NATIONAL POLYTECHNIC INSTITUTE

SCHOOL OF CHEMICAL ENGINEERING E
EXTRACTIVE INDUSTRIES

D EPARTMENT PETROLEUM CHEMICAL ENGINEERING

THESIS FOR THE TITLE PETROLEUM
CHEMICAL ENGINEER

# thermodynamic properties server open source based on cubic equations of state

PRESENTS: Lira
Abad Conde

Under the supervision of Dr.
Rogelio Sotelo Buoys

Mexico DF

OCTOBER 2007

# Thanks

During the development of this work I have received a lot of support, so I hope to meet the expectations of the people who supported me all this time, so I need first to acknowledge that anything Dr. Rogelio Sotelo Buoys by his patience in the direction of the thesis.

I also thank Dr. Enrique Arce Media and Dr. Oliver Silva Guadalupe for their help in programming OllinTS, Esther Solano Fragoso and Lourdes Reyes Martinez de Simsci Esscor for your time to clarify my doubts and give me study material. A Eduardo Luna Ortiz for giving the necessary documentation.

I thank especially Elizondo Raul Cota for helping me for their help in the study of Sim42.

And basically my family ...

*But we are not our thoughts. When we identify with them, we cease to be ourselves. Thoughts are, we do not. The sound of an empty mind is the noise they make the words of the questioner. "Where a thought arises and what is it?" Is a set of words that should not respond with another set of words. From one thought another thought arises and so on to infinity ...*

*"The teacher and Magas", A. Jodorowsky*

# Index

Summary..... ..... ..... ..... III

1. Introduction..... ..... ..... .... 1

1.1 Motivation ..... ..... ..... ..... 2

1.2 Structure of thesis ..... ..... ..... ..... 4

2 General ..... ..... ..... ..... 5

2.1 History of simulation ..... ..... ..... ..... 5

2.2 General simulation models ..... ..... ..... 7

..... 2.3 Simulation Styles ..... ..... ..... 9

2.3.1 Qualitative and quantitative simulation ..... ..... ..... 9

2.3.1.1 Simulation steady state ..... ..... ..... 10

2.3.1.2 Simulation dynamic state ..... ..... ..... 10

2.3.1.3 Methods solution simulators ..... ..... .....eleven

2.3.1.3.1 oriented global Simulators or equations ..... .....eleven

2.3.1.3.1.1 Main features of global simulators ..... 11

2.3.1.3.2 modular Simulators ..... ..... ..... 12

2.3.1.3.2.1 Main features of the modular simulators ..... 13

2.4 Components of a simulator ..... ..... ..... 13

2.4.1 Structure of a general purpose simulator ..... ..... ..... 14

2.4.2 Interface of the basic components of a simulator ..... ..... 16

2.5 Steps for developing a process simulator ..... . ..... 18

3 Thermodynamic properties ..... ..... ..... twenty

Chemical equilibrium ..... 3.1 ..... ..... .....twenty

3.2 Phase Equilibria ..... ..... ..... .....twenty-one

3.2.1 equilibrium systems ..... ..... ..... ..... 22

3.2.2 Fugacidad ..... ..... ..... ..... 22

3.2.3 Activity coefficient ..... ..... ..... 2. 3

3.2.4 Vapor pressure ..... ..... . ..... 2. 3

3.2.4.1 Antoine equation ..... ..... .. ..... 2. 3

3.2.4.2 FrostKalkwarfThodos Equation (Harlacher) ..... ..... ..... 24

3.2.5 Thermodynamic properties ..... ..... ..... 24

3.2.6 Heat Capacity ..... ..... ..... ..... 24

3.2.7 Enthalpy ..... ..... ..... ..... 25

3.2.8 Entropy ..... ..... ..... . 25

3.2.9 Internal energy ..... ..... ..... . 26

3.2.10 Gibbs free energy ..... ..... ..... 26

3.2.11 Helmholtz free energy ..... ..... ..... 27

3.3 Equations of State ..... ..... ..... ..... 27

3.4 Cubic Equations of State ..... ..... ..... . 29

3.4.1 RedlichKwong (1949) ..... 31
3.4.2 RedlichKwong (1949) Simplified ..... 32
3.4.3 SoaveRedlichKwong (1972) ..... 33
3.4.4 PengRobinson (1976) ..... 3. 4
3.4.5 Isothermal Flash (liquid-vapor equilibrium) ..... 35
4 Development of OllinTS program ..... 38
4.1 The programming language ..... 38
4.2 Structure thermodynamic properties server ..... 40
4.3 virtual structure OllinTS ..... 41
4.4 Database ..... 43
4.5 Considerations programming ..... 43
4.6 Main algorithms OllinTS ..... 44
Solver 4.6.1 ..... Four. Five
4.6.2 Flash ..... 46
4.6.3 FracVap ..... 47
4.6.4 FracTemp ..... 48
4.6.5 HenTemp ..... fifty
4.6.6 HenPre ..... 52
4.6.7 Isotermic ..... 54
4.6.8 Thermal ..... 60
4.6.9 ZG ..... 62
4.6.10 ZL ..... 63
4.6.11 Zo ..... 65
4.6.12 Thermo.Calc ..... 66
4.7 Errors ..... 66
4.8 Commands OllinTS ..... 67
4.8.1 Administrator Commands ..... 67
4.8.2 Commands thermodynamic case ..... 69
4.9 Calculation of balance fasesEjemplo ..... 69
5 cases of application ..... 73
Case 1. Construction of a phase diagram ..... 73
Case 2. Calculate the vapor pressure ..... 76
Case 3. Design of a phase separator LV ..... 80
Conclusions and recommendations ..... 83
Appendix A: Analysis of results ..... 85
Appendix B: Creating database ..... 89
Appendix C: Installation procedure in Windows XP ..... 96
Appendix D: Nomenclature OllinTS ..... 99
Appendix E: Glossary ..... 105

## Summary

In this work the development of a, based on cubic equations of state (Redlich-Kwong, Soave-Redlich-Kwong and Peng-Robinson), with the aim that can be used as a scientific tool server thermodynamic properties is presented free distribution main reason that have promoted the realization of this work, since most of the programs that are used by chemical engineers in the industrial, scientific and academic area are proprietary programs or restricted license.

The thermodynamic properties server written in Python 2.5 language based on cubic equations of state, due to the ease of estimating other state properties from the compressibility factors. In addition to these thermodynamic calculations for a small number of data you are needed.

The server properties result of this work has been called OllinTS, which allows the estimation of phase equilibrium mixtures also calculates the thermodynamic properties such as enthalpy, entropy, internal energy, free energy, free energy Helmholtz. All these values are calculated for each mixture and the pure components. OllinTS is able to estimate the phase equilibrium knowing the composition of the mixture and any of the following pairs of variables.

1. Pressure-Temperature
2. Pressure-Enthalpy
3. Pressure-Vapor Fraction
4. Steam temperature-Fraction
5. Temperature-enthalpy

The results obtained from the application program developed for observing the flexibility of the cubic equation of state in the application to solving the problem of industrial processes and petroleum industry, besides the ease with which it used in the calculation of phase equilibrium from parameters that are not directly related to the cubic equation of state. Although, currently the program only considers the equilibrium liquid-vapor phase. In addition, OllinTS has no ability to solve problems in at least one compound is in the critical region and is not able to identify retrograde condensation.

## Boards

Table 1. Nomenclature ..... SAW
Table 2. Subscripts ..... VII
Table 3. Greek letters ..... VII
Table 4. Historical Timeline simulating chemical processes [3] ..... 5
Table 4. Historical Timeline simulating chemical processes (continued) ..... 6
Table 4. Historical Timeline simulating chemical processes (continued) ..... 7
Table 5. Models of the state equations ..... 29
Table 6 Specifications Redlich-Kwong equation ..... 31
Table 7. Specifications Redlich-Kwong equation Simplified ..... 32
Table 8. Specifications Soave Redlich-Kwong equation-..... 33
Table 9. Specifications Peng-Robinson equation ..... 3. 4
Table 10. Main algorithms OllinTS ..... 44
Table 11. Composition of the mixture of hydrocarbons ..... 70
Table 12.Composición in the balance ..... 72
Table 13. Composition of the mixture ..... 73
Table 14. Composition of the mixture of aromatic ..... 80
Table 15. Composition of the mixture ..... 85
Table 16. Phase equilibria of the hydrocarbon mixture at 282 K ..... 86
Table 17. Phase Equilibria of the hydrocarbon mixture at 244.76 K ..... 87
Table 18. Phase Equilibria of the hydrocarbon mixture at 297.13 K ..... 88
Table 19. Nomenclature OllinTS ..... 99
Table 19. Nomenclature OllinTS (continued) ..... 100
Table 19. Nomenclature OllinTS (continued) ..... 101
Table 20. Nomenclature database ..... 102
Table 20. Nomenclature database (continued) ..... 103
Table 20. Nomenclature database (continued) ..... 104

## figures

Figure 1. Simulation Styles ..... 9
Figure 2. Relationship of mathematical models ..... 14
Figure 3. Typical structure of a simulator ..... 16
Figure 4. Development Process simulators chemical processes ..... 19
Figure 5. Separator flash ..... 36
Figure 6. Based Programs OllinTS ..... 41
Figure 7. Basic structure of the program OllinTS ..... 42

Figure 8. Scheme work OllinTS ..... ..... ..... 42

Figure 9. Algorithm Solver ..... ..... . .....Four. Five

Figure 10. Flash Algorithm ..... ..... ..... ..... 46

Figure 11. Algorithm FracVap ..... ..... ..... ..... 47

Figure 12. Algorithm FracTemp ..... ..... ..... .... 48

Figure 13. Algorithm FracTemp (continued) ..... ..... ..... 49

Figure 14. Algorithm HenPre ..... ..... ..... ..... fifty

Figure 15. Algorithm HenPre (continued) ..... ..... ..... 51

Figure 16. Algorithm HenPre ..... ..... ..... ..... 52

Figure 17. Algorithm HenPre (continued) ..... ..... .. .. 53

Figure 18. Algorithm for the model Isotermic Redlich Kwong ..... ... ..... 54

Figure 19. Algorithm for the model Isotermic Redlich Kwong (continued) ..... ... .. 55

Figure 20. Algorithm for SoaveRedlichKwong model Isotermic ..... ..... .... 56

Figure 21. Algorithm for SoaveRedlichKwong Isotermic model (continued) 57 .....

Figure 22. Model Algorithm Isotermic PengRobinson ..... .. ..... 58

Figure 23. Model Algorithm Isotermic PengRobinson (continued) ..... ..... 59

Algorithm Figure 24. Thermal ..... ..... ..... ..... 60

Figure 25. Thermal Algorithm (continued) ..... ..... . .. 61

Figure 26. Algorithm ZG ..... ..... ..... ..... 62

Figure 27. Algorithm ZL ..... ..... ..... ..... 63

Figure 28. Algorithm ZL (continued) ..... ..... ..... 64

Figure 29. Algorithm Zo ..... ..... ..... ..... 65

Figure 30. Algorithm Thermo.Calc ..... ..... ..... 66

Figure 31.Separador by flash ..... ..... ..... 70

Figure 32. Graph phase equilibrium ..... ..... ..... 76

Figure 33. Vapor pressure of butane ..... ..... ... ..... 79

Figure 34. Vapor pressure of n-heptane ..... ..... . ..... 79

Figure 35. Glyph Ollin ..... ..... ..... ..... 106

*Table 1. Nomenclature*

| Symbol | Meaning |
|---|---|
| $P$ | Pressure |
| $T$ | Temperature |
| $V$ | molar volume |
| $R$ | Gas constant |
| $x_i$ | mole fraction of the liquid phase |
| $Y_i$ | mole fraction of the gas phase |
| $z_i$ | mole fraction of the mixture |
| $Z$ | Compressibility factor |
| $F$ | fleetingness |
| $P_V$ | Vapor pressure |
| $C_p$ | heat capacity at constant pressure |
| $C_V$ | heat capacity at constant volume |
| $H$ | enthalpy |
| $OR$ | Internal energy |
| $S$ | entropy |
| $G$ | Gibbs free energy |
| $TO$ | Helmholtz free energy |
| • $C_p$ | residual heat capacity at constant pressure |
| • $C_V$ | residual heat capacity at constant volume |
| • $H$ | residual enthalpy |
| • $OR$ | residual internal energy |
| • $S$ | residual entropy |
| • $G$ | Gibbs free energy of residual |
| • $TO$ | Power residual Helmholtz free |
| $Q$ | Energy flow |
| $F$ | molar feed flow |
| $L$ | molar flow of liquid |
| $V$ | Molar Steam Flow |
| $Tr$ | reduced temperature |
| $D$ | Pipe Diameter |
| $v$ | Speed |
| $Re$ | Reynolds number |

*Table 2. Subscripts*

| Symbol | Meaning |
|---|---|
| *i* | Composite Index |
| *L* | Liquid phase |
| *V* | Vapor phase |

*Table 3. Greek letters*

| Symbol | Meaning |
|---|---|
| • | Fugacity coefficient |
| • | Activity coefficient |
| • | chemical potential, fluid viscosity |
| π | Number of phases |
| • | vaporised fraction |
| • | acentric factor |
| • | Coefficient of thermal expansion |
| Ω | Parameter of the cubic equation of state |
| • | Fluid density |

# 1 Introduction

Simulators chemical processes are the tool that allows the chemical engineer to obtain information from a chemical plant quickly and reliably to a certain point. Before the advent of computer calculations that are available today they simulators chemical processes were performed manually, which was repetitive and tedious. In addition, these calculations demanded a lot of man hours. With the advent of computers began to automate calculations ( *eg* vapor pressure) by simple programs. Shortly after the development of new and better programming languages and the increasing capacity of computers these programs were integrated to solve complex operations and unit processes, eg flash separators. The next step in the development of chemical process simulation was the integration of programs representing the unit operations so they could exchange information, so you can simulate a chemical plant in one step.

To represent the unit operations and fluid behavior simulators use mathematical models constructed from experimental data obtained by numerical methods regressions. Therefore simulators can be considered as predictive tools.

Currently simulators chemical processes are present throughout the life of a chemical plant. During the design are used to calculate the flow of the plant, the dimensions of the equipment, risk analysis, analysis of the evolution of disturbances. During operation they can assist in energy and material balances unit or globally, in adjusting the control variables ( *eg* when there are changes in input conditions and / or product specifications), downsizing of the plant, analysis of the impact on the environment.

Simulators to apply to the processing plants should be analyzed carefully

simulation results, compared and interpreted with respect to the history of the plant to prove that the parameters have been properly adjusted to simulate the behavior of the plant and thus be applied for predicting the optimum operating conditions, scaling plant or disturbance analysis.

Process simulators are highly profitable and relatively easy to operate thanks to have tools for process optimization, analysis of the behavior of unit operations and also provide graphical environments to develop simulations of a simple and convenient way.

### 1.1 Motivation

Within the overall development of programs such as simulators chemical process development are two philosophies:

- Free Software [1] (free software)
- not free software [2] ( *ie* proprietary software)

Most free software is developed by international teams cooperating through free association. The teams are made up of individuals with a wide variety of motivations. In addition to these groups, there are specialized companies that support the development of projects to offer products and services related to projects that sponsor services. Most of these projects have mainly two types of license:

- GNU General Public License (GPL)
- BSD style licenses (Berkeley Software Distribution)

Under this philosophy of free software development there are several programs for scientific computing, for example, molecular Monte Carlo simulation, combustion analysis, virtual laboratories, arrangement of heat exchangers. Most of these projects can be viewed and downloaded at the www.sourceforge.net page.

In the area of simulation of chemical processes there are several open source projects, of which stand out are as follows.

- **Sim42:** It is a non-sequential modular simulator written in "Python" under a BSD license, the project was founded in 2000 at the University of Calgary, Canada, with the collaboration of Virtual companies Materials®. The latest version that was released is 2.0, the

which provides commands and interface with a graphical interface. The problem is that it depends Sim42 server thermodynamic properties of Virtual Materials®, resulting then only partially free. Currently due to dependence on Sim42 project Virtual Materials® and lack of community support, the project has stalled. At the moment there is no official website of the project, but the source code is still available online or page: http://sourceforge.net/projets/ollints

- **OpSim:** founded in March 2006 under a GNU GPL project, which aims to develop a sequential modular simulator with its own server and thermodynamic properties similar to HYSYS graphical interface. He is currently in the planning stage and raises it to be written in "Free Pascal". http://sourceforge.net/projects/opsim.

- **Open** Chemasim: It is a sequential modular chemical process simulator written in Fortran that has its own server thermodynamic properties. It is developed by BASF and ITT University of Stuttgart. Your license is more restrictive because it only allows the distribution of the program and its source among its partner community code. http://chemasim.itt.uni-stuttgart.de.

It based on the current state of Sim42 project, this paper intends to initiate the development of a server thermodynamic properties, which can be applied as an educational tool or as a base module for the development of specific applications such as equipment design, and future work standards CAPE-OPEN apply [1] for use in chemical process simulators. The server properties allow Virtual Server to replace the Materials® in Sim42.

The importance of thermodynamic properties server within the simulation of chemical processes because it provides information on the thermodynamic and transport properties of fluids considered within a simulation. For example, the case of a flash separator server provides the property value of the vaporized fraction, the composition and flow rate of the resulting currents; Another case is that of a changing simple heat where you want to know the amount of energy moving, for which the server properties provides the value of the enthalpy at the inlet and outlet of the exchanger, assuming you know the value of the pressure and temperature.

1 For more information see Section 2.4.2

## 1.2 Structure of thesis

At **Episode 2** A review of the history of chemical simulators of different styles of solving chemical process simulators processes performed also describes the different sections that make up the simulators.

At **Chapter 3** A review of the basic definitions phase equilibrium is made and the mathematical models used to calculate the phase equilibrium, and describes some of the qualities of these models are reviewed.

The application of the state equations for calculating the phase equilibrium as well as the consideration taken into account in developing algorithms to OllinTS program component are described in **Chapter 4,** in addition to the commands that were created to interact with the server properties they are explained.

For server capabilities developed properties, and shows how it can be used to develop other programs in specific applications during **Chapter 5.**

Finally, conclusions and recommendations for future work on the area of phase equilibrium and especially on OllinTS are cited.

# 2 Overview

## 2.1 History of simulation

Simulators chemical processes are the result of the union of idea and efforts, that through computer programs tasks that would typically perform chemical engineer manually are made also has the advantage that using computer programs saves time and errors caused by human error during the execution of calculations manually are avoided. Through Table 4 a chronological history analysis divided by decades of development of simulators chemical processes is performed.

*Table 4. Historical Timeline simulating chemical processes [3]*

| | |
|---|---|
| **1950-1959** | |
| **1950** | Rose & Williams [3] written in machine language program the first column binary multi-stage distillation. |
| **1953** | H. Hachmuth [4] suggests represent a fractionator, a diagram expressing fractions of products in the dome and bottom, of the feed components |
| **1957** | MF Nagiev [5] developed the mathematical theory of material balances in chemical processes complex and multiple recirculation state.<br>Donald I. Rubin [6] is the energy and material balances of a chemical system to permanent regime in three stages, which are: formulation node-current relationship, determining the optimal solution sequence the node, and calculations leading to a balance. |
| **1960-1969** | **"It was automation"** |
| **1961** | MA Vela [4] introduced the concept of fractions as "the proportion of a single component which is separated from a stream for some operation"; for troubleshooting recirculation. |
| **1962** | Rosen EM [7] describes the material balance as the solution of a large nonlinear system of simultaneous equations or as a problem of nonlinear programming. He introduced the distribution coefficients for problems with multi-stage phase separation. |

*Table 4. Historical Timeline simulating chemical processes (continued)*

| **1960- 1969** | **"It was automation"** |
|---|---|
| **1963** | MG Kesler and PR Griffiths [8] describe the method of direct substitution for solving simulation calculations in chemical processes. |
| **1964** | LM Naphtali [9] presents a method for solving simultaneous equations representing material balances with or without heat balances for an arbitrary process.<br>Sargent and RWH AW Westerbeg [10] made a description of the program general purpose computer SPEED-UP[2] for the analysis and design of complex processes. |
| **1965** | HA Lindahl [11] explains how to prepare and use simulation to define a process represented in terms of the variables of the equipment. |
| **1966** | Wooyoung Lee, James H. Dale F. Christensen and Rudd [12] presents original works for the selection of design variables and simplification of processes and calculations propose four algorithms that are adaptable to simulation systems.<br><br>Dr. Wang YL and A. Paul Oleston [13] form a society to develop a distillation scheme known as SP03 (Separation Program 3) |
| **1967** | In articles published by RW Sargent [14] makes a literature review of progress in modeling and optimization of processes that have been developed and discussed aspects of input languages for the user in general purpose programs, organization of calculations and the integration of optimization methods. |
| **1968** | Shoei Komatsu [15] gives a description of the mathematical model linearization applied to plant design hydro alkylation.<br><br>LB Evans DG Steward and CR Sprague [16] review contemporary literature and fall into three general areas design for computers: structure data specific problem solving technique calculations thereof and the user's communication with the system. |
| **1970-1979** | **"It was the generalization"** |
| **1973** | And M. L. K'hat Shacham [17] is a review of literature relating to the simulation of chemical processes, by classifying the programs for specific and general simulation, also classifies management methods for calculating recirculation systems.<br><br>Pho and L. Lapidus [18] propose a basic algorithm break (BTA) for recirculating, which identifies and eliminates unnecessary current directly on a diagram, conducting a reduction of the problem and solution algorithms to present the optimal design of heat exchanger networks. |
| **1974** | the first simulator FLOWTRAN chemical processes [19] on the market. |
| **1975** | RL Motard, M. Shacham and EM Rosen [20] make a review of the various techniques for process simulation and note that research has focused on the dynamic simulation and process optimization. |
| **1980-1985 "It was user"** | |
| **1981** | programs simulation of chemical processes for the PC (DESIGN II, ASPEN, SIMSCI [PROII], HYSIM, and CHEMCAD) [21] are released. |

2 In computing, speed-ud refers to the faster is a parallel algorithm with respect to its algorithm corresponding sequential.

*Table 4. Historical Timeline simulating chemical processes (continued)*

| | |
|---|---|
| **1986-1989** | **"It was integration"** |
| | Some providers simulation programs provide the ability to add own home or modules programmed in FORTRAN [22]. |
| **1990-1999** | **"The era of granulation: based software components"** |
| **nineteen ninety-five** | CAPE-OPEN project whose main objective is to create standards for interface modules in process simulation programs is based. |
| **2000** | The philosophy of free software begins to penetrate different areas of development that are unrelated to operating systems appear multiple scientific computing projects including open source chemical process simulator Sim42. |
| **2006** | the OpSim project is created, its aim is to develop a process simulator open source server with its own properties. |

Table 4 shows that simulators chemical processes have been developed in parallel with the advances in computers manner. Hardware, operating systems and programming languages are characteristics that define or limit the complexity of the programs and mathematical models used.

## 2.2 General Simulation Models

Physical models are mathematical representations of a phenomenon or for study can be classified by the written description of the system. Although there are many different ways to classify simulation models bearing in mind that at a given time, a single model can belong to more than one classification, presented below [23]:

1. **Physical models.** This are the prototype models and pilot or scale models.

   a) prototype models: These models are most faithfully represent all the attributes of the actual physical system. This is a physical model on which can be experienced in such a way that the risk and uncertainty associated with predicting the development of real physical system is minimal. Examples of these models are the first atomic bomb and prototype aircraft.

   b) pilot or scale models: They are a small size version of the actual physical system and contain many of the attributes of this; pilot models can physically scaled up to industrial level. In these models the principles of similarity prey. Examples of these models are: chemical process plants miniature (pilot plant) scale dams, wind tunnels and architectural construction models.

**2. symbolic models.** Symbolic models are of different types and are listed continuation.

a) Models of block diagrams, flowcharts, signals and transfer functions: these models are widely used in electronic engineering and control.

b) The graphs, tables and monograms: are frequent forms of expression models.

c) functional and procedural models: these models are implemented through algorithms and logic operation sequences. These economic models are included.

**3. Analogue models.** These are an operational version of the actual physical system behavior and is implemented using mathematical analogies of physical systems as distillation towers, pumps and compressors.

**4. Mathematical models.** This is the most common form of representation systems is through one or more mathematical expressions. Examples are equations vapor pressure or heat capacity Reynolds equation describing flow turbulence.

$$re = \frac{\bullet\ dv}{\bullet} \tag{1}$$

The last two kinds of models are the most used today, given the features and versatility of computers and calculators, hence the importance of mathematical models. For the application of symbolic simulation models to build and review the algorithms they are also used.

From the point of view of the phenomena being studied, models can also be classified as deterministic or stochastic.

**5. Deterministic models:** We believe that in which the equations depend on parameters and variables known with certainty, ie that there is no uncertainty or probability laws associated with them.

**6. Stochastic models.** Certain variables are subject to uncertainty, which may be expressed by probability distribution functions. In this case, therefore, also the model results are associated with a probability law (for example the use of graphics and nome grams).

## 2.3 Simulation Styles

Simulation styles can be classified according to:

- Process characteristics (qualitative and quantitative)
- Type of analysis with respect to the (dynamic or steady) time
- Type process to simulate (continuous or batch).
- Method of solving the problem (modular and global).
- Resolution sequence thereof (sequential or non-sequential).

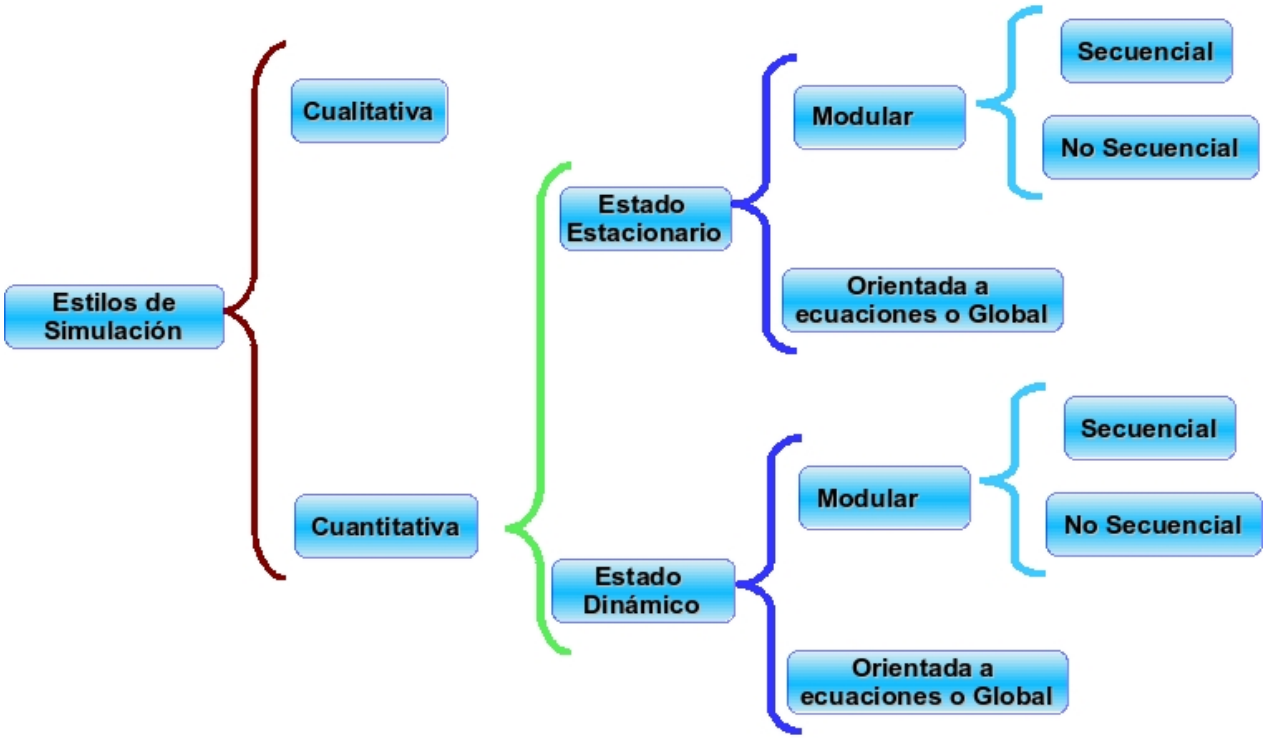



*Figure 1. Simulation Styles*

Then these concepts are expanded.

#### 2.3.1 Qualitative and Quantitative Simulation

The *qualitative simulation*[3] It applied in cases where it is desired to predict the behavior of a variable that has a regular behavior, or predict the propagation of disturbances through a process or system. It is widely used in the field of control and automation as well as safety and fault detection.

3 called qualitative values of a variable, unlike the numerical value (quantitative), its sign; either all or well with respect to a given value or reference. Therefore, generally working with values such as (+, -, 0). [19]

In the *quantitative simulation* numerically it describes the behavior of a process or system, through the mathematical model that represents. Depending on the characteristics simulation energy and material balances are performed, the phase equilibria and thermodynamic calculated. The quantitative simulation mainly include simulation in steady state and dynamic state simulation.

### 2.3.1.1 Simulation steady state

Simulating steady-state or steady state is one in which there is a variation of the variables with respect to time, as they are considered constant, but considering the variation with respect to space being modeled, for example in a heat exchanger it is considered only the temperature variation through the equipment or time. It is also considered that the variables of interest can be analyzed as lumped[4] or distributed parameters[5]. In most simulators lumped models are used.

$$\frac{dT}{dt} = 0 \tag{2}$$

### 2.3.1.2 Simulation dynamic state

In the simulation in dynamic state it is considered that the energy and material balances (consequently the phase equilibria and thermodynamic), with respect to the variation of a variable of interest versus time are made. These may be the concentration of reactants in a batch reactor (batch), the evolution has a disturbance in a chemical process among others. Models representing these systems are built considering differential equations with respect to time. In the case of a heater temperature variation varies with respect to the amount of flow and the heat capacity of the fluid passing through the equipment.

$$\frac{dT}{dt} = \frac{H}{Cp\,F} \tag{3}$$

In turn, the simulation steady state and dynamic can be classified according to the methodology of global and modular solution.

