

# PYTERMO - AN OPEN-SOURCE PYTHON PROGRAM FOR THERMODYNAMIC PROPERTIES DETERMINATION, USING THE LEE-KESLER GENERALIZED EQUATION OF STATE.

**Rafael Sene de Lima**

Departamento de Engenharia Mecânica, Faculdade de Engenharia de Ilha Solteira  
Av. Brasil Centro, n.56, 15385-000, Ilha Solteira - SP – Brazil.

[rafaelsene@gmail.com](mailto:rafaelsene@gmail.com)

**Emanuel Rocha Woiski**

Departamento de Engenharia Mecânica, Faculdade de Engenharia de Ilha Solteira  
Av. Brasil Centro, n.56, 15385-000, Ilha Solteira - SP – Brazil.

[woiski@gmail.com](mailto:woiski@gmail.com)

**Abstract.** In project, analysis and optimization of thermal systems it is very important to be able to automatically obtain the thermodynamic properties of the working fluids at any time as the systems evolve between states. Unfortunately, property evaluation could be very onerous if done through thermodynamic chart lookups. However a quite accurate answer, at least for pure substances, can be obtained, determining the thermodynamic properties using a suitable generalized equation of state and some computational methods. According to that idea, Lee and Kesler suggested a modification of the Benedict-Webb-Rubin Equation of State, able to correlate the thermodynamic behavior of pure substances in compressed liquid, superheated vapor and liquid-vapor mixture states, from their critical temperature, critical pressure and Pitzer's acentric factor, as well as the coordinates of the actual state. There are some packages in the literature employing the same foundations, almost all closed source. The present code has been developed in Python as part of a much larger simulation program, although it may be used standalone. Python has been chosen as a multiplatform object-oriented open-source programming language, exhibiting very readable and powerful syntax. In order to validate both the methodology and the code, several results from the literature are compared, for water, hydrocarbons and refrigerants.

**Keywords.** Thermal systems, Thermodynamic properties, Generalized equation of state, Python language.

## 1. Introduction

The procedure of obtaining thermodynamic properties by means of thermodynamic lookup tables has been very common among students and engineers. However, that error-prone procedure delays the attainment of important data for a thermodynamic analysis, so the solution would be getting such data automatically employing some software. There are some available packages able to provide thermodynamic properties, e.g. EES – *Engineering Equations Solver*, CATT2 – *Computer Aided Thermodynamic Tables*, FLUIDPROP and NIST-REFPROP, where, from those, only the last one is open source and the second from last is actually free.

The present work describes an open source and free *Python* application development for thermodynamic properties determination. Through Pytermo the student, or engineer, will be able to understand the structure behind the construction of a thermodynamic chart, mainly because the chosen language to develop Pytermo was the Python programming language.

To elucidate the significant productivity gain in program construction with Python, it is enough to state that a code in Python is two to ten times smaller than one in C/C++, moreover, Python possess a concise and easy reading syntax, allowing to the user a straightforward source code manipulation in the desired way.

## 2. The Lee & Kesler Equation of State

One way for utilization of an equation of state (EOS) for pure fluids is determining the EOS parameters fitting them to experimental data for each individual fluid. However, instead of setting particular parameters for each fluid, an EOS can be extended to broad classes of fluids by expressing the temperature and pressure in terms of the critical properties  $T_c$  and  $P_c$  and at least one additional parameter such as the acentric factor,  $\omega$ . The result is a generalized form of EOS.

By means of the Pitzer acentric factor, Pitzer et. al. (1955), it is possible to calculate the compressibility factor  $z$  of a pure fluid as a linear function of the acentric factor,  $\omega$ , as follows,

$$z = z^{(0)} + \omega \frac{(z^{(r)} - z^{(0)})}{\omega^{(r)}} \quad (1)$$

where the superscripts  $(0)$  and  $(r)$  refer to a simple and a reference fluid respectively.

