APPLICATION OF COMPUTER SIMULATION AND REGRESSION TECHNIQUES TO THE DEVELOPMENT OF AN EQUATION OF STATE FOR R134a AND VAPOR PRESSURE EQUATIONS FOR CARBON DIOXIDE, PROPANE AND R134a

by

TAPAN BHATT

A thesis submitted to the Faculty and Board of Trustees of the Colorado School of Mines in partial fulfillment of the requirements for the degree of Master of Science (Chemical Engineering and Petroleum Refining).

Golden, Colorado
Date 24th AUG '93

Signed:
Tapan Bhatt

Approved: James F. Ely
Dr. James F. Ely
Thesis Advisor

Golden, Colorado
Date 8/24/93

Dr. Robert Baldwin
Professor and Head
Department of
ChemicalEngineering
& Petroleum Refining

ABSTRACT

Accurate prediction of thermophysical properties is essential to design effective and efficient systems. Equations of state valid over a wide range of conditions are needed to predict various thermophysical properties accurately. The work presented here aims at developing such equations using complex regression and computer simulation techniques. The primary regression technique used for equation of state development was the stepwise regression technique. Multiproperty fitting is used in conjunction with the stepwise regression method to develop a new equation of state. The equation of state is developed for a refrigerant R134a, which is an alternative refrigerant. Results and statistical comparisons indicate the new equation to be comparable with existing equations of state for R134a. Also the new equation has the advantage of fewer terms and Helmholtz free energy form, which makes it more convenient to derive properties like C_p, C_v, sound velocity, etc.

Simulated annealing is another regression technique studied in detail here. It is used to develop vapor pressure equations for carbon dioxide, propane and R134a. The vapor pressure comparisons indicate the new equations to be quite comparable to the existing equations. Detailed graphs and statistical comparisons are presented for all the models developed in this study.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACKNOWLEDGEMENTS

I would like to thank Dr. James F. Ely for offering me the opportunity to work with him and for his constant support and guidance during the course of this work. I am grateful to the Department of Energy for providing the funding for this project under Grant No DE-FG02-92ER14294. I would like to show my appreciation for my committee members, Dr. Sloan and Dr. Way.

I am grateful to my parents for their understanding and encouragement. Special thanks to my friends for their moral support.

1. INTRODUCTION

1.1 General Background

Chemical engineering is an applied science in that it utilizes conservation principles and other theoretical results to solve practical problems. Unfortunately, apart from conservation principles, there are few exact theoretical results available to the practicing engineer. For example, many exact results are known in transport phenomena involving laminar flow, but few are known for more common turbulent regimes. Similarly, in thermodynamics, many exact results are known for ideal gases but few are available for compressed gases or liquids. In order to perform accurate and innovative process design, it is therefore necessary to develop high accuracy empirical models for physical behavior. In the case of thermodynamics, this means equations of state and, for example, mixing rules.

The aim of this work is to explore the application of computer simulations to systematically develop highly accurate empirical equations of state which are valid over a broad range of thermodynamic conditions. In addition to the wide range equation of state work, vapor pressure equations have been developed for propane, carbon dioxide and R134a. The fluid primarily studied in this work is, in fact, refrigerant R134a (1,1,1,2-tetrafluoroethane). Refrigerant R134a has generated a lot of interest among refrigeration engineers and thermodynamicists because of its minimal detrimental environmental impact. In particular, one of the most widely used refrigerants across the world today is R12 (dichlorodifluoromethane). R12 is a chloro-fluorocarbon(CFC), a class of chemical compounds that has been implicated as a cause of depletion of the ozone layer and resulting global warming. In order to minimize the detrimental effects of refrigerants on the environment, a treaty known as the Montreal Protocol has been enacted. This treaty proposes a rigorous schedule for phasing out use of the CFCs all around the world. R134a has emerged as a leading substitute for R12, because its thermodynamic properties are similar to those of R12 and it has low acute and chronic toxicity and almost zero ozone depletion potential. As a direct consequence of this, R134a will be used as a working fluid in refrigeration systems such as home refrigerators and automobile air conditioners. For effective and efficient design of these refrigeration systems, accurate prediction of thermophysical properties of R134a is required. Because of this, there has been a great interest in measuring and predicting the thermophysical properties of R134a.

In this work, an innovative approach is used to develop an equation of state for R134a. A modified stepwise regression algorithm is being used in conjunction with constrained multiproperty fitting to develop an accurate equation of state. Studies were also performed using the simulated annealing method as a possible alternative to the stepwise regression method. In the remainder of this section, the theory and practical applications of the simulated annealing method, stepwise regression method and multiproperty fitting will be reviewed and discussed.

1.2 Regression Analysis

As the need for finding optimal solutions to complex engineering problems grows, so does the need to find more efficient techniques to accomplish that goal. To that end, computer simulation, optimization and regression techniques have become an integral part of present day engineering practice. Linear optimization problems (e.g., problems which have a linear dependence on model parameters) can be solved using linear regression techniques[1,2] better known as the least squares algorithm. That is to say that the least squares algorithm[3] provides an optimal solution to the linear optimization problem given the functional form and weighing scheme. The most common application of the least squares method in chemical engineering is that of finding the best possible solution to represent the given data using a known functional form. Using least squares in exceedingly complex, multivariate problems such as those of process optimization, equation of state modelling, design of electronic circuits etc., leads to different correlations based upon the assumed functional forms. Due to this, numerous functional forms would have to be examined to find the optimal representation. This introduces an effective non-linearity into the problem. Unfortunately, when the optimization problem is non-linear, least-squares-like approaches do not guarantee an optimal solution. That is to say that each step in a non-linear least squares optimization moves in the direction of a minimum of the objective function, but can be "trapped" in local minima.

In this work we have explored a solution to the multiple functional form problem using a less well known adaptation of linear regression known as the stepwise regression method. Stepwise regression is very efficient in solving multivariate optimization problems, but in its native form cannot move "uphill" in non-linear optimization process.

The basic form of a linear regression equation is as follows:

$$y = \Sigma a_m f_m(\mathbf{x}) \qquad (1.1)$$

where y is the dependent variable, $\mathbf{x}$ is the set of independent variables, f_m are the functional forms, and a_m are the coefficients of the functional forms. The goal of regression analysis is to determine these a_m coefficients. By doing so, the aim is to arrive at an equation which best represents the data.

Regression analysis could be described in general terms by the following main points[4] :

- Determine the most important contributory terms $f_m(\mathbf{x})$ from a general expression $y = y(\mathbf{x}_i)$.
- Evaluate the coefficients for the selected terms.
- Examine the significance of each of the selected terms using statistical tools.
- Select the best equation based on the results of the statistical tests.

The stepwise regression method is a systematic application of the regression technique described above. Since it has linear regression at its core, a detailed explanation of the least squares method is presented before discussing the stepwise regression technique.

1.2.1 Linear Regression

Consider the following regression equation

$$y=\sum_{m=1}^{M} a_m f_m(\boldsymbol{x}) \tag{1.2}$$

where y is the dependent variable, $f_m(\mathbf{x})$ denotes a functional form, a_m is the coefficient of the m^{th} term, **x** denotes the K independent variables and M is the number of terms in the equation. If there are N experimental data points, then the weighted sum of squares of deviations between eqn. (1.2) and experimental observations can be calculated as

$$S = \sum_{n=1}^{N} w_n \, (y_{obs}-y_{calc})_n^2 \tag{1.3}$$

where S is the weighted sum of squares, w_n is the weight of the n^{th} data point, y_{obs} is the observed value of dependent variable, and y_{calc} is the calculated value of the dependent variable from eqn. (1.2). The weights are usually calculated from the variance of the dependent variable using the Gaussian error propagation formula :

$$w_n = \frac{1}{\sigma_{I_n}^2} \tag{1.4}$$

where

$$\sigma_{I_n}^2 = \sigma_{Y_n}^2 + \sum_{k=1}^{K} \left(\frac{\partial y}{\partial x_{nk}} \sigma_{x_{nk}} \right)^2 \tag{1.5}$$

σ_Y is the error in the dependent variable and σ_X is the error in the independent variable.

Minimizing eqn (1.3) with respect to the set $\{a_m\}$, one finds M normal equations in M unknowns.

$$\sum_{n=1}^{N} w_n f_{nm'} \left(\sum_{m=1}^{M} f_{nm} a_m \right) = \sum_{n=1}^{N} w_n y_{n,obs} f_{nm'} \tag{1.6}$$

Rearranging equation (1.6), we find,

$$\sum_{m=1}^{M} a_m \sum_{n=1}^{N} w_n f_{nm} f_{nm'} = \sum_{n=1}^{N} w_n y_{n,obs} f_{nm'} \tag{1.7}$$

where m′ ranges from 1 to M. This equation can be written in matrix notation as

$$[\boldsymbol{F}]\ [\boldsymbol{A}] = [\boldsymbol{Y}] \tag{1.8}$$

[F] is a square matrix of order M where each element of the matrix F is given by

$$F_{ij} = \sum_{n=1}^{N} w_n f_{ni} f_{nj} \qquad (1.9)$$

[Y] is a column matrix whose elements are given by,

$$Y_i = \sum_{n=1}^{N} w_n \, y_{n,obs} \, f_{ni} \qquad (i=1,2..,M) \qquad (1.10)$$

and [A] is a column matrix of order M whose elements are the unknown coefficients a_j's.

1.2.2 Constraints

Constraints are an integral part of any regression technique and are primarily added to obtain a better fit at or around a specific point. Consider an thermodynamic equation of state, where the most difficult region to fit is around the critical point. This problem can be rectified by adding constraints which force the equation to pass through the critical point, i.e., forcing the equation to have $p=p_c(\rho_c,\tau_c)$, $(\partial p/\partial \rho)_c=0$, and $(\partial^2 p/\partial^2 \rho)_c=0$ where c denotes the critical point.

If L is the number of constraints and M is the number of normal equations, constraints can be added either by eliminating L equations from the M normal equations or by using the method of Lagrangian undetermined multipliers proposed by McCarty and Hust[5]. Mathematically, the constraints are given by,

$$\sum_{m=1}^{M} a_m f_{lm} = c_l \qquad (l=1,2,\ldots\ldots,L) \tag{1.11}$$

where f_{lm} is the value of the functional form evaluated at the constraint conditions and c_l is the actual value of the l^{th} constraint. In the method of Lagrangian multipliers one forms an augmented weighted sum of squares given by

$$Q = S + \sum_{l=1}^{L} \lambda_l \left(\sum_{m=1}^{M} a_m f_{lm} - c_l \right) \tag{1.12}$$

where Q is the augmented weighted sum of squares with constraints, S is given by eqn. (1.3) and λ_l is a Lagrange multiplier.

In order to minimize Q, the constraints have be satisfied exactly and hence the second term on the right hand side of equation (1.12) will be zero. Thus, when Q is a minimum S is also minimized. Performing the differentiation of Q with respect to the unknown coefficients a_m, and setting the derivative equal to zero, one finds

$$\frac{\partial S}{\partial a_m} + \sum_{l=1}^{L} \lambda_l f_{lm} = 0 \qquad (m=1,2,.....,M) \qquad (1.13)$$

where from eqn. (1.3)

$$\frac{\partial S}{\partial a_m} = \sum_{n=1}^{N} 2w_n \left(\sum_{m=1}^{M} a_m f_{nm} - Y_n\right) f_{nm} \qquad (1.14)$$

Substituting eqn. (1.13) and eqn. (1.14) into eqn. (1.12), one finds

$$\sum_{n=1}^{N} 2w_n \left(\sum_{m=1}^{M} a_m f_{nm} - Y_n\right) f_{nm'} + \sum_{l=1}^{L} \lambda_l f_{lm'} = 0 \qquad (1.15)$$

After expansion and simplification, the following equation is obtained,

$$\sum_{m=1}^{M} a_m \sum_{n=1}^{N} w_n f_{nm} f_{nm'} + \sum_{l=1}^{L} \lambda_l f_{lm'} = \sum_{N=1}^{N} w_n f_{nm'} Y_n \qquad (1.16)$$

$$m' = 1,2,......,M$$

The set of M+L equations defined by eqn. (1.16) and eqn. (1.11) can be solved simultaneously to get the coefficients a_m and Lagrange multipliers, λ_l. The sum of squares will be minimized when the constraints are satisfied exactly. As a result, the values of the Lagrange multipliers are not needed.