4 considers a variation of the variables of interest in a space, such as the variation of the concentration to along the longitudinal and radial of a tubular reactor. [19]

5 considers the variables of interest is uniformly distributed in space, for example, consider the concentration is uniform in a tubular reactor. [19]

### 2.3.1.3 Methods solution simulators

Solution methods simulators is divided according to the way in which models representing the process or system to simulate arise.

1. Global simulators or oriented equations
2. modular simulators (sequential, non-sequential)

#### 2.3.1.3.1 oriented global Simulators or equations

Under the approach *global simulation* or *oriented equations,* the mathematical model representing the process building a large simultaneous equations representing the whole plant or to simulate arises. Thus the problem solving results in a large simultaneous equations, so highly nonlinear general. As an example may be mentioned that typical simulation problems distillation columns rigorous methods where the system of equations may contain more than a thousand variables.

The main problem associated with global resolution or philosophy oriented equations is the system convergence and consistency of the solutions to be found. Systems are highly nonlinear as corresponding to models of chemical plants which can, for example, to produce multiple solutions. In addition, the numerical solution for large systems requires appropriate initializations, ie close to an environment of the solution, otherwise it may present inconsistencies or may not come to converge [19] .Históricamente, these difficulties have been the cause that has limited the development this type of simulators. plus the size of the matrix constructed requires large memory capacities and processor were not readily available at first.

An important advantage is that a quadratic rate of convergence can be achieved, since there is only one level of iteration. In addition, since the system raises oriented equations, it is possible to incorporate expressions restriction to define optimization problems directly. The following summarizes the most important features of global simulators.

##### 2.3.1.3.1.1 Main features of global simulators

1. Each team is represented by equations that model. The model is the integration of all subsystems.

**2.** the distinction between process variables and operating parameters disappears, therefore simplify design problems.

**3.** simultaneous solution of the resulting set (nonlinear) algebraic equations.

**Four.** Faster convergence.

**5.** You need better initialization (the bigger the problem to be solved).

**6.** A greater complexity, lower reliability in the results and convergence problems (Unphysical solutions).

**7.** Plus difficult use by "non-specialists".

**8.** Three levels iteration

i. calculations physiochemical.

ii. Resolution matrix.

iii. optimization

### 2.3.1.3.2 modular Simulators

In the *modular simulators* for each unit operation a mathematical model is proposed which is programmed as a module in which the code is optimized for each particular case, these modules are approximately the same philosophy as the unit operations, that is, each team pump, valve or heat exchangers.

In these simulations a flowchart (flowsheet) process which translates to a similar diagram, flowchart call information (DFI) is created [ 19]. This diagram is mathematically a digraph, in which nodes are connected equipment modules one by one. There is also the possibility of creating new operations by operations that are programmed into the simulator.

In the modular philosophy, when the modules are oriented physical flow systems, to build the flow chart it is a sequential simulator. Furthermore a nonsequential simulator is one in which unit operations are resolved when determined to have sufficient information and results are propagated back and forth in the flowchart, if it contains recycles will be necessary to have a procedure iterative calculation to solve the balance of the entire process. This simulation method is slower than oriented equations below summarizes the most important features.

### 2.3.1.3.2.1 Main features of the modular simulators

**1.** Library modules (equipment)

**2.** Diagram from flow (Flowsheet) equivalent to one graph or digraph oriented

**3.** Order fixed resolution (iterations)

**Four.** Four levels iteration

i. Physico-chemical calculations.

ii. Modules (Operations and unit processes).

iii. Iteration variables (recycles).

iv. optimization

**5.** methods from robust convergence (Direct Replacement, Wegstein, etc).

**6.** The information entered by the user is easy to correct and interpret as it relates with equipment or streams.

**7.** The problems design (selection parameters) are more difficult to resolve.

**8.** Little bit versatile, but very flexible, very reliable and very robust.

## 2.4 Components of a simulator

To describe a process simulator will need to know three main aspects that can be classified as follows:

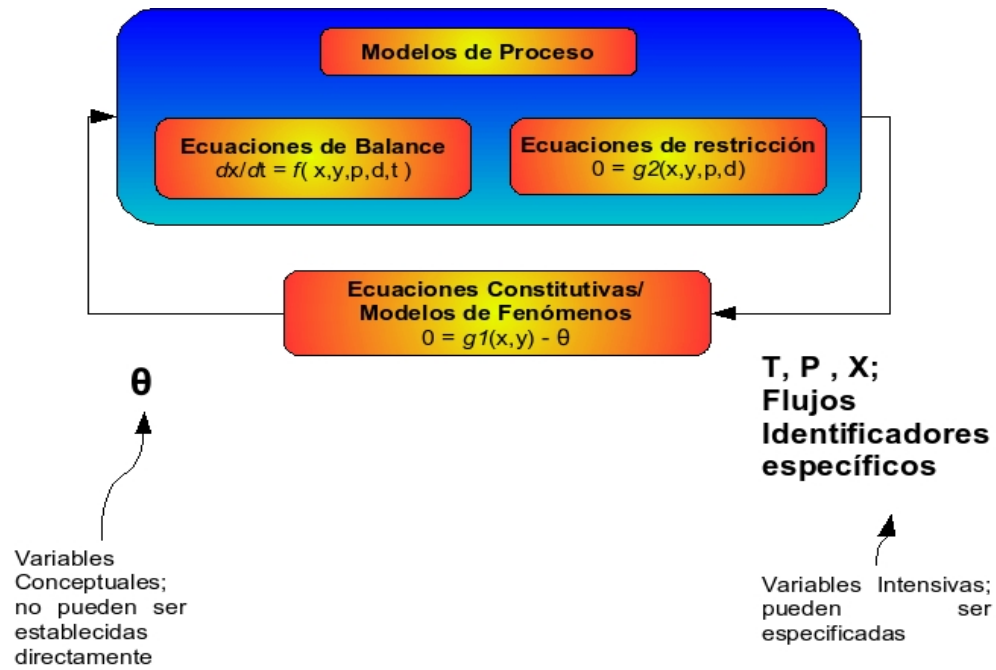
➢ Process data: They are all related data simulation was carried out by such data streams and characteristics of the equipment to simulate turning out to be the best way to check for inconsistencies, storage and its location and the possibility that data can move from one place to another within the program when required .

➢ Engineering or mathematical models: This includes the design of accurate mathematical models physical equipment and establishing correlations for calculation of thermodynamic and physical properties.

➢ Numerical methods or algorithms: These are required in the calculations and recirculation also in the iterative solutions of the most complex units. the use of optimization methods is also necessary to carry out the order of execution of the modules.

From the standpoint of Russel [24], mathematical models product / process typically consists of three types of equations: balance equations, equations constitutive equations and restrictive.

Where the process diagram represents the balance equations, the model represents the thermodynamic properties of constitutive equations group while the equilibrium conditions are imposed through restrictive equations. See Figure 2.




*Where: x = vector of real intensive variables, y = whole specifications, design specifications p = (optimization), d = entry specified variables (optimization) = Vector of constituent variables*

***θ***

*Figure 2. Relationship of mathematical models*

### 2.4.1 Structure of a general purpose simulator

The structure of a general purpose simulator is closely related to the types of models used for their construction, so that simulators consist of perfectly definable independent sections:

**1.** The central logic or general logic simulator also known as administrative section or executive. This can be further divided into:

to) Input Section

b) Output section results

c) General logic section

Administrator functions are:

➢ The analysis of the degrees of freedom globally and per team.

➢ Determining the sequence of calculation (linear or recirculation) depending on the flow chart DFI information[6].

➢ The input section functions as a user input interface. In most commercial simulators this is done by representing the flowchart of process that is equivalent to the real plant, and transformed into a flow of information.

➢ The output section shows the user the results to be understandable, so graphics or tables, and if displays errors and inconsistencies that occur in the data entered by the user or the iteration process also provides the ability to save the results in a database and if necessary perform a chart with respect to a variable of interest.

➢ The administrator can also perform the integration of the results, a program equipment design (CAD).

1. The section responsible for the estimation of physicochemical properties or server thermodynamic properties, contains a library that provides applets representing the behavior of the compounds used during the simulation. This section is the most time-consuming computing and presenting greater difficulty scheduling.

2. -es library unit operations each of said modules representing behavior valves, exchangers, distillation columns, splitters, flash or COMPRESSORS- provides methods for representing the behavior of each of the unit operations in modular style simulation cases that can occur usually optimized.

3. Mathematical tools section provides algorithms, as may be the numerical methods and matrix operations, to solve the thermodynamic or unit operations models.

6 Information Flow Diagram is the interpretation of the simulator for Process Flow Diagram.

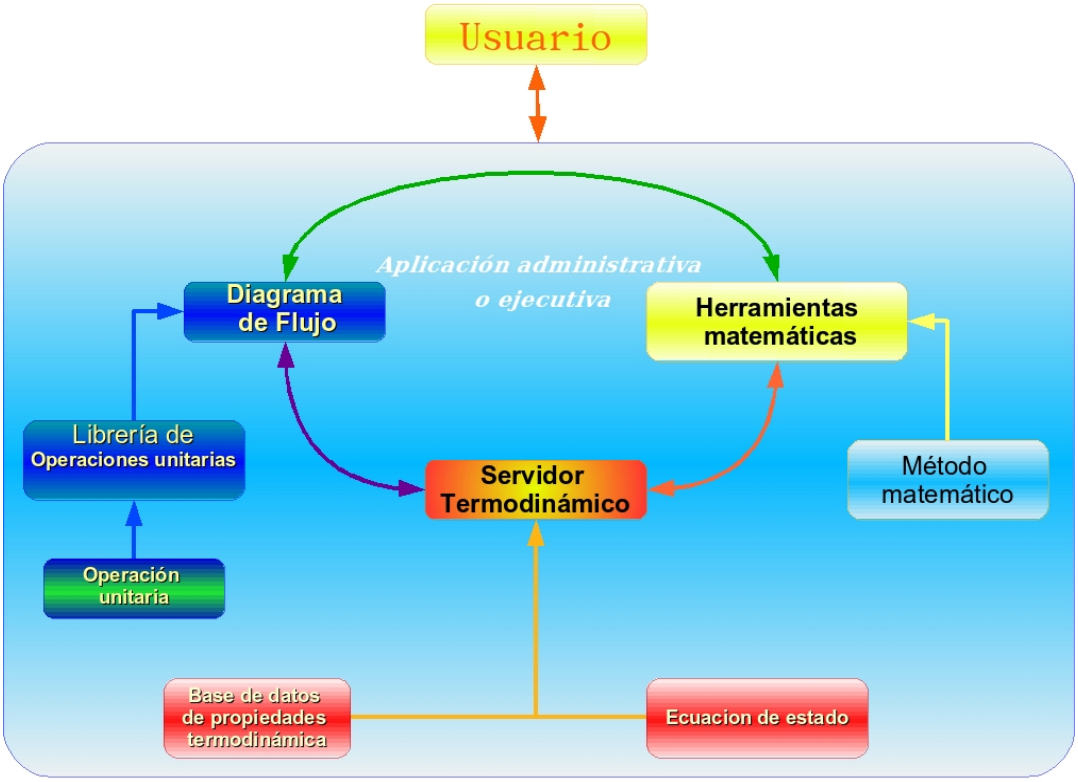



*Figure 3. Typical structure of a simulator*

The way in which the simulator modules interact, determine its internal architecture that can be:

1. Fixed structure: here the different sections that make up the simulator are already defined and integrated, ie the executive program can not be replaced or modified, or a third party can not build extra modules to integrate the simulator.
2. Variable structure: the user determines that he is reliable supplier or fits to their needs, for this there are standards for communication between different modules.

### 2.4.2 Interface of the basic components of a simulator

Object-oriented programming and interface development[7] standard such as COM and CORBA for the Windows operating system for UNIX-based operating systems enable software developers to create applications focused on specific areas of process simulation [25].

In the practice of building a simulator of chemical processes from binding of applications

7 Interface: in the context of software refers to the specific nomenclature and type of parameters used in the definition of the methods set out a program.

different developers or suppliers, is known as "core components simulation" [26]

The advantages of applying basic components are:

- Companies can focus on specific points of the market can develop more robust and reliable applications.
- Companies can acquire the best simulation tools that best suit their specific needs.
- They can adhere models of development or home to commercial applications
- Changes can be made in specific components without changing the target application. disadvantages:
- Do not there are real standards for component interfaces chemical engineering [27]. In other words, the various components must first solve the communication channels to exchange information.
- Sometimes when a component is applied the interface may cause design problems cause interference with the functions of the simulator. This type of problem is difficult to overcome, because the different software manufacturers do not provide the source code of their applications; therefore the only entity with the probability of making changes is the manufacturer.

The resolution of communication between different applications is solved by standardizing interfaces of the different components. To this this type of application is called "Open Architecture Programs" ( "open software architecture [8] ").

The currently known for the standardization of the interface component project is the CAPE-OPEN [28] project. CAPE for its acronym in English "Computer Aided Process Engineering", Lien & Perris [29] defines it as: "CAPE is the application of a modeling system that was used to study processes as an integrated whole, from the point of view development, design and operation "

CAPE-OPEN project is an effort involving companies, academic institutions and software manufacturers. Part of the funds for this project are provided by the European Economic Community. CAPE-OPEN is not the only project of standardization; There are also other initiatives such as OPC and pdXi [30].

8 explains the term "Open Architecture Programs" is not related to the term "open source software".

The first step towards the standardization of interfaces is the definition of basic objects that make up a process simulator. A typical components including division unit operations, thermodynamic unit, logic unit and user interface. These areas are covered by the "CAPE-OPEN" project.

## 2.5 Steps for developing a process simulator

Simulators chemical processes are developed based on mathematical models representing the behavior of fluids and plant equipment process, for which are needed to establish the mathematical relationships of the behavior of properties characterizing the fluids with respect to other measurable properties obtained through the laboratory or directly process plant. The process for constructing mathematical models can be divided according to the scientific method in the following steps:

**1.** Note: At this stage the observation and measurement of the phenomenon takes place in space where it occurs, system performance data are collected to construct the mathematical model

**2.** Hypothesis During this stage the initial mathematical model is constructed based on the data collected.

**3.** Experimentation: calculating the parameters defining the mathematical model is performed from laboratory data obtained by means of the model formulated and laboratory data error calculation is performed to demonstrate that the model is correct formulated.

**Four.** Theory: Having validated the mathematical model, we proceed to compare the results of model with respect to data of the process plant, proceed perform mathematical model corrections and recalculate data for the model predictions are more accurate.

**5.** Law: Finally with the validated mathematical model is necessary to define its limits application using data from the process plant, to be used during plant operation process, operator training me in designing control systems.

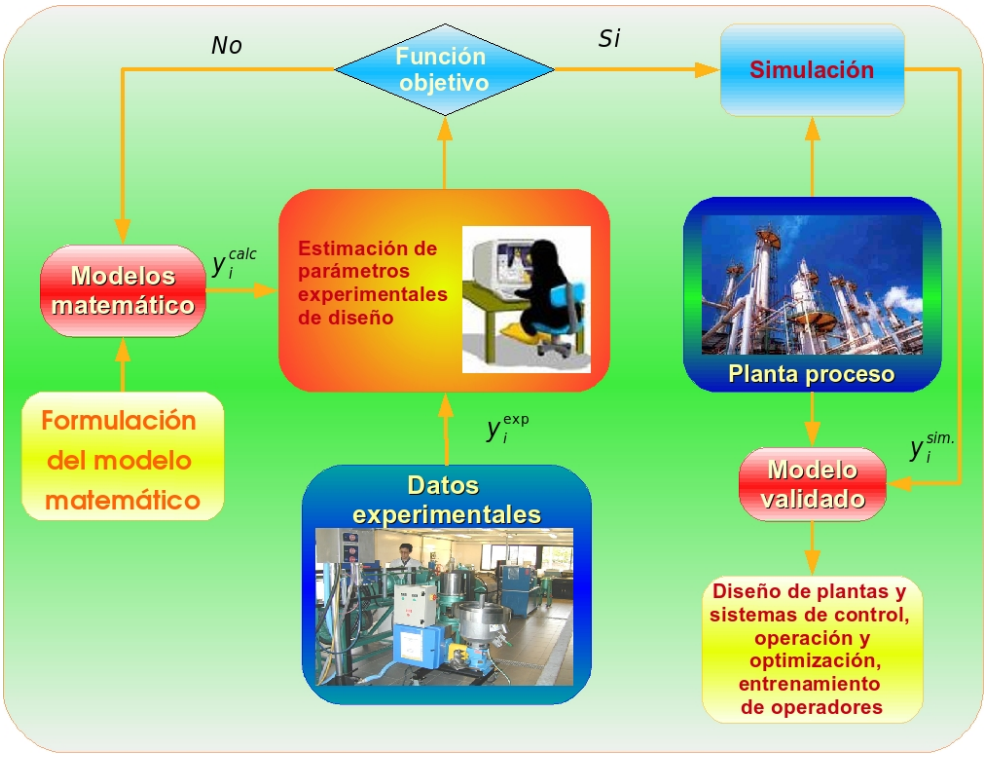



*Figure 4. Development Process simulators chemical processes*

As described in this chapter, chemical process simulators are developed based on mathematical models for unit operations and process fluids. Mathematical models of unit operations mostly need information on the properties of the fluids they work, these properties are estimated, for example by state equations described in the next chapter.

# 3 thermodynamic properties

Thermodynamic properties ( *eg* enthalpy, entropy, temperature) of any substance depend on the nature of the molecules thereof [ 19]. The thermodynamic properties of mixtures are usually calculated in a steady state phase and / or chemical. Calculating the phase equilibrium and chemical equilibrium are the most important problems in the simulation of chemical [31] processes, due to the difficulty of performing these calculations in complex mixtures and the importance values of the concentration of each of the phases considered in the values of thermodynamic properties, or if one takes into account a case where a gas phase reaction where consider simultaneously the chemical and phase equilibrium.

The usefulness of methods for estimating the equilibrium of phases present and used in this work is the ease with which to estimate the thermodynamic properties of mixtures from experimental parameters that are based on readily measurable properties, such as pressure and the temperature. These methods use mathematical models representing the behavior of the substances from experimental statistical parameters that are available in a database, so they are easily programmable.

### 3.1 Chemical Equilibrium

The chemical equilibrium is observed when the concentrations of two or more substances remain constant, ie, the chemical equilibrium occurs when the moles do not change. Initially, the reactants combine to form products, but there comes a time when the number of products is large enough so that they react together to form the initial becoming reactive. Thus two reactions simultaneously pass, forward and reverse, equilibrium is reached when the reactants are transformed into products with the same speed with which the products again become reactive.

When steady state is reached, the concentrations of the reactants and products are in a constant numerical relationship. Experimentally it is found that the concentrations of the substances involved in an equilibrium system are related by the following mathematical expression:

$$aA \bullet bB \Leftrightarrow DC \bullet dD \tag{4}$$

$$K = \frac{[C]_{c[} D]_{d}}{[ B]_{b[} TO]_{to}} \tag{5}$$

In this expression the numerator is the product of the equilibrium concentrations of the products, each raised to the exponent of the number of moles of each product shown in the chemical equation

The denominator and numerator are the product of the equilibrium concentrations of the reactants, each raised to an exponent equal to the number of moles of each reagent shown in the chemical equation.

The ratio K between the numerator and the denominator is the equilibrium constant of the reaction. If the value of K is less than 1, the direct reaction occurs only to a small extent before reaching equilibrium. That is, the reagents do not react at all. However, if the value of K is greater than 1, the direct reaction is strongly favored; this is,

the original reagents become

products.

It is concluded that:

- If K <1, the formation of reagents is favored.
- If K> 1, the product formation is favored.

## 3.2 Phase Equilibria

The phase equilibrium is reached when conversion rates between two two phases are the same in each direction. Speeds are equal when the molar free energy of the substance is the same in each stage and therefore no tendency to change in any direction.

The first method estimates the nonideal behavior of each phase by state equations (Redlich-Kwnog, Peng-Robinson) and the second method predicts the vapor-liquid behavior from the estimation: To estimate the phase equilibrium two methods are known of the liquid phase, either by activity coefficients models or models that provide liquid phase

functional relationship for the same (UNIFAC[9] UNIQUAC[10]).

### 3.2.1 equilibrium systems

In a closed heterogeneous system, consisting of n components π phases, each phase can be regarded as an open system with respect to the overall system. It is considered that the system is in equilibrium if:

$$T_{I} = T_{II} = \ldots = T \quad (6)$$

$$P_{I} = P_{II} = \ldots = P \quad (7)$$

$$\mu_{I} = \mu_{II} = \ldots = \mu \quad (8)$$

This means that when the temperature, pressure and chemical potential are equal in each of the phases the system is in equilibrium.

The criteria for balancing systems can be expressed by an equality between the fugacity of each phase for each of the components of a mixture.

$$f_{iv} = f_{iL} \quad (9)$$

This equality establishes the criteria by which the phase equilibrium in mixtures is estimated.

### 3.2.2 Fugacidad

fleetingness • $f$ • It is a mathematical equivalent pressure which replaced in the equations of ideal substances makes them valid for actual behavior. A ratio of the fugacity between the vapor pressure of a substance is known as fugacity coefficient •••

, Which can be calculated using the following equations are expressed in terms of composition, pressure, volume and temperature properties obtainable by cubic equations of state.

$$RT \ln \phi_i = \int_V^{\infty} \left[ \left( \frac{\partial P}{\partial n_i} \right)_{T, n_{j \neq i}} - \frac{RT}{V} \right] dV - RT \ln Z \quad (10)$$

$$\phi_i = \frac{f_i}{P_{Saw}} \quad (\text{eleven})$$

$$f_i = x_i \phi_i P \quad (12)$$

9 It is currently the most convenient method for evaluating the activity coefficients in mixtures standardized method liquids

10 UNIQUAC model is used to calculate the excess free energy and activity coefficients of gases and liquids, systems used in less than ideal quite good result

### 3.2.3 Activity coefficient

The activity coefficient ••• It is the mathematical equivalent of the molar concentration, is a non-ideality correction of the compounds in the mixture. The fugacity coefficient can be calculated by a ratio of the fugacity coefficients.

$$\bullet_{i} = \frac{\bullet_{i}}{\bullet_{io}} \qquad (13)$$

To describe a relationship of composition between the liquid phase and gas distribution coefficient is used • $K_i$•

It can be calculated by various methods, as expressed in the following equation.

$$K_i = \frac{Y_i}{X_i} = \frac{\bullet_{iL} F_{i,L\ or}}{\bullet_{iL} P = \bullet_{iL}} \qquad (14)$$

When considering that the mixture behaves ideally, you can equal the activity coefficient ••• and fugacity coefficient ••• 1, then we can estimate the distribution coefficient as a ratio of the vapor pressure of each component of the system pressure.

$$K_i = \frac{P_V}{P} \qquad \text{(fifteen)}$$

### 3.2.4 Vapor pressure

It is defined as the pressure exerted by a vapor in equilibrium with its liquid or solid that originates at a certain temperature; its value is independent of the amounts of liquid and vapor present while simultaneously exist. The vapor pressure of a pure compound can be calculated via hyperbolic equations Antoine and Frost-Kalkwarf-Thodos described below.

#### 3.2.4.1 Antoine Equation

The correlation of Antoine (1888) is a hyperbolic equation for estimating the vapor pressure of a pure substance.

$$\ln \bullet P_V \bullet = TO - \frac{B}{T \bullet C} \qquad (16)$$

Where A, B and C are the coefficients in tables available for different units of pressure and temperature.

### 3.2.4.2 Equation Frost-Kalkwarf-Thodos (Harlacher)

Frost-correlation Kalkwarf-Thodos is a correction Antoine equation, calculates the more accurately the value of the vapor pressure. Although needs more parameters .:

$$\ln \bullet P_V \bullet = TO - \frac{B}{T \bullet C \ln \bullet T \bullet \bullet D \bullet \frac{P_V}{T_2} \bullet} \tag{17}$$

Wherein A, B, C and D are the coefficients available in tables for different units of pressure and temperature.

### 3.2.5 Thermodynamic properties

The thermodynamic properties of state quantized energy state and aggregation of substances, for calculation can be divided into 4 parts: Lead State • $h_{ref}$ • , contribution ideal • $h_{or}$ • , internal residual energy • $or$ • $_{beef}$ • and the depending portion to the compressibility factor • $h_{\cdot z}$ • [32]. Based on the case of the enthalpy we have:

$$h = h_{ref} \bullet h_{or} \bullet \bullet \; or_{beef} \bullet h_z \tag{18}$$

For this work it is considered that at T = 0 K and P = 1 atm values $h_{ref}$Y $or$ • $_{beef}$may considered null, therefore only you have the ideal contribution and the dependent part of the compressibility factor, the residual considered. Then the thermodynamic properties common status are defined and corrections for calculation are set.

## 3.2.6 Heat Capacity

This property indicates the degree of difficulty of a substance to undergo changes in temperature under the supply of heat. It can be interpreted as a measure of thermal inertia.

$$C_p = to \bullet bT \bullet cT_2 \bullet dT_2 \tag{19}$$

The corrected heat capacity is expressed by the following equation.

$$\bullet \quad C_p = C_p^{real} - C_{p\,ideal} \tag{twenty}$$

Where a, b, c, d are constants for the equation of the heat capacity available in tables,

- $C_P$It is the contribution of the residual heat capacity.

### 3.2.7 Enthalpy

Enthalpy is the amount of energy of a thermodynamic system that it can exchange with their environment. The contribution of the ideal enthalpy can be calculated by the following equation:

$$H \bullet T \bullet_{Ideal} = \int_{T_{or}}^{T} Cp_i \bullet T \bullet dT \qquad \text{(twenty-one)}$$

To calculate the enthalpy of a gas phase mixture using the following equation:

$$H_V \bullet T \bullet_{Ideal} = \sum_{i=1}^{n} Y_i \int_{T_o = 0\,°K}^{T} Cp_i \bullet T \bullet dT = \sum_{i=1}^{n} Y_i \bullet to_i T \bullet \frac{b_i T^2}{2} \bullet \frac{c_i T^3}{3} \bullet \frac{d_i T^4}{4} \bullet \qquad (22)$$

Considering the correction of the residual can be written:

$$\bullet \quad H_V = H_{Vactual} - H_{Videal} \qquad (2.\ 3)$$

To calculate the enthalpy of the liquid phase mixture, a correction of the enthalpy of gas phase median value of heat of vaporization and the critical temperature is realized.

$$H_L \bullet T \bullet_{Ideal} = \sum_{i=1}^{n} x_i \int_{T_o = 0\,°K}^{T} Cp_i \bullet T \bullet dT - HV_{li} \bullet \frac{T - T_{Ci} \bullet}{\bullet T_{Bi} - T_{Ci} \bullet]^{0.38}} \qquad (24)$$

Applying the correction of residual property we have.

$$\bullet \quad H_L = H_{LReal} - H_{lideal} \qquad (25)$$

Where $HV_i$ It is is latent heat of vaporization, $T_C$ It is the critical temperature and $T_B$ is the Normal boiling temperature.

### 3.2.8 Entropy

It describes the irreversible entropy or the degree of disorder that have molecules that comprise thermodynamic systems.

$$S \bullet T \bullet_{Ideal} = \int_{T_{or}}^{T} \frac{Cp_i \bullet T \bullet}{T\, dT} \qquad (26)$$

The entropy contribution to the gas phase is calculated by the following equations:

$$S \bullet T \bullet_{Saw}^{Ideal} = \int_{T_o = 1\,K}^{T} \frac{Cp_i \bullet T \bullet}{DT\, T} = \bullet to_i \log \bullet T \bullet\bullet b_i \bullet T - 1 \bullet\bullet c_i \bullet \frac{T - 1 \bullet^2}{2} \bullet d_i \bullet \frac{T - 1 \bullet^3}{3} \bullet \qquad (27)$$

$$S_{Videal} = \sum_{i=i}^{n} S_{Saw}^{ideal} Y_i \bullet R \sum_{i=1}^{n} Y_i \ln \bullet Y_i \bullet \qquad (28)$$

$$\bullet \quad S_V = S_{Vactual} - S_{Videal} \qquad (29)$$

To calculate the value of the entropy phase liquid:

$$S \cdot T \cdot_{Li}^{Ideal} = \sum_{i=1}^{n} x_i \int_{T_o = 1\,K}^{T} \frac{Cp_i}{T}\, dT - \frac{h_{Saw}}{T} \cdot \frac{T}{\left[\cdot \cdot T - B_i T - C_i T \cdot C_i \cdot\right]^{0.38}} \tag{30}$$

$$S_{Ideal\,L} = \sum_{i=i}^{n} S_{Li}^{ideal}\, x_i \cdot R \sum_{i=1}^{n} x_i \ln \cdot x_i \cdot \tag{31}$$

- $S_L = S_{LReal} - S_L^{ideal}$ (32)

### 3.2.9 Internal energy

L to internal energy of the material or system is the result of the kinetic energy of the molecules or atoms constituting it, of energies of rotation, translation and vibration, in addition to energy intermolecular potential due to the forces of gravity type , electromagnetic and nuclear, which together constitute the fundamental interactions.

$$U = H - PV \tag{33}$$

To determine the contribution of domestic energy in gas phase we use the following equations.

$$OR_{Videal} = H_V - PV_V \tag{3. 4}$$

- $OR_V = OR_{Vactual} - OR_{Videal}$ (35)

Calculating the contribution of the internal energy the liquid phase is made to the following equations:

$$OR_{lideal} = H_L - PV_L \tag{36}$$

- $OR_L = OR_L^{real} - OR_{Lideal}$ (37)

### 3.2.10 Gibbs free energy

The Gibbs free energy is a thermodynamic potential, ie a function of extensive state with power units, represents the equilibrium condition and spontaneity of a chemical reaction (at constant pressure and temperature). Its formula is:

$$G = H - ST \tag{38}$$

The contribution of Gibbs free energy of the gas phase is expressed mathematically as:

$$G_{Videal} = H_V - S_V T \tag{39}$$

- $G_V = G_{Vactual} - G_{Videal}$ (40)

For the liquid phase has:

$$G_L^{Ideal} = H_L - S_L T \quad (41)$$

- $$G_L = G_L^{real} - G_{L\ ideal} \quad (42)$$

### Helmholtz free energy 3.2.11

The Helmholtz free energy is calculated by the following equation.

$$A = U - ST \quad (43)$$

Calculating the Helmholtz free energy of the gas phase it is made by the following equations:

$$TO_{Videal} = OR_V - S_V T \quad (44)$$

- $$TO_V = TO_{Vactual} - TO_{Videal}$$ ( Four. Five)

For the liquid phase has:

$$TO_L^{Ideal} = OR_L - S_L T \quad (46)$$

- $$TO_L = TO_L^{real} - TO_{lideal} \quad (47)$$

## 3.3 Equations of State

The state equations representing the state of matter according to their intensive properties such as temperature, pressure, molar volume, density, among others. The most important use of an equation of state is to predict the state and the thermodynamic properties of gases and liquids. The state equations apply to normal fluids (rare gases, nitrogen, oxygen, carbon monoxide and hydrocarbons), carbon dioxide, hydrogen sulfide, hydrogen and slightly polar substances also can work at high pressures [33].