In 1975, Lee and Kesler (1975), seeking to improve the Pitzer results for fluids involved in hydrocarbon processing, developed an analytical implementation for  $z(v'_r, T_r)$ , using a modified Benedict-Webb-Rubin (1942) (BWR) EOS, with 12 empirical constants and using  $\omega = 0$  and  $\omega = 0.3978$  as the simple and reference fluids acentric factors respectively. The Lee-Kesler EOS is:

$$z = \left( \frac{P_r v'_r}{T_r} \right) = 1 + \frac{B}{v'_r} + \frac{C}{v'^2_r} + \frac{D}{v'^5_r} + \frac{c_4}{T_r^3 v'^2_r} \left( \beta + \frac{\gamma}{v'^2_r} \right) \exp \left( -\frac{\gamma}{v'^2_r} \right)$$

$$B = b_1 - \frac{b_2}{T_r} - \frac{b_3}{T_r^2} - \frac{b_4}{T_r^3}$$

$$C = c_1 - \frac{c_2}{T_r} + \frac{c_3}{T_r^3}$$

$$D = d_1 + \frac{d_2}{T_r}$$
(2)

where  $z$  is the compressibility factor and  $T_r$ ,  $P_r$  and  $v'_r$  are the reduced temperature, pressure and specific volume respectively, given by

$$T_r = \frac{T}{T_c}, \quad P_r = \frac{P}{P_c}, \quad v'_r = \frac{vP_c}{RT_c}$$
(3)

and  $T$ ,  $P$ ,  $v$  and  $R$  are the temperature, pressure, specific volume and the ideal gas constant respectively. The remaining parameters are the foregoing mentioned empirical constants obtained by Lee & Kesler (1975), disposed at Tab. (1) for simple and reference fluids, in which octane has been chosen as reference fluid.

Table 1 – Lee & Kesler EOS constants for simple and reference fluids.

Constant	Simple Fluid	Reference Fluid
$b_1$	0,1181193	0,2026579
$b_2$	0,265728	0,331511
$b_3$	0,154790	0,027655
$b_4$	0,030323	0,203488
$c_1$	0,0236744	0,0313385
$c_2$	0,0186984	0,0503618
$c_3$	0,0	0,016901
$c_4$	0,042724	0,041577
$d_1 \times 10^4$	0,155488	0,48736
$d_2 \times 10^4$	0,623689	0,0740336
$\beta$	0,65392	1,226
$\gamma$	0,03754	0,060167

The thermodynamic properties enthalpy and entropy are expressed into a dimensionless form called the entropy and enthalpy departures, and can be expressed as a linear functions of the acentric factor,  $\omega$ , as in Eq. (1). According to Lee and Kesler (1975), the enthalpy and entropy departures and the fugacity coefficient are expressed, using the Lee & Kesler EOS, Eq.(2), as,

$$\frac{h - h^*}{RT_c} = T_r \left\{ z - 1 - \frac{b_2 + \frac{2b_3}{T_r} + \frac{3b_4}{T_r^2}}{T_r v'_r} - \frac{c_2 - \frac{3c_3}{T_r^2}}{2T_r v'^2_r} + \frac{d_2}{5T_r v'^5_r} + 3E \right\}$$
(4)

$$\frac{s - s^*}{R} = -\ln \left( \frac{P}{P^*} \right) + \ln(z) - \frac{b_1 + \frac{b_3}{T_r^2} + \frac{2b_4}{T_r^3}}{v'_r} - \frac{c_1 - \frac{2c_3}{T_r^3}}{2v'^2_r} - \frac{d_1}{5v'^5_r} + 2E$$
(5)

$$\ln \left( \frac{f}{P} \right) = z - 1 - \ln(z) + \frac{B}{v'_r} + \frac{C}{2v'^2_r} + \frac{D}{5v'^5_r} + E,$$
(6)

where

$$E = \frac{c_4}{2T_r^3 \gamma} \left\{ \beta + 1 - \left( \beta + 1 + \frac{\gamma}{v_r'^2} \right) \exp \left( -\frac{\gamma}{v_r'^2} \right) \right\}$$

and  $\ln(f/P)$  is the fugacity coefficient to be described in the Saturation section.

