precision arithmetic. The formulation is valid for the range of  $m$  from 1 to 64 (which is more than 8 times greater than the largest value of  $m$  considered for a pure fluid in the work of Gross and Sadowski<sup>1</sup>). To answer the rhetorical question posed at the beginning of the results section, iterative VLE calculations for pure fluids from the canonical PC-SAFT equation of state without association for the range of  $m$  in this work are now obsolete; they have been replaced by the superancillary approach developed in this work.

The advent of these superancillary equations should make many thermodynamic calculations for pure fluids much faster. For instance the I-PC-SAFT EOS<sup>21</sup> could use these curves for the VLE calculations for pure fluids, which would simultaneously improve the reliability and speed of the library. The calculations with the PC-SAFT equation of state in `CoolProp` could be similarly improved. Although the superancillaries are for pure fluids, they are also useful for mixtures as they are used to obtain the starting point for tracing isothermal phase equilibria in binary mixtures.<sup>14,24</sup> Such a method has been used to fit interaction parameters for fuel-air mixtures that could be used for sustainable aviation fuels.<sup>25</sup> Inclusion of

the contributions for dipoles<sup>26</sup> or quadrupoles<sup>27</sup> should be considered, and also the association term, but those may not be possible to include in a general superancillary structure due to the curse of dimensionality. A feasible option could be to develop a superancillary equation for a given model (for instance a fixed association energy) that includes dipole and/or quadrupolar contributions, as that would have all the upsides of the superancillaries, but it would not be generic in the same way that the superancillaries are in this work.

## 9 Supplementary Material

The supplementary information includes a derivation of the critical extrapolation approach. Additional information, deposited at <https://doi.org/10.18434/mds2-2713> includes:

- The C++ code used to build all the expansions and test the implementations
- The implementation in C++, as a C++ header and implementation file
- The data files for the expansions and the critical curves, scripts used to construct the Python module

## Acknowledgments

The foundational work in the Chebyshev approach was developed in tandem with Bradley Alpert at NIST.

## Literature Cited

- (1) Gross, J.; Sadowski, G. Perturbed-Chain SAFT: An Equation of State Based on a Perturbation Theory for Chain Molecules. *Ind. Eng. Chem. Res.* **2001**, *40*, 1244–1260, DOI: 10.1021/i0003887.
- (2) Yelash, L.; Müller, M.; Paul, W.; Binder, K. Artificial multiple criticality and phase equilibria: an investigation of the PC-SAFT approach. *Phys. Chem.*

Table 3: Check values at nominal reduced temperatures of 0.9 and 0.7 for two values of  $m$ . Values with a subscript of SA were obtained from the superancillary equation, and without a subscript were obtained from extended precision iterative calculations.

$m$	$\tilde{T}$	$\tilde{\rho}$	$\tilde{\rho}_{SA}$	$\tilde{\rho}'$	$\tilde{\rho}'_{SA}$
1	1.1481738529594	0.578301305568109	0.578301305568106	0.0813666179283029	0.081366617928304
1	0.893024107857271	0.742270101982427	0.742270101982426	0.0140855386238737	0.014085538623874
64	3.79621938847095	0.0060603811080974	0.00606038110809739	5.02223276262297e-06	5.0222327626229e-06
64	2.952615079922	0.00927005575174637	0.00927005575174636	3.25402208337437e-11	3.25402208337432e-11

- Chem. Phys.* **2005**, 7, 3728, DOI: 10.1039/b509101m.
- (3) Polishuk, I. Standardized Critical Point-Based Numerical Solution of Statistical Association Fluid Theory Parameters: The Perturbed Chain-Statistical Association Fluid Theory Equation of State Revisited. *Ind. Eng. Chem. Res.* **2014**, 53, 14127–14141, DOI: 10.1021/ie502633e.
- (4) Privat, R.; Gani, R.; Jaubert, J.-N. Are safe results obtained when the PC-SAFT equation of state is applied to ordinary pure chemicals? *Fluid Phase Equilib.* **2010**, 295, 76–92, DOI: 10.1016/j.fluid.2010.03.041.
- (5) Lemmon, E. W.; Bell, I. H.; Huber, M. L.; McLinden, M. O. NIST Standard Reference Database 23: Reference Fluid Thermodynamic and Transport Properties-REFPROP, Version 10.0, National Institute of Standards and Technology. <http://www.nist.gov/srd/nist23.cfm>, 2018.
- (6) Bell, I. H.; Wronski, J.; Quoilin, S.; Lemort, V. Pure and Pseudo-pure Fluid Thermophysical Property Evaluation and the Open-Source Thermophysical Property Library CoolProp. *Ind. Eng. Chem. Res.* **2014**, 53, 2498–2508, DOI: 10.1021/ie4033999.
- (7) Wagner, W.; Pruß, A. The IAPWS Formulation 1995 for the Thermodynamic Properties of Ordinary Water Substance for General and Scientific Use. *J. Phys. Chem. Ref. Data* **2002**, 31, 387–535, DOI: 10.1063/1.1461829.
- (8) Span, R.; Lemmon, E. W.; Jacobsen, R. T.; Wagner, W.; Yokozeki, A. A Reference Equation of State for the Thermodynamic Properties of Nitrogen for Temperatures from 63.151 to 1000 K and Pressures to 2200 MPa. *J. Phys. Chem. Ref. Data* **2000**, 29, 1361–1433, DOI: 10.1063/1.1349047.
- (9) Lemmon, E. W.; McLinden, M. O.; Wagner, W. Thermodynamic Properties of Propane. III. A Reference Equation of State for Temperatures from the Melting Line to 650 K and Pressures up to 1000 MPa. *J. Chem. Eng. Data* **2009**, 54, 3141–3180, DOI: 10.1021/je900217v.
- (10) Bell, I. H.; Alpert, B. K. Efficient and Precise Representation of Pure Fluid Phase Equilibria with Chebyshev Expansions. *Int. J. Thermophys.* **2021**, 42, 75, DOI: 10.1007/s10765-021-02824-x.
- (11) Bell, I. H.; Deiters, U. K. Superancillary Equations for Cubic Equations of State. *Ind. Eng. Chem. Res.* **2021**, 60, 9983–9991, DOI: 10.1021/acs.iecr.1c00847.
- (12) Boyd, J. P. Finding the Zeros of a Univariate Equation: Proxy Rootfinders, Chebyshev Interpolation, and the Companion Matrix. *SIAM Rev.* **2013**, 55, 375–396, DOI: 10.1137/110838297.
- (13) Gross, J.; Sadowski, G. Reply to Comment on "Perturbed-Chain SAFT: An Equation of State Based on a Perturbation Theory for Chain Molecules". *Ind. Eng. Chem. Res.* **2019**, 58, 5744–5745, DOI: 10.1021/acs.iecr.9b01515.