Some advantages of the state equations based on the equations activity coefficients are:

1. continuity in the critical region is assured, although it is difficult to achieve convergence.
2. They can be calculated a variety of properties [34].

Among the most important properties that can be calculated using the equations of state are:

- Compressibility factor
- fleetingness
- Vapor pressure

- Density
- State properties
  - A-Aº, Helmholtz free energy
  - S-Sº Entropìa
  - H-Br: Enthalpy
  - U-Uº: Internal Energy
  - G-Gª: Gibbs free energy

Although the state equations has its limitations, as they generally fail in predicting liquid density and can not be applied to:

- polar compounds
- Large molecules (polymers)
- electrolytes

The first equation which registry is the Van der Waals equation, which was introduced in 1873 as a semi theoretical development of the ideal gas equation, the equation is as follows:

$$P = \frac{RT}{V - b - \frac{to}{V^2}} \qquad (48)$$

The Van der Waals equation is classified as a cubic equation of state which are derived corrections considering many factors such as the interaction of molecules to be applied to a greater number of cases. Where the constant *"B"* It is a correction for the volume occupied by the molecules, and the term $a / v_2$ is a correction that takes into account the intermolecular forces of attraction.

The terms " *to* " Y " *b* "They are obtained by observing the general behavior of gases on the critical isotherm, passing through the turning point in the critical point. For the Van der Waals equation can mathematically express this condition as follows.

$$\left(\frac{dP}{dV}\right)_T = 0 = \left(\frac{d_2 P}{d_2 V}\right)_T \qquad (49)$$

simultaneously solving both equations for " *to* " Y " *b* ", Y applying this condition obtained based on the critical properties:

$$a = \frac{27}{64} \frac{R_2 T_{c2}}{P_c} \tag{fifty}$$

$$b = \frac{RT_c}{8P_c} \tag{51}$$

The numerical value of the compressibility factor at the critical point for the Van der Waals equation is defined by the following equation.

$$Z_c = \frac{P_c V_c}{RT_c} = \frac{3}{8} \tag{52}$$

In this paper we consider only the equations described below.

## 3.4 Cubic Equations of State

Cubic equations of state are very popular in the industry because of its ability to give good results for a large number of components relatively easily. [32]. Recommended for non-polar low molecular weight compounds. Most of these equations are corrections of the Van der Waals equation, which can be generalized as follows:

$$P = \frac{RT}{V - b} - \frac{to\, V_2 - UBV \bullet}{wb_2} \tag{53}$$

Where:

$$a = a_c \bullet\bullet T \bullet \tag{54}$$

*Table 5. Models of the state equations*

| ***Model*** | ***or*** | ***w*** | $to_c$ | ***b*** |
|---|---|---|---|---|
| RK | 1 | 0 | $\bullet \frac{_{TO} R_2 T_{ci}^{2.5}}{P_{ci}}$ | $\bullet \frac{_B RT_{ci}}{P_{ci}}$ |
| SRK | 1 | 0 | $\bullet \frac{_{TO} R_2 T_{ci}^{2}}{P_{ci}}$ | $\bullet \frac{_B RT_{ci}}{P_{ci}}$ |
| PR | 2 | - 1 | $\bullet \frac{_{TO} R_2 T_{ci}^{2}}{P_{ci}}$ | $\bullet \frac{_B RT_{ci}}{P_{ci}}$ |

The cubic equation of state can also be represented as a function of the compressibility factor:

$$Z_3 \cdot\cdot 1 \cdot B - uB_2 \cdot Z_2 \cdot\cdot TO \cdot wB_2 - uB - uB_2 \cdot Z - AB - wB_2 - wB = 0 \tag{55}$$

Where:

$$TO_i - \frac{to_i P}{\cdot RT \cdot_2} \tag{56}$$

$$B = \frac{bP}{RT} \tag{57}$$

For the calculation of the residuals from the cubic equation of state the following equations are used:

$$\cdot\ A = \frac{ab \bullet or_2 - \ln 4w\ 2Z \cdot B \cdot \overline{or - \bullet or_2 - 4w \cdot}}{2Z \cdot B \cdot or \cdot \bullet or_2 - 4w \cdot\cdot RT \ln \cdot \underline{Z - B \cdot}} \tag{58}$$

$$\cdot\ S = R \ln \cdot Z - B \cdot\cdot \frac{1}{b \bullet or_2 - 4w} \frac{gives}{dT \ln 2Z - B \cdot or \cdot \bullet or_2 - 4w \cdot} \tag{59}$$

$$\cdot\ H = \cdot TO \cdot T \cdot S \cdot RT \cdot Z - 1 \cdot \tag{60}$$

$$\cdot\ C_v = T \int_{\infty}^{v} \bullet \frac{\partial_2 P}{\partial T_2} \bullet dV - R \tag{61}$$

$$\cdot\ C_p = \cdot C_v - T \bullet \frac{\partial T \bullet_v}{\partial P} - \frac{\partial P}{\bullet \frac{\partial P}{\partial V} \bullet_T}^{2} \tag{62}$$

$$\bullet \frac{\partial P}{\partial T} \bullet_v = \frac{RT}{V - b - gives T} \frac{1}{V_2 \cdot UBV \cdot wb_2} \tag{63}$$

$$\bullet \frac{\partial P}{\partial V} \bullet_T = \frac{RT}{\cdot V - b \cdot_2 - \text{the 2nd}} \frac{}{V_2 \cdot VUBV \cdot wb_2 \cdot_2} \tag{64}$$

$$\int \bullet \frac{\partial_2 P}{\partial T_2} \bullet = \frac{}{ddT_2 to} \frac{1}{b \bullet or_2 - \ln 4w} \frac{}{2Z \cdot B \cdot or \cdot \bullet or_2 - 4w \cdot\cdot} \tag{65}$$

The mechanical properties are calculated by the following equations:

isothermal compressibility.

$$Iso = \frac{1}{P} - \frac{1}{Z} \bullet \frac{\partial P}{\partial T} \bullet_T \tag{66}$$

Coefficient of thermal expansion.

$$\cdot = \frac{1}{V} \bullet \frac{\partial V}{\partial T} \bullet_P \tag{67}$$

### 3.4.1 Redlich-Kwong (1949)

The Redlich-Kwong equation is suitable for calculating the properties of the gas phase when the ratio between the pressure and the critical pressure is less than half the ratio between temperature and the critical temperature $P/P_c \bullet 1/2 \bullet T/T_c \bullet$.

*Table 6 Specifications of equation RedlichKwong*

| $\bullet_{TO}$ | $\bullet_{B}$ | $\alpha(T)$ | $Z_c$ Mechanics |
|---|---|---|---|
| 0.42748 | 0.08664 | $\frac{1}{\bullet \sqrt{T}}$ | $\frac{1}{3}$ |

Mixing rules Van Der Waals:

$$a = \sum\sum \bullet 1 - k_{ij} \bullet \bullet to_i \overline{to_j x_i x_j} \tag{68}$$

$$b = \sum x_i b_i \tag{69}$$

$$k_{ij} = k_{ij}^{or} \bullet \frac{k_{ij}^{1}}{T \bullet k_{ij}^{2}} \ln T \tag{70}$$

Fugacity coefficient for pure compounds:

$$\ln \bullet\bullet\bullet = Z - 1 - \ln \bullet Z - B \bullet\bullet \frac{TO}{B} \ln [1 \bullet \frac{B}{Z}] \tag{71}$$

Fugacity coefficient for mixtures:

$$\ln \bullet\bullet\bullet = \bullet \ln \bullet Z - B \bullet\bullet \frac{B_i}{B} \bullet Z - 1 \bullet\bullet \frac{TO}{B} [\frac{B_i}{B} - 2 \bullet \frac{TO}{TO}] \ln \left( 1 \bullet \frac{B}{Z} \right) \tag{72}$$

Mixing rules:

$$A = \sum\sum \bullet 1 - k_{ij} \bullet \bullet TO_i \overline{TO_j x_i x_j} \tag{73}$$

$$B = \sum x_i B_i \tag{74}$$

$$TO_i = \bullet \overline{aa_i} - \sum_j Y_k to_{ij} = Y_1 to_{i1} \bullet Y_2 to_{i2} \bullet Y_3 to_{i3} \bullet \ldots = 2 \sum_{j=1}^{nc} x_j TO_{ij} \tag{75}$$

First and second derivative of the function "a":

$$\frac{gives}{dT} = - \frac{TO_c}{2T \bullet \sqrt{T}} \tag{76}$$

$$\frac{gives}{dT^2} = \frac{3}{4} \frac{to_c}{T^2 \bullet \sqrt{T}} \tag{77}$$

### 3.4.2 Redlich-Kwong (1949) Simplified

The difference to the standard model equation RK lies in the considerations of the mixing rules for calculating the fugacity coefficients.

*Table 7. Specifications equation RedlichKwong Simplified*

| • TO | • B | α (T) | Zc Mechanics |
|---|---|---|---|
| 0.42748 | 0.08664 | 1 / • T | 1 3 |

Rules mixing Van Der Waals:

$$a = \sum\sum \bullet 1 - k_{ij} \bullet \bullet to_i to_j x_i x_j \quad (78)$$

$$b = \sum x_i b_i \quad (79)$$

$$k_{ij} = k_{ij}^{or} \bullet \frac{k_{ij}^{1}}{T \bullet k_{ij}} {}_{2} \ln T \quad (80)$$

Fugacity coefficient for pure compounds:

$$\ln \bullet\bullet_i \bullet = Z_i - 1 - \ln \bullet Z_i - B_i \bullet\bullet \frac{TO_i}{B_i} \ln [1 \bullet \frac{B_i}{Z_{ij}} \quad (81)$$

Fugacity coefficient for mixtures:

$$\ln \bullet\bullet_i \bullet = \bullet \ln \bullet Z - B \bullet\bullet \frac{B_i}{B \bullet Z - 1 \bullet\bullet \frac{TO}{B} [\frac{B_i}{B} - 2 \bullet \frac{TO_i}{TO}] \ln \bullet 1 \bullet \frac{B}{B} \bullet} \quad (82)$$

Mixing rules:

$$A = \bullet \sum Y_i \bullet \overline{TO_i \bullet_2} \quad (83)$$

$$B = \sum Y_i B_i \quad (84)$$

First and second derivative " *to* ":

$$\frac{gives}{dT = - \frac{\overline{TO_c \bullet T}}{2}} \quad (85)$$

$$\frac{gives}{dT_2 = \frac{3}{4} \frac{to_c}{T_2 \bullet \overline{T}}} \quad (86)$$

**Soave Redlich-3.4.3-Kwong (1972)**

In 1972 Soave replaced the term *to / • T* the Redlich-Kwong equation by expression α (T, ω) as a function of temperature and acentric factor. The α function was designed to reconcile data vapor pressures of hydrocarbons.

*Table 8. Specifications Soave Redlich-Kwong equation-*

| • TO | • B | α (T) | Zc Mechanics |
|---|---|---|---|
| 0.42748 | 0.08664 | •1 • Fwi• 1 - • Tri• •2 | 1 3 |

Mixing rules Van Der Waals:

$$a = \sum\sum \bullet 1 - k_{ij} \bullet \bullet to_i to_j x_i x_j \quad (87)$$

$$b = \sum x_i b_i \quad (88)$$

$$k_{ij} = k_{ij} \quad or \bullet \frac{k_{ij}^{1}}{T \bullet k_{ij}} \quad {}_2 \ln T \quad (89)$$

$$F_{wi} = 0.48 \bullet 1{,}574\, w_i - 0.176\, w_{i2} \quad (90)$$

Fugacity coefficient for pure compounds:

$$\ln \bullet\bullet_i \bullet = Z - 1 - \ln \bullet Z - B \bullet\bullet \frac{TO}{} \quad B \ln [1 \bullet \frac{B}{} Z] \quad (91)$$

Fugacity coefficient for mixtures:

$$\ln \bullet\bullet_i \bullet = \bullet \ln \bullet Z - B \bullet\bullet \frac{B_i}{} \quad B \bullet Z - 1 \bullet\bullet \frac{TO}{} B [\frac{B}{} - \frac{2}{} to \bullet \sum_j Y_j \bullet to \bullet\bullet_{ij} \ln \bullet 1 \bullet \frac{B}{} Z \bullet \quad (92)$$

Mixing rules:

$$A = \sum\sum \bullet 1 - k_{ij} \bullet \bullet TO_i TO_j x_i x_j \quad (93)$$

$$B = \sum x_i B_i \quad (94)$$

$$\bullet to \bullet\bullet_{ij} = \bullet 1 - k_{ij} \bullet \bullet \bullet \overline{aa \bullet_{ii} \bullet aa \bullet_{jj}} \quad (95)$$

First and second derivative of the function " *to* ":

$$\frac{gives}{dT = a_c F_w} \quad \frac{F_w \bullet T_r - F_w - 1}{T_c \bullet T_r} \quad (96)$$

$$\frac{gives_2}{dT_2 = to_c F_w^2} \quad \frac{fw \bullet F_w \bullet 1 \bullet}{T_c T \bullet T_r} \quad (97)$$

### 3.4.4 Peng-Robinson (1976)

The equation of Peng-Robinson was developed in 1976 to meet the following objectives:

**1.** The parameters should be expressed in terms of the critical properties and the acentric factor.

**2.** The model should be reasonably accurate near the critical point, particularly for calculations of the compressibility factor and liquid density.

**3.** The mixing rules should not use more than one parameter of binary interactions also be independent of pressure, temperature and composition.

**Four.** The equation should be applicable to all calculations of all fluid properties in natural gas processes.

Generally equation Peng-Robinson gives similar results to the equation SoaveRedlich-Kwong, but it is much better in predicting the densities of many compounds, especially non-polar liquid phase.

The equation Peng-Robinson as good results as Soave-Redlich-Kwong systems of light and heavy hydrocarbons (> $C_5$ ) and mixtures of nonpolar light hydrocarbons with SH2 (up to 25 mol% $SH_2$) and of $H_2$ with light hydrocarbons.

Near the critical region equation Soave-Redlich-Kwong It provides less accurate results than equation Peng-Robinson, n or is very good l to accurately predict the values of the two equations for the density of liquid, as they are usually 10% to 20% below the experimental values.

*Table 9. Specifications equation PengRobinson*

| • TO | • B | α (T) | $Z_c$ Mechanics |
|---|---|---|---|
| 0.457236 | 0.0777969 | •1 • $F_{wi}$• 1 - • $T_{ri}$• •2 | 0.30740 |

Mixing rules Van Der Waals:

$$a = \sum\sum \bullet 1 - k_{ij} \bullet \bullet to_i to_j x_i x_j \quad (98)$$

$$b = \sum x_i b_i \quad (99)$$

$$k_{ij} = k_{ij}^{or} \bullet \frac{k_{ij}^{1}}{T \bullet k_{ij}} {}_2 \ln T \quad (100)$$

$$F_{wi} = 0.37464 + 1.574226\, w_i - 0.26992\, w_{i2} \qquad if\ w_i + 0.5 \tag{101}$$

$$F_{wi} = .3796 + 1.4850\, w_i - 0.644\, w_{i2} + 0.01666\, w_{i3} \qquad if\ w_i + 0.5 \tag{102}$$

Fugacity coefficient for pure compounds:

$$\ln \phi_i = Z - 1 - \ln + Z - B \phi\ TO \frac{}{2 + 2\, B \ln [\frac{Z \phi 1 + \phi 2 \phi B}{}]} \tag{103}$$

Fugacity coefficient for mixtures:

$$\ln \phi_i = + \ln + Z - B \phi \frac{B_i}{} \quad B + Z - 1 \phi \frac{TO}{} B [\frac{B}{} - \frac{2}{} to + \sum_j Y_j + to \phi_{ij} \ln [\frac{Z \phi 1 + + 2 + B}{Z \phi 1 - + 2 + B}] \tag{104}$$

Mixing rules:

$$A = \sum\sum + 1 - k_{ij} + + TO_i \overline{TO_j x_i x_j} \tag{105}$$

$$B = \sum x_i B_i \tag{106}$$

$$+ to \phi_{ij} = + 1 - k_{ij} + + \overline{aa + ii + aa + jj} \tag{107}$$

First and second derivative of the function " $to$ ":

$$\frac{gives}{dT = a_c F_w} \quad \frac{F_w + \overline{T_r} - F_w - 1}{T_c + \overline{T_r}} \tag{108}$$

$$\frac{gives_2}{dT_2 = to_c F_w^2} \quad \frac{fw + F_w + 1 +}{T_c T + \overline{T_r}} \tag{109}$$

### 3.4.5 Flash Isothermal (vapor-liquid equilibrium)

Flash equipment is the simplest application where the phase equilibrium is presented and to analyze the importance of proper estimation of phase equilibrium in the simulation of chemical processes. In this equipment, it is to determine the value of the vaporized fraction and the composition in each phase. For this work envisage two phases: liquid and gas. Furthermore it raises the pressure and temperature are known, since these considerations can be written material balances for each component and global.

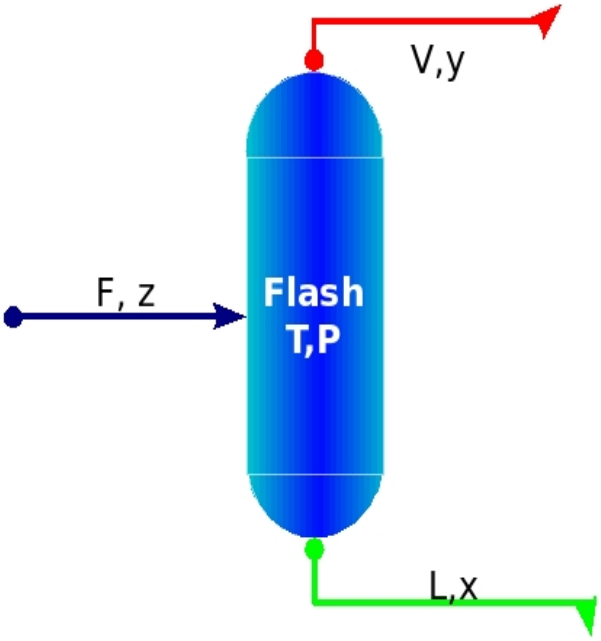
Material balance

$$F = L + V \tag{110}$$

Mass balance by component

$$Fz_i = L\, x_i + V\, and_i \tag{111}$$

Energy balance




$$FH_F \bullet Q = VH_V \bullet LH_L \quad (112)$$

Where:

*F:* molar feed flow

*zi:* molar feed composition

*L:* molar flow of liquid

*xi:* Molar composition of liquid

*V:* molar steam flow yi: steam molar composition

*Q:* Energy flow

$H_{F, L, V}$: Enthalpy stream

*Figure 5. Separator flash*

**If the vaporized fraction defined ••• as the ratio of steam flow ( *V)* on the feed rate ( *F).***

$$\bullet = \frac{V}{F} \quad (113)$$

Subsequently, operating the mass balance for each component and making the appropriate replacements the following equations are obtained:

mole fraction liquid phase.

$$x_i = \frac{z_i}{\bullet \cdot K_i - 1 \bullet\bullet 1} \quad (114)$$

Mol fraction gas phase.

$$Y_i = \frac{z_i K_i}{\bullet \cdot K_i - 1 \bullet\bullet 1} \quad (115)$$

It is assumed that in equilibrium:

$$\sum_{i\,nc} Y_i - \sum_{i\,nc} x_i = 0 \quad (116)$$

Applying the equations 111 and 112 into equation 113 can write the equation Rachford Rice, which allows us to calculate the vaporized fraction and the composition in each phase.

$$\sum_{i}^{nc} \frac{\cdot K_i - 1 \cdot z_i}{\cdot \cdot K_i - 1 \cdot\cdot 1 = 0} \tag{117}$$

How cubic equations of state and the equation Rachford Rice integrated to predict phase equilibrium is described in the next chapter. In addition the characteristics of the program developed in this work as well as the considerations taken into account when selecting the programming language and during program development, distribution modules that comprise the program are described final.

# 4 OllinTS development program

Throughout this chapter an analysis is performed considerations that were taken into account during development thermodynamic properties server called OllinTS. The development objective of the program is to calculate the phase equilibrium to isothermal and isobaric conditions, in addition to estimating the typical thermodynamic properties that characterize the substances with which a chemical engineer works, for example equilibrium constant, thermal conductivity, viscosity, heat capacity or vapor pressure. Also in this chapter the most important sections that make up the server properties are described. If any person wishes to make some modifications or additions to the program, it is recommended to study the program directly from the source code.

To choose the programming language are taken into account all the needs and objectives of thermodynamic properties server. Among these points is the format of the database and mathematical problems such as finding the roots of the cubic equation, but especially the programs selected to develop the server properties must be licensed open source for OllinTS it can be distributed freely.

## 4.1 programming language

The programming language you choose must be multi-platform, with a system of access to databases format *I Structured Query anguage*[eleven] SQL) mathematical tools for handling numeric and matrices arrangements, with the ability to display results using a graphical, plus during programming easy to edit or debug, and especially that all programs that are used have a license code open.

All requirements satisfactorily covers the Python programming language in which

11 Structured Query Language (Structured Query Language) is a declarative language access to databases Relational to specify various types of operations on them [35].

Many groups have developed a lot of modules that make it more useful, among them are Numerical, pysqlite and Pylab, which are basic and useful in the development of this work.

Currently the thermodynamic properties server is based on the following programs and programming languages for development:

- Python [36]: It is an interpreted language, which saves considerable time in the program development because no compilation and linking is necessary. The interpreter can be used interactively, which makes it easy to experiment with features of language, writing disposable test programs or functions during program development. It is also a very useful calculator.

  Python allows you to split the program into reusable modules from other programs written in Python. It comes with a large collection of standard modules that can be used as a basis for programs. There are also modules that provide input and output (I / S) file, system calls, etc. plus it is available for all platforms or operating systems.

- SQLite [37]: It is a database management system compliant relational database with ACID, which is contained in a small bookstore in C. SQLite is a public domain project created by D. Richard Hipp. The program uses SQLite's functionality through simple calls to subroutines and functions. So that the program can access this file uses the extension pySqlite2.

- Numeric [38]: This Python extension provides tools for manipulating numerical arrays and matrices multi-dimencionales, has a library of mathematical functions. It is written in C language to increase the speed of calculations.
- Matplotlib [39]: Pyhton Extension for 2D representation arrangements or lists of data in a variety of formats also allows plotting of pictures, enlarge areas of interest and save the resulting graphics as graphic images.

Computer equipment for working under Windows XP operating system is recommended to use the Enthoug Python program in which all the above programs are already included and configured so that many conflicts are avoided interface. Through the following figure shown in schematic graph form, as are programs are coupled to develop the program OllinTS using as a base to Python.

## Structure 4.2 server properties thermodynamic

The thermodynamic properties server is designed so that when being changed or expanded the possibilities of causing conflicts in other modules or mathematical models are minimized. For this the OllinTS program has been divided into sections:

- **administrative section:** Within this section is a program that performs the functions of thermodynamic models and store the values of the constants obtained from the database of the corresponding compounds. The administrator also provides the program running "as a thermodynamic case", in which the conditions to which it is desired to determine the phase equilibria and thermodynamic properties of the blend are defined. In addition, the administrator enables access to thermodynamic models to the database, links the thermodynamic model Thermodynamic has solved cases.

- **Database:** This section provides the functions to access the database so that at the end of the consultation the result is a numerical arrangement easily applied in the calculations.

- **Section equations of state:** Contains programs for solving cubic equations of state, the main function of these programs are calculating the distribution coefficients.

- **Tools section:** Within this module a program for interpolating Lagrange, which turns out to be very useful during the iterative calculations are available.
- **Flash section:** It includes the program to perform the calculation of the vaporized fraction to the case of liquid-phase equilibrium Vapor
- **Section thermodynamic equations:** Within this section include programs of mathematical models used for calculating the vapor pressure and state properties are expressed in terms of the equations of state.

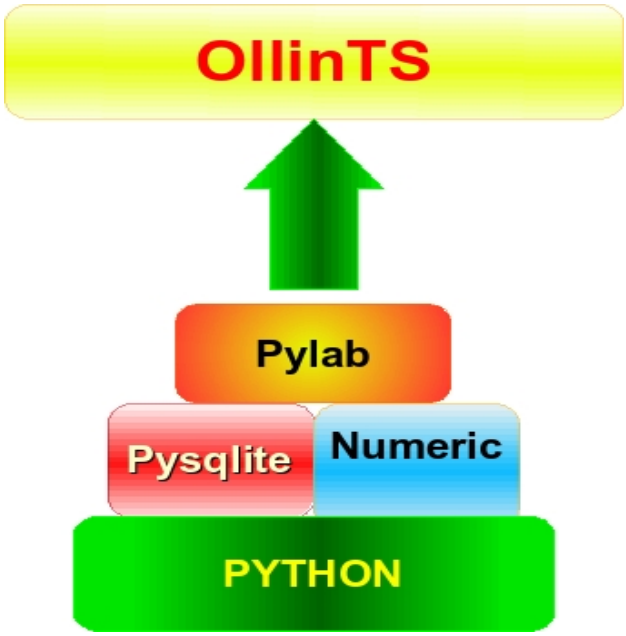



*Figure 6. Based Programs OllinTS*

## 4.3 virtual structure OllinTS

Figure # 7 shows the distribution of the modules that make up OllinTS at the time of execution described. In this figure it is also expressed the direction of flow of information and levels of interaction between the different sections. Through this figure it can be shown that the administrator has no direct control over the numerical methods or sequences calculation for calculating the phase equilibrium because the administrator only functions as a user interface.

OllinTS within the program that functions as a thermodynamically case or problem to solve, can be interpreted as a stream of material. The administrator has the ability to use two or more thermodynamic models (Figure No. 8), although it has priority the first model is created. When two or more thermodynamic models need to specify in which cases will thermodynamic method with which they will be resolved.

They have also been taken into account situations where they initially created the thermodynamic case and then a thermodynamic model. The administrator performs automatically connecting the case with thermodynamic thermodynamic model.

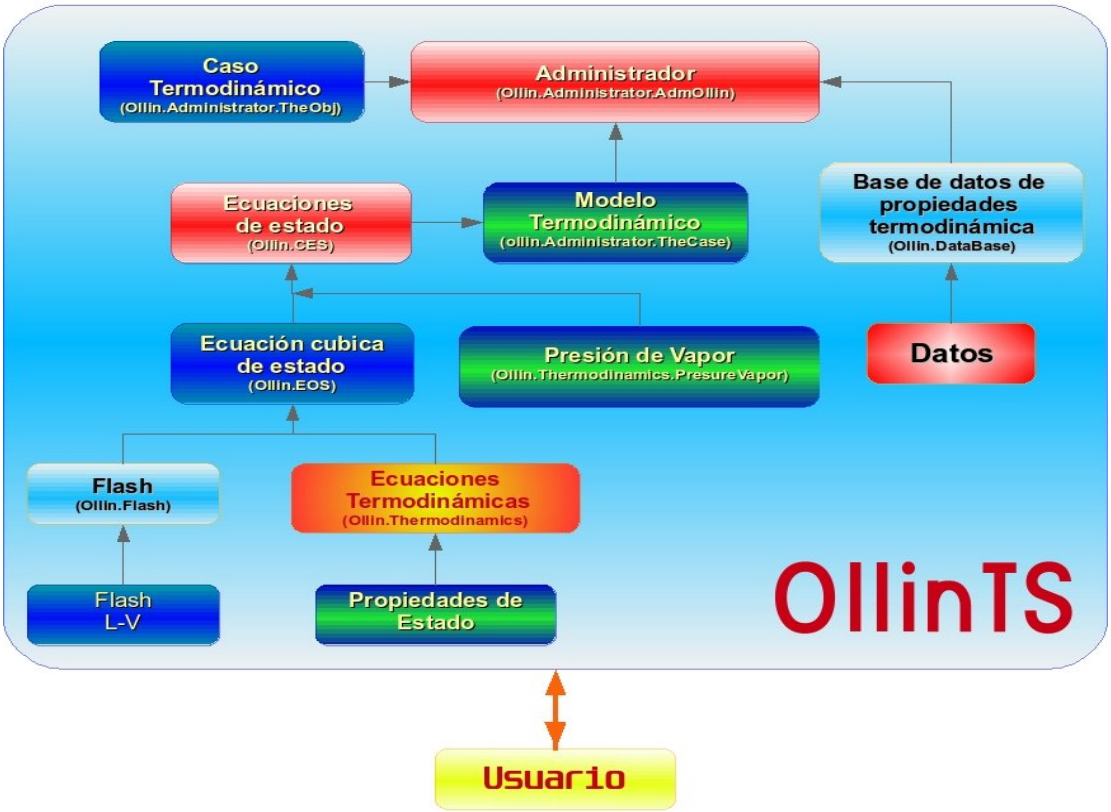



*Figure 7. Basic structure of the program OllinTS*

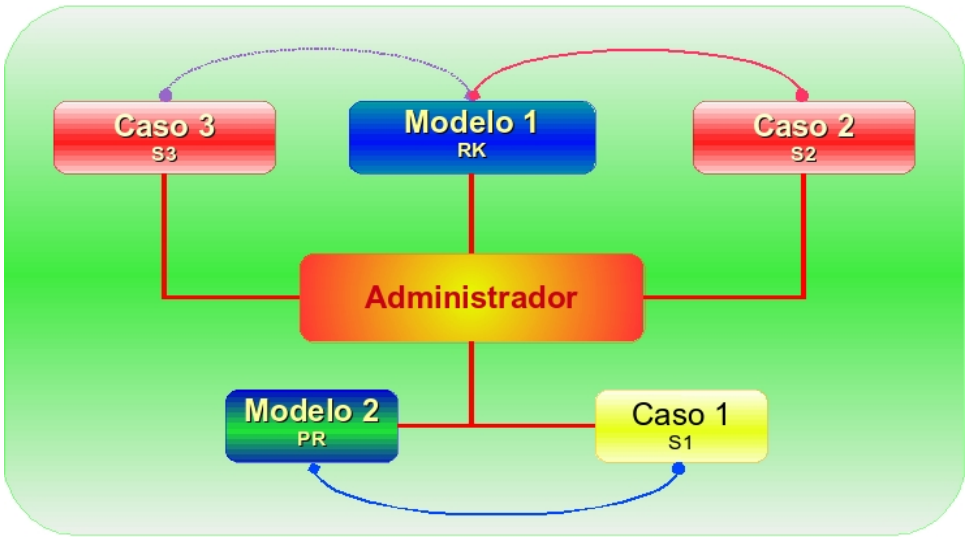



*Figure 8. Scheme work OllinTS*

## 4.4 Database

The database currently has 468 entries between chemical elements and compounds, some of them can not be applied to equations state due to the lack of information from some of its critical properties. All these data have been extracted from Appendix A of the book "The Properties of Gases and Liquids, [40]" These data can be downloaded into a spreadsheet format of the page: *http://www.tak2000.co*

Because the units where is the original database is a mixture of different systems of units, all have been performed corresponding to the international system conversions. In cases where it is not possible to convert directly into the database, as Antoine constants, the conversions of units is performed within the corresponding algorithm.