### 3. Methodology

#### 3.1. Obtaining $v_r'$ from $T_r$ and $P_r$

Once Eq. (2) is implicit in  $T_r$  and  $v_r'$ , it is initially necessary to use an efficient numerical root-finding method to obtain  $v_r'$  as a function of  $T_r$  and  $P_r$  and then determine the remaining properties. The Newton-Raphson method was chosen to obtain  $v_r'(T_r, P_r)$ . First Eq. (2) is rewritten as,

$$0 = -\left( \frac{P_r v_r'}{T_r} \right) + 1 + \frac{B}{v_r'} + \frac{C}{v_r'^2} + \frac{D}{v_r'^5} + \frac{c_4}{T_r^3 v_r'^2} \left( \beta + \frac{\gamma}{v_r'^2} \right) \exp \left( -\frac{\gamma}{v_r'^2} \right) \quad (7)$$

In order to obtain  $v_r'(T_r, P_r)$  from Eq. (7), two initial guesses are necessary for the Newton Raphson method convergence of  $v_r'$ , a large value for superheated vapor and a small value for compressed liquid. In this work, the selected initial guesses are 0,008 and 20 respectively. After solving the Eq. (7) using the constants at Tab. (1) for the simple and the reference fluid, is possible to calculate  $z(T_r, P_r, \omega)$  using the Eq. (1).

#### 3.2. Saturation

From the Gibbs function, important property for the fluid phase change at equilibrium, the fugacity,  $f$ , is defined, which has a dimensionless form called fugacity coefficient,  $\ln(f/P)$ . In two-phase region, the liquid and vapor fugacities are the same. Therefore, in a dimensionless form,

$$\ln \left( \frac{f}{P} \right) (T_r, P_r, v_r')_l = \ln \left( \frac{f}{P} \right) (T_r, P_r, v_r')_v, \quad (8)$$

but  $v_r' = v_r'(T_r, P_r)$ , hence

$$\ln \left( \frac{f}{P} \right) (T_r, P_r, v_r'(T_r, P_r))_l = \ln \left( \frac{f}{P} \right) (T_r, P_r, v_r'(T_r, P_r))_v, \quad (9)$$

As mentioned before, the initial guesses for the Newton Raphson method convergence of  $v_r'(T_r, P_r)_{(v)}$  and  $v_r'(T_r, P_r)_{(l)}$  are 20 and 0,008 respectively. Rewriting the Eq. (9) in the same form of Eq. (7) give,

$$\ln \left( \frac{f}{P} \right) (T_r, P_r, v_r'(T_r, P_r))_l - \ln \left( \frac{f}{P} \right) (T_r, P_r, v_r'(T_r, P_r))_v = 0 \quad (10)$$

This equality permits to numerically calculate  $T_r(P_r)$  or  $P_r(T_r)$  in the two-phase region of a pure substance. Again, a numerical root finding method is required to obtain  $T_r(P_r)$  or  $P_r(T_r)$  and the Newton Raphson was chosen. However, here the initial guesses are pressure or temperature values, so, a good initial guess is obtained using the vapor pressure correlation proposed by Lee & Kesler (1975), as a function of the acentric factor  $\omega$ , Eq. (11),

$$\ln(P_r) = 5.92714 - \frac{6.09648}{T_r} - 1.28862 \ln(T_r) + 0.169347 T_r^6 + \omega \left( 15.2518 - \frac{15.6875}{T_r} - 13.4721 \ln(T_r) + 0.43577 T_r^6 \right) \quad (11)$$