1.2.3 STEPWISE REGRESSION

The conventional least squares algorithm[6,7] is one in which the sum of squares between the experimental and the calculated value is minimized to find the best possible equation for a given set of functional forms. This minimum value is then used to find the coefficients of the terms in the equation. Stepwise regression is an application of the linear regression formalism presented in the previous section. In particular, it is used to address the question of which functional forms, $f_m(x)$ should be used to best represent a set of data. In the stepwise regression method[4], a collection of terms known as the "bank of terms" is generated. This bank of terms consists of all the possible terms which could be a part of the equation. The stepwise regression method selects those terms from the bank of terms which are statistically most significant.

The stepwise regression algorithm has the ability to add one term at a time to the equation. It also has the ability to exchange terms within the equation with those not in the equation but in the data bank, with the ultimate goal of reducing the sum of squares. This in turn leads to a reduction of intercorrelation between terms and results in a better equation.

In the stepwise regression method, the regression matrix [**B**] is generated,

$$[B] = \begin{bmatrix} [F] & [Y] \\ [Y]^T & [S] \end{bmatrix} \qquad (1.17)$$

where [**F**] is the MXM matrix given in eqn. (1.9), $[\mathbf{Y}]^T$ is the transpose of [**Y**] given in eqn. (1.10), [**S**] is a 1X1 matrix whose value is the weighted sum of squares for the residues of the dependent variable. Thus, [**B**] is a (M+1)X(M+1) matrix. Note that in the case of stepwise regression, M is the number of terms in the bank of terms.

The regression matrix is generated in the above manner in order to facilitate the efficient calculation of the sum of squares, i.e., the overall sum of squares can be directly obtained from the element $b_{m+1,m+1}$. The elements of the matrix [**Y**] are used to generate the change in the sum of squares upon adding the m^{th} term to the equation. Moreover, the coefficients of the terms in the equation can be directly calculated from the elements of the [**Y**] matrix.

A constrained stepwise regression matrix can be constructed from eqns. (1.16) and (1.17). Here a vector [**C**] of the order X can be constructed. The regression matrix upon adding the constraints becomes,

$$[B] = \begin{bmatrix} [F] & [C] & [Y] \\ [C]^T & [O] & [YC] \\ [Y]^T & [YC]^T & [S] \end{bmatrix} \qquad (1.18)$$

where [**B**] is a square matrix of the order M+L+1, [**O**] is a null square matrix of the order equal to the number of constraints, [**C**] is LXM matrix whose elements are functional forms under the constraint, [**YC**] is a 1XL matrix whose elements are the actual values of the constraints. In the regression matrix in eqn (1.18), the M would be the total number of terms included in the bank of terms.

The matrix generated in eqn. (1.18) is a square and symmetric matrix and hence, the operations like addition, deletion and exchange of terms can be carried out only on one half of the matrix. This is unlike the stepwise regression methods proposed by de Reuck & Armstrong[8], Draper & Smith[9] and Efroymson[10], where a unit square matrix of the same order as the regression matrix has to be generated. A column matrix [**T**] of the order of M+L is also generated to keep track of the terms being added and deleted from the equation. If the term is present in the equation, the corresponding value in the matrix [**T**] is 1, whereas if the term is not present in the equation, its value is 0.

The stepwise regression method can be explained in the following steps.

- Initial Selection of Terms

The search in the stepwise regression algorithm is usually started with a small set of preselected terms. These are the terms which are most likely to be present in the equation. If such terms are not known, the search could also be started using a randomly selected set of terms. Once the initial terms are selected and their regression matrix is initialized, a search is initiated for another term which will provide the greatest reduction in the overall sum of squares. This is found by calculating the quantities q_i for each term in the bank of terms.

$$q_i = b_{NMAX,NMAX} - \frac{b_{NMAX,i} \times b_{i,NMAX}}{b_{I,I}} \tag{1.19}$$

where q_i is the change in the sum of squares for the regression if the ith term were added to the equation and NMAX=M+L+1.

- Addition of Terms

The term which reduces the standard deviation by the greatest amount as determined from eqn. (1.19) is then added to the equation using the algorithm for adding a term. This algorithm is shown in Table 1.1.

- Statistical Tests

After a term has been added, comprehensive statistical tests are performed to test the significance of the term and 0the equation as a whole. These include the Student-t[11] test to examine the significance of each term and the Fisher F[11] test to examine the significance of the equation as a whole. As these tests are an integral part of the stepwise regression method, a brief discussion of the tests is included here for completeness.

For any equation, it is important to examine the significance of each term in the equation while executing the stepwise regression algorithm. In this context, a statistically significant coefficient is one which differs from zero with a statistical probability S of at least 99.99% and hence cannot be omitted from the equation. The student-t distribution is used to test the significance of a term

Table 1.1 Transformation of the regression matrix upon addition of the kth term.

Transformation	*Condition*
$b_{ij} = 1/b_{kk}$	$i=j=k$
$b_{ij} = b_{ij}/b_{kk}$	$i=k,\ j<k$ or $j=k,\ i>k$
$b_{ij}=b_{ij} + (-1)^{1+T[i]}\ b^2_{ik}/b_{kk}$	$i=j>k$
$b_{ij}=b_{ij} + (-1)^{1+T[i]}\ b^2_{ki}/b_{kk}$	$i=j<k$
$b_{ij} = b_{ij} - b_{ik}b_{jk}/b_{kk}$	$i>k, j>k, i\neq j$
$b_{ij} = b_{ij} - b_{ik}b_{kj}/b_{kk}$	$i>k, j<k, i\neq j$
$b_{ij} = b_{ij} - b_{ki}b_{kj}/b_{kk}$	$i<k, j<k, i\neq j$
$1 \preceq i \preceq NMAX, \quad 1 \preceq j \preceq NMAX, \quad i \succeq j$	

already present in the equation. It is initially assumed that all the coefficients of the terms present in the equation are zero. Let a_i be the coefficient of the i^{th} term. The null hypothesis would be that $a_i=0$. According to definition, if the statistical probability of the test applied to the null hypothesis is greater than a specified value, then the hypothesis is rejected, otherwise, it is accepted.

For the coefficient a_i, the Student-t statistic is given by,

$$t_i = \frac{a_i}{\sigma_{n_i}} \tag{1.20}$$

The standard deviation, σ_i, of each coefficient a_i already in the equation is given by

$$\sigma_{n_i} = [b_{LL}\, b_{ii}/(N-M_a)]^{\frac{1}{2}} \tag{1.21}$$

where M_a is the number of terms already present in the equation and N is the number of data points.

The statistical probability that the coefficient a_i differs from zero can be obtained from the probability of the Student-t distribution,

$$S_i = \left(\frac{1}{\sqrt{\pi\nu}}\right)\ [\Gamma(\frac{\nu+1}{2})/\Gamma(\nu/2)] \int_{-t_i}^{t_i} (1+\frac{t^2}{\nu})^{-(\nu+1)/2} dt \tag{1.22}$$

where υ is degrees of freedom and is equal to the difference between the number of data points and number of terms in the equation and the t_i's are obtained from eqn. (1.20)

If the value of S_i is greater than a user specified value (around 99.99%), then the value of the coefficient being tested is significantly different from zero and hence the zero hypothesis is rejected. The terms for which the zero hypothesis is rejected are retained in the equation. On the other hand, if the probability of any of the terms in the equation were less than the user specified value, then that term would be eliminated from the equation using the deletion procedure described below. The whole purpose of this test is to minimize the intercorrelation between terms. Intercorrelation between terms means that different variables/terms exhibit the same behavior in regard to the regression of the quantity to be determined. So, in this case, if a term loses its significance, it is eliminated and thus reducing the probability of intercorrelation.

- Deletion

The terms which have already been added to the equation can also be deleted from the equation, if they do not satisfy the rigorous statistical tests outlined above. The deletion process can be summarized in the steps shown in Table 1.2.

- Fisher F : A Test of significance for the equation

Once all the individually insignificant terms have been deleted, the equation is statistically tested as a whole. This kind of testing is performed using the Fisher-F test. The Fisher-F test compares variances of different equations for the same population.

Table 1.2 Transformation of the regression matrix upon deletion of the k^{th} term.

Transformation	*Condition*
$b_{ij} = 1/b_{kk}$	$i=j=k$
$b_{ij} = b_{ij}/b_{kk}$	$i=k,\ j<k$ or $j=k,\ i>k$
$b_{ij}=b_{ij} + (-1)^{2+T[i]}\ b^2_{ik}/b_{kk}$	$i=j>k$
$b_{ij}=b_{ij} + (-1)^{2+T[i]}\ b^2_{ki}/b_{kk}$	$i=j<k$
$b_{ij} = b_{ij} + b_{ik}b_{jk}/b_{kk}$	$i>k, j>k, i\neq j$
$b_{ij} = b_{ij} + b_{ik}b_{kj}/b_{kk}$	$i>k, j<k, i\neq j$
$b_{ij} = b_{ij} + b_{ki}b_{kj}/b_{kk}$	$i<k, j<k, i\neq j$
$1 \le i \le NMAX, \quad 1 \le j \le NMAX, \quad i \ge j$	

Let V_1 be the variance of the first equation and V_2 be the variance of a second equation. The second equation is obtained by eliminating the term with the smallest t value. The zero hypothesis here would be that these variances are not significantly different. In other words, if the probability of the F-statistic is greater than a certain user specified value, then the null hypothesis is rejected, otherwise, it is accepted. The variance can be calculated from the matrix [B] by

$$V_1 = b_{LL}/(N-M_a) \tag{1.23}$$

and V_2 is calculated by

$$V_2 = (\ b_{LL}+\frac{b_{Li}b_{Li}}{b_{ii}})/(\ M-N_a+1) \tag{1.24}$$

where N_a is the number of terms in the equation.

The statistical probability of the Fisher F distribution can be calculated using the following equation.

$$S = [\ \Gamma(\frac{\nu_1+\nu_2}{2}/\ \Gamma(\frac{\nu_2}{2})\ \Gamma(\frac{\nu_1}{2})]\ \ (\nu_2)^{\nu_2/2}\ \nu_{\nu/2}$$

$$\times \int_0^F [\ f^{(\nu_2-2)/2}/\ (\nu_1+\nu_2 f)^{(\nu_1+\nu_2)/2}]\, df \tag{1.25}$$

where υ_1 = Degrees of freedom of actual equation, υ_2 = Degrees of freedom of equation with eliminated term and $F = V_1/V_2$.

If the probability S is greater than a user defined value (around 99.99%) then, the variances for the two equations differ significantly and the null hypothesis should be rejected. This means that deleting the term with the lowest Student-t values deteriorates the equation and that term should not be deleted. If the probability S is less than the user specified value then, both equations are quite comparable the null hypothesis should be accepted. This means that deleting the term with the lowest Student-t value does not affect the equation.