## 4.5 Programming Considerations

To avoid errors logical while running OllinTS and facilitate programming have taken into account the following considerations:

- fraction mol minimum It is acceptable $1x10^{-8}$ this restriction is controlled by the input OllinTS specification thermodynamic cases.
- To calculate concentration in each phase for any event that is present, is set to the maximum will be vaporized fraction of 0.99999 and the minimum $1x10^{-6}$.
- In applying the rules of mixing, when not count the values of the coefficients of interaction, these are considered zero.
- When the solution of the cubic equation of state for the liquid phase is less than or equal to zero, then the root of the cubic equation is used as if it were a gas.
- Due what constants for the Antoine equation available in the database are given for use with units of millimeters of mercury pressure, the program representing this equation automatically converts millimeters mercury pascals kilo and vice versa as appropriate.
- Within algorithms solution numerical method Newton-Raphson are used, so we need to know the derivative of the objective function corresponding to the cubic equation of state based compressibility factor and the equation for Rice Rachford

calculating the vaporized fraction, which are presented below.

$$\frac{df \cdot z \cdot}{dz = 3Z_2 - 2 \cdot 1 \cdot B - uB_2 \cdot Z \cdot\cdot TO \cdot wB_2 - uB - uB_2 \cdot} \tag{118}$$

$$\frac{df \cdot\cdot\cdot}{d \cdot = \sum_{i=i}^{NC} \frac{\cdot k_i - 1 \cdot_2 z_i}{\cdot\cdot k_i - 1 \cdot\cdot\cdot 1 \cdot_2}} \tag{119}$$

## 4.6 Main algorithms OllinTS

Below present diagrams of the main algorithms of programs that make up OllinTS, which meet in order to calculate the phase equilibria and thermodynamic properties. Diagrams presented algorithms are not entirely the actual algorithms, as these only try represent the sequence of calculations performed due to the complexities of these algorithms represent actual shape.

*Table 10. Main algorithms OllinTS*

| Algorithm | Description |
|---|---|
| Solver | Determines the method for estimating the phase equilibrium. |
| Flash | Determines whether it is possible to calculate the value of the vaporized fraction. |
| FracVap | Calculates the value of the vaporized fraction distribution coefficients. |
| FracTemp | Estimates the pressure at which the vaporized fraction and specified temperature occurs. |
| HenTemp | From the enthalpy and the temperature estimated value estimated phase equilibrium |
| HenPre | Calculates the temperature at which the value of the enthalpy and pressure is specified. |
| Isotermic | Calculates the phase equilibrium using cubic equations of state data having as temperature and pressure. |
| Thermal | Calculate the thermal properties of the mixture and makes correction by residual properties. |
| ZG | Performs the calculation of the compressibility factor for a gas |
| ZL | Calculates the compressibility factor for a liquid |
| Zo | Make an initial estimate of the compressibility factor |
| Thermo.Calc | Calculate the heat capacity, enthalpy and entropy of the ideal gas |

### 4.6.1 Solver

this algorithm HE which it is responsible for determining he method or he method for calculating the phase equilibrium.

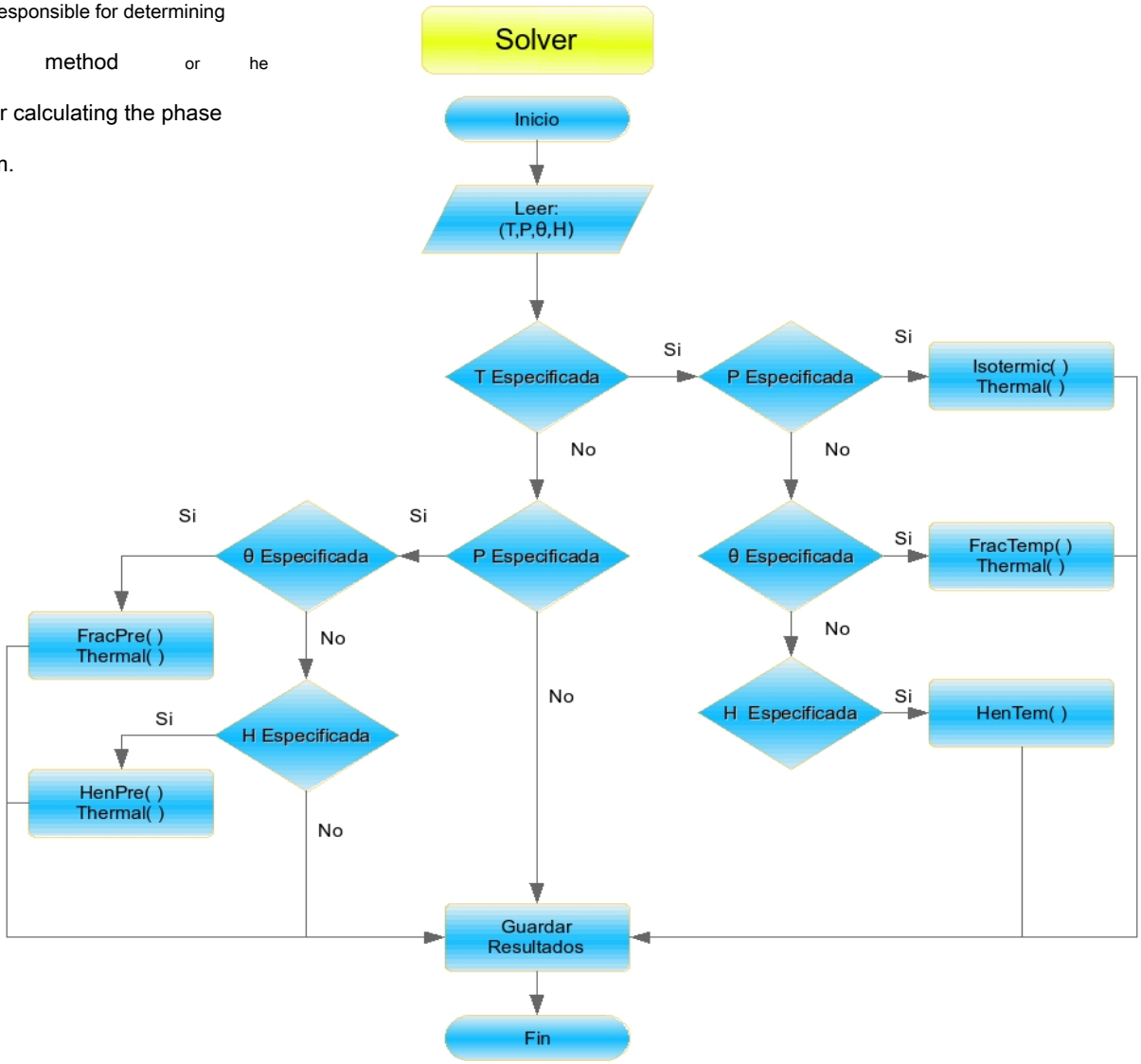



*Figure 9. Algorithm Solver*

### 4.6.2 Flash

This module determines the sequence of calculation for the value of the vaporised fractionated from the distribution coefficients and the overall mole fraction.

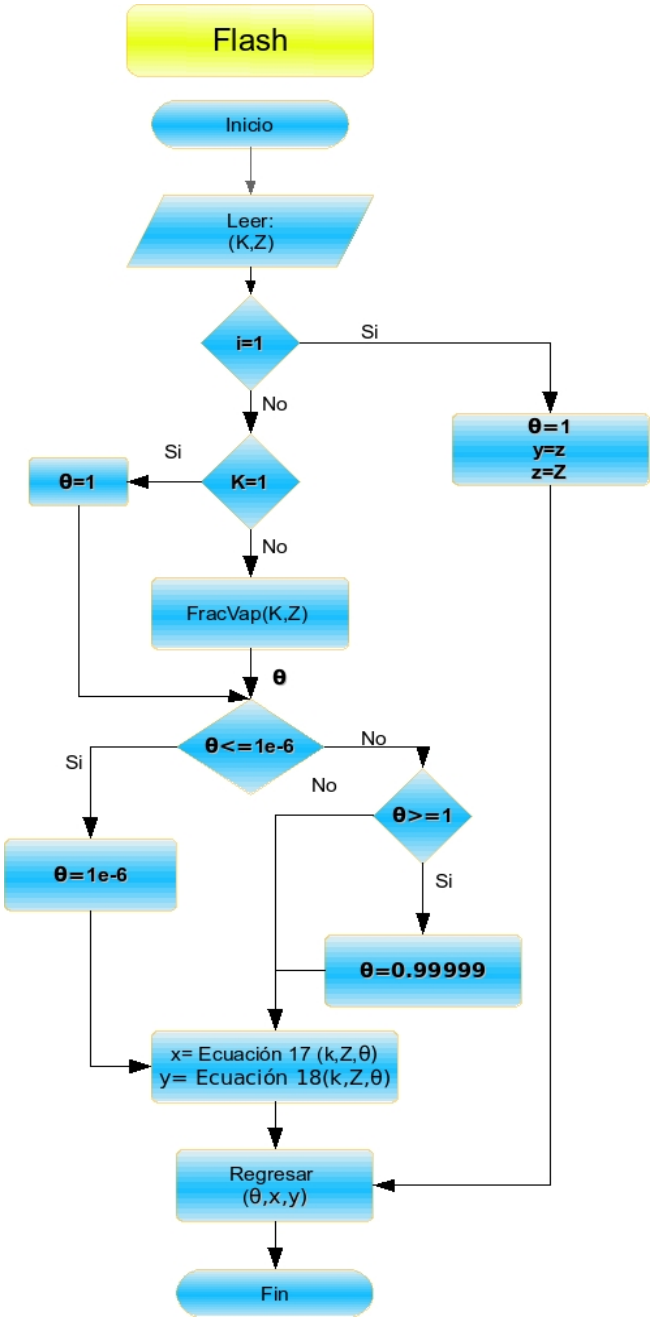



*Figure 10. Flash Algorithm*

### 4.6.3 FracVap

This algorithm is responsible for calculating the value of the fraction vaporized by the numerical method "Newton-Raphson" as data having the distribution coefficient and concentration of the mixture.

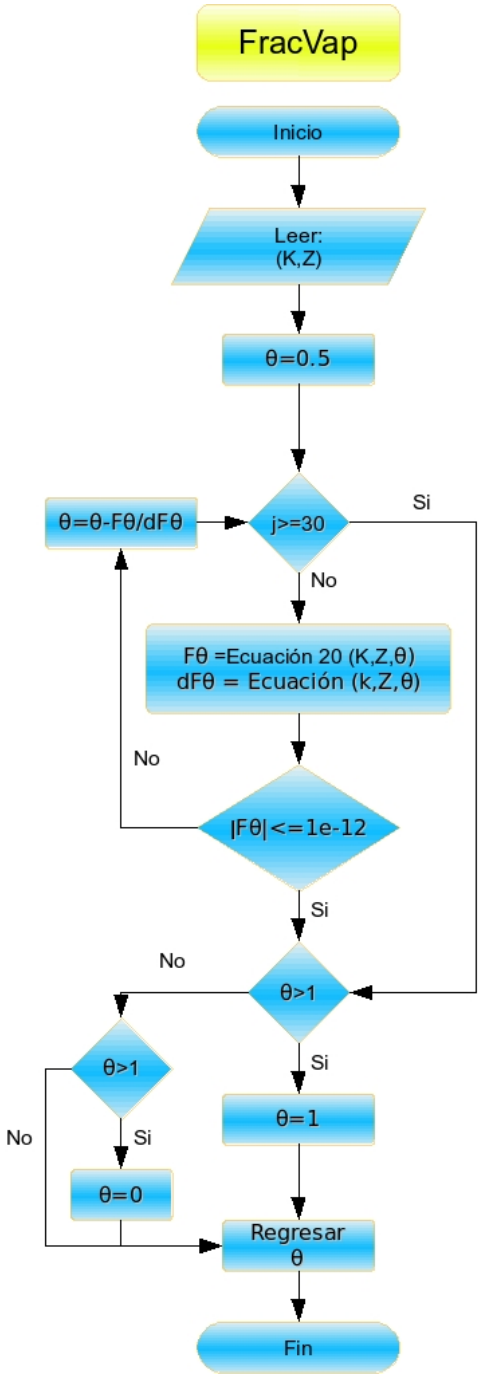



*Figure 11. Algorithm FracVap*

### 4.6.4 FracTemp

Through from East Thermodynamic algorithm cases are solved where HE specifies the mole fraction, the temperature Y the fraction Vaporized mixture.

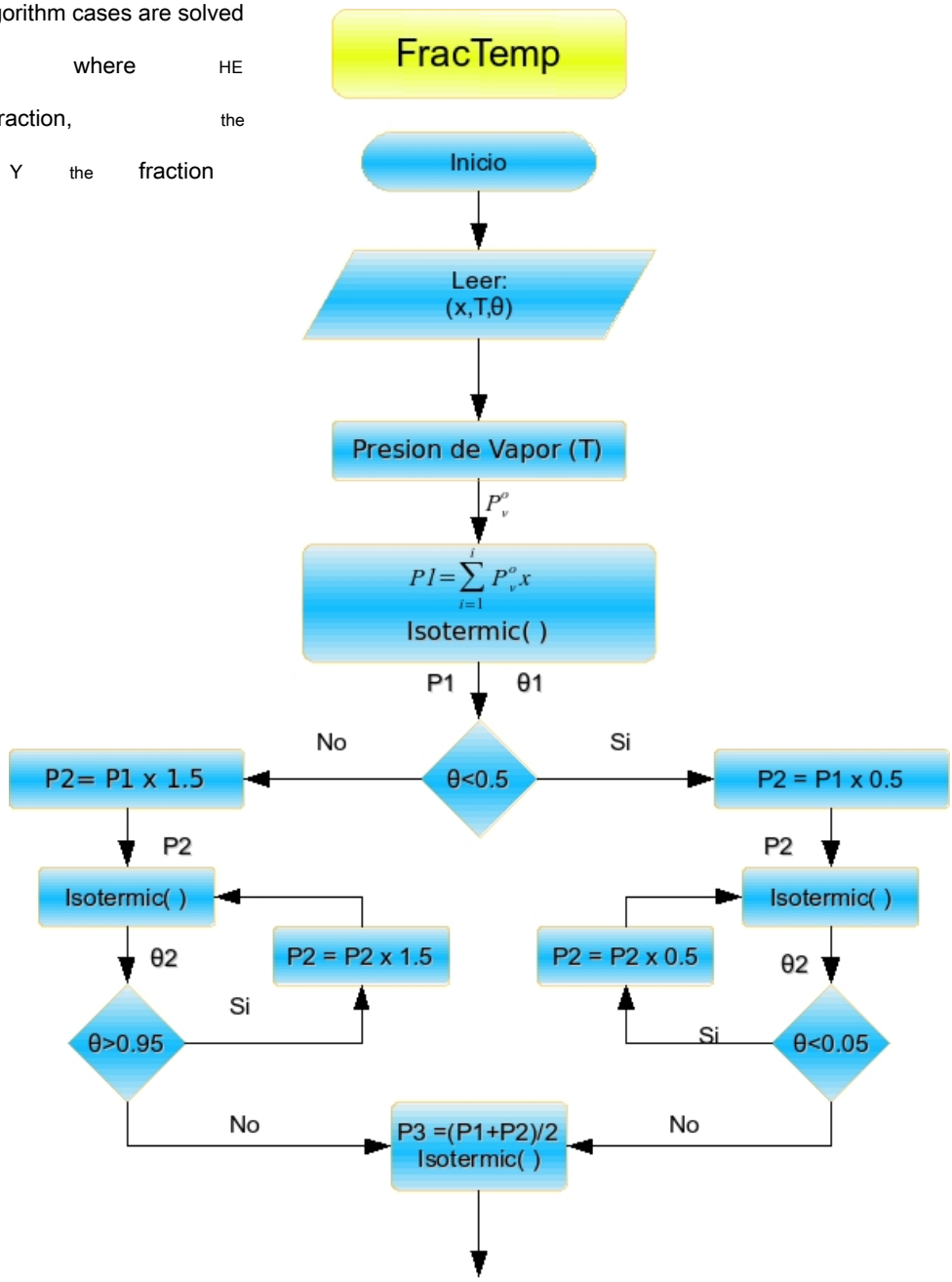



*Figure 12. Algorithm FracTemp*

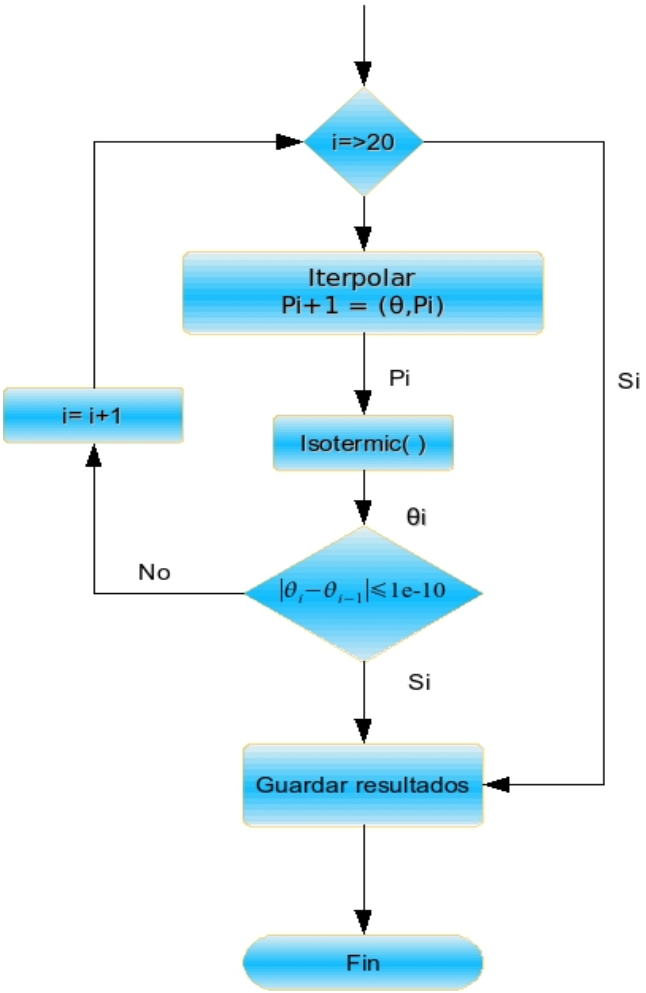



*Figure 13. Algorithm FracTemp (continued)*

### 4.6.5 HenTemp

This algorithm describes the procedure performed to calculate the pressure at which equilibrium exists phase to the enthalpy and temperature specified.

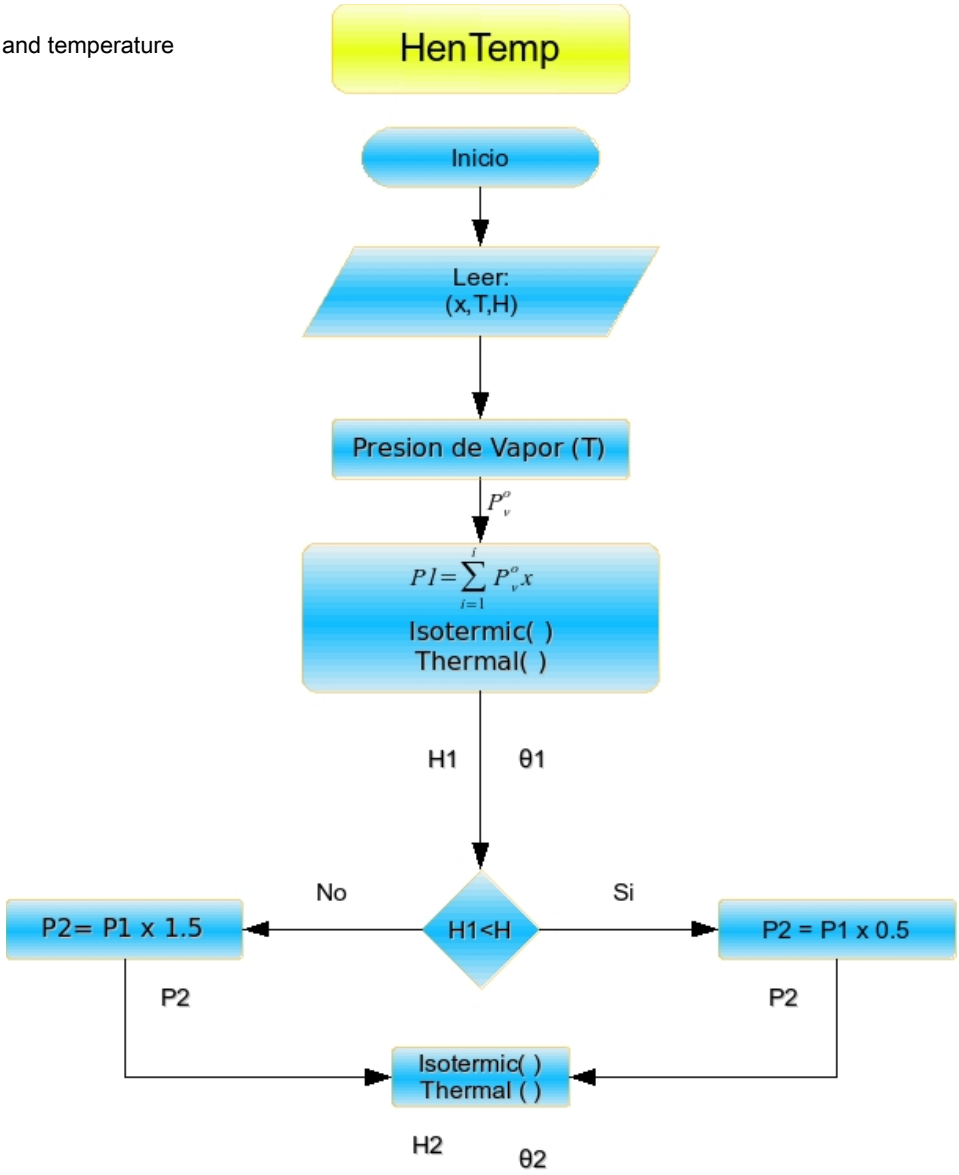



*Figure 14. Algorithm HenPre*

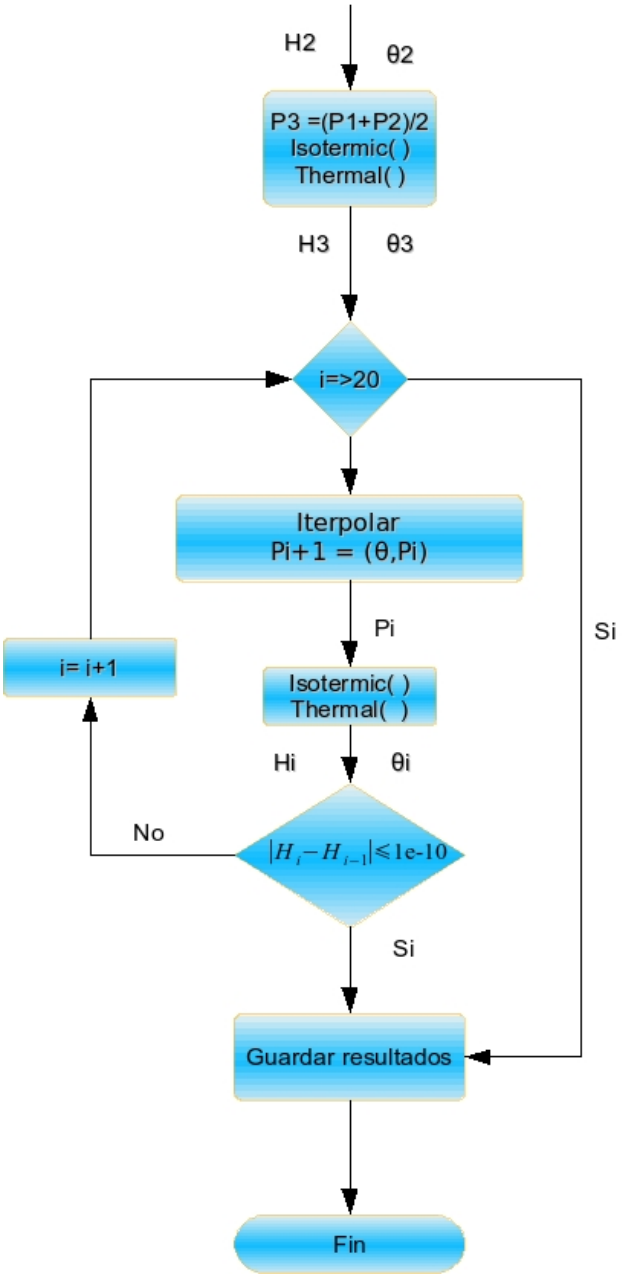



*Figure 15. Algorithm HenPre (continued)*

### 4.6.6 HenPre

Through this module is solve the case Thermodynamic which is needed in determining the value of the fraction vaporized Y the temperature at which there is a phase equilibrium, they have as the mole fraction data, pressure and enthalpy of mixing.