After numerically solving the Eq. (10) using the constants at Tab. (1) for the simple and the reference fluid, it is possible to fit a correlation for a generalized vapor pressure curve, as a function of the acentric factor  $\omega$ . The correlation has been divided in two intervals for better results, one for  $0.30 < Tr < 0.90$  (Eq. 12a) and another for  $0.91 < Tr < 0.99$  (Eq. 12b):

$$\ln(P_r) = -279,317936.T_r^6 + 1254,437017.T_r^5 - 2352,620771.T_r^4 + 2381,469001.T_r^3 - 1397,498918.T_r^2 + 470,032214.T_r - 76,510885 + \omega(-288,430735.T_r^6 + 1358,027380.T_r^5 - 2702,962460.T_r^4 + 2949,232448.T_r^3 - 1895,177848.T_r^2 + 699,430958.T_r - 120,134988) \quad (12a)$$

$$\ln(Pr) = 16568,705307.Tr^6 - 93006,477660.Tr^5 + 217547,712342.Tr^4 - 271393,368496.Tr^3 + 190430,818981.Tr^2 + 71246,168407.Tr - 11098,777935 + \omega(700541,840273.Tr^6 - 3968329,771123.Tr^5 + 9364624,636528.Tr^4 - 11783936,678735.Tr^3 + 8339306,105560.Tr^2 - 3146896,451088.Tr - 494690,320166) \quad (12b)$$

It should be stressed that the vapor pressure correlation in Eqs. (12a) and (12b) developed in this work and which has never been published before, exhibits a much better behavior along a greater range of inputs than Eq. (11), as will be demonstrated in the Results section.

An illustrative procedures diagram is depicted in Fig.(1).

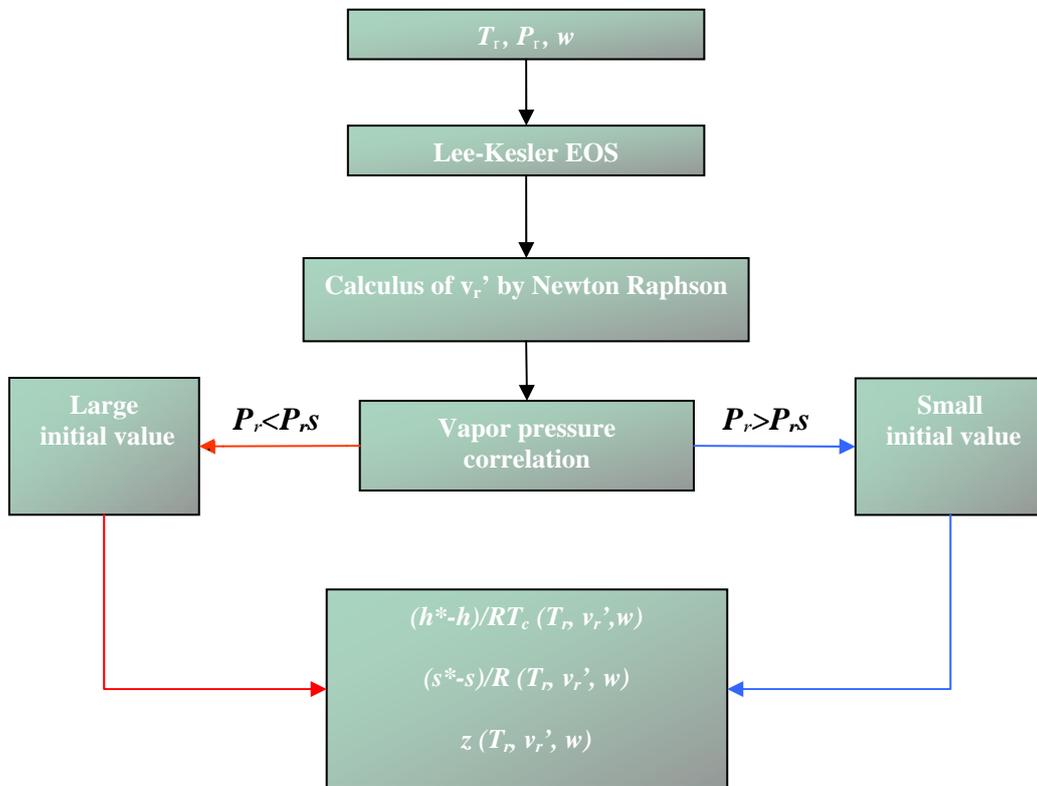


Figure 1 –Diagram illustrating the internal working.