- (14) Bell, I. H.; Deiters, U. K.; Leal, A. M. M. Implementing an Equation of State Without Derivatives: teqp. *Ind. Eng. Chem. Res.* **2022**, *61*, 6010–6027, DOI: 10.1021/acs.iecr.2c00237.
- (15) Mohr, P. J.; Newell, D. B.; Taylor, B. N.; Tiesinga, E. Data and analysis for the CODATA 2017 special fundamental constants adjustment. *Metrologia* **2018**, *55*, 125–146, DOI: 10.1088/1681-7575/aa99bc.
- (16) Bell, I. H.; Alpert, B. K. Exceptionally reliable density-solving algorithms for multiparameter mixture models from Chebyshev expansion rootfinding. *Fluid Phase Equilib.* **2018**, *476B*, 89–102, DOI: 10.1016/j.fluid.2018.04.026.
- (17) Bell, I. H.; Alpert, B. K.; Bouck, L. ChebTools: C++11 (and Python) tools for working with Chebyshev expansions. *J. Open Source Soft.* **2018**, *3*, 569, DOI: 10.21105/joss.00569.
- (18) Battles, Z.; Trefethen, L. N. An Extension Of MATLAB To Continuous Functions And Operators. *SIAM J. Sci. Comput.* **2004**, *25*, 1743–1770, DOI: 10.1137/S1064827503430126.
- (19) Olver, S.; Townsend, A. A practical framework for infinite-dimensional linear algebra. Proceedings of the 1st Workshop for High Performance Technical Computing in Dynamic Languages – HPTCDL ‘14; New Orleans, Louisiana, USA. 2014.
- (20) Thol, M.; Rutkai, G.; Köster, A.; Lustig, R.; Span, R.; Vrabec, J. Equation of State for the Lennard-Jones Fluid. *J. Phys. Chem. Ref. Data* **2016**, *45*, 023101, DOI: 10.1063/1.4945000.
- (21) Moine, E.; Piña-Martinez, A.; Jaubert, J.-N.; Sirjean, B.; Privat, R. I-PC-SAFT: An Industrialized Version of the Volume-Translated PC-SAFT Equation of State for Pure Components, Resulting from Experience Acquired All through the Years on the Parameterization of SAFT-Type and Cubic Models. *Ind. Eng. Chem. Res.* **2019**, *58*, 20815–20827, DOI: 10.1021/acs.iecr.9b04660.
- (22) Walker, P. J.; Yew, H.-W.; Riedemann, A. Clapeyron.jl: An extensible, open-source fluid-thermodynamics toolkit. *Ind. Eng. Chem. Res.* **2022**, *61*, 7130–7153, DOI: 10.1021/acs.iecr.2c00326.
- (23) Jakob, W.; Rhineland, J.; Moldovan, D. pybind11 – Seamless operability between C++11 and Python. 2016 (Accessed July 13, 2022); <https://github.com/pybind/pybind11>.
- (24) Bell, I. H.; Deiters, U. K. On the construction of binary mixture  $p$ - $x$  and  $T$ - $x$  diagrams from isochoric thermodynamics. *AIChE J.* **2018**, *64*, 2745–2757, DOI: 10.1002/aic.16074.
- (25) Rowane, A.; Bell, I. 395d Modeling of Nitrogen + Fuel Phase Behavior Using the Helmholtz-Energy-Explicit Equation of State. 2022 AIChE Annual Meeting, Phoenix, AZ. 2022.
- (26) Gross, J.; Vrabec, J. An equation-of-state contribution for polar components: Dipolar molecules. *AIChE J.* **2006**, *52*, 1194–1204, DOI: 10.1002/aic.10683.
- (27) Gross, J. An equation-of-state contribution for polar components: Quadrupolar molecules. *AIChE J.* **2005**, *51*, 2556–2568, DOI: 10.1002/aic.10502.

# TOC Graphic