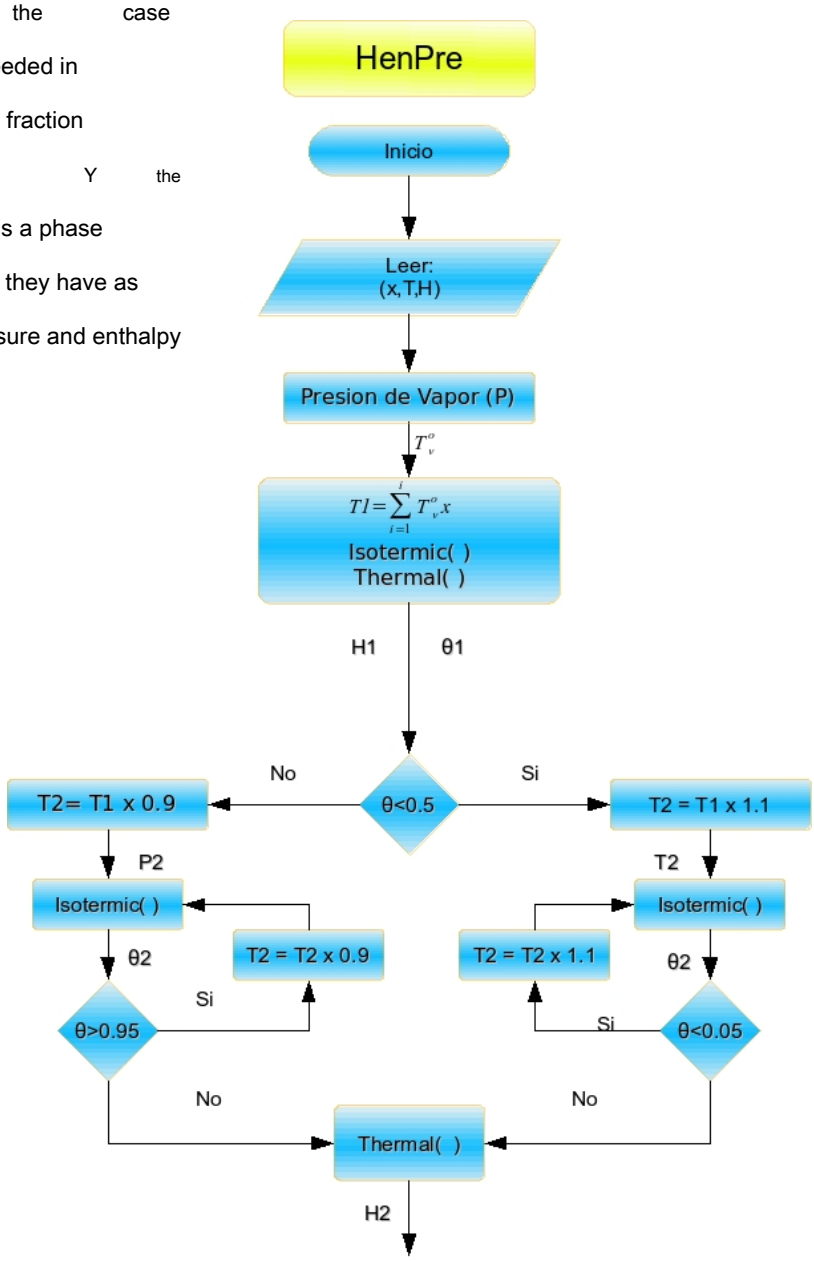



*Figure 16. Algorithm HenPre*

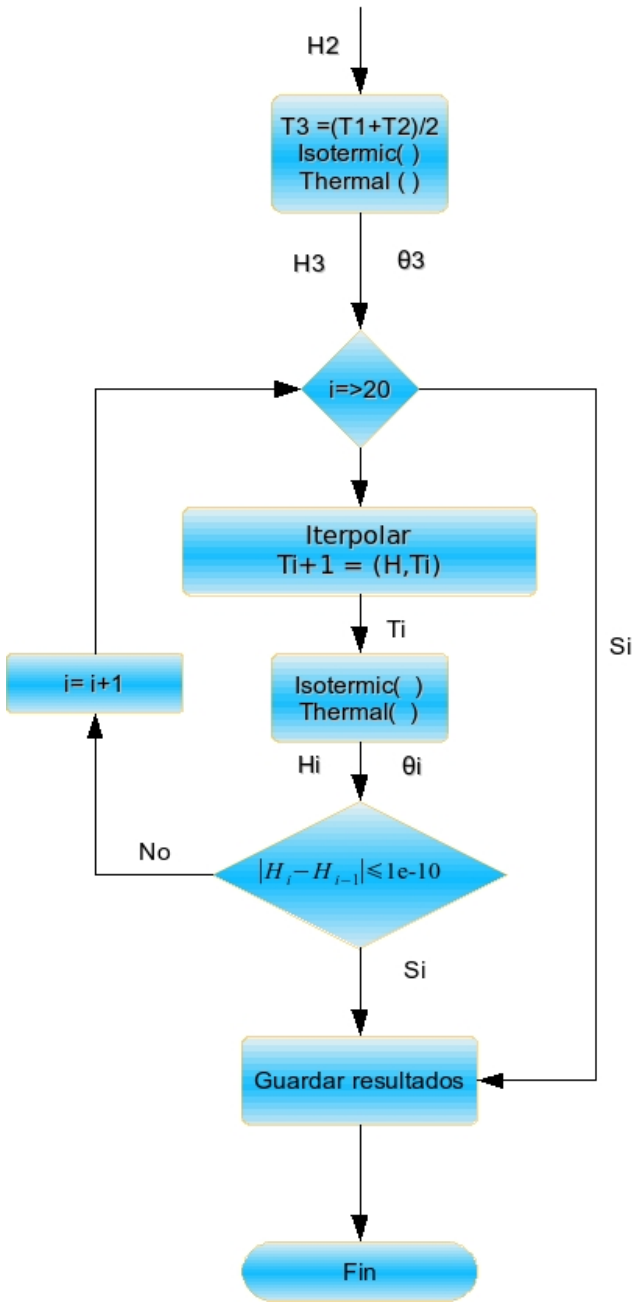



*Figure 17. Algorithm HenPre (continued)*

### 4.6.7 Isotermic

In the following algorithms describe the sequence of calculations for the solution phase equilibrium through the cubic equations of state, when the concentration data are established, pressure and temperature of the mixture.

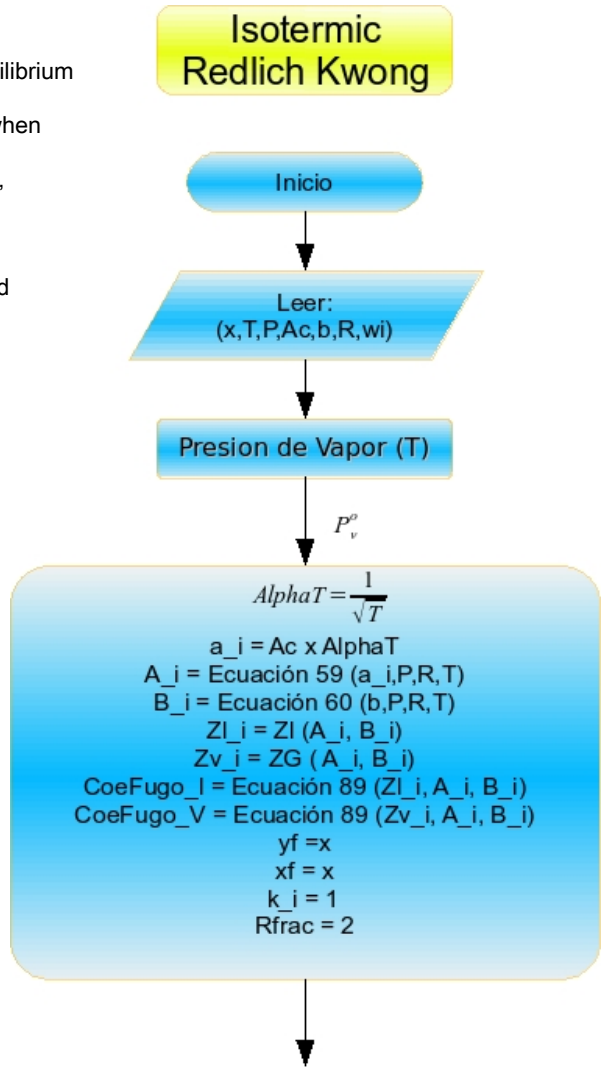



*Figure 18. Algorithm for Isotermic Redlich Kwong model*

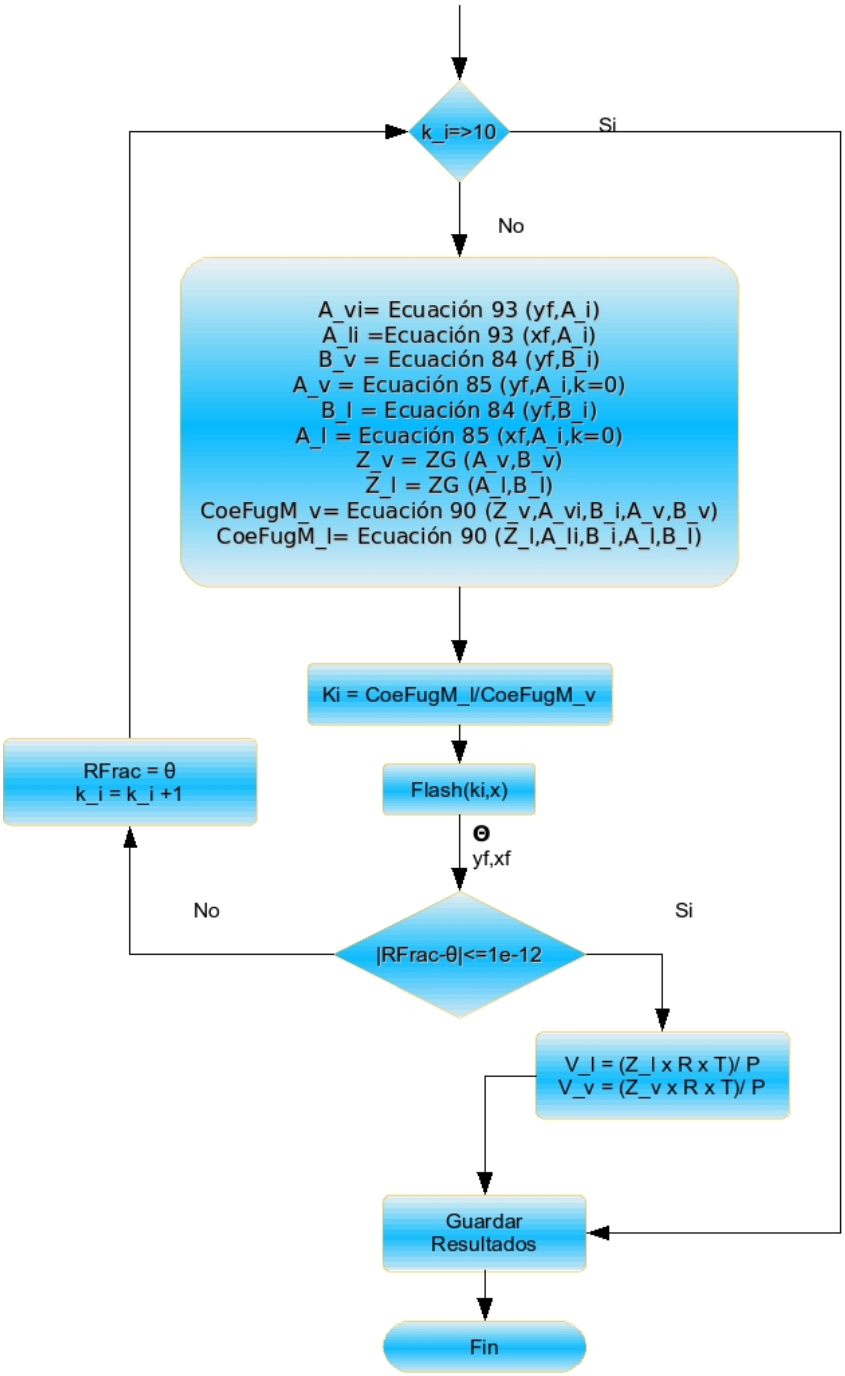



*Figure 19. Algorithm for Redlich model Isotermic Kwong (continued)*

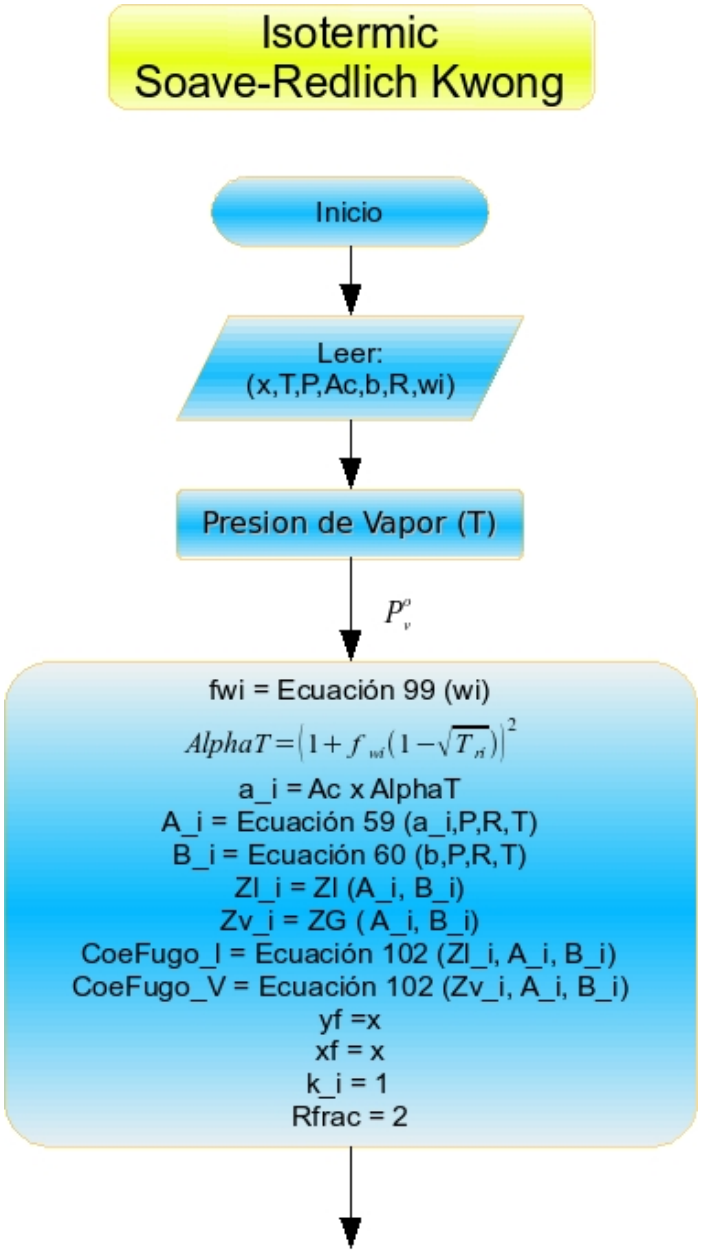



*Figure 20. Algorithm for the Soave model Isotermic RedlichKwong*

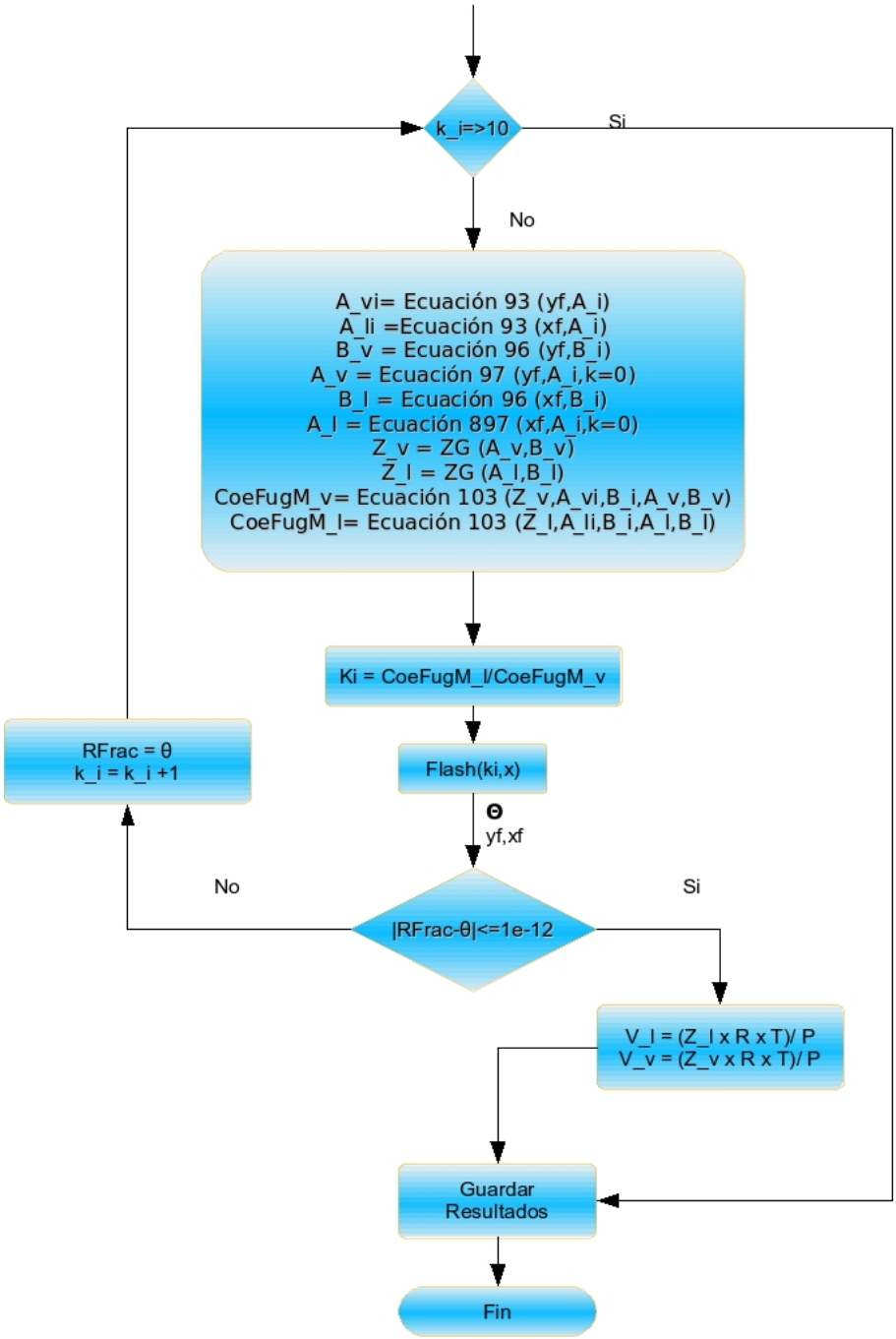



*Figure 21. Algorithm for the model Isotermic SoaveRedlichKwong (continuation)*

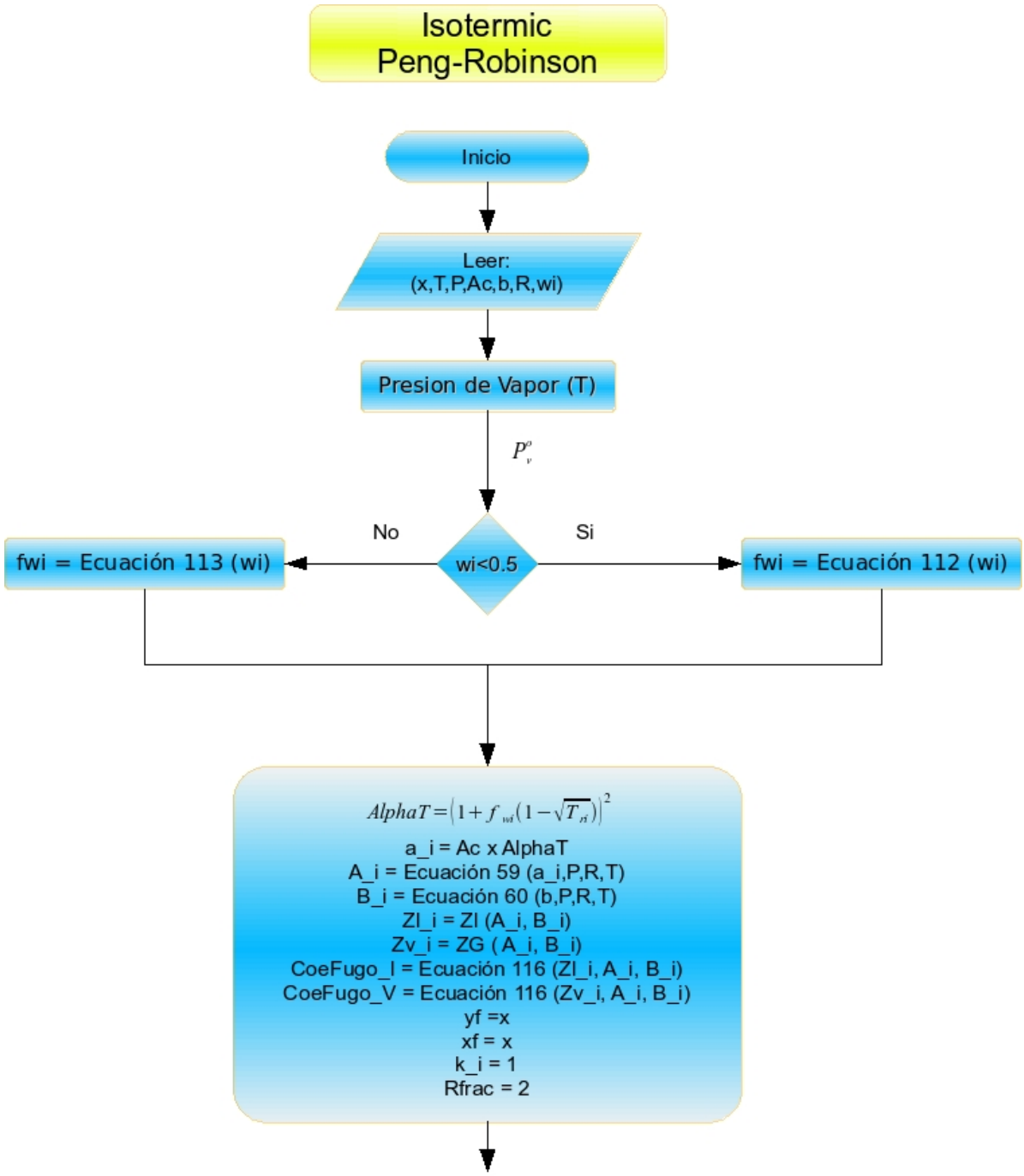



*Figure 22. Model Algorithm Isotermic PengRobinson*

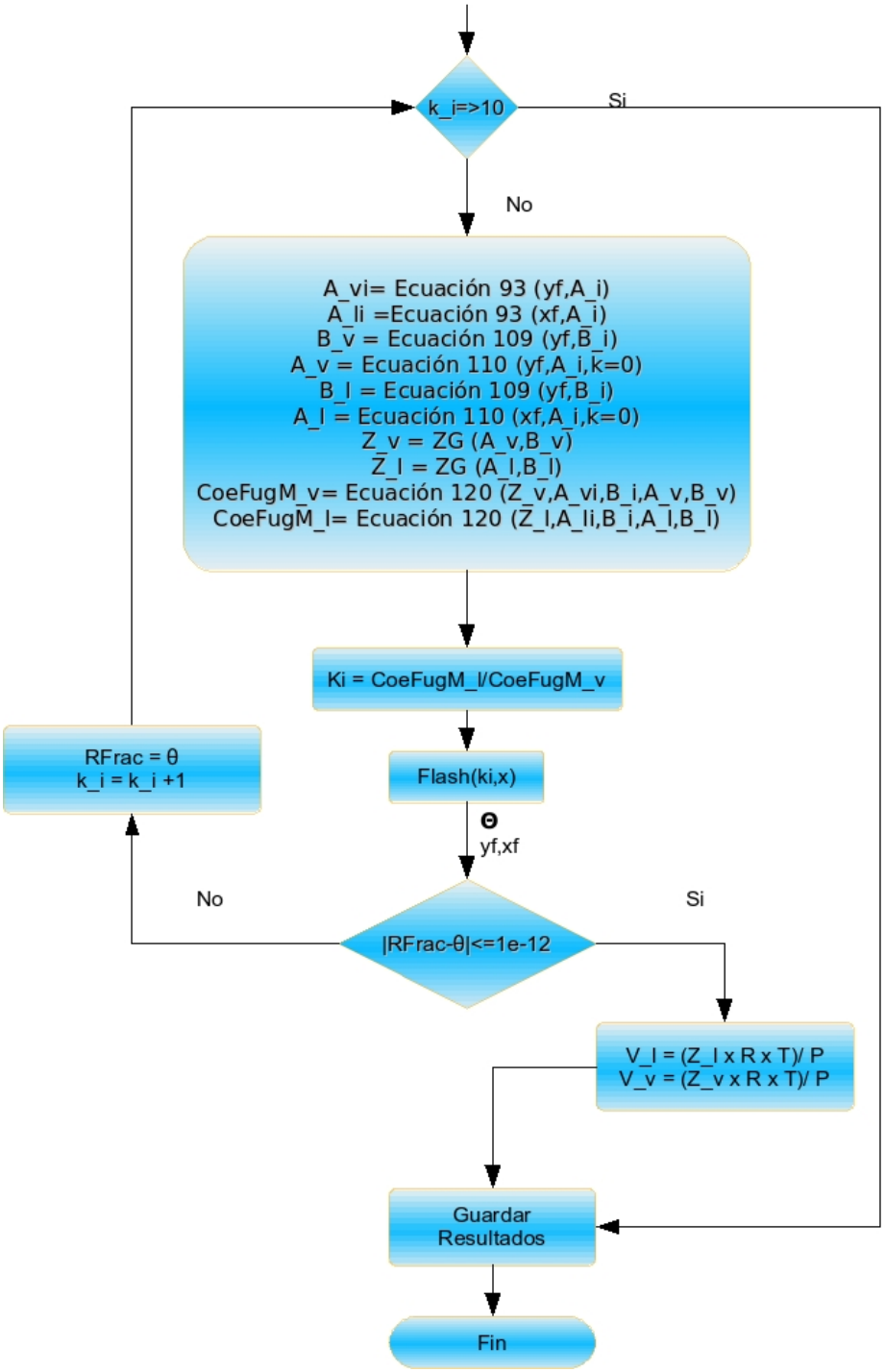



*Figure 23. Model Algorithm Isotermic PengRobinson*

*(continuation)*

### 4.6.8 Thermal

When we calculated the phase equilibrium Thermal it is responsible for performing calculations the properties thermodynamic, including the correction by properties waste.

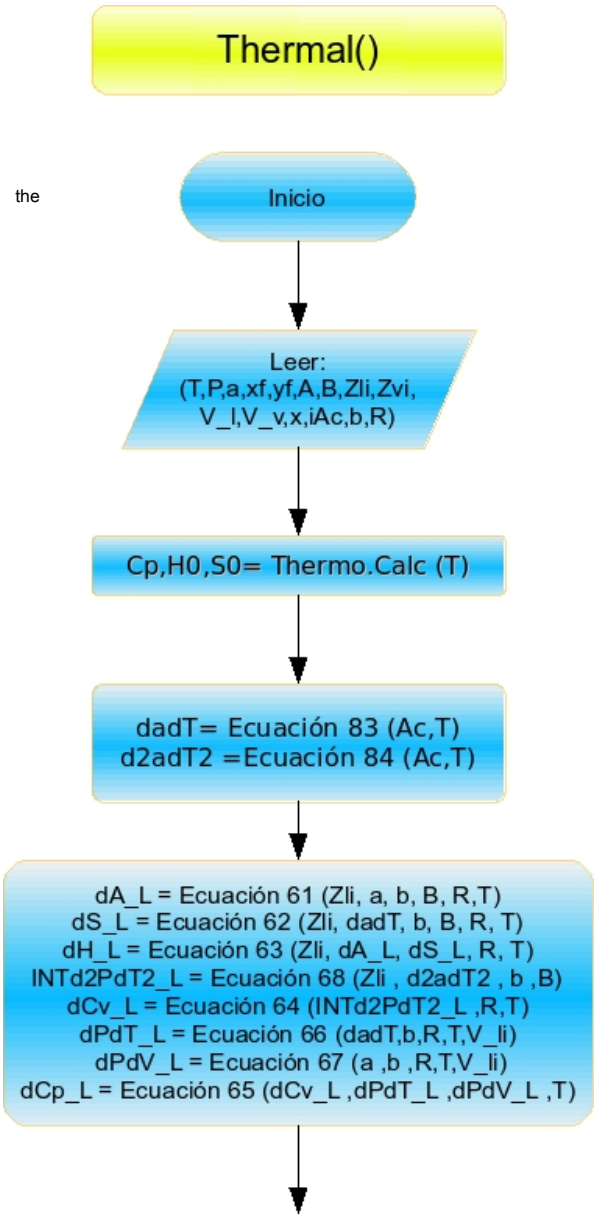



*Figure 24. Thermal Algorithm*

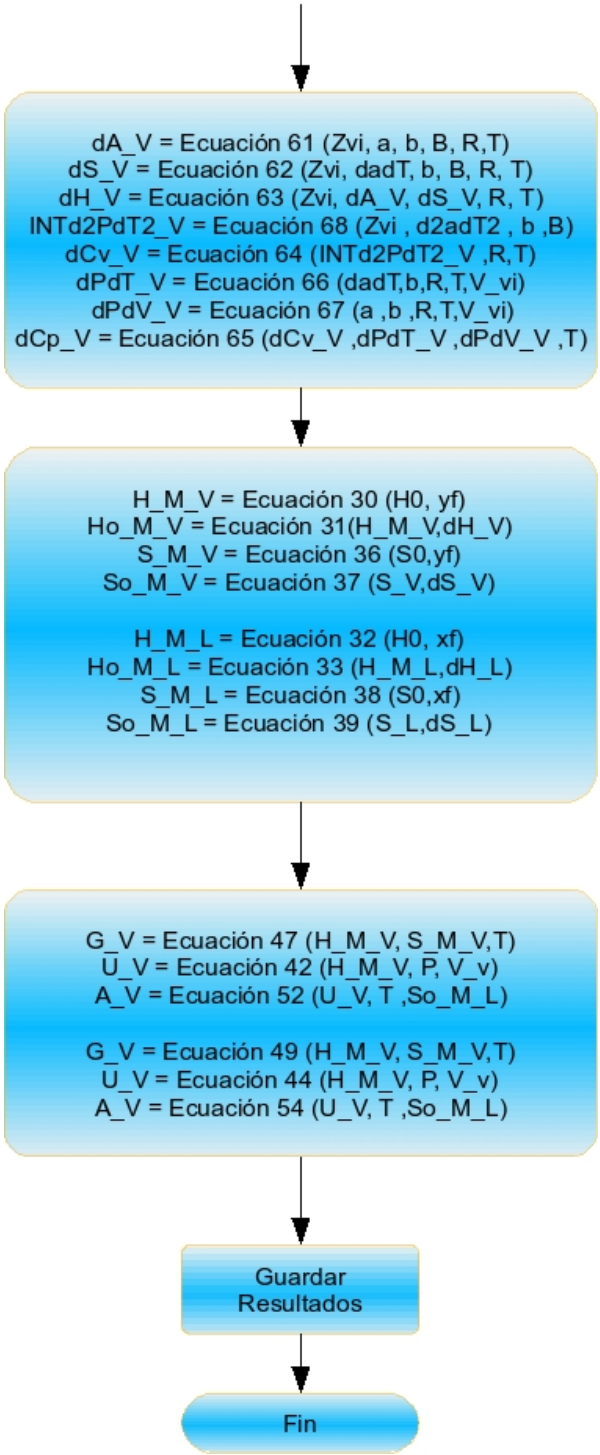



*Figure 25. Thermal Algorithm (continued)*

### 4.6.9 ZG

Through this algorithm calculating the value of the compressibility factor of a gas by means of the numerical method of "Newton-Raphson" is performed.