## 4. Results

### 4.1 The Pytermo package

The Pytermo package was entirely developed in the Python programming language, from the source code until the graphical interface, which has been built using the wxPython library. Pytermo has a user-friendly graphical interface allowing for results either in the dimensional or the dimensionless form. The Pytermo obtained results are output in a grid fashion, which allows the user to easily determine either a single thermodynamic state or several thermodynamic states by just by keeping one property fixed and changing another thermodynamic property to obtain a sequential result.

A graphical interface of Pytermo is depicted in Fig. (2).



Figure 2 – A dialog for data entrance of Pytermo for dimensional thermodynamic properties determination in a range format.

The window “planilha - dimensional” is disclosed after pushing the “substância” button, located at the top left side, and it is used when it is desirable to obtain dimensional results in a sequential form. In that window, one property must have three input values, the initial, the final and a step value, represented in the dialog by “Valor inicial”, “Valor final” and “Passo” respectively. The other property has a single input value. The result for this example is shown at Fig. (3).

	T(K)	P(MPa)	h(kJ/kg)	s(kJ/kgK)	x(01)
1	484.961426513	2.0	2000.0	4.70336519641	0.444386122889
2	489.839262453	2.2	2000.0	4.6828040129	0.435376985597
3	494.394672958	2.4	2000.0	4.66434177399	0.426808468645
4	498.673496791	2.6	2000.0	4.64762558051	0.418615157882
5	502.711953722	2.8	2000.0	4.63238316851	0.410745187698
6	506.539204673	3.0	2000.0	4.6184002406	0.403156660253
7	510.179107686	3.2	2000.0	4.60550521443	0.395815179727
8	513.65145613	3.4	2000.0	4.59355865988	0.388692107334
9	516.972872876	3.6	2000.0	4.58244579893	0.381763297008
10	520.15746957	3.8	2000.0	4.57207106303	0.375008160354
11	523.21734166	4.0	2000.0	4.56235406754	0.368408962555

Figure 3 – A grid for output data of Pytermo for dimensional thermodynamic properties determination in a range format.

If a single result is desired the dialog is represented by Fig. (4).