The primary goal of this test is to get rid of small intercorrelations between the terms. Sometimes, the Student-t test is not able to detect small intercorrelations between terms and hence the analysis of variance method is used to detect it and try to eliminate it. The reduction of intercorrelation between contributory terms achieved by this test is essential in finding optimal solutions.

- Exchange of Terms

Terms are added and deleted using the procedures described above until no term can be found that reduces the sum of squares. At this point an attempt is made to exchange terms. Here, each term in the equation, except the last one added is exchanged with all the remaining terms in the bank of terms. The effects of exchanging terms can be directly observed by looking at the standard deviation which can be calculated from the Table 1.3.

It is thus not necessary to transform the whole matrix

[**B**]. If a particular exchange leads to a minimization in the sum of squares, then the actual exchange is carried out. In the exchange, the term in the equation is first deleted using the method for deleting, and the term outside the equation is then added using the algorithm for adding a term. Statistical tests are again performed on the terms and the equation. The exchange of terms is a new feature added to the stepwise regression method by Wagner[4]. The whole stepwise regression

Table 1.3 Calculation of the Sum of Squares of an Equation in which term n already in the Equation is exchanged for term m

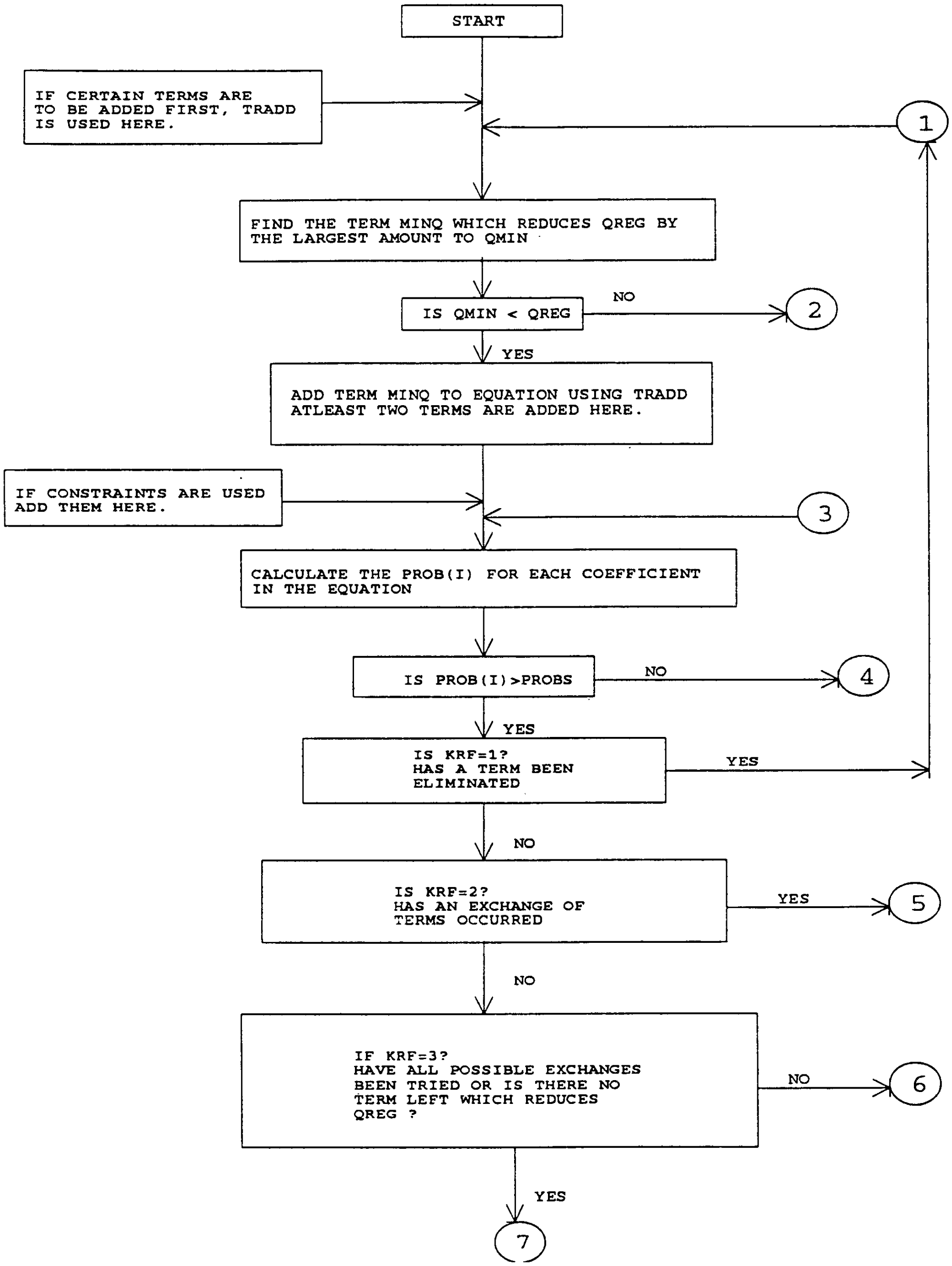
Condition	
$m > n$	$m < n$
$c_m = b_{Lm} + b_{Ln}b_{mn}/b_{nn}$	$c_m = b_{Lm} + b_{Ln}b_{mn}/b_{nn}$
$d_m = b_{mm} + b^2_{mn}/b_{nn}$	$d_m = b_{mm} + b^2_{mn}/b_{nn}$
$AA = b_L + b^2_{Ln}/b_{nn}$	
$SS = AA - c^2_m/d_m$	

method can be summarized the flowsheet shown in Figure 1.1. An example which better illustrates the stepwise regression algorithm is given in Appendix 1.

1.2.4 Applications

The stepwise regression method has found numerous applications in engineering and science. Wagner[4] first applied this method for development of a vapor pressure equation for nitrogen and water. de Reuck & Armstrong[8] used this method to develop an equation of state for propylene. Wagner[14] also used the stepwise regression method in conjunction with mutation[12,13] to generate an extremely accurate equation of state for water. Jacobsen and co-workers[15] have also applied the stepwise regression methodology to the development of equations of state for air, R22, and R12.

In this work, the stepwise regression method is being applied to development of an equation of state for refrigerant R134a. In order to develop a good equation of state, it has to accurately predict all thermophysical properties such as PVT,



1.1a Flowsheet for the stepwise regression method

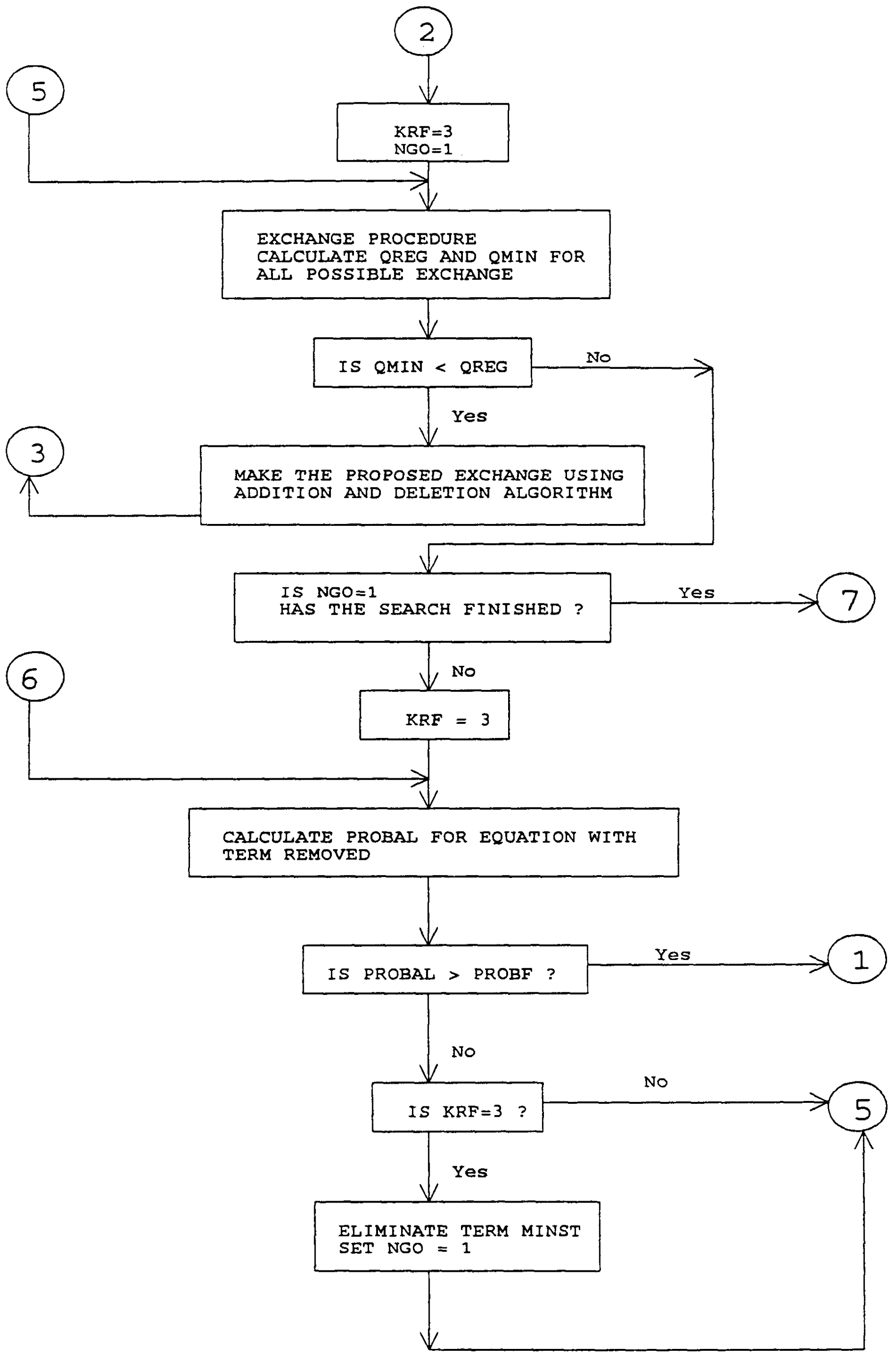


Fig 1.1b Flowsheet for the stepwise regression method.

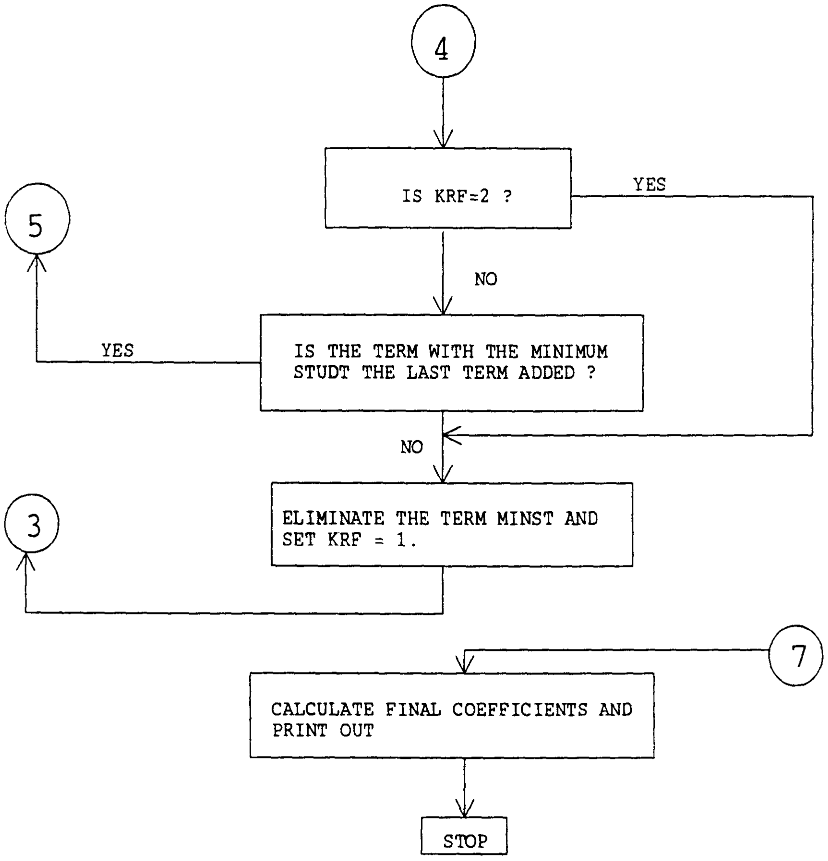


Fig 1.1c Flowsheet for the stepwise regression method.

sound velocity, C_p, C_v, second virial data and the saturation boundary. This involves fitting the equation to different kinds of data and is known as multiproperty regression. Multiproperty regression is used in conjunction with the stepwise regression method to develop a new equation of state, which predicts different properties accurately. The next section contains a detailed discussion of multiproperty regression technique.