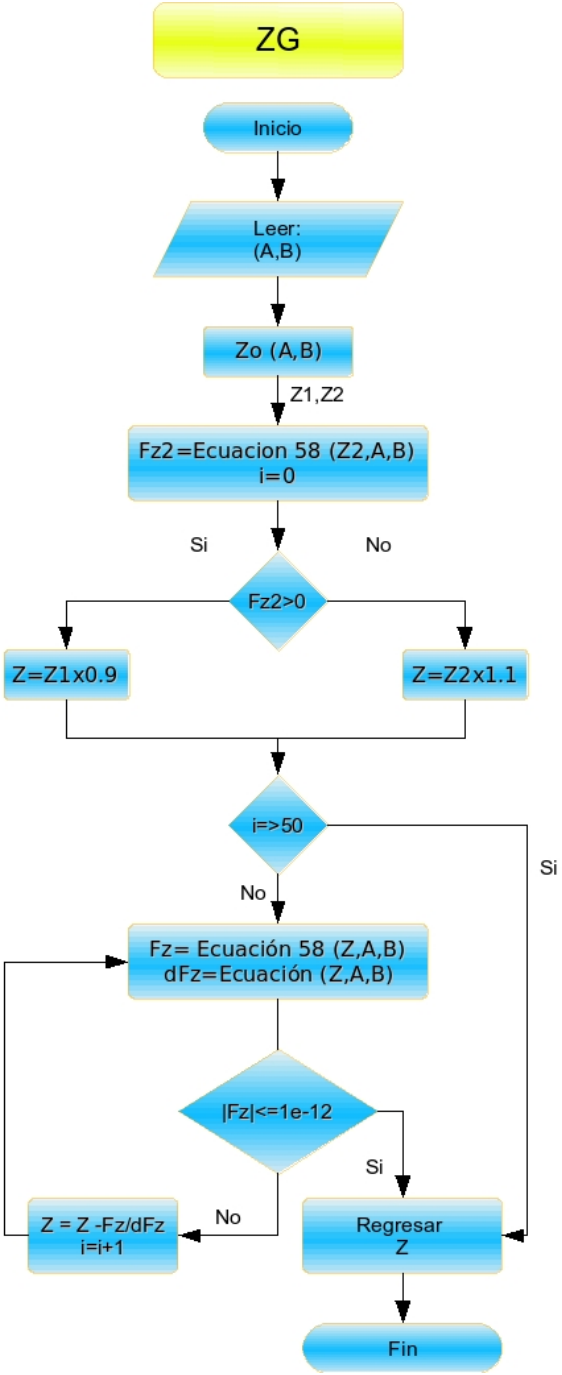



*Figure 26. Algorithm ZG*

### 4.6.10 ZL

Using this algorithm is possible to calculate the value of the compressibility factor of a liquid,

the numerical method of "Newton-Raphson" is used to reach the solution.

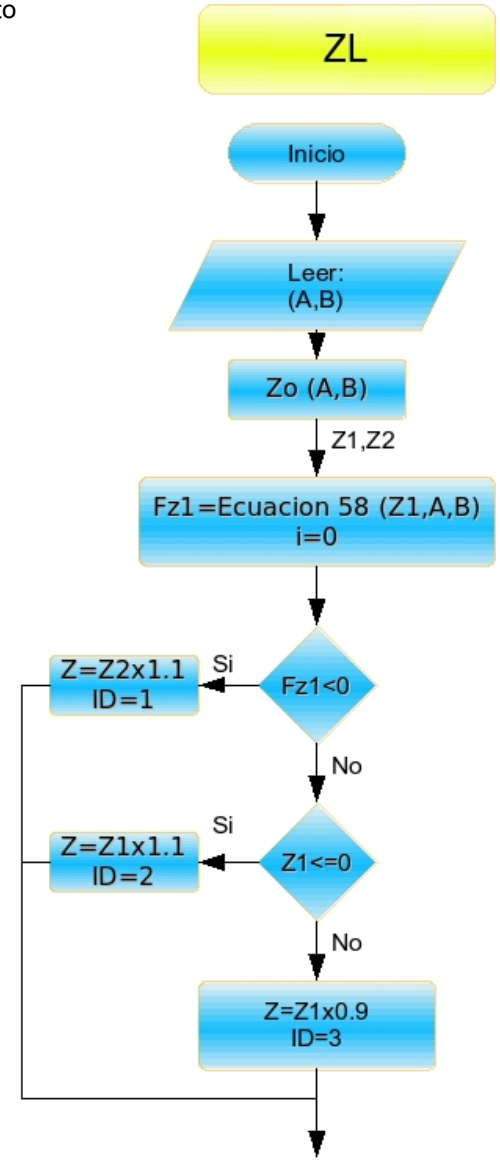



*Figure 27. Algorithm ZL*

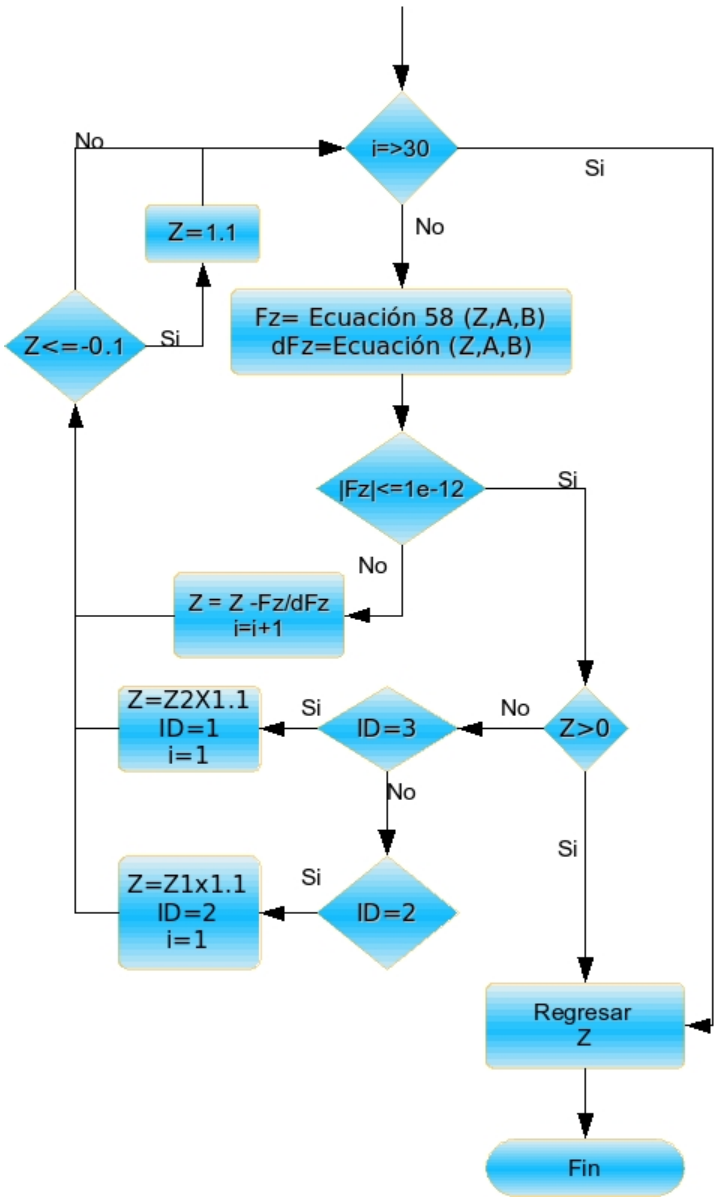



*Figure 28. Algorithm ZL (continued)*

### 4.6.11 Zo

This algorithm serves to locate the inflection points in the cubic equation of state based on the compressibility factor. These data are used as initial values in the iterations of the algorithms ZL and ZG.

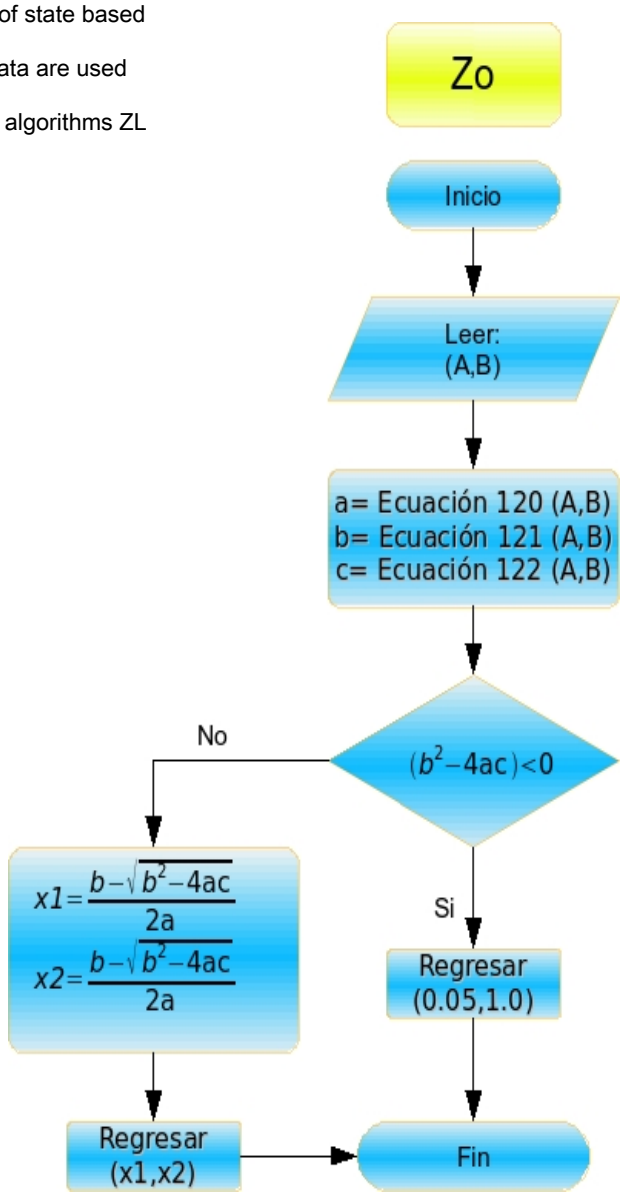



*Figure 29. Algorithm Zo*

To simplify East algorithm the following equations are used.

$$a = 3 \tag{120}$$

$$b = -2 \cdot 1 \cdot B - uB_2 \cdot \tag{121}$$

$$c = \cdot TO \cdot wB_2 - uB - uB_2 \cdot \tag{122}$$

### 4.6.12 Thermo.Calc

This program calculates the values of the heat capacity, enthalpy and entropy of an ideal gas.

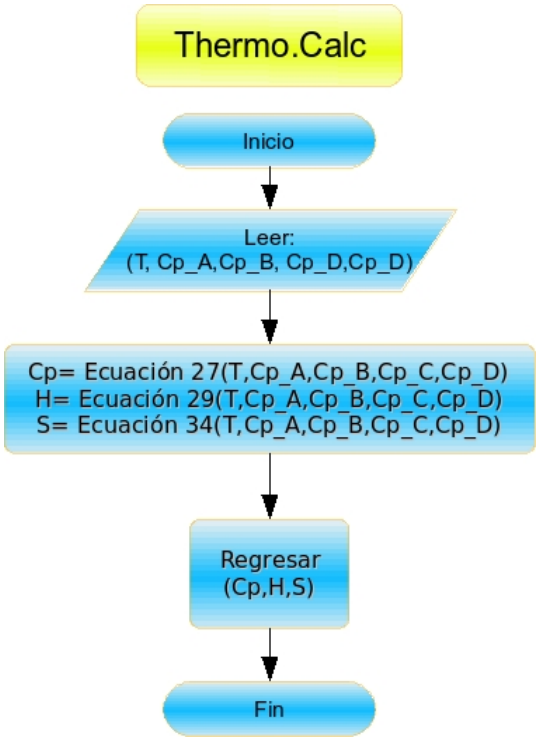



*Figure 30. Algorithm Thermo.Calc*

## 4.7 Errors

During the execution of server properties you may occur some errors that may result has been entered incorrectly some of the most common data occurs when the number of data fractions of the mixture does not match the number of compounds that are specified in the thermodynamic model.

In addition we must consider that there is a possibility that the equilibrium conditions is outside the area where three real roots to the cubic equation of state there. For the moment OllinTS it does not have the ability to predict these situations, so they may have inconsistent values.

## 4.8 Commands OllinTS

OllinTS is programmed to resolve cases where necessarily the mole fraction of the mixture of specified which derive the following cases:

1. Pressure- Temperature
2. Pressure-Enthalpy
3. Pressure-Vapor Fraction
4. Temperature- Vapor fraction
5. Temperature- enthalpy

Commands described below must be applied from an object can be a server or a thermodynamic property case. The syntax for these commands is that these should be written after a point separating the object to which point ( "Object". "Command").

### 4.8.1 Administrator Commands

This section object commands to which it is applied " **Ollin** "( *Ollin.Solve () )* which is the server properties.

- **AddModel (** TO , B , C ): Add a thermodynamic model administrator and returns the address of which can model created can be saved as a variable to enter the model created faster by following these specifications:

  TO Specifies the name by which the thermodynamic model is identified and which is stored in the administrator, this name should be a string.

  B : specifies the equation status to use thermodynamic model, the options are as follows.

  RKS = Redlich-Kwong-Simplicado

  Redlich-Kwong RK = Soave-Redlich =

  SRK-Kwong PR = Peng-Robinson

  When it omitted East server properties argument uses the default model RK

  C : Designates the equation by which the vapor pressure within the thermodynamic model is calculated. OllinTS currently has a database for equations

  *Antonie* Y *Harlacher,* even though the latter is not recommended, as the

constants available in our database the results are incorrect. If this argument is omitted equation is used *Antoine.*

- **Add ( TO , B ):** adds one or more compounds that are available in the database, when the component is missing or does not match the database, a search for compounds having the name some equivalence and printed on the screen is performed . The name should be written in CAPITAL LETTERS. Because the components can not be repeated, repeated compounds are ignored.

  TO : specifies compounds which they are to be added.

  B : Specifies the model thermodynamic to which the component is added.

- **Remove ( TO , B ):** eliminates one or more compounds which are available in the thermodynamic model, in the absence of the compound an error message is printed. Similarly, the name should be capitalized.

  TO Specifies the compound or compounds to be removed.

  B Specifies the thermodynamic model to which the component is disposed.

- **AddCase ( TO , B ):** adds a thermodynamic administrator and returns the address of the event created if you can saved a variable for quicker access, the case is created under the following specifications:

  TO : It's the name with which the administrator will identify this case.

  B : It is the name of the thermodynamic model which connects the thermodynamic case, when you omit this argument, the administrator automatically connect this with the first thermodynamic model in the manager added.

- **LoadConst ( TO ):** Load all necessary for the thermodynamic model constants specified by the argument "TO" , if not specified constants are loaded for all models that have been created.

- **Connect ( TO , B ):** connect a thermodynamic model [ TO] , a thermodynamically case [ B] . This command is used when you need to change the thermodynamic model which Thermodynamic case is resolved.

- **Solve ( TO ):** solve If the thermodynamic specified by the argument "TO" . When not specified, all possible cases are solved, if not defined equilibrium conditions prints an error.

- **Summary( TO ):** Print the results of a case decided by the argument specifying "TO" , if that is not the case resolved it prints an error message.
- **Comp ():** prints compounds which are in thermodynamic models

### 4.8.2 Commands thermodynamic case

For this section object to which I apply commands is determined by the name assigned to the thermodynamic case.

- **P ( TO ):** Specifies the value of the pressure by the argument "TO" in Kpa.
- **T ( TO ):** Specifies the value of the temperature the argument "TO" in K.
- **FracVap ( TO ):** specifies vaporized fraction defined by the argument "TO"
- **SetX ( TO ):** specifies he value of the overall mole fraction defined by the argument "TO" , must be in list form, this command normalizes concentration.
- **Rx ( ):** calculates fraction overall from the vaporized fraction mole and the concentrations of each of the phases
- **reset ( ):** Borra all intensive properties which define the conditions that are used to calculate the Balance phase.
- **CasePrint ():** Print values general intensive properties.
- **XPrint ( ):** Print the values of the equilibrium concentrations
- **Get ( TO ):** Came back a fix the values specified by the argument "TO" , Which should be stored in a variable or used immediately. should not be confused with screen printing results. The names of the variables are listed in the appendix.

As a result we have a server properties able to calculate the phase equilibrium and basic thermodynamic properties that can be used to build more complex programs.

### 4.9 Calculation of phase equilibrium-Example

The procedure for the server application properties in an example of calculating the phase equilibrium described.

Example 1. Phase equilibrium pressure and defined temperature.

For the hydrocarbon mixture of Table 10 is desired to know the temperature to which must operate so that the flash vaporized fraction is 0.5, at a pressure 101,325 Kpa.

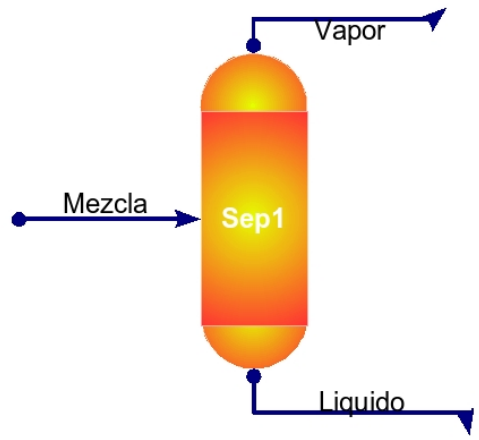



*Figure 31.Separador by flash*

*Table 11. Composition of the mixture of hydrocarbons*

| Compound | Mol fraction |
|---|---|
| ethane | 0.05 |
| Propane | 0.15 |
| N-Butane | 0.25 |
| N-Pentane | 0.20 |
| N-Hexane | 0.35 |

Application Procedure: Commands can be executed directly on the command console Python *Home> All Programs> Python 2.4> Python (Comand line)* or you can type commands and store in a simple text file with the extension py to run together.

From the console command is invoked Python OllinTS

```
>>> ollin.Administrator.AdmOllin import from Olin
```

*Loading Data Base data.db | .....*

Create a thermodynamic models and naming

```
>>> PR = Ollin.AddModel ( "PR", "PR", "Antoine")
```

Add the compounds of the mixture

```
>>> Ollin.Add ([ "ETHANE", "PROPANE", "NBUTANE", "NPENTANE", "NHEXANE"], "PR")
```

*ETHANE component 100 was add to 132 PROPANE PengRobinson component was add to 181 NBUTANE PengRobinson component was add to 223 NPENTANE PengRobinson component was add to 271 NHEXANE PengRobinson component was add to PengRobinson*

Thermodynamic creates cases and store them in a variable

```
>>> S1 = Ollin.AddCase ( "S1")
```

Defines the composition of thermodynamic case

```
>>> S1.SetX ([0.05,0.15,0.25,0.20,0.35])
```

Specifies the equilibrium conditions

```
>>> S1.FracVap (0.5)
>>> S1.P (101,325)
```

Solves the case

```
>>> Ollin.Solve ()
```

*Solving S1 ... Presure*

*Defined*

*…Defined FracVap*

Print results

```
>>> Ollin.Resumen ()
```

*…::Summary ::..*

*FracVap = 0.5000 KPa*

*Press Temp = 101,325*

*295,273 = Z L K = 0.006*

*V = 0.977 Z = 0.492 Z*

*Enthalpy KJ / kgmol = 2607.51497846 Entropy*

*KJ / Kg KgmolK = 134.195509151 MolWt / kgmol*

*= 67,241 MolWt L Kg / kgmol = 76,538 MolWt V*

*Kg / kgmol = 57945*

*…:: Component :: .. << >> << Vap Liq Fraction Fraction >> ETHANE*

*==> 0.0046 | ____ | PROPANE 0.0954 ==> 0.0395 | ____ | 0.2605*

*NBUTANE ==> 0.1585 | ____ | .3415*

*NPENTANE ==> 0.2335 | ____ | 0.1665 NHEXANE ==> 0.5640 |*
*____ | .1360*

To access the results of a thermodynamic case the "Get" which specify the name of the variable command is used. For example, for the values of the molar volume of the gas phase that is in the variable "Vvi" the following command is executed:

```
>>> S1.Get print ( "Vvi")
    [24.09860111 23.92423415 23.67905276 23.34669897 22.90005727]
```

Results: The temperature at which the equipment must operate flash is 295 273 K, these conditions the mixture the more heavy compounds are concentrated.

Table 12.Composición in the balance

| Component | Liquid phase composition | Gas phase composition |
|---|---|---|
| ethane | 0.0046 | .0954 |
| Propane | 0.0395 | .2605 |
| N-Butane | .1585 | .3415 |
| N-Pentane | .2335 | .1665 |
| N-Hexane | .5640 | .1360 |

To demonstrate the effectiveness of the values in Appendix A comparison of calculations performed by a server OllinTS regarding commercial properties is performed.

# 5 Application cases

This chapter has OllinTS capabilities through its application as a basis for more specific programs such as the construction of graphics or design teams demonstrated. For further details you should review the documentation for Python. For Windows XP Python is in *Home> All Programs> Python 2.4> Python (Comand line)*

## Case 1. Construction of a phase diagram

If described as OlliTS it can be used to construct the phase diagram where the evolution of the vaporized fraction of a hydrocarbon mixture is expressed from the bubble point at the dew point. This example is within OllinTS folder in the Examples section under the name *diagrama.py.*

*Table 13. Composition of the mixture*

| Compound | Mol fraction |
|---|---|
| ethane | 0.05 |
| Propane | 0.15 |
| N-Butane | 0.25 |
| N-Pentane | 0.20 |
| N-Hexane | 0.35 |

Application Procedure: The phase diagram is calculated for a pressure of 101,325 kPa, approximately bubble temperature is 240 K and pressure sprayed is 300 K. For the data needed to construct the diagram is calculated balance a range of 2 K.

Then the source code described to construct the phase diagram:

Invokes OllinTS and pylab

```
ollin.Administrator.AdmOllin from import Ollin from pylab
import *
```

Creates the thermodynamic model and adds the compounds

```
RK = Ollin.AddModel ( "RK", "RK", "Antoine")
Ollin.Add ([ "ETHANE" "PROPANE" "NBUTANE" "NPENTANE" "NHEXANE"], "RK")
```

Creates a current and defines the composition and pressure of stream

```
S1 = Ollin.AddCase ( "S1")
S1.SetX ([0.05,0.15,0.25,0.20,0.35]) S1.P
(101,325)
```

Sets the range of phase equilibrium calculation

```
plot_x = range (240,300,2)
```

Define the variables where the data is saved

```
plot_y0 = []
plot_y1 = []
plot_y2 = []
plot_y3 = []
plot_y4 = []
plot_y5 = []
```

Starts calculating the equilibrium in the balance range

```
for T in plot_x:
```

Defines the temperature and settle the balance

```
   S1.T (T)
Ollin.Solve ()
```

Retrieves the values of the concentration of the gas phase at equilibrium

```
f = S1.Get ( "f")
```

Saves the results in the variables

```
plot_y0.append (f [0])
plot_y1.append (f [1])
plot_y2.append (f [2])
plot_y3.append (f [3])
plot_y4.append (f [4])
plot_y5.append (S1.Get ( "FracVap"))
```

End of calculations

Grafica results

```
plot (plot_x, plot_y0) plot
(plot_x, plot_y1) plot
(plot_x, plot_y2) plot
(plot_x, plot_y3) plot
(plot_x, plot_y4) plot
(plot_x, plot_y5)
```

Defines the characteristics diagram

```
axis ([244,300,0,1]) grid
(True)
titles = RK.library titles.append (
"FracVap") legend (titles)

title ( "Vapor Fraction Vs T, Y vs. T") xlabel ( 'T
(K)') ylabel ( 'and FracVap')
```

Diagram shows

```
Show()
```

The resulting plot of the execution of this code represents the evolution of the concentration of hydrocarbons in the gas phase.

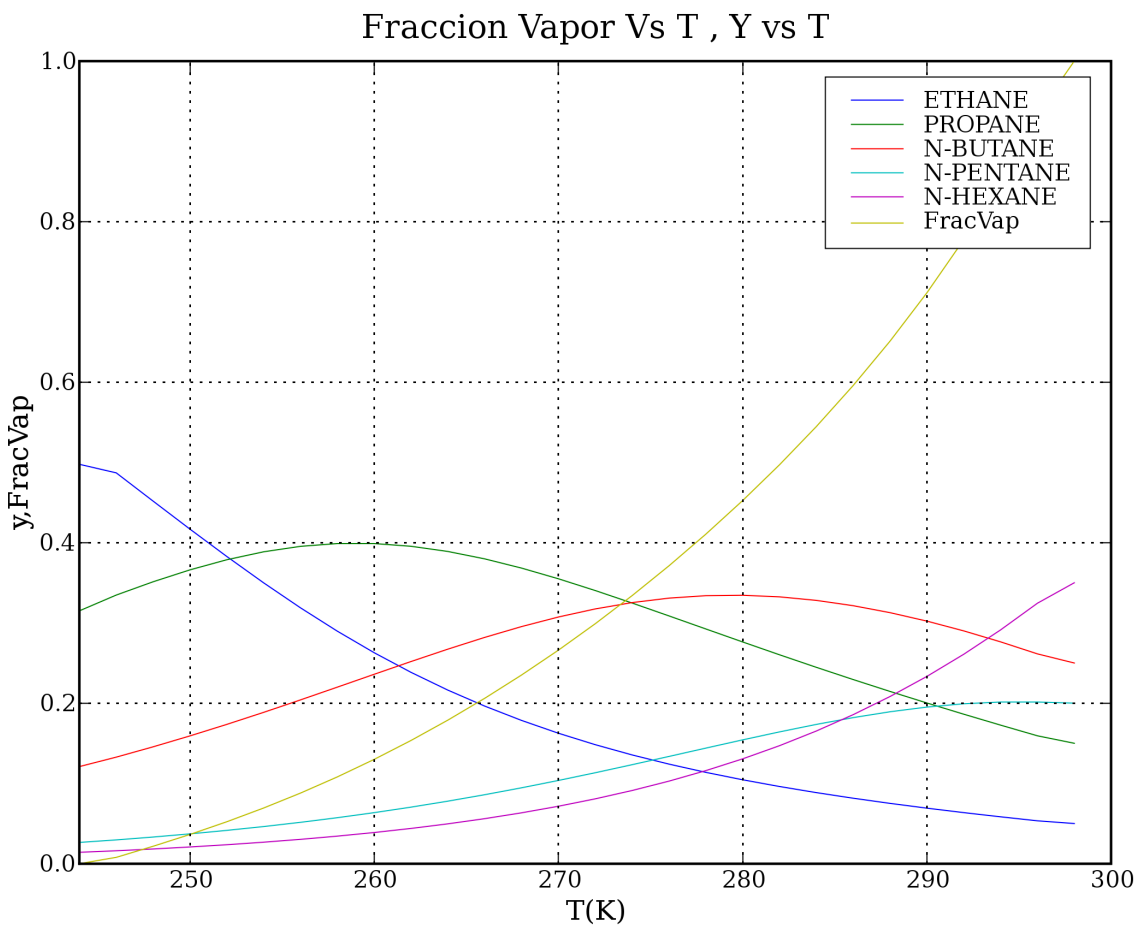



*Figure 32. Graph phase equilibria*

Results: In the diagram one can see that there is a high presence of ethane in the bubble point and with increasing vaporized fraction and the presence of going up until the time that the composition in the same initial mixing .

**Case 2. Calculate the vapor pressure**

With the help server properties can calculate the value of the vapor pressure corrected in the following example values of the vapor pressure calculated using the Antoine equation and the equation of Peng-Robinson for N-Butane is graphed and N-Heptane. This example is within OllinTS folder in the Examples section under the name *Presionv.py.*

Application Procedure: For equation Peng-Robinson vapor pressure is defined in a pure compound as the point where the value of fugacities for each phase is the same. The code by which you perform these graphs is written below.

Invoke OllinTS, pylab and lagrange interpolation tool

```
ollin.Administrator.AdmOllin from import Ollin from
ollin.Tools.tools import lagrange from pylab import *
```

Create a model and a thermodynamic case

```
RK = Ollin.AddModel ( "RK", "PR") S1 =
Ollin.AddCase ( "S1")
Ollin.Add ([ "NHEPTANE"], "RK") S1.SetX
([1])
```

Defines the temperature range

```
Ti = range (300,450,10)
```

Create the variables to store the results.

```
Ppi = []
Ppv = []
```

Starts calculating the vapor pressure

```
for T in Ti:
```

Create the variables to store the iterations

```
   df = []
P = []
```

Define initial conditions and settle the balance

```
   S1.P (101,325)
S1.T (T) Ollin.Solve
()
```

Retrieving baseline

```
   Pvi = S1.Get ( "PreVap") [0]
Ppi.append (PVI) fl = S1.Get (
"fl_i") [0] = fv S1.Get ( "fv_i") [0]
P.append (S1 .get ( "P"))
df.append (flfv)
```

```
   S1.P (PVI)
Ollin.Solve ()
   fl = S1.Get ( "fl_i") [0] = fv
S1.Get ( "fv_i") [0] P.append
(S1.Get ( "P")) df.append (flfv)
```

Calculating the error value by fugacities

```
   E = flfv
```

Starts iterations to calculate the vapor pressure

```
   while abs (E)> 1e3: Pi =
lagrange (df, P, 0) print S1.P Pi
(Pi) Ollin.Solve () fl = S1.Get (
"fl_i") [0] = fv S1.Get ( "fv_i") [0]
P.append (S1.Get ( "P")) E =
flfv
```

Adds the results to the list of actual values

```
      df.append (E)
Ppv.append (Pi)
```

Plot the results and defines the characteristics of the graph

```
plot (Ti, Ppi) plot
(Ti, Ppv) grid
(True)
titles = [ "Antoine", "PengRobinson"] legend
(titles)
title ( "vapor pressure of n-heptane") ylabel ( 'P
(Kpa)')
xlabel ( 'Temperature (K)')
```

Chart shows

```
Show()
```

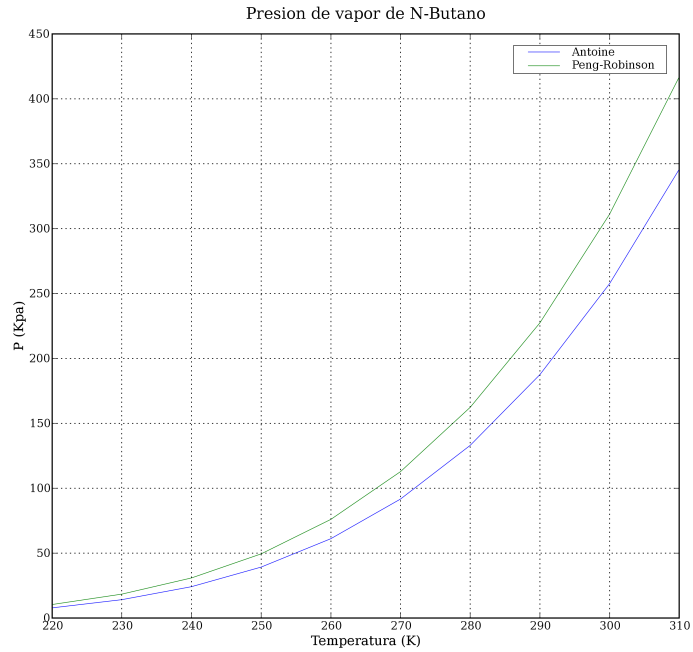



*Figure 33. Vapor pressure of n-butane*

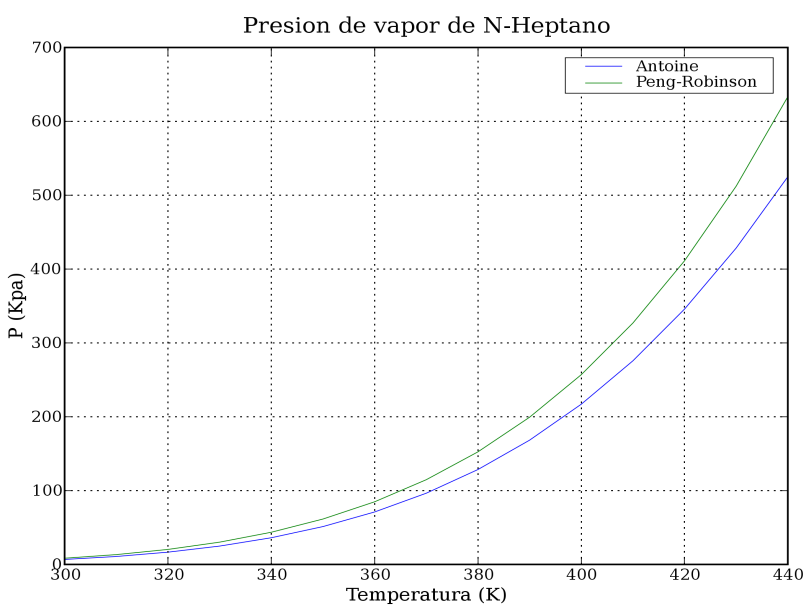



*Figure 34. Vapor pressure of n-heptane*

Results: In both graphs shown as the value of the vapor pressure calculated by the equation cubic state is greater than that calculated by the Antoine equation, besides the deviation between both methods is becoming wider as it approaches the critical point.