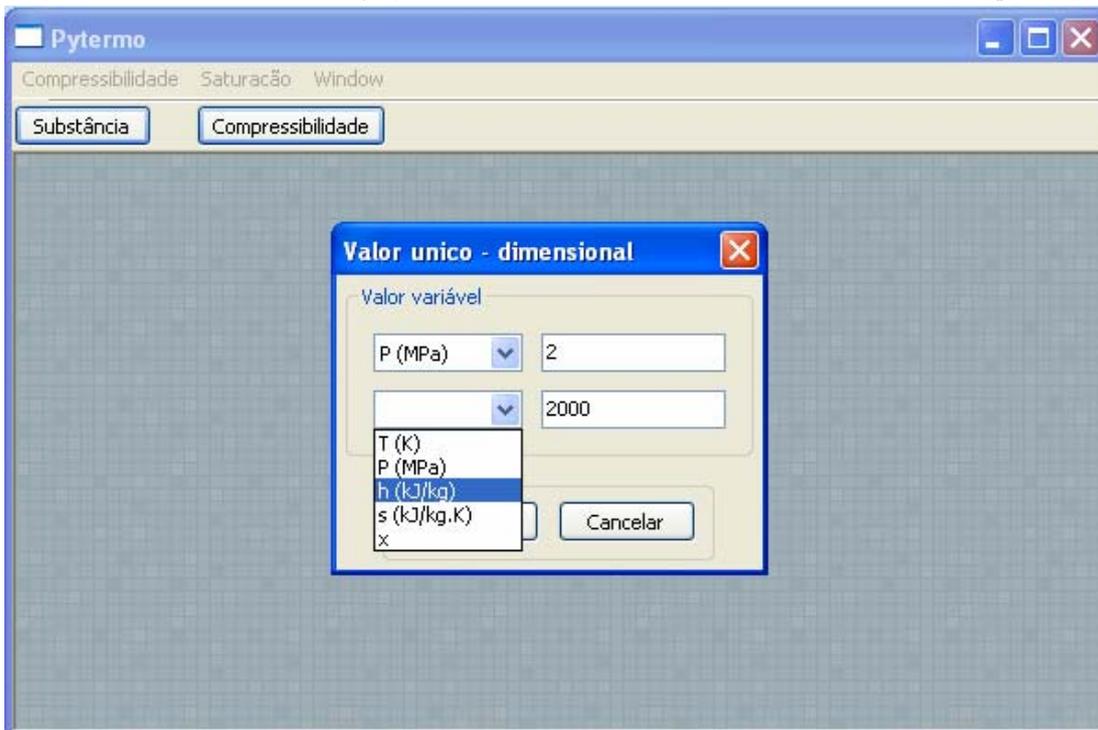


Figure 4 – A dialog for data entrance of Pytermo for dimensional thermodynamic properties determination in a single format.

The grid for results is,

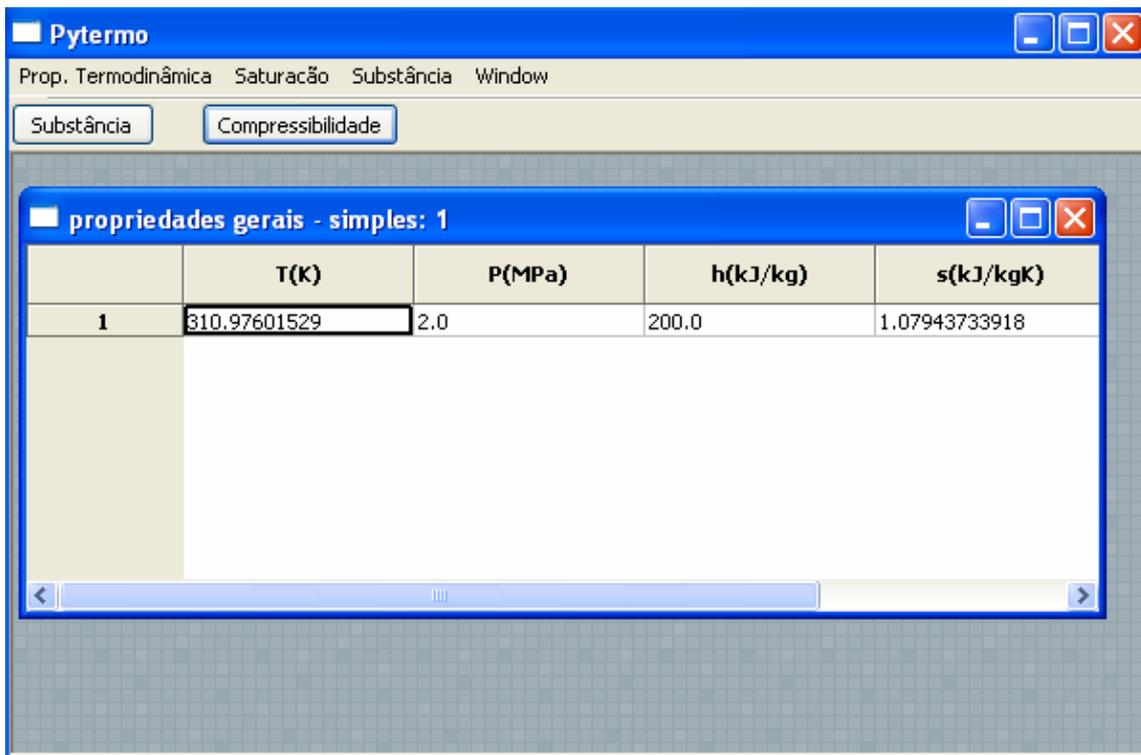


Figure 5 – A grid for results of Pytermo for dimensional thermodynamic properties determination in a single format.