1.3 MULTIPROPERTY REGRESSION

In modern engineering and science, the need for finding accurate equations over a wide range of conditions is growing rapidly. Very frequently, these equations are used for deriving other properties that are obtained through differentiations or integrations of the underlying equation. It is therefore imperative that an equation represent the derived data accurately. Multiproperty regression[17] is a direct consequence of this need. Although multiproperty regression can be used in any linear or non-linear regression technique, more often than not, the method is used with the linear least squares[16].

Consider the example of a thermodynamic equation of state from which a large number of properties can derived. These include isochoric heat capacity, isobaric heat capacity, second virial coefficient, enthalpy, etc. Ideally, an equation of state should be accurate not only in PVT surface representation, but also in the representation of these derived properties. Conventionally, equations of state have been fitted only to one kind of data, namely PVT data. As a result, these equations have frequently been inaccurate in representing the derived properties. At present, focus is shifting more towards equations which fit second virial, sound velocity, etc. data simultaneously and thereby developing very accurate equations of state.

1.3.1 Theory

Let there be j =1,2,....,J different kinds of data sets which are incorporated in the least squares method. For each of these J different data sets, the following set of linear equations can be constructed.

$$[\mathbf{F}_1][\mathbf{a}] = [\mathbf{Y}_1]$$

$$[\mathbf{F}_2][\mathbf{a}] = [\mathbf{Y}_2]$$

$$\ldots\ldots\ldots\ldots\ldots$$

$$[\mathbf{F}_J][\mathbf{a}] = [\mathbf{Y}_J]$$

We wish to minimize the weighted sum of squares for all data types in the fitting process. Adding all the above equations, one finds

$$[[\mathbf{F}_1]]+[\mathbf{F}_2]+\dots\dots\dots+[\mathbf{F}_j]][\mathbf{a}] = [\mathbf{Y}_1]+[\mathbf{Y}_2]+\dots+[\mathbf{Y}_j] \qquad (1.26)$$

This equation can be further written as

$$[F][a] = [Y] \qquad (1.27)$$

In multiproperty fitting, the constraints can again be added by the method of Lagrange multipliers.

1.3.2 Linearization of Non-Linear Functions

A number of times in thermodynamics, we encounter properties which have nonlinear dependence on the equation of state parameters. In order to use these properties in a linear regression method like the stepwise regression, the non-linear terms must be linearized in some way. One method of linearizing non-linear thermodynamic properties dependence is to calculate non-linear part of the term using an existing equation of state and thus eliminating the need for finding the value of the non-linear parameter. A good example is the incorporation of the sound velocity in a linear regression calculation. The sound velocity is defined as

$$w^2(\rho, T) = \frac{RTC_p(\rho, T)}{M_w C_v(\rho, T)} \left(\frac{\partial P}{\partial \rho}\right)_T \qquad (1.28)$$

The sound velocity can be used in a linear regression by generating the $(\partial P/\partial \rho)$ data which has linear dependence on the equation of state parameters and by independently calculating the non-linear ratio C_p/C_v from a previous fit. Since C_p/C_v is relatively slowly varying property, this process (e.g., getting a better EOS and regenerating $\partial P/\partial \rho$ converges rapidly. In this way, stepwise regression method could be applied to non-linear forms after some manipulations.

1.3.3 Weighing

Multiproperty fitting is extremely sensitive to weighing, thus the weighing scheme incorporated plays a very important role in finding an optimal solution to the problem. In multiproperty fitting, complications arise from the fact that there are two kinds of weights which have to be considered:

1. Weight for each individual data point of a given type.
2. Weight for type of data.

Finding the right balance between these weights is essential to obtaining a good result. There is no accepted prescription for finding the correct balance of weights for a given data set. Because of this, literally hundreds of weighing schemes are tried in conjunction with the regression process.

The multiproperty regression technique has found numerous applications in thermodynaics. Huber and Ely[18] used this method develop a new equation of state for R134a. Huber and McLinden[19] used this method to develop a new equation of state for R134a in pressure explicit form.

1.4 <u>Simulated Annealing</u>

One problem with the stepwise regression approach is that it can find local minima in the function space spanning the bank of terms. Other optimization methods can in principle overcome this difficulty. A primary method in this category is known as simulated annealing optimization.

Simulated annealing is a general method for treating a broad class of complex, multivariate optimization problems. It has found wide applications in physical sciences and engineering, but has been sparingly used in conjunction with multiproperty regression in thermodynamics. The simulated annealing algorithm was first proposed by Kirkpatrick *et al.* (1983)[20], who drew an analogy between the annealing process which attempts to minimize the energy state of a pure solid and the mathematical optimization of a complex system. Simulated annealing has been applied by Dolan *et al.*[25] to find optimal and near optimal solutions to process network design problems that have combinatorially large number of feasible designs. It has been successfully used for heat exchanger network design by Dolan *et al*[26]. This method has also found numerous other applications from genetics[27] to design of complicated electronic circuits.

In this work, simulated annealing algorithm is used to generate vapor pressure equations for pure fluids. Simulated annealing is becoming increasingly popular for solving complex multivariable optimization problems. The mathematical theory of Markov chains (Hammersley & Handscomb)[24] suggests that, in the optimization context, simulated annealing is capable of producing the global minimum independent of the initial guess. Several proofs have been proposed, which establish that if the number of moves is infinite at each temperature, simulated annealing produces a globally optimum solution. In practice, however, one cannot guarantee that the solution obtained by simulated annealing in a finite amount of time is the rigorous optimum, but formal results indicate that a sufficiently slow annealing schedule will provide an optimal or a nearly optimal solution which is independent of the initial guess.

Consider a system that has the set $\{S_1, S_2, S_3....., S_N\}$ of possible states. If X_t is the state of the system at time t, the system is a Markov chain if the probability of observing X_t is independent of all previous states except for its immediate predecessor X_{t-1}. Mathematically,

$$P(X_t=S_j \mid X_{t_1}=S_{i_1}, S_{i_2}, ..., S_{i_{t-1}}) = P(X_t=S_j \mid X_{t-1}=S_{i_{t-1}}) \qquad (1.29)$$

where P denotes the probability of observation. The above equation indicates that for a Markov chain, the probability that the system is in state S_j given that other system states at different time intervals were $S_{i,1}, S_{i,2}$, etc., is equal to the probability that a system is in state S_j at time t given that system was in state $S_{i,t-1}$ at time t-1. A simpler explanation is that the system is independent of all other states except the previous state.

Simulated annealing generates a Markov chain of points of the objective function which depends only on the previous state of the system. For example, the chain of points could be a series of equations for a regression problem. It accepts and rejects randomly generated "moves" on the basis of a probability related to an annealing temperature. It can accept moves which change the value of the objective function in the direction of the desired optimum, as well as moves which change the value of the objective function in the opposite direction. Thus, for a global minimization problem, a move that increases the value of the objective function (an uphill move) may be accepted as a part of the full series of moves for which the general trend is to reduce the value of the objective function. In this way, simulated annealing is able to explore the full solution space and find solutions which are independent of the starting point. This means that probability of getting trapped in a local minima is much lower as compared to other optimization techniques.

Simulated annealing has proven to be a practical method for solving combinatorially large optimization problems. This method was first applied successfully to the problem of the N-city travelling salesman by Kirkpatrick[20]. Several other applications of simulated annealing have also been explored. The next section deals with the theory behind simulated annealing and its underlying metropolis algorithm.

1.4.1 Monte Carlo simulations

Simulated annealing is based on Monte Carlo[23] simulation techniques developed by Metropolis *et al.* (1953)[22] to study the statistical mechanics of condensed systems. Statistical mechanics as applied to condensed systems is a method for obtaining aggregate properties arising from large number of atoms found in condensed matter. Because of the high density of molecules, only the most probable behavior of the system in thermal equilibrium at a given temperature can be observed. A fundamental question in statistical mechanics concerns what happens to the system in the limit of low temperature. Ground states and configurations close to them in energy are extremely rare among all the configurations of a macroscopic body, yet they dominate its properties at low temperatures because as T is lowered, the Boltzmann distribution collapses into the lowest energy state or states. Finding the low temperature state of a system when a method for calculation of its energy is given is an optimization problem not unlike those encountered in combinatorial optimization. Iterative improvement applied to such problems is much like microscopic rearrangement processes modeled by statistical mechanics, with the cost function playing the role of energy.

In Monte Carlo simulations of condensed systems, the configurational energy E, which the system possesses by the virtue of molecule positions and which is determined from the intermolecular potentials, is calculated. E is given completely by a set of $\{r_i\}$, $i = 1,....,N$, where r_i denotes the position of the molecule i in the system. Also, one can define the configurational energy of a system of N molecules by the 3N dimensional vector.

$$\Gamma^N = (r_1, r_2, ..., r_N) \tag{1.30}$$

The energy E is a function of Γ^N, *viz.*,

$$E = E(\Gamma^N) \tag{1.31}$$

The objective of the Metropolis algorithm is to generate a series of points in configuration space Γ_i^N, which are distributed according to the canonical ensemble probability density

$$P(\Gamma^N) \propto \exp(-E(\Gamma^N)/k_bT) \tag{1.32}$$

where k_b is the Boltzmann Constant.

The Metropolis algorithm is a rigorously correct formalism for generating a Markov chain[21] of points in configuration space that are asymtotically distributed according to (1.32) and are independent of the initial point. The basics of the Metropolis algorithm are as follows:

1. Initialize the position of the molecules.
2. Randomly choose a molecule and move it at random to a position.
3. Calculate $\Delta E = E(\Gamma^{N,New}) - E(\Gamma^{N,Old})$. If $\Delta E < 0$, then accept the configuration. Otherwise accept it with a probability of $\exp(-\Delta E/k_bT)$.
4. Iterate over steps 2 and 3.

The theory behind the Metropolis algorithm is beyond the scope of this work, but a more detailed explanation of the algorithm can be found in Hammersley & Handscomb[24]. The following inferences can be drawn about the Metropolis algorithm.

- The algorithm has the capability of climbing out of a local minima in the energy function since uphill moves are permissible.
- The proportion of uphill moves accepted increases with temperature.

An example of the simulated annealing process is shown in Fig 1.2.