## Case 3. Design of a phase separator LV

At the exit of a reactor producing benzene from toluene, it has a separator type flash phases of which one wishes to know the vaporized fraction and size of equipment to recover benzene to a flow of the mixture of 1919,605 kgmol / hr to a temperature of 311.15 ° K and a pressure of 3206,062 kPa. The feed composition is as follows:

*Table 14. Composition of the mixture of aromatic*

| Compound | Mol fraction |
|---|---|
| Hydrogen | 0.36602 |
| Methane | 0.54813 |
| Benzene | 0.062618 |
| toluene | 0.021503 |
| diphenyl | 0.000945 |

Application Procedure: Flash tank dimensions are determined by the volume of the liquid being processed, establishing a residence time of 5 minutes. For vertical tank it is recommended that the height of the tank is the height occupied by the liquid plus three times the diameter and height ratio of the diameter is 4. This example is within OllinTS folder in the Examples section under the TanqueFlash name. *py.*

So the length of the flash tank will be:

$$L = 3D \cdot V_L \quad \cdot D_2 \tag{18}$$

And the diameter:

$$D = \frac{L}{4} \tag{18}$$

Pooling and operating both equations is obtained:

$$L = 3 \cdot \sqrt{\frac{256V_L}{\cdot}} \tag{18}$$

Where:

L = Length Tank

D = diameter of tank

$V_L$ = Fluid volume residing in the tank

The script to solve the problem posed is described in detail below, in this example OllinTS besides, are needed to invoke the numerical value of π, the array variable power method.

solving OllinTS have this with the following code:

Invoke OllinTS, the constant π, and varying array

```
ollin.Administrator.AdmOllin from Olin import from
Numeric import array, power, pi
```

Create a thermodynamic model and defines the components

```
PR = Ollin.AddModel ( "PR", "PR")
Ollin.Add ([ "HYDROGEN", "METHANE", "BENZENE", "TOLUENE", "DIPHENYL"], "PR")
```

Thermodynamic case creates and defines the conditions

```
S1 = Ollin.AddCase ( "S1")
S1.SetX ([0.366021,0.548913,0.062618,0.021503,0.000945]) S1.T (38 +
273.15) S1.P (3206.062)
```

Thermodynamic solves the case and prints the results

```
Ollin.Solve ( "S1")
Ollin.Resumen ( "S1")
```

Calculate the liquid flow rate, the volume resident, the length and diameter of the tank

```
L = (1S1.Get ( "FracVap")) * 1919.605
Gv = (L * S1.Get ( "MolWt_l")) / (S1.Get ( "LiqDen") * 60) Vr = Gv * 5

Lon = power ((256 * Vr / pi), 0.333333) Dia = lon
/ 4
```

the results are printed

```
print "longuitud" Lon print
"Diameter" Dia
```

Output the executing this code is as follows:

```
Loading Data Base data.db | .........

OllinTS has-been loaded

HYDROGEN component 19 was add to PengRobinson component
was 61 METHANE add to PengRobinson component was 242
BENZENE add to PengRobinson component was 286 TOLUENE add
to PengRobinson component was 429 DIPHENYL add to
PengRobinson Solving S1 ...

Defined Temperature
…Presure Defined

 …:: Summary :: .. FracVap     =
0.9138 Temp Press KPa = K
= 311,150 3206,062 ZL


              = 0132
ZV            = 1.031 Z
             = 0.954
Enthalpy kJ / kgmol = 6499.86626874 Entropy KJ
/ KgmolK = 236.251421265 MolWt    Kg / kgmol =
16,562 MolWt L kg / kgmol = 74,482 V MolWt kg /
kgmol = 11098

…:: Component :: ..          << >> << Vap Liq Fraction Fraction >>
HYDROGEN              ==>     0.0164              |____ |        .3990
METHANE               ==>     .1051               |____ |        .5908
BENZENE               ==>     .6317               |____ |        0.0089
TOLUENE               ==>     .2359               |____ |        0.0013
DIPHENYL              ==>     0.0110              |____ |        0.0000


Longuitud (m) 4.6729013708
Diameter (m) 1.1682253427
```

Results: At flash tank operating shows that the highest amount of hydrogen and methane is in the gas phase, so that benzene is in the liquid phase. Also are the dimensions of a diameter of 1.1682 and a height of 4.6729 meters. effectively,

# Conclusions and recommendations

During this development work is presented based on cubic equations of state to calculate the phase equilibria and thermodynamic properties typical characterizing fluids, called OllinTS having the following qualities server thermodynamic properties.

OllinTS allows the estimation of phase equilibrium mixtures also calculates the thermodynamic properties such as enthalpy, entropy, internal energy, free energy, Helmholtz free energy for the mixture and each of the pure components.

OllinTS is able to estimate the phase equilibrium knowing the composition of the mixture and any of the following pairs of variables.

1. Pressure-Temperature
2. Pressure-Enthalpy
3. Pressure-Vapor Fraction
4. Steam temperature-Fraction
5. Temperature-enthalpy

OllinTS has great relevance in the field of chemistry, academic and scientific engineering, since I was developed under an open source philosophy which allows both students and academics modify mathematical models of the state equations and solution algorithms phase equilibrium.

Since proprietary programs developed over the area of simulation of chemical processes are not affordable today for a student and causes the scientific community fall into illegality to make modifications to the solution algorithms of proprietary programs because their licenses remains the restricted source code, so OllinTS provides the basis for an alternative that will allow free themselves from dependence on proprietary programs, because it can be distributed, used and modified freely.

Within the subject area OllinTS it allows students simply analyze the effect of varying pressure and temperature on the phase equilibrium and a lots of thermodynamic properties.

As demonstrated OllinTS can be used as a base module for specific programs as it can be adjusted or manipulated according to the needs of the problem and the solution can be tailored to specific programs for specific problems programs.

In addition, by developing server thermodynamic properties has been shown that scientific tools developed by open source is like the case of breakthrough Python and maturity have to start being adopted and implantas in chemical engineering courses.

It is expected that future collaboration with the user community in the development of server properties, so finely can be used in simulators or as a base of development for specific applications such as the design of equipment in a reliable way. As recommended: Consider the case where one or more of the compounds forming the mixture exceeds some of their critical properties, improved algorithms to consider the behavior of the variables to enter a region retrograde condensation include rules mixing more rigorous estimation methods include the interaction coefficients to enhance the prediction of phase equilibrium. Although,

# Appendix A: Performance

To demonstrate the effectiveness of the calculations performed by OllinTS a comparison phase equilibrium for a mixture of hydrocarbons, for this phase equilibrium at different temperatures and 1 atm = 101325 Kpa by Redlich-Kwong equation is calculated, the results is performed They are compared with the results server thermodynamic property of Virtual Materials Group (VMG).

*Table 15. Composition of the mixture*

| Compound | Mol fraction |
|---|---|
| ethane | 0.05 |
| Propane | 0.15 |
| N-Butane | 0.25 |
| N-Pentane | 0.20 |
| N-Hexane | 0.35 |

*Table 16. Phase equilibria of the hydrocarbon mixture at 282 K*

| **Property** | VMG | **OllinTS** | **Error = OllinTS-VMG** |
|---|---|---|---|
| Vapor fraction | 0.5000 | .4971 | 0.0029 |
| z | .4884 | .4875 | 0.0010 |
| Composition | Liquid phase | Liquid phase | Liquid phase |
| ethane | 0.00438 | 0.00441 | 0.00003 |
| Propane | 0.04080 | 0.04104 | 0.00024 |
| N-Butane | 0.16821 | 0.16860 | 0.00039 |
| N-Pentane | 0.23584 | 0.23539 | 0.00045 |
| N-Hexane | 0.55078 | 0.55062 | 0.00016 |
| Composition | Gas phase | Gas phase | Gas phase |
| ethane | .0956 | .0961 | 0.0005 |
| Propane | .2590 | .2602 | 0.0012 |
| N-Butane | .3318 | .3324 | 0.0006 |
| N-Pentane | .1642 | .1642 | 0.0000 |
| N-Hexane | .1492 | .1470 | 0.0022 |

*Table 17. Phase Equilibria of the hydrocarbon mixture at 244.76 K*

| **Property** | VMG | **OllinTS** | **Error = OllinTS-VMG** |
|---|---|---|---|
| Vapor fraction | 0.0000 | 0.0000 | 0.0000 |
| Z | 0.00573 | 0.00577 | 0.00004 |
| Composition | Liquid phase | Liquid phase | Liquid phase |
| ethane | 0.0500 | 0.0500 | 0.0000 |
| Propane | 0.1500 | 0.1500 | 0.0000 |
| N-Butane | 0.2500 | 0.2500 | 0.0000 |
| N-Pentane | 0.2000 | 0.2000 | 0.0000 |
| N-Hexane | 0.3500 | 0.3500 | 0.0000 |
| Composition | Gas phase | Gas phase | Gas phase |
| ethane | 0.50790 | 0.50720 | 0.00070 |
| Propane | 0.32420 | 0.32364 | 0.00056 |
| N-Butane | 0.12528 | 0.12528 | 0.00010 |
| N-Pentane | 0.02753 | 0.02757 | 0.00004 |
| N-Hexane | 0.01506 | 0.02757 | 0.01250 |

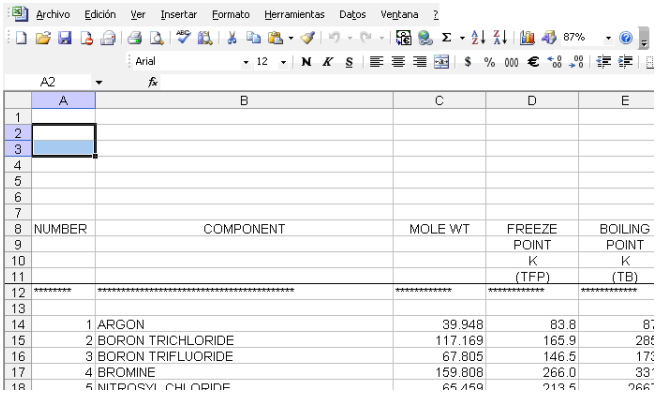
*Table 18. Phase Equilibria of the hydrocarbon mixture at 297.13 K*

| **Property** | VMG | **OllinTS** | **Error = OllinTS-VMG** |
|---|---|---|---|
| Vapor fraction | 1.00000 | 0.98579 | 0.01421 |
| Z | 0.96735 | 0.95762 | 0.00973 |
| Composition | Liquid phase | Liquid phase | Liquid phase |
| ethane | 0.00182 | 0.00185 | 0.00003 |
| Propane | 0.01695 | 0.01720 | 0.00025 |
| N-Butane | 0.08281 | 0.08360 | 0.00079 |
| N-Pentane | 0.17315 | 0.17307 | 0.00008 |
| N-Hexane | 0.72528 | 0.72429 | 0.00099 |
| Composition | Gas phase | Gas phase | Gas phase |
| ethane | 0.0500 | 0.05069 | 0.00069 |
| Propane | 0.1500 | 0.15191 | 0.00191 |
| N-Butane | 0.2500 | 0.25240 | 0.00240 |
| N-Pentane | 0.2000 | 0.20038 | 0.00038 |
| N-Hexane | 0.3500 | 0.34460 | 0.00540 |

Through these Tables it can be concluded that the server thermodynamic properties developed in this work is reliable because the results show that the error between acceptable programs.

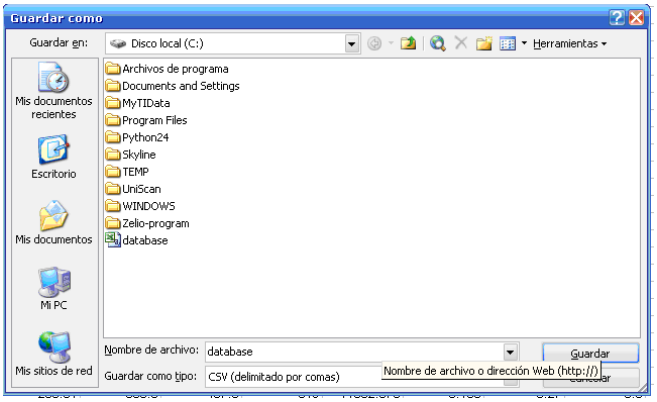
# Appendix B: Creating Database

Following the procedure for creating a SQL database format from a spreadsheet with the help of a GUI application for SQLite called "SQLite Data Browser" [41].

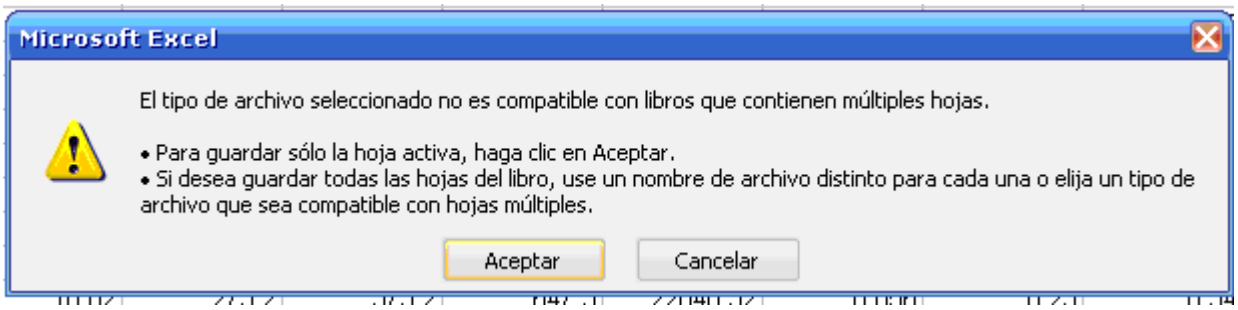


| NUMBER | COMPONENT | MOLE WT | FREEZE POINT K (TFP) | BOILING POINT K (TB) |
|---|---|---|---|---|
| ******** | ********** | ********** | ********** | ********** |
| 1 | ARGON | 39.948 | 83.8 | 87 |
| 2 | BORON TRICHLORIDE | 117.169 | 165.9 | 285 |
| 3 | BORON TRIFLUORIDE | 67.805 | 146.5 | 173 |
| 4 | BROMINE | 159.808 | 266.0 | 331 |
| 5 | NITROSYL CHLORIDE | 65.459 | 213.5 | 266 |

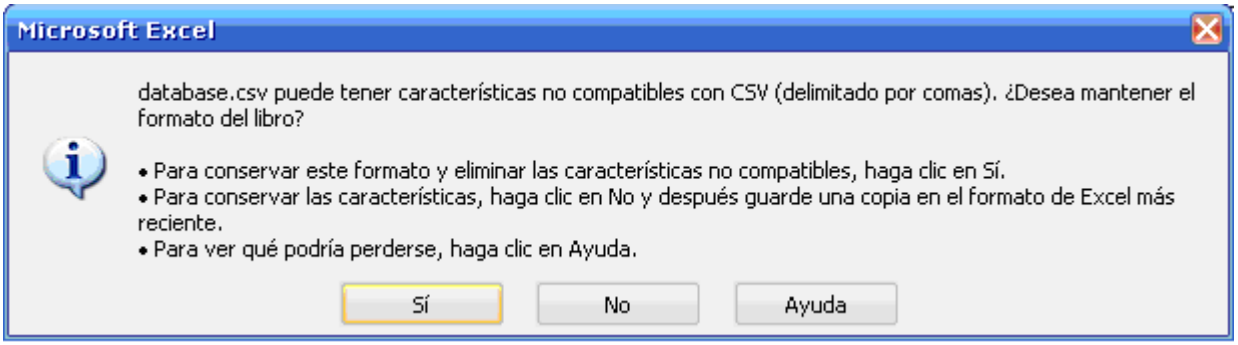
Initially available spreadsheet containing all necessary data which will be stored in a blank worksheet, the columns should not have any extra information. For example the name of the column.



Save the new spreadsheet with the CSV (comma delimited) which can be used by SQLite Database Browser

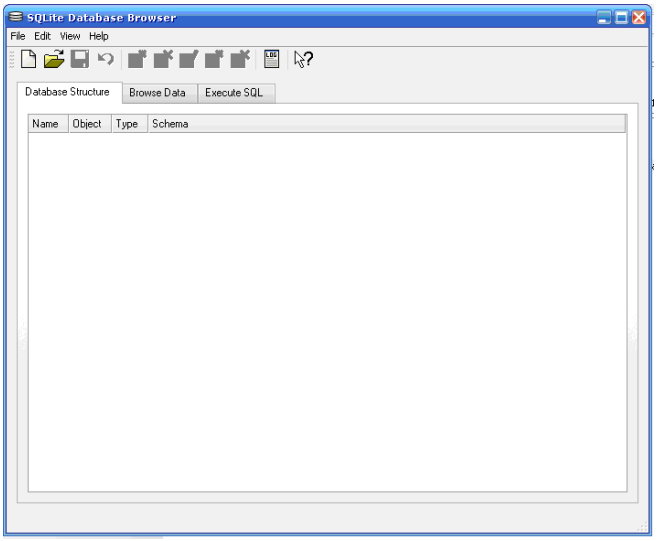



This window appears to warn us that we are selecting a format that does not support multiple sheets when the spreadsheet is saved. the "OK" option is selected to continue the process.

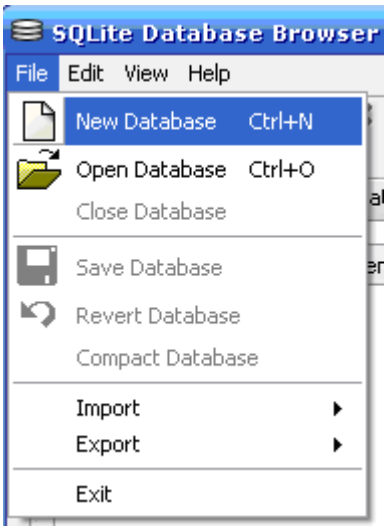



When this sale appears, the "Yes" is pressed to save the database in CSV format SQLite Database Browser.

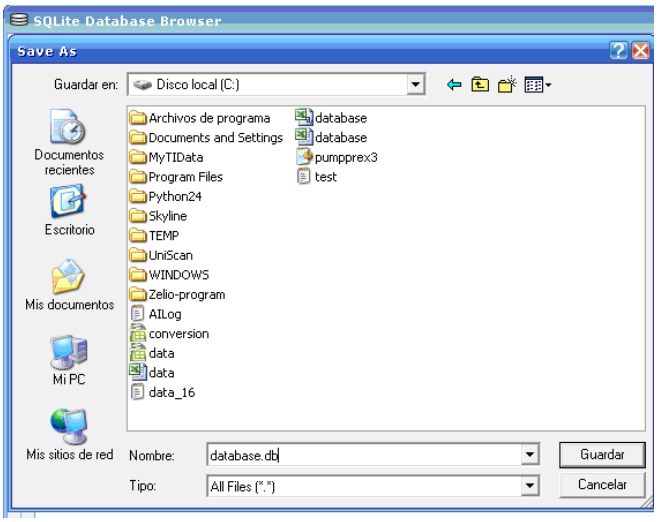
Once you've created the CSV file is run SQLite Database Browser, which is also an open source program. It can be downloaded from the website www.souceforge.net



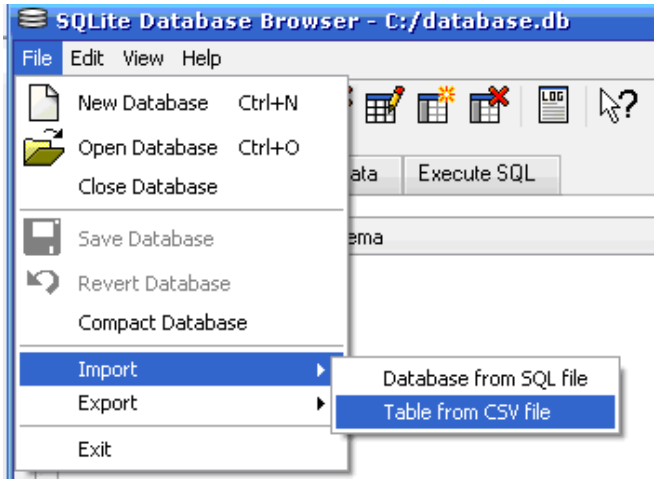

SQLite Database Browser lets you create the database in a very easy way, because they do not need to know SQL. Besides it is very close to the appearance of a spreadsheet.



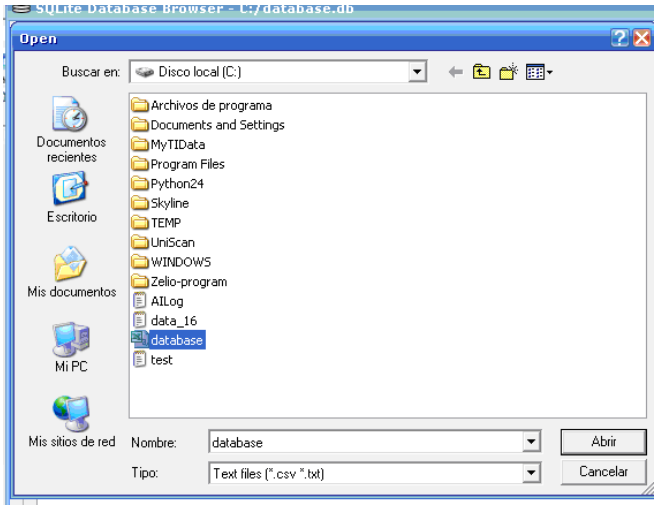

Here is a database where insert our information is created. This can be done through the menu "File> New DataBase [Ctrl + N]"



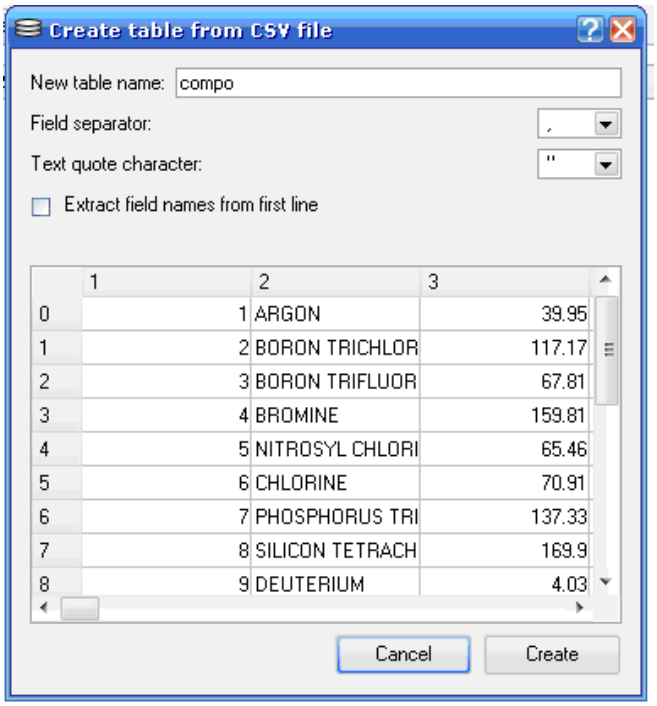

By selecting the Save option this window you can select the directory and name for the database appears. For this example the name database.db used.



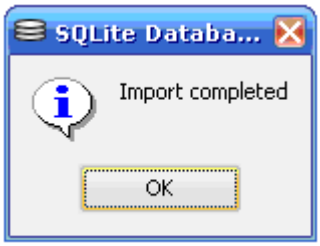

Then the spreadsheet format created previously imported. To select the directory and file name to import the File> Import> Table from CSV file menu is selected.



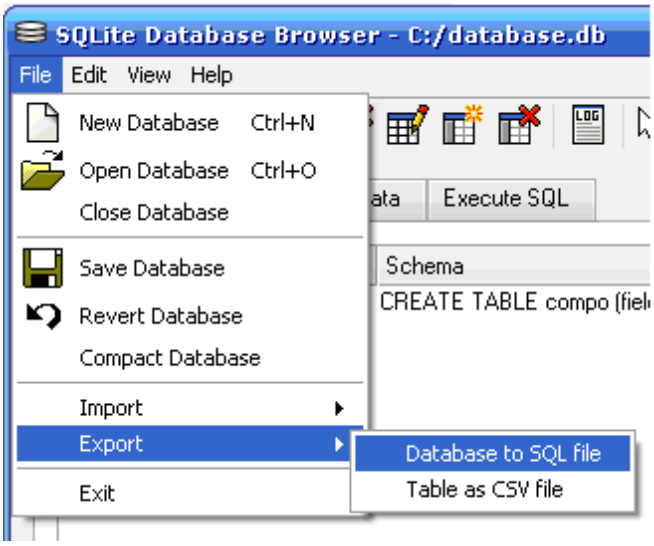

Through this window the directory and file name is selected. For this example the file has the name database.csv and finally open option is selected.




This window shows the result of the process of importing the CSV file. In the "New table name:" name "compo" is written, this is the name by which OllinTS accesses the database. To create the base for imported s data select the "Create" option

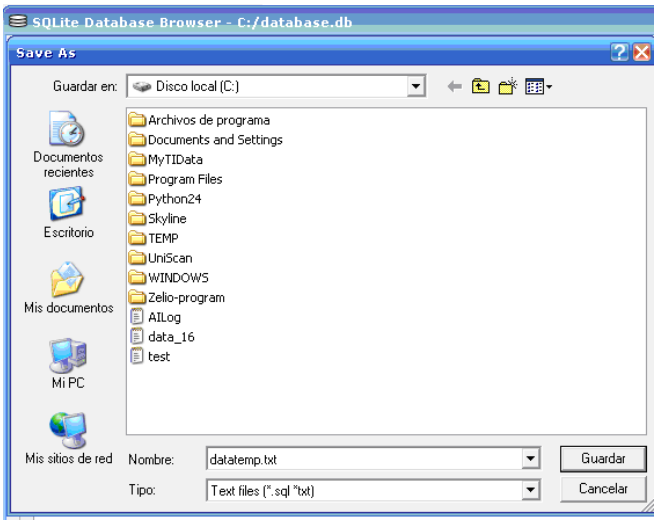



This sale confirms that imported information. Now we proceed to give the correct information to the fields of the database names.



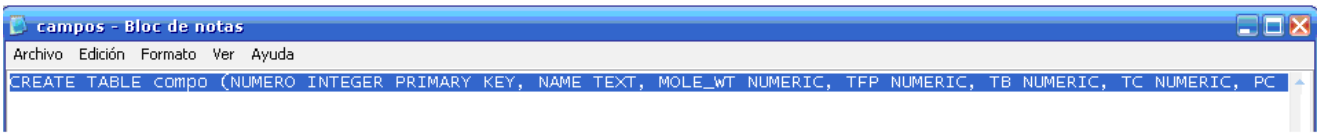

database in SQL format is exported, to change the name of the fields more easily.

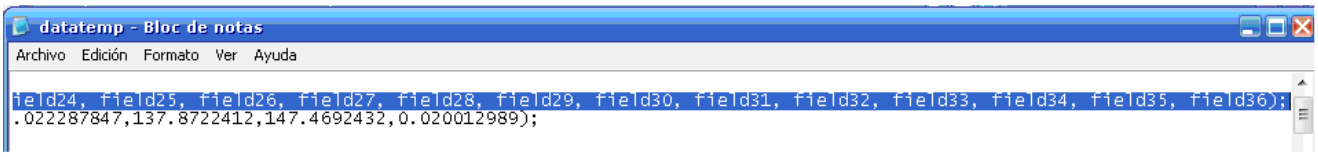
select the action: File> Export> Database to SQL file.