If is desired to obtain dimensionless results, the dialog for data entrance will be:

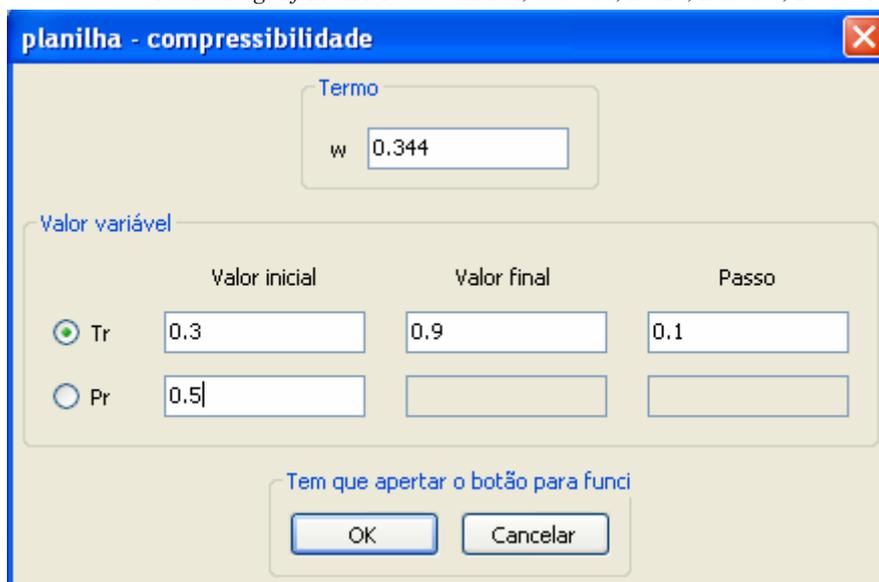


Figure 6 – A dialog for data entrance of Pytermo for dimensionless thermodynamic properties determination in a single format.

Again, the dialog “planilha - dimensional” is shown when the user clicks at the “substância” button, located at the top left side.

#### 4.3 Tests for some substances

Comparisons for hydrocarbons and refrigerants were made between results using this work and NIST REFPROP 6.01. The Tabs. (2) and (3) shows the average deviation, standard deviation and the maximum deviation for each analyzed fluid.

Table 2 – Percentage deviations for enthalpy.

ENTHALPY			
Substance	Average deviation (%)	Standard deviation (%)	Maximum deviation (%)
Ammonia	5.963	7.914	24.023
butane	1.308	1.887	13.306
propane	1.642	2.907	18.945
ethane	1.113	2.074	11.638
R12	2.810	3.270	11.308
R134a	1.006	1.770	18.308

Table 3 – Percentage deviations for entropy.

ENTROPY			
Substance	Average Relative Deviation (%)	Standard Relative Deviation (%)	Maximum Deviation (%)
Ammonia	5.929	7.828	22.642
butane	1.612	1.257	12.598
propane	1.316	2.199	12.691
ethane	1.061	1.878	8.823
R12	2.900	3.279	12.211
R134a	0.993	1.772	15.308

Finally, in order to show that the Eq. (12) produces more accurate values than Eq. (11), the average deviations, comparing with the standard steam pressure table, taken from Wylen, Sonntag and Borgnakke (1955), have been plotted in the Fig. (7).