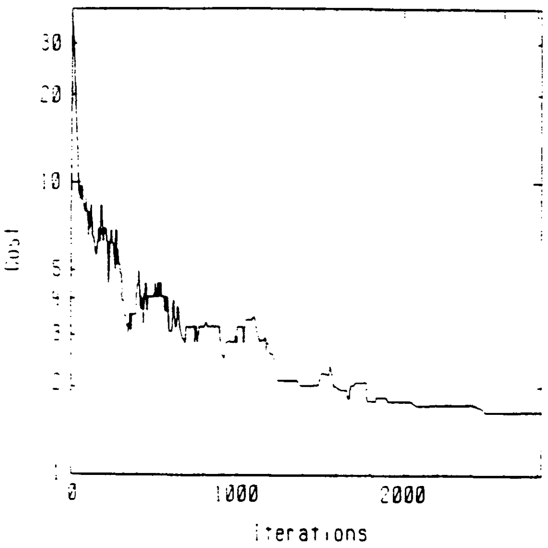


Fig 1.2 Cost as a function of iteration number (Reproduced from Ellen *et al.*[28])

1.4.2 Theory of Simulated Annealing Optimization

The process of identifying a set of variables x = $(x_1, x_2, \ldots\ldots, x_N)$ which globally minimize an objective function C(x) is referred to as multivariable optimization. Kirpartrick[20] proposed that multivariable optimization problems involving a large number of variables might be solved efficiently using a generalization of the canonical ensemble Monte Carlo algorithm. The "cost" function or the objective function C(x) in multivariable optimization is assumed to be the equivalent of energy in a physical system. Similarly, an analogy can be drawn between the configuration point $\{\Gamma^N\}$ and the vector x. Since as explained before, if a physical system is annealed to an extremely low temperature, it will eventually find itself in the lowest energy configuration, Kirkpatrick & co-workers[20] suggested that by performing a similar "annealing" procedure on multivariable optimization problem, the vector $\{\mathbf{x}\}$ which globally optimized the objective function could be located.

The application of this kind of procedure necessitates finding an analog of a physical system's temperature in the multivariable optimization problem. Kirkpatrick introduced the concept of simulated annealing temperature T_{sa}, which has the units of the objective function and is used to control the probability for accepting uphill moves in the optimization process. The simulated annealing temperature is initially set to a high value so that the probability of accepting changes in the vector {x} is also high. The value of T_{sa} is progressively reduced according to an annealing schedule. As the value of T_{sa} reduces, the proportion of accepted uphill moves goes down until no more moves are accepted. This indicates that all attempted moves are uphill in cost and that the temperature is low enough that these moves are no longer accepted. In order to find the global minimum for this optimization, the annealing has to be done in extremely small steps. The simulated annealing algorithm of Kirkpatrick is given as follows :

1. Initialize $\mathbf{x}$ and T_{sa}.
2. Choose an element of $\mathbf{x}$ at random and change it to a new value.
3. Examine $\Delta C = C(\mathbf{x}_{new}) - C(\mathbf{x}_{old})$. If $\Delta C < 0$ then accept the change. Otherwise accept it with a probability of $\exp(-\Delta C/T_{sa})$, i.e. if $\Delta C > 0$.
4. Repeat steps (2) & (3) while reducing T_{sa} according to the annealing schedule.

$$T_{sa}^{(n+1)} = \xi \; T_{sa}^{(n)} \qquad (1.33)$$

ξ = Annealing Schedule Parameter and $\xi > 0$ and slightly less than 1.

This algorithm is analogous to the Metropolis algorithm for Monte Carlo simulation.

2. APPLICATION OF SIMULATED ANNEALING METHOD TO DEVELOPMENT OF VAPOR PRESSURE EQUATIONS

In this work, an attempt has been made to examine the feasibility of simulated annealing as a method for developing accurate vapor pressure equations. The ultimate goal was to apply the simulated annealing algorithm to developing equations of state for pure fluids. This chapter deals with the application of the simulated annealing method to develop vapor pressure equations for carbon dioxide, propane and refrigerant R134a.

As discussed in the Introduction, both stepwise regression and simulated annealing require that a bank of terms be constructed first. This bank of terms contains all the possible terms which could be a part of the equation. In this case, since the pressure is a function of temperature only, the terms in the bank of terms are actually different exponents of the temperature. For the development of vapor pressure equations, the number of terms in the bank of terms was 16. The choice of these terms is very important as terms with a small probability of being in the equation add to the computing time. Usually this set of terms is selected from previous vapor pressure equations. For this study, the following form of the vapor pressure equation was used :

$$\ln\left(\frac{P}{P_c}\right) = \frac{f(\tau)}{(1 - \tau)} \tag{2.1}$$

where P_c is critical pressure, $\tau=1-T/T_c$, T_c is the critical temperature, and P is the vapor pressure. The terms contained in the bank of terms are shown in Table 2.1.

Vapor pressure equations were initially generated from random combinations of the set of terms given in Table 2.1. The sum of squares of these resultant equations was then subjected to simulated annealing in order to obtain the best possible vapor pressure equation. The algorithm for vapor pressure used in this case is shown in Fig. 2.1. The maximum number of terms which could be a part of the equation can be specified by the user. Random numbers associated with every term are then generated. If the value of the random number for the term is greater than some user specified value, then that term is accepted. These terms are accepted until the total number of terms in the equation is equal to the value specified maximum value. The selected terms are

Table 2.1 Terms in the Bank of Terms for the Vapor Pressure Equation.

1	τ
2	$\tau^{1.9}$
3	$\tau^{6.5}$
4	$\tau^{2.5}$
5	$\tau^{1.5}$
6	$\tau^{7.5}$
7	τ^{3}
8	τ^{4}
9	$\tau^{5.5}$
10	τ^{7}
11	$\tau^{8.5}$
12	τ^{6}
13	$\tau^{3.5}$
14	τ^{8}
15	$(1-\tau)^{2}$
16	τ^{-1}

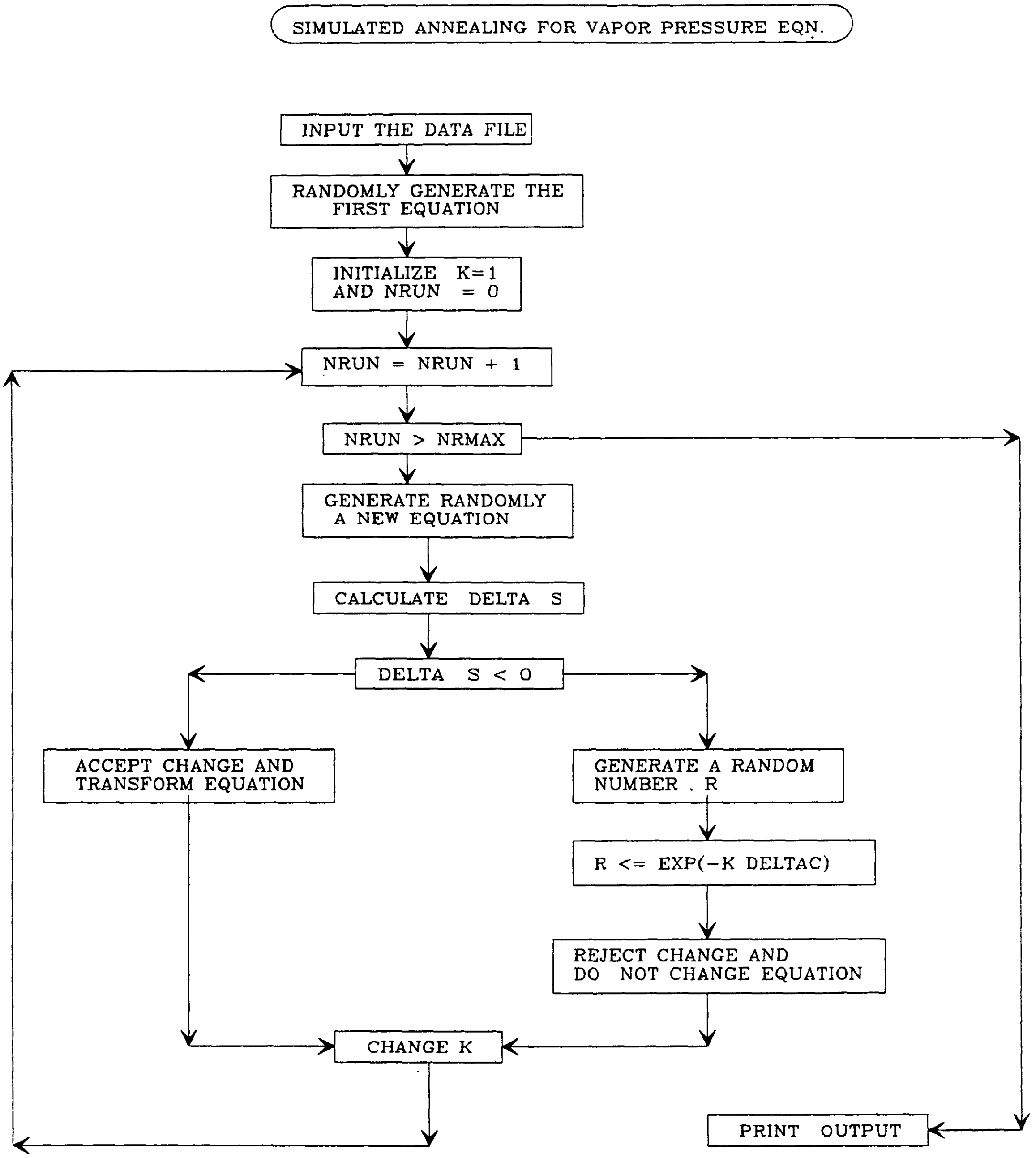


FIG 2.1 Flowsheet for the simulated annealing method algorithm.

then subjected to a linear least squares fit from which the overall sum of squares can be determined. The calculated sum of squares is compared with the previous sum of squares and if there is a reduction in sum of squares then the change is accepted, otherwise, the Metropolis algorithm is applied. This whole process continues until the total number of iterations is equal to some value NMAX.

The simulated annealing algorithm was applied to develop vapor pressure equations for carbon dioxide, propane and R134a. The number terms in the equations was limited to six, but in principle more terms could be added to the equation. Experimental vapor pressure data were collected and a Gaussian weighing scheme was used. The errors in the measurements for temperature (σ_T) and pressure (σ_P) were assumed to be as follows :

$$\sigma_T = 10^{-3}\ T_i$$

$$\sigma_P = 3 \cdot 10^{-4}\ P_i$$

where T_i and P_i are experimental temperature and pressure respectively.

Using the above errors, the weights were generated for each data point. The weight WT_i for each point is given by,

$$WT_i = \frac{1}{\sqrt{\sigma_T^2 + \sigma_P^2}} \tag{2.3}$$

These weights were then used along with the data points to generate the vapor pressure equations. In the next section, equations for carbon dioxide, propane and R134a are presented. The computer program for simulated annealing is given in Appendix II.

2.1 Carbon Dioxide

The CO_2 vapor pressure equation was generated using the data sets shown in Table 2.2. Upon using the above data, the following equation was generated by the simulated annealing algorithm for CO_2. The resulting coefficients are given in Table 2.3.

$$f(\tau) = G_1\tau + G_2\tau^{1.5} + G_3\tau^4 + G_4\tau^7 + G_5\tau^{3.5} + G_6(1-\tau)^2 \tag{2.4}$$

where $f(\tau) = (1-\tau)\ \ln(P/P_c)$, $\tau=1-T/T_c$, $T_c=304.25$ K and $P_c=73.7642$ bar.