In selling the name given to the database in the format "txt" for this example the name is "datatemp.txt" option and select "Save"

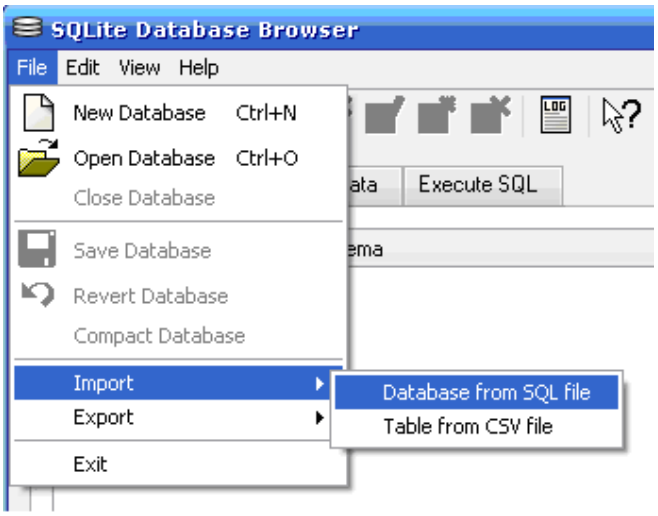
This window indicates that the information is to successfully exported.



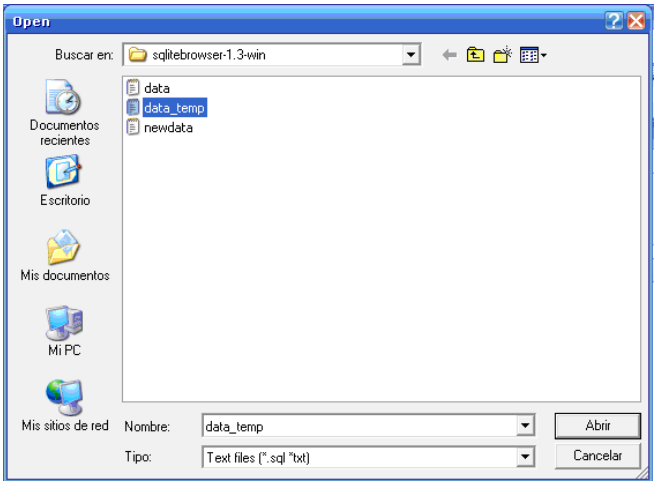




Then the "datatemp.txt" file and the "campos.txt" file containing the name opens

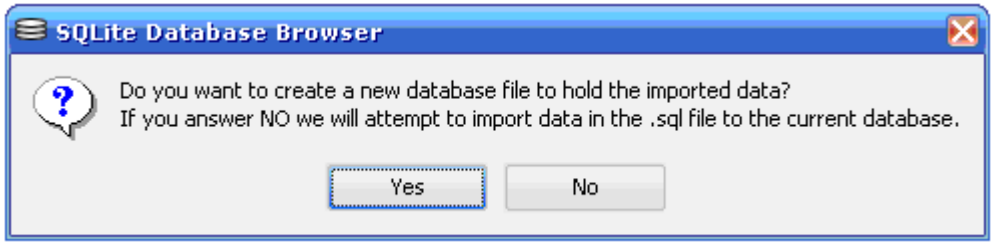
fields such as the need OllinTS which the line that begins with "CREATE" and ends with "copy; "I'll replace the file" datatemp.txt "by the line that starts with" CREATE "and ends in"; ". At the end we keep the "datatemp.txt" file.



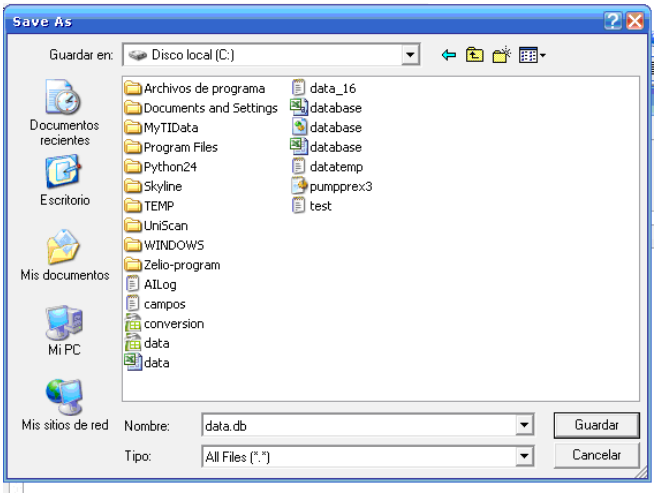

To import the corrected database, select it from the menu SQLite Data Browser: File> Import> DataBase from SQL file.



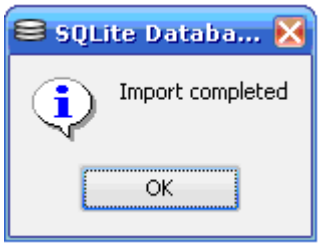

Through this window the modified file "datatemp.txt" is selected, and "Open" option is selected.



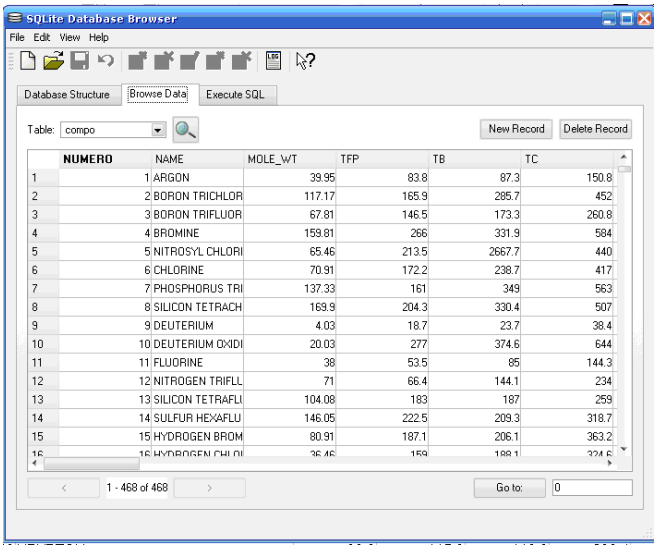

This window only appears when you have open another database, for which "YES" is selected to save the information into a new file.




The new file is given the name "data.db" because this is the name with which OllinTS calls the container file database.




This window confirms that the creation of the database completed successfully.




| | NUMERO | NAME | MOLE_WT | TFP | TB | TC |
|---|---|---|---|---|---|---|
| 1 | 1 | ARGON | 39.95 | 83.8 | 87.3 | 150.8 |
| 2 | 2 | BORON TRICHLOR | 117.17 | 165.9 | 285.7 | 452 |
| 3 | 3 | BORON TRIFLUOR | 67.81 | 146.5 | 173.3 | 260.8 |
| 4 | 4 | BROMINE | 159.81 | 266 | 331.9 | 584 |
| 5 | 5 | NITROSYL CHLORI | 65.46 | 213.5 | 2667.7 | 440 |
| 6 | 6 | CHLORINE | 70.91 | 172.2 | 238.7 | 417 |
| 7 | 7 | PHOSPHORUS TRI | 137.33 | 161 | 349 | 563 |
| 8 | 8 | SILICON TETRACH | 169.9 | 204.3 | 330.4 | 507 |
| 9 | 9 | DEUTERIUM | 4.03 | 18.7 | 23.7 | 38.4 |
| 10 | 10 | DEUTERIUM OXIDI | 20.03 | 277 | 374.6 | 644 |
| 11 | 11 | FLUORINE | 38 | 53.5 | 85 | 144.3 |
| 12 | 12 | NITROGEN TRIFLL | 71 | 66.4 | 144.1 | 234 |
| 13 | 13 | SILICON TETRAFLI | 104.08 | 183 | 187 | 259 |
| 14 | 14 | SULFUR HEXAFLU | 146.05 | 222.5 | 209.3 | 318.7 |
| 15 | 15 | HYDROGEN BROM | 80.91 | 187.1 | 206.1 | 363.2 |

In the end I could check that the database is complete and no errors in the "Browse Data" tab

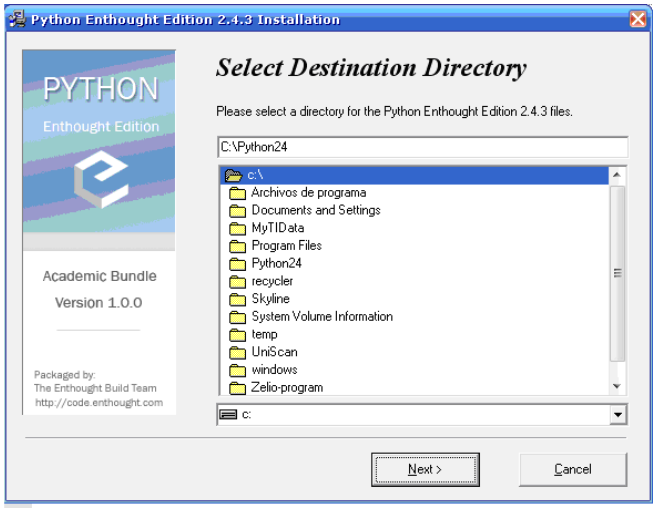
The new database must be copied to the DataBase folder inside the folder OllinTS.

for the operating system Windows XP is C: \ Python24 \ OllinTS \ DataBase

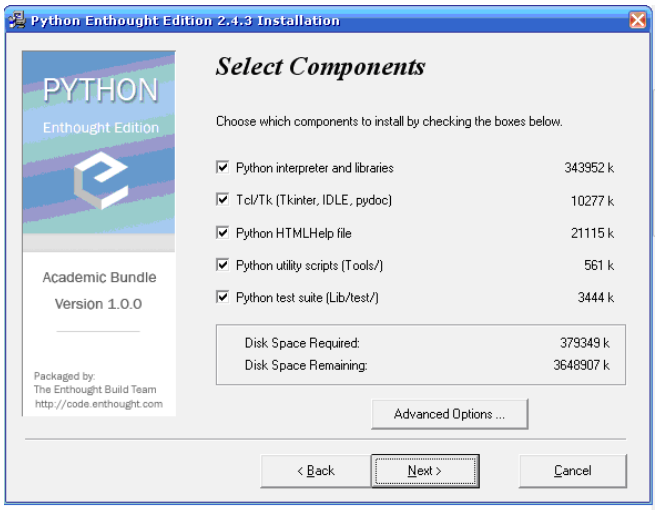
# Appendix C: Installation Procedure in Windows XP

OllinTS for installing Windows will use a modified version of Pyhton (Python Enthought Edition) which includes all necessary libraries to run OllinTS, this version can be downloaded from the website http://code.enthought.com/enthon/ . We will describe the process for installing Python and OllinTS.

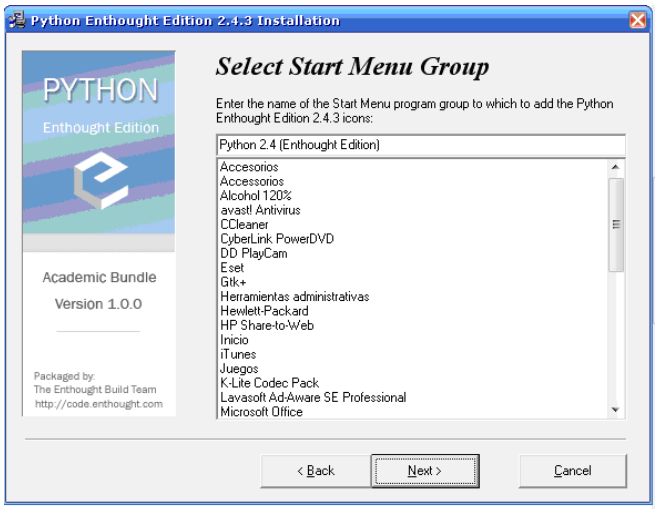
Run the installation program Python named "enthon-python2.4-1.0.0".



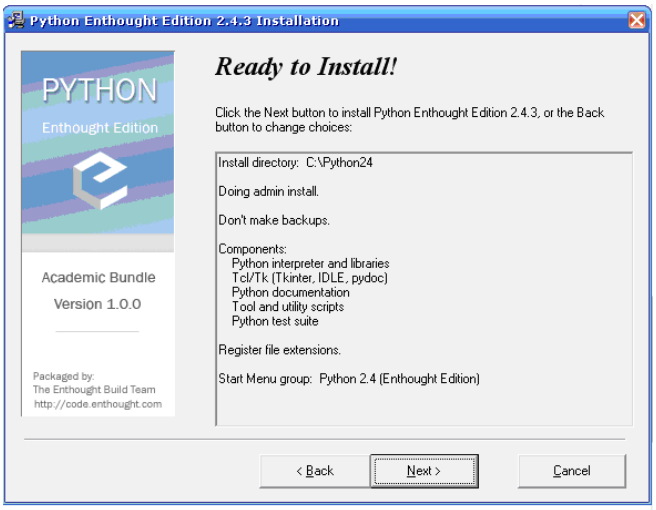

the installation direction is selected, it should be left as is set ( C: \ Python24 ) ,
Select "Next"



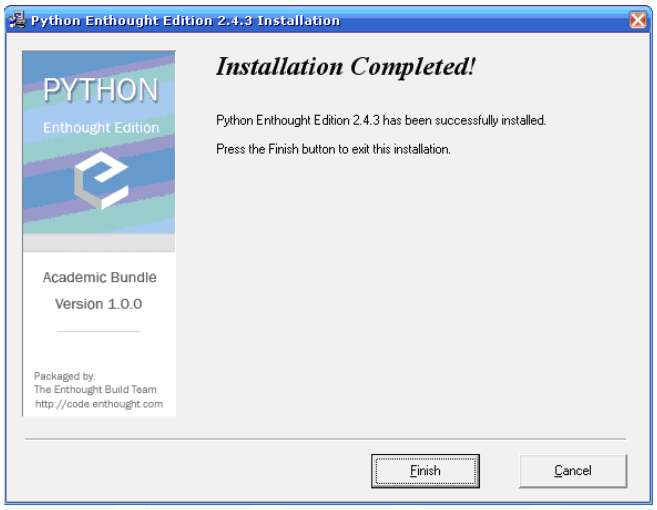

This window components installed with Python are selected, all boxes are selected. Select "Next"



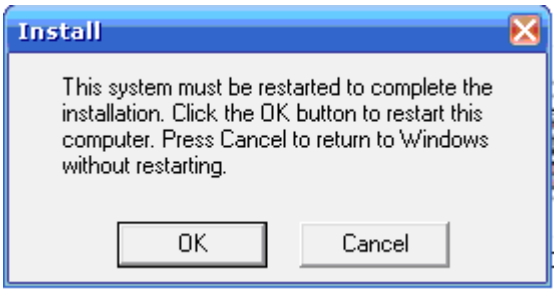

In this sale the group where shortcuts were created is selected, it is recommended to use the default. Select "Next"




In this window selected options before installing shown. Select "Next"




This window confirms that Python is installed.




Finally the computer is rebooted to complete the installation.

After the installation of Python, we proceed to copy the folder to the folder OllinTS Python, which is the route C: \ Python 24 \.

Now you can run OllinTS.

# Appendix D: Nomenclature OllinTS

*Table 19. Nomenclature OllinTS*

| Variable | units | Specifications |
|---|---|---|
| T | $^{o}K$ | Temperature |
| P | *kpa* | Pressure |
| FracVap | - | vaporised fraction |
| xf | - | Mol fraction liquid fas |
| f | - | Mol fraction gas phase |
| x | - | mole fraction of the mixture |
| zl | - | Compressibility factor liquid phase |
| ZV | - | Gas phase compressibility factor |
| CoefPureVap | - | Fugacity coefficient of numerical pure compounds gas- phase Arrangement |
| CoefMixVap | - | Fugacity coefficient of compounds in gas- phase mixture numerical Arrangement |
| CoefMixVLiq | - | Fugacity coefficient of the compounds in admixture Liquid phase numerical Arrangement |
| VVI | $\frac{M^3}{kgmol}$ | Volume numerical gas- phase pure compounds Arrangement |
| VLI | $\frac{M^3}{kgmol}$ | Volume of Liquid phase pure compounds numerical Arrangement |
| Vv | $\frac{M^3}{kgmol}$ | Gas phase volume |

*Table 19. Nomenclature OllinTS (continued)*

| Variable | units | Specifications |
|---|---|---|
| vl | $\frac{M^3}{kgmol}$ | Volume of liquid phase pure compounds |
| ActivityVap | - | Activity coefficient of the gas phase pure compounds - numerical Arrangement |
| ActivityLiq | - | Activity coefficient of liquid phase pure compounds - numerical Arrangement |
| PreVap | *KPa* | Vapor pressure - numeric array |
| Ki | - | Coefficient of distribution numeric array |
| AlphaT | - | Temperature function |
| Tr | - | reduced temperature |
| fw | - | Acentric factor function and reduced temperature - numerical Arrangement |
| to | - | Factor "a" for the cubic equation of state |
| TO | - | A factor for the cubic equation of state |
| B | - | Factor B for the cubic equation of state |
| DADT | - | First derivative of the factor "a" for the cubic equation of state |
| d2adT2 | - | Second factor derived from "a" to the cubic equation of state |
| MolWt | $\frac{kg}{kgmol}$ | average molecular mass of the mixture |
| MolWt_l | $\frac{kg}{kgmol}$ | average molecular weight of the liquid phase |
| MolWt_v | $\frac{kg}{kgmol}$ | average molecular mass of the vapor phase |
| LiqDen | $\frac{kg}{M^3}$ | average liquid density |
| Cp_v | $\frac{KJ}{Kgmol^oK}$ | heat capacity at constant pressure vapor-phase numerical Arrangement |
| Cv_v | $\frac{KJ}{Kgmol^oK}$ | heat capacity at constant volume vapor-phase numerical Arrangement |

*Table 19. Nomenclature OllinTS (continued)*

| Variable | units | Specifications |
|---|---|---|
| HF | $\frac{KJ}{kgmol}$ | Standard Power Training |
| GF | $\frac{KJ}{kgmol}$ | Gibbs free energy of formation |
| G | $\frac{KJ}{kgmol}$ | Gibbs free energy mix |
| H | $\frac{KJ}{kgmol}$ | Enthalpy of mixing |
| S | $\frac{KJ}{kgmol}$ | Entropy mix |
| OR | $\frac{KJ}{kgmol}$ | internal energy of the mixture |
| Afree | $\frac{KJ}{kgmol}$ | Helmholtz free energy |
| G_v | $\frac{KJ}{kgmol}$ | Gibbs free energy of gas phase |
| G_L | $\frac{KJ}{kgmol}$ | Gibbs free energy liquid phase |
| H_v | $\frac{KJ}{kgmol}$ | Gas phase enthalpy |
| h_l | $\frac{KJ}{kgmol}$ | free enthalpy of the liquid phase |
| S_v | $\frac{KJ}{Kgmol^{o}K}$ | Entropy gas phase |
| S_L | $\frac{KJ}{Kgmol^{o}K}$ | Free entropy of the liquid phase |
| U_v | $\frac{KJ}{kgmol}$ | internal energy of the gas phase |
| U_l | $\frac{KJ}{kgmol}$ | internal energy of the liquid phase |
| AFree_v | $\frac{KJ}{kgmol}$ | Helmholtz free energy of the gas phase |
| AFree_l | $\frac{KJ}{kgmol}$ | Helmholtz free energy of the liquid phase |

*Table 20. Nomenclature database*

| OllinTS name Ollin.DataBase.SysData | Name Database | Description | units |
|---|---|---|---|
| ZC | ZC | Critical compressibility factor | - |
| OMEGA | OMEGA | Pitzer acentric factor | - |
| *LIQDEN* | *LIQDEN* | liquid density | |
| *TDEN* | *TDEN* | Temperature of the liquid density | *° K* |
| *DIM* | *DIM* | Dipole momentum | - |
| CP_A | CP_A | Coefficient of heat capacity of ideal gas A | $\frac{KJ}{Kgmol^{o}K}$ |
| CP_B | CP_B | Coefficient of heat capacity of ideal gas B | $\frac{KJ}{Kgmol^{o}K}$ |
| CP_C | CP_C | Coefficient of heat capacity of ideal gas C | $\frac{KJ}{Kgmol^{o}K}$ |
| CP_D | CP_D | Coefficient of heat capacity of ideal gas D | $\frac{KJ}{Kgmol^{o}K}$ |
| VL_B | VISC_LIQ_B | Viscosity coefficient of fluid B | *Cp* |
| *VL_C* | *VISC_LIQ_C* | Viscosity coefficient of liquid C | *Cp* |
| *DELHF* | *DEL_HF* | Energy standard training | $\frac{KJ}{kgmol}$ |
| *ANTA* | *ANTOINE_VP_TO* | A coefficient for Antoine equation for | *MmHg P = T = ° K* |

*Table 20. Nomenclature database (continued)*

| OllinTS name Ollin.DataBase.SysData | Name Database | Description | units |
|---|---|---|---|
| *ANT_B* | *ANTOINE_VP_B* | Coefficient B for for Antoine equation | *MmHg P = T = °K* |
| *ANT_C* | *ANTOINE_VP_C* | Coefficient C for Antoine equation | *MmHg P = T = °K* |
| *ANT_MAX* | *TMAX* | maximum temperature for Antoine equation | *°K* |
| *ANT_MIN* | *TMIN* | minimum temperature for Antoine equation | *°K* |
| *HAR_A* | *HARLACHER_VP_A* | A coefficient for equation for Harlacher | *MmHg P = T = °K* |
| *HAR_B* | *HARLACHER_VP_B* | Coefficient B for the equation for Harlacher | *MmHg P = T = °K* |
| *HAR_C* | *HARLACHER_VP_C* | Coefficient C for equation Harlacher | *MmHg P = T = °K* |
| *HAR_D* | *HARLACHAR_VP_D* | Coefficient D for equation Harlacher | *MmHg P = T = °K* |
| *HV* | *HV* | Standard vaporization heat | $\frac{KJ}{kgmol}$ |
| *RK_A* | *RK_ac* | ac constant for the Redlich-Kwong case | - |
| *RK_B* | *RK_b* | b constant for the case Redlich-Kwong | - |

*Table 20. Nomenclature database (continued)*

| OllinTS name Ollin.DataBase.SysData | Name Database | Description | units |
|---|---|---|---|
| *SRK_A* | *SRK_ac* | ac constant Soave-Redlich case-Kwong | - |
| *PR_A* | *PR_ac* | ac constant for the case Peng-Robinson | - |
| *PR_B* | *PR_b* | b constant for the case Peng-Robinson | - |

# Appendix E: Glossary

- **Free Software [1] (English *free software):*** It is software that, once obtained, can be used, copied, studied, modified and redistributed freely. There is no need to apply any license and whose exploitation rights are for all mankind, because it belongs to everyone equally. Anyone can make use of it, provided for lawful purposes and consigning its original authorship. This software would be one whose author gives to mankind or whose copyrights have expired. If an author conditional use under a license, however weak it may be, it is no longer public.

- **not free software [2] (** also called proprietary software, proprietary software, proprietary software, software with proprietary or proprietary software): Refers to any software where users have limited possibilities to use, modify or redistribute it (with or without modification), or whose source code is not available or access to it is restricted.

For the Foundation for Free Software (FSF) this concept applies to any software that is not free or only is partially (semi-free), or because their use, redistribution or modification is prohibited, or requires express permission of the owner of the software . Non-free software in a natural or legal person owns the copyright on software denying or granting at the same time, the rights to use the program for any purpose; to study how the program works and adapt it to their own needs; distribute copies; or to improve the program and make public improvements.

- **GNU General Public License (GPL):** The author retains the copyright (copyright), and permits redistribution and modification under terms designed to ensure that all modified versions of the software remain under the more restrictive terms of the GNU GPL itself. This makes it not impossible to create a product with parts unlicensed GPL: the

assembly must be GPL.

- **BSD style licenses ( *Berkeley Software Distribution):*** The author, under such licenses, maintains copyright protection only for the disclaimer of warranty and require proper attribution of authorship in derivative works, but allows free redistribution and modification, even if such work their ownership. They are very permissive, so that are easily absorbed when mixed with the GNU GPL with whom they are compatible.

- **OllinTS:** Thermo Server Ollin Ollin is a Nahuatl word meaning movement, the cosmic. In the glyphs it is symbolized by a figure that resembles a butterfly with open wings.

*Figure 35. Glyph Ollin*

## Bibliography

1: Wikipedia, Free Software, http://es.wikipedia.org/wiki/Software_libre , Last Accessed June 2007.

2: Wikipedia, owner Software, http://es.wikipedia.org/wiki/Software_no_libre , Last Accessed June 2007.

3: Svrcek W., Mahoney D. & Young B., A realtime approach to process control, Industrial Engineering Chemistry Research, No. 8, V. 39, 2000, pp. 29983006.

4: Candle M.A., Use fraction for recycle balance. Part I. Fractions separated. Part II. Types of seprations, Hydrocarbon Processing and Petroleum Refiner, No. 56, V. 40, May-June 1961.

5: Naviev M. F., Material balance in complez and recycle multistage chemical Processes, Chemical Engineering Progress, No. 6, V. 53, 6 June 1957, pp. 297,303.

6: Donald I. Rubin, Generalized Material Balance, Chemical Engineering Progress Symposium Series, No. 3758, 1962, pp. 5461.

7: E. M. Rosen, A method for machine computation prerforming Material balances, Chemical Engineering Progress, No. 10, V. 58, October 1962, pp. 6973.

8: Kesler M. G., A computer simulation process for sistem, Proc. American Petroleum Institute, Vo. 43, Sec. III, 1963, pp. 4956.

9: Naphtali L. M., Process heat and materials balances, Chemical Engineering Progress, No. 9, V. 60, September 1964, pp. 7074.

10: R. W. H. Sargent, Sppeadup in chemocal Engineering desing, Trans. INSTN. Chem. Engrs., Vol. 42, 1964, pp T190T197

11: A. H. Lindahl, Refinig Improve Operation With Process Simulation, Chemical Engineering Progress, N. 4, V. 61, April 1965, pp. 7781.

12: Wooyoung Lee, Desing Variable selection process to simplify calculation, American Institute of Chemical Engineers Journal, November 1966, pp. 11841190.

13: Lake Forest, SIMCCIESSCOR 19,672,007, Invensys, December 2006.

14: R. W. H. Sargent, Integrated Desing and Optimization of progress, Chemical Engineering Progress, No. 4, V. 63, September 1967.

15: Komatsu Shoei, Aplication of linearization to desing of a hidroalkylation plant, Industrial and Engineering chemestry, No. 2, V. 60, February 1968, pp. 3643.

16: Evans LB, Computer Aided Chemical Process desing, Chemical Enginering Progress, No. 4, V. 64, April 1968, pp. 3946.

17: K'hat L., Chemical Process simualtion, Proc. Tech. International, Nos. 1,2,3,4 and 5, Vol. 64, Part. I, II, III, January, February, March, April, May 1973, pp. 3537.

18: Pho TK, Topics in computer aided desing: Part I.'An Optimum Algorithm for recycle system tearing, American Institute of Chemical Engineers Journal, No. 6, V. 19, November 1972.

19: Scenna NJ, Modeling, Simulation and Optimization of Chemical Processes, Edit. National Technological University, Ed. 1st, Buenos Aires, Argentina, 1999, Cap. V, VI, VII, VIII, IX, http://www.modeladoeningenieria.edu.ar/libros/modeinge/modinge_f.htm , Last access August 2007.

20: Motard R. L., Chemical Process Simulation Steady State, American Institute of Chemical Engineers Journal, No. 3, V. 21, May 1975, pp. 417,436,

21: A Chemical Engineering Timeline, http://www.pafko.com/history/h_time.html , Last visit in July 2006.

22: Kakhu A. et al. , An open source architecture for process modeling and modelbased applications, Third International Conference on Process Operation fundations of computerAided, 1998, pp. 518524, 1998.

23: Neri H., simulation of chemical processes, ESIQUIEIPN, Professional Thesis, 1976, Cap. I.

24: Rafiqul Gani, Property modeling and simulation for product and process desing, ElselvierFluid phase equilibria, 2002, pp. 4359.

25: Wiese, B. S., Chemical engineering application and software components for the next century, Presented at Chemputers, Houston, TX, 1998.

26: Sim, DW, Component Software Engineering: Lego for Process Engineers, http://www.hyprotech.com/support/papers/engineer/default.asp , Last access February 2005.

27: P. D. Edwards & G Merkel, Impact of an open architecture environment on the design of software components for process modeling, International conference on intelligent systems in process engineering,

1996, pp. 307-310.

28: L. Wedel, A case study in componentbased process simulation, Foundations of Computer-Aided Process Operations (FOCAPO), 2000, pp. 494-497.

29: Lien, K. & T. Perris, Future directions for research CAPE, Computers & Chemical Engineering, 2000, p. 1551.

30: Braunschweig, B. L., Pantelides, C., Britt, H. I. & S. Sama, Open architectures for process modeling software: current status and future perspectives, 5th International Conference on Foundations of computer- aided design process, 1996, pp. 220-235.

31: Yeow Peng Lee, Gade Pandu Rangaiah, Rein Luus, Phase and chemical equilibrium calculation by direct search optimization, Computer and Chemical Engeneering, 1999, pp. 11831191

32: I. And G. S. Melaaent Owren, Computers chem. Engn., How do inacurancies of enthalpy and vaporliquid equilibrium baseload LNG plant influence calculation desing ?, No.1, V. 20, 1995, pp. 111

33: René BañaresAlcántara, Arthur W. Westerberg and Michael D. Rychener, Development of an expert system for physical property prediction, Computers & Chemical Engineering, No. 2, V. 9, 1985, pp. 127142

34: RC Reid, J.M. Prausnitz & T. K. Sherwood, The property of gases and liquids, Edit. McGraw-Hill, Ed. 3, 1977, Apendix A.

35: Wikipedia, SQL, http://es.wikipedia.org/wiki/SQL , Last Access Agosto2007.

36: www.python.org, Last Access May2007.

37: www.sqlite.org , Last accessed May 2007.

38: http://numpy.scipy.org , Last Access May2007.

39: http://matplotlib.sourceforge.net/ , Last Access May2007.

40: RC Reid, J.M. Prausnitz & T. K. Sherwood, The property of gases and liquids, Edit. McGraw-Hill, Ed. 3, 1977, Apendix A.

41: SQlite Data Browser, http://sqlitebrowser.sourceforge.net/ , Last Access Junio2007.