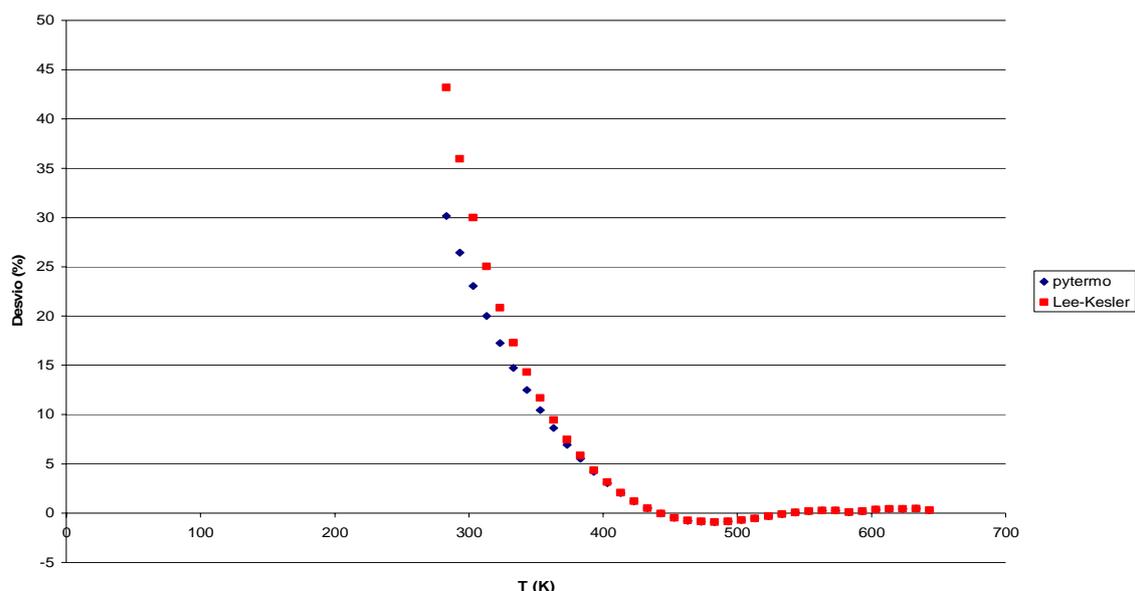


Figure 7 – Comparison of the steam pressure curves average deviations between Lee & Kesler and the present work for water.

## 5. Discussion of results.

Due to Lee & Kesler EOS complexity, there are convergence problems for some values of  $T_r$  and  $P_r$  as for the regions near to the critical point ( $T_r = 1$ ,  $P_r = 1$ ) and at compressed liquid values, especially for very low reduced pressures. Caution is necessary when the temperature approaches the assumed reference state ( $s_l = 0$  and  $h_l = 0$  for  $T = 233.15$  K), because residual values there tend to zero and any absolute deviation results in a large relative deviation.

Is necessary to observe that the maximum deviation values at Tabs. (2) and (3) occur in the *compressed liquid region*, since Lee & Kesler equation behaves much better in the vapor and gas phase. Additionally, since these algorithms have been developed in a modular fashion using Python with Object Orientation, a change in the EOS is quite feasible, allowing for much better results in the case of polar substances, for example. On fact, simulation of mixtures using suitable enhanced EOSs and pseudo-critical models is also being worked upon.

## 6. References

- Wylen, G. V.; Sonntag R.; Borgnakke C. “Fundamentos da Termodinâmica Clássica”. 9.ed. São Paulo: Edgard Blucher, 1995. 580 p.
- Lee, B. I.; Kesler, M. G. “A Generalized Thermodynamic Correlation Based on Three-Parameter Corresponding States”. AIChE Journal, New Jersey. v.21, 1975.
- Pitzer, K.S. et al., *J. Am. Chem.Soc.*, 77, 3439, 1955.
- Çengel, Y. A.; Boles, M. A. “Thermodynamics an Engineering Approach” 3. ed. Princeton: McGrawHill, 1997. 1010 p.
- Rossum, G. V. “Python Programming Language”. 2003. available in: <[Http://Www.Python.Org/](http://www.python.org/)> e <<http://docs.python.org/ext/ext.html>>. Accessed in: 15 dec. 2005.
- Oliphant, T. E. “SciPy Tutorial” 2004. disponível em: <<http://www.scipy.org/scipy.org/documentation/>> Acesso em: 15 dez. 2005.
- McLinden, M. O.; Klein, S. A.; Lemmon, E. W.; Peskin, A. P. “REFPROP Thermodynamic and Transport Properties of Refrigerants and Refrigerant Mixtures”. NIST Standard Database 23, Washington DC. v.6.01, 1998.
- Rossum, G. “Python Programming Language”. Stichting Mathematisch Centrum, Netherlands, version 2.3.5., Copyright (c) 2001-2005 Python Software Foundation.