Table 2.2 Data Sets Used for Vapor Pressure Comparisons for Carbon Dioxide

No	DATA SET	RANGE K	Points
1.	Meyers and Van Dusen[29]	216 - 304	67
2.	Schmidt and Thomas [30]	293 - 303	8
3.	Cook[31]	295 - 303	5
4.	Michels[32]	217 - 276	19
5.	Jenkin and Pye[33]	222 - 276	22
6.	Cook[34]	293 - 304	15
7.	Levelt-Sengers and Chen[35]	267 - 304	38
8.	Michels,Blaise and Michels[36]	276 - 304	8
9.	Vukalovich[37]	238 - 298	12
10.	Kirillin[38]	283 - 303	3
11.	Amagat[39]	272 - 305	42
12.	Holste[40]	250 - 303	12
13.	Fernandez-Fassnacht *et al.*[41]	216 - 243	21

Table 2.3 Coefficients for CO_2 Vapor Pressure Equation

No	Coefficient
G_1	-7.0317262402
G_2	1.5759670861
G_3	7.7267263208 10^{01}
G_4	-3.3119067390 10^{02}
G_5	-4.5769933009 10^{01}
G_6	2.4789223571 10^{-03}

The equation thus developed was then compared with the equations developed by Ely *et al.*[42] The comparisons are shown in Figure 2.2. This graph demonstrates that the equation obtained with simulated annealing is in good agreement with the data and other equations for CO_2. The statistical comparisons for both equations are shown in Table 2.4. For the statistical test, AAD is the absolute average deviation, RMS is the root mean square of the deviations and BIAS is the average deviation.

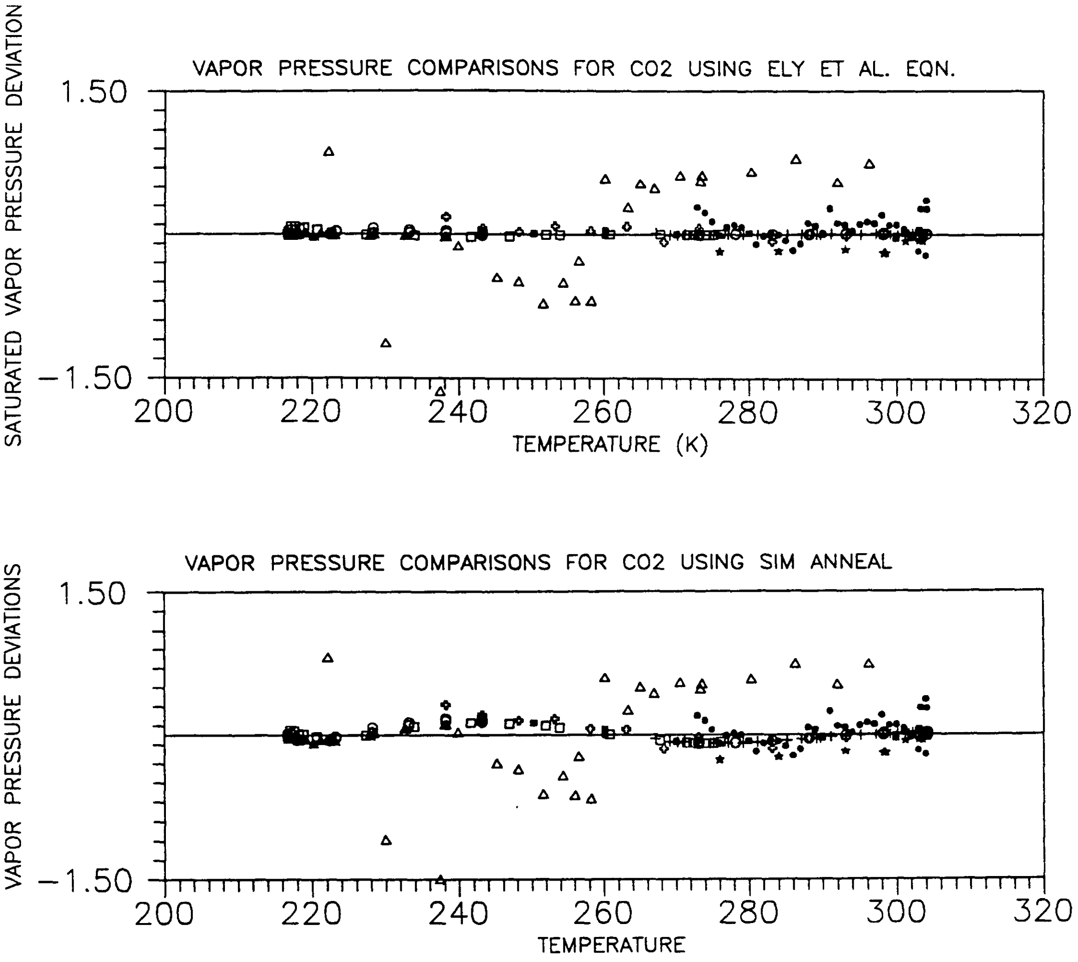


FIG 2.2 Comparison of experimental vapor pressures with calculated values for CO_2.
(○) Meyers and Van Dusen; (□) Michels; (Δ) Jenkin and Pye ; (+) Levenlt-Sengers and Chem; (★) Michels, Blaise and Michels; (╬) Vukalovich; (•) Amagat; (▪) Holste; (▲) Fernandez-Fassnacht.

Table 2.4 Statistical comparisons for carbon dioxide for all data.

Equation	AAD, %	BIAS,%	RMS, %
Ely *et al.*	0.097	$8.554\ 10^{-3}$	0.222
THIS WORK	0.11	$-2.136\ 10^{-4}$	0.207

2.2 Propane

The data used for developing a vapor pressure equation for propane are summarized in Table 2.5. The vapor pressure curve for propane is very different from that of CO_2 in that it is very long, i.e., from a triple point of 85.47 K to a critical point of 369.85 K. For comparison, carbon dioxide's liquid-vapor region ranges form 274-304 K.

The equation developed for propane using simulated annealing is,

$$f(\tau) = G_1\tau + G_2\tau^{2.5} + G_3\tau^{1.5} + G_4\tau^{7.5} + G_5\tau^{7} + G_6\tau^{8.5} \qquad (2.5)$$

where $f(\tau)$ related to the vapor pressure given by eqn. (2.1) and $\tau = 1 - T/T_c$, T_c=369.85 K and P_c = 4.24746 MPa.

Table 2.5 Data Sets Used for Vapor Pressure Comparisons for Propane

No	DATA SET	RANGE K	Points
1.	Tickner and Flossing[43]	105.35-164.75	13
2.	Carruth and Kobayashi[44]	94.54-178.65	12
3.	Maass and Wright[45]	229.75-250.05	6
4.	Dana *et al.*[46]	210.33-323.49	22
5.	Sage, Schaafsma and Lacey[47]	293.93-368.76	20
6.	Beattie *et al.*[48]	323.15-348.15	2
7.	Kemp and Egan[49]	166.19-231.46	12
8.	Deschner and Brown[50]	301.95-368.97	28
9.	Reamer, Sage and Lacey[51]	313.48-368.72	8
10.	Cherney, Marchman and York[52]	303.15-323.15	2
11.	Clegg and Rowlinson[53]	323.15-368.15	8
12.	Helgeson and Sage[54]	277.59-360.93	16
13.	Gilliland and Scheeline[55]	314.81-357.59	3
14.	Burrell and Robertson[56]	148.95-229.05	16
15.	Thomas and Harrison[57]	258.15-369.75	24
16.	Mousa[58]	334.78-369.74	11
17.	Kratze[59]	311.97-367.58	13
18.	Teichmann[60]	324.75-363.42	15

The coefficients for the equation are shown in the Table 2.6.

The comparisons for the vapor pressure equation with the equation developed by Goodwin[61] are shown in Figure 2.3. The statistical comparisons for both the equations for propane are shown in Table 2.7 and 2.8.

Table 2.6 Coefficients of the Vapor Pressure Equation for Propane

No	Coefficients
G_1	-6.7967627955
G_2	-1.8438894757
G_3	1.6737415605
G_4	4.6524895833 10^{01}
G_5	-3.3948963988 10^{01}
G_6	-1.4652247024 10^{01}

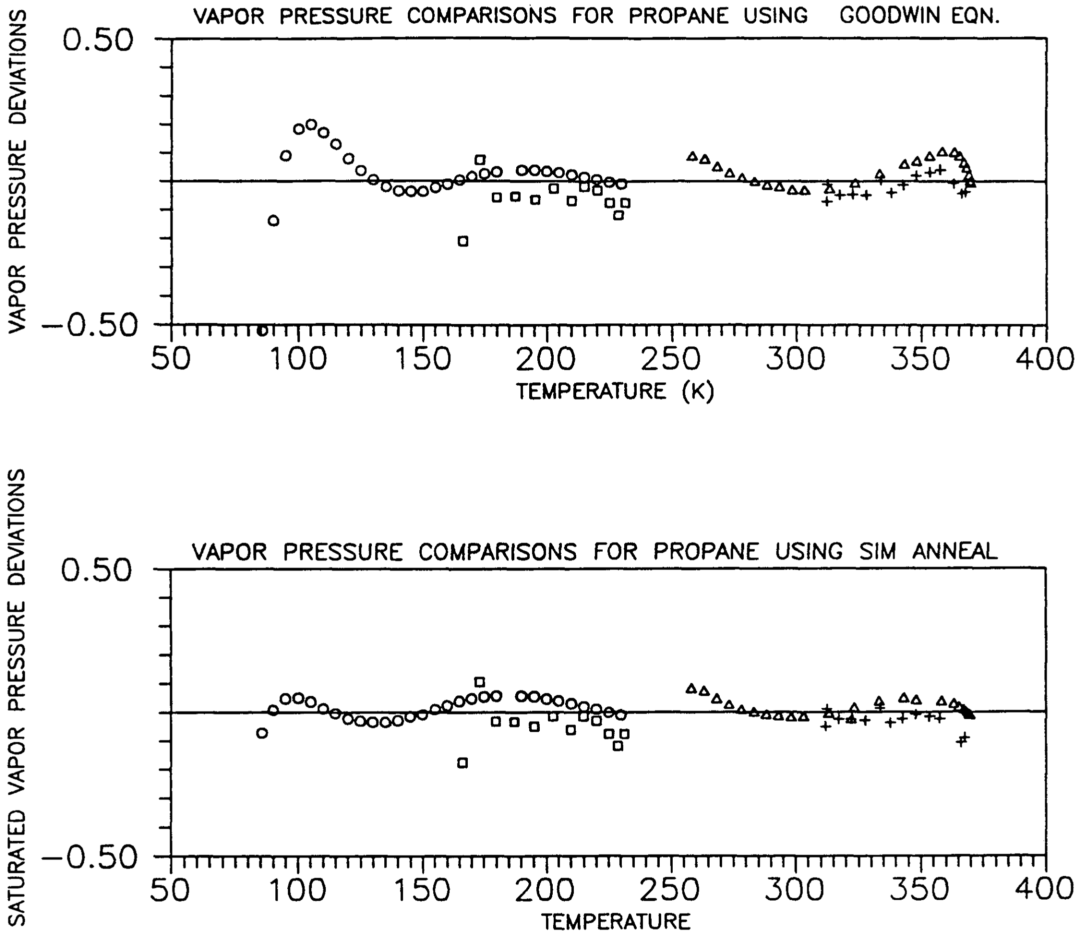


FIG 2.3a Comparison of experimental vapor pressures with calculated values for Propane.
(○) Thermal Loops; (□) Kemp; (Δ) Thomas ; (+) Kratze.

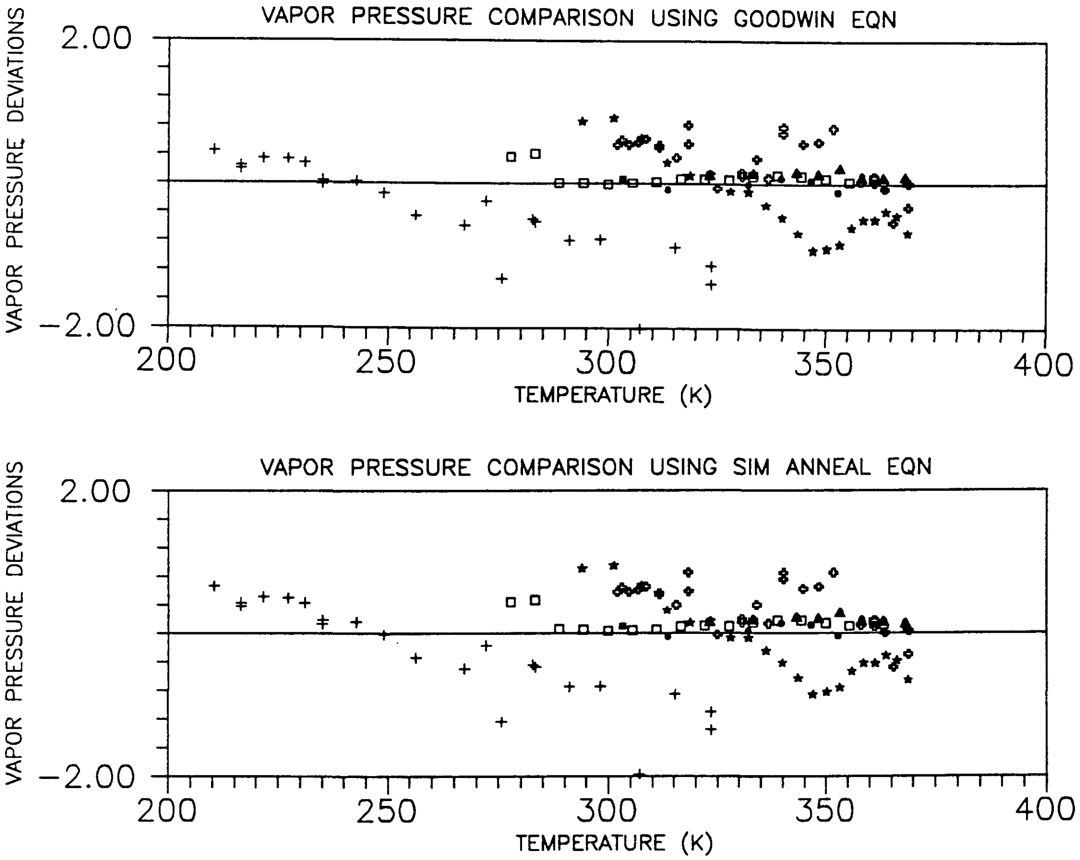


FIG 2.3b Comparison of experimental vapor pressures with calculated values for Propane.
(Δ)Maass; (+)Dana; (★) Sage; (╬)Deschner; (•)Reamer; (■) Cherney; (▲) Clegg; (□) Helgeson.

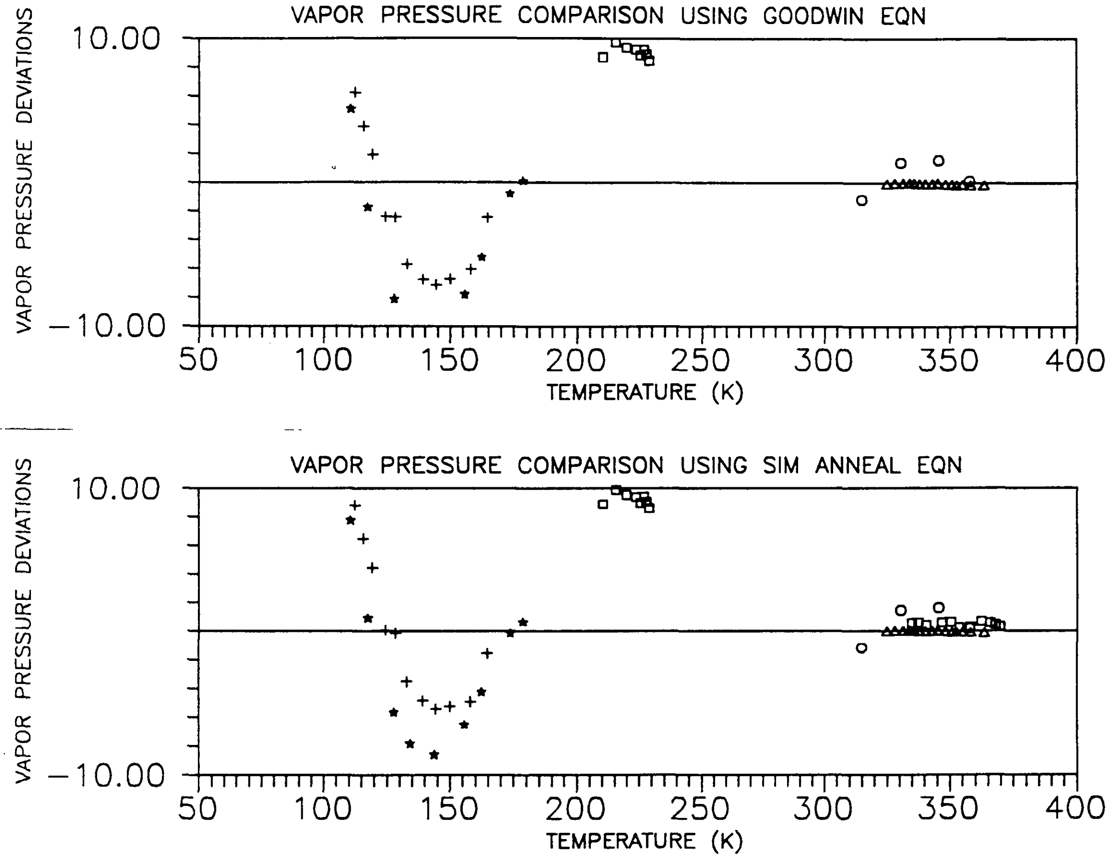


Fig 2.3c Comparison of experimental vapor pressures with calculated values for propane.
(○)Gilliland; (□) Burrell; (Δ)Teichmann; (+)Tickner; (★)Carruth.

Table 2.7 Statistical comparisons for propane for all data.

Equation	AAD,%	BIAS,%	RMS,%
Goodwin	1.898	1.284	4.889
Sim Anneal	1.887	1.273	4.889

Table 2.8 Statistical comparisons for propane for fitted data.

Equation	AAD,%	BIAS,%	RMS,%
Goodwin	$5.5855\ 10^{-2}$	$2.879\ 10^{-3}$	$8.85\ 10^{-2}$
Sim Anneal	$3.775\ 10^{-2}$	$-2.044\ 10^{-3}$	$4.912\ 10^{-2}$

2.3 R134a

The data sets used for fitting R134a (tetrafluoroethane) are shown in Table 2.9. The following equation was then generated using the above data sets shown in Table 2.9. The coefficients for the terms are given in the Table 2.10.

Table 2.9 Data Sets Used for Vapor Pressure Comparisons for R134a.

No	DATA SET	RANGE K	Points
1.	Goodwin and Weber[62]	214 - 373	79
2.	Arita *et al.*[63]	273 - 323	3
3.	Zhu, and Wu [64]	279 - 363	43
4.	Baehr and Tillner-Roth[65]	303 - 374	37
5.	Weber[66]	313 - 373	22
6.	Wilson & and Basu[67]	210 - 369	32
7.	Piao *et al.*[68]	308 - 374	51
8.	Kubota, *et al.*[69]	253 - 373	25
9.	Nishiumi H and Yokoyama[70]	247 - 373	16
10.	Magee and Howley[71]	180 - 350	19
11.	Morrison and Ward[72]	268 - 374	12
12.	Niesen[73]	312 - 371	13
13.	Maezawa[74]	279 - 350	14
14.	Baroncini[75]	243 - 258	61
15.	Fukushima[76]	262 - 372	41

Table 2.10 Coefficients of the Vapor Pressure Eqn. for R134a

No	Coefficients
G_1	-7.6559241549
G_2	-8.0208389582 10^{-01}
G_3	1.9507847451
G_4	-2.8803710775
G_5	-1.9400505510
G_6	1.5870481334 10^{-03}

$$f(\tau) = G_1\tau + G_2\tau^{1.9} + G_3\tau^{1.5} + G_4\tau^3 + G_5\tau^5 + G_6(1-\tau)^2 \qquad (2.6)$$

where $f(\tau)$ is given by eqn. (2.1), $\tau = 1 - T/T_c$, T_c= 374.265 K, and P_c= 40.603 bars.

The above equation developed using the simulated annealing algorithm was then compared with another vapor pressure equation for R134a given by Huber and McLinden[19]. The comparisons are shown in Figure 2.4. The graph shows that the new equation compares favorably with the existing equations. The statistical comparisons for the R134a data set are shown in Table 2.11.

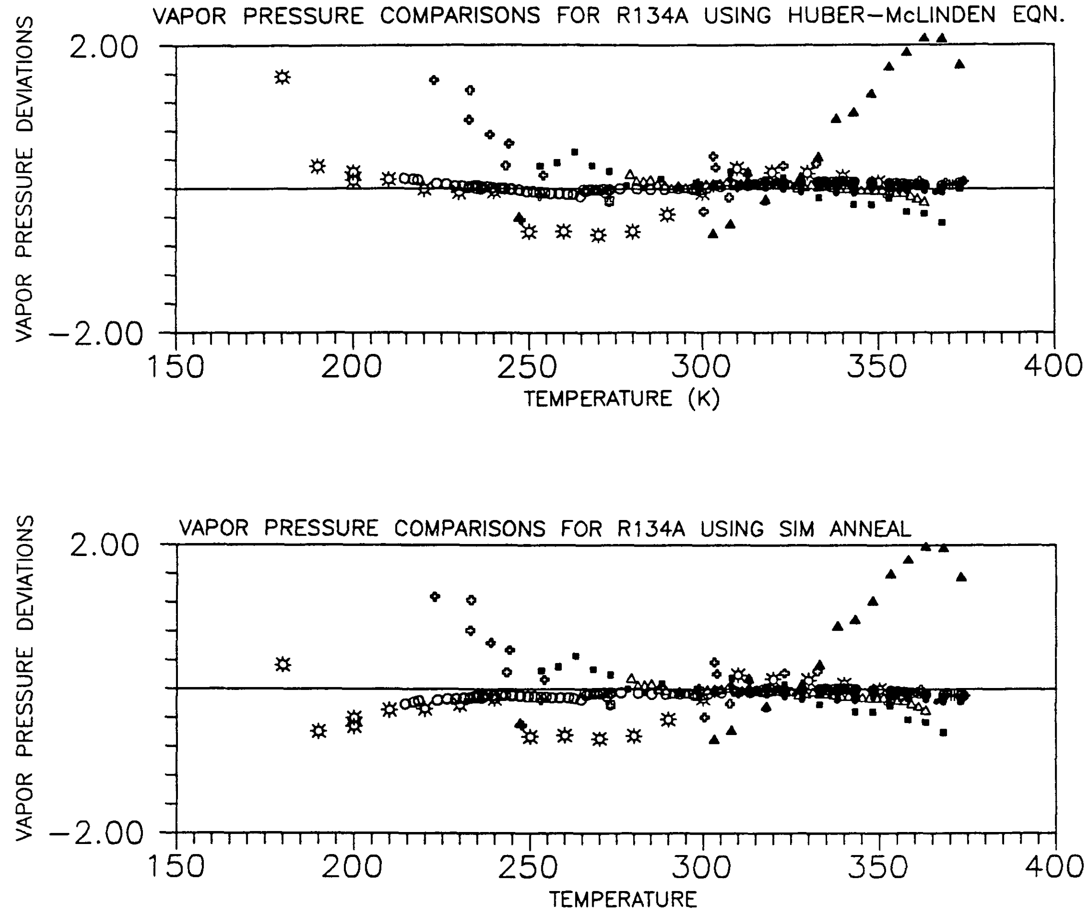


FIG 2.4a Comparison of experimental vapor pressures with calculated values for R134a.
(○) Goodwin and Weber; (□) Arita et.al.; (Δ) Zhu and Wu ; (+) Baehr and Tillner-Roth; (★) Weber; (╬) Wilson and Basu; (●) Piao et al.; (■) Nishuimi and Yokoyama; (▲)Magee and Howley; (✲) Morrison and Ward.

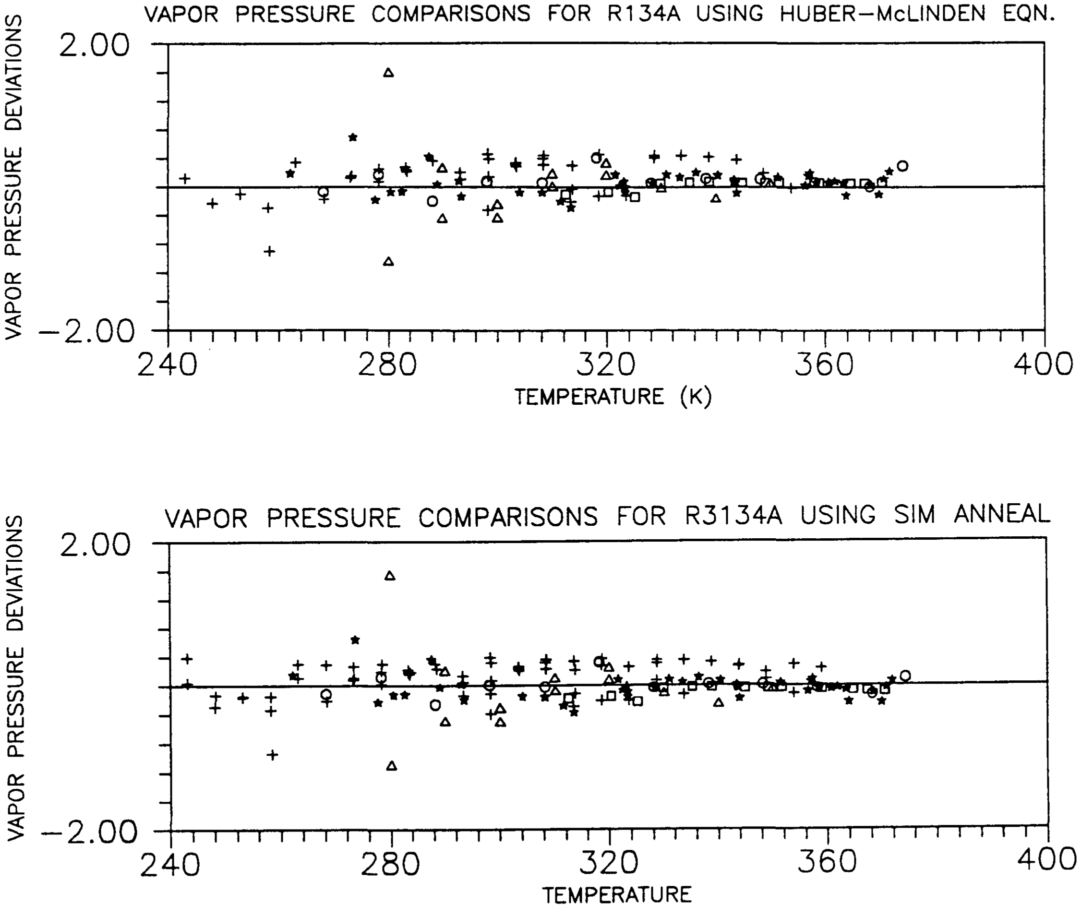


FIG 2.4b Comparison of experimental vapor pressures with calculated values for R134a.
(○) Niesen; (□) Maezawa; (Δ) Baroncini; (+) Fukushima.

Table 2.11 : Statistical Comparisons for R134a Data for all data.

Equation	AAD,%	BIAS,%	RMS,%
McLinden	0.179	0.108	0.337
Sim Anneal	0.177	$5.93\ 10^{-4}$	0.326

The overall comparisons for the vapor pressure equations shown in Table 2.12.

2.4 Summary

Simulated annealing method seems to have numerous potential applications in thermodynamics for developing vapor pressure equations and equations of state. One of the primary disadvantages of simulated annealing experienced in this work was the time factor. Simulated annealing took a lot of time to find a good equation of state. One of the reasons for this problem is that simulated annealing was used in conjunction with the linear least squares method. If simulated annealing could be made more efficient, it can applied to developing new equations of state with significant ease.

FIG. 2.12 STATISTICAL COMPARISONS FOR VAPOR PRESSURE EQUATIONS

EQUATIONS	AAD	RMS
CARBON DIOXIDE		
ELY *et al.*	0.097	0.222
SIM ANNEAL	0.110	0.207
PROPANE		
GOODWIN	0.058	0.085
SIM ANNEAL	0.037	0.049
R134a		
McLINDEN	0.179	0.337
SIM ANNEAL	0.177	0.326

3. APPLICATION OF STEPWISE REGRESSION METHOD TO THE DEVELOPMENT OF A NEW EQUATION OF STATE FOR R134a

In this work, the stepwise regression method is used in conjunction with multiproperty fitting to develop two new equations of state SRM-20 and SRM-29, for the refrigerant R134a (tetrafluoroethane). In the development of these equations of state PVT, second virial, sound velocity, C_V, C_p, and saturation boundary data were used. In this chapter, the fitting techniques for all these different kinds of data and the results will be discussed.

The fitting form conventionally used for most equations of state involves pressure as a function of temperature and density, in other words, $P = P(\rho, T)$. In this pressure explicit form, not all functions of ρ and T can be used while retaining analytic closed forms for derived properties. This is because some properties require integration of the equation of state and not all possible functional forms are integrable. This problem can be eliminated to a great extent by using the Helmholtz free energy form. This is due to the fact that all of the thermodynamic properties can be obtained from the Helmholtz free energy through differentiation. The Helmholtz free energy is used (as opposed to the Gibbs free energy)

keep, maintain

since it is a natural function of density and temperature i.e., $f(\rho,T)$. This function can further be divided into the contribution due to the ideal gas $f^0(\rho,T)$ and the residual part $f^r(\rho,T)$. This relationship can be expressed as,

$$f(\rho,T) = f^0(\rho,T) + f^r(\rho,T) \tag{3.1}$$

The ideal contribution in this case $f^0(\rho,T)$ is obtained by integrating the total differential of the ideal gas Helmholtz free energy.

$$dA^* = -p^*dV - S^*dT$$

from some reference state (ρ_0,T_0) to the state of interest (ρ,T). S^* in the above equation is obtained from the ideal-gas heat capacity correlation developed by McLinden *et al.*

$$C_p = a + bT + cT^2 \tag{3.2}$$

where $a = 1.94006$, $b = 2.58531\ 10^{-1}$, $c =-1.29665\ 10^{-4}$, T is the temperature in kelvin.

Once the ideal part of the equation is obtained, the residual of the Helmholtz free energy has to be determined. The function to be determined here is the dimensionless molar Helmholtz free energy as a function of the dimensionless density $\acute{o}=\rho/\rho_c$ and reciprocal temperature $\tau=T_c/T$. Defining $\Phi = f/RT$ this can be written as

$$\frac{f(\delta,\tau)}{RT} = \frac{f^0(\delta,\tau)}{RT} + \frac{f^r(\delta,\tau)}{RT} \tag{3.3}$$

$$\Phi(\delta,\tau) = \Phi^0(\delta,\tau) + \Phi^r(\delta,\tau) \tag{3.4}$$

The stepwise regression method is used to obtain the residual Φ^r.

3.1 Bank of Terms

The stepwise regression method requires a bank of terms. This bank of terms comprises of all the possible functional forms which could be a part of the equation. The general form of the equation for R134a was assumed to be the following:

$$\Phi^r = \Sigma_{n=1}^{N} a_n \delta^i \tau^j \exp(-\delta^k) \tag{3.5}$$

In this study, the bank of terms consisted of 178 terms which are listed in the Appendix III. The computer program for stepwise regression is given in Appendix IV. In fitting properties derivatives of Φ^r will be required. These are denoted by subscripts. For example, the mixed second derivative of Φ^r is given by

$$\Phi^r_{\delta\tau} = \partial^2\Phi^r/\partial\delta\partial\tau$$

3.2 Fitting

Fitting in the stepwise regression algorithm is done by minimizing the overall weighted sum of squares, which includes the residuals for all the different data sets. Let y_{exp} be the experimental value of any property which is used for fitting purposes and y_{calc} be the value calculated from the equation developed. The following sum of squares is then minimized:

$$\chi^2 = \Sigma_{j=1}^{J} \chi_j^2 \tag{3.6}$$

$$\chi_j^2 = \Sigma_{m=1}^{M_j} \left(\frac{[y_{calc} - y_{\mathrm{exp}}]^2}{\sigma_{\mathrm{exp}}^2} \right)_{m,j} \tag{3.7}$$

In the above equation, the subscript j corresponds to the jth data set and the index m corresponds to the mth data point in the jth data set. The σ_{exp} corresponds to the Gaussian error propagation formula and is given by

$$\sigma_{j,m}^2 = \left[\left(\frac{\partial \Delta y}{\partial x}\right)_{y,z}^2 \sigma_x^2 + \left(\frac{\partial \Delta y}{\partial z}\right)_{y,x}^2 \sigma_z^2 + \left(\frac{\partial \Delta y}{\partial y}\right)_{z,x}^2 \sigma_y^2 \right]_{j,m} \tag{3.8}$$

3.3 Nonlinear Properties

As discussed earlier, the step wise regression method is a linear regression method and cannot be applied to residuals of properties which have nonlinear dependence on the function parameters. While fitting the data for finding residual Helmholtz free energy, nonlinear terms are encountered in the C_p fit and sound velocity fit.

$$\frac{C_p(\delta,\tau)}{R}=-\tau^2(\Phi^0_{\tau\tau}+\Phi^r_{\tau\tau})+\frac{(1+\delta\Phi^r_\delta-\delta\tau\Phi^r_{\delta\tau})^2}{1+2\delta\Phi^r_\delta+\delta^2\Phi^r_{\delta\delta}} \qquad (3.9)$$

In this case, the second part of the C_p equation is a nonlinear part and cannot be incorporated directly in stepwise regression. In order to get around this problem, the value of the non-linear part is first calculated using an equation developed previously. The value of that function is then subtracted from the experimental value of C_p giving a function which can be fitted by the stepwise regression method. The sound velocity calculations also involve nonlinear functions as shown below. $\partial P/\partial \rho$ is needed to fit the sound velocity data.

$$w^2(\rho,T)=\frac{RTC_p(\rho,T)}{M_w C_v(\rho,T)}\left(\frac{\partial P}{\partial \rho}\right)_T \qquad (3.10)$$

3.4.1 PVT Data

The following equation expresses the relationship between the pressure and the Helmholtz free energy,

$$p = -\left(\frac{\partial f}{\partial v}\right)_T \tag{3.11}$$

This can further be written in the reduced form as

$$\frac{p(\delta,\tau)}{\rho RT} = 1 + \delta\Phi_\delta^r \tag{3.12}$$

The data sets used for the PVT fit are shown in Table. 3.1.

3.4.2 Saturated Vapor Pressure Data

The saturated vapor pressure data are incorporated in the fit by using the Gibbs constraint. The fact that the Gibbs free energy for the liquid phase is equal to that for the gas phase along the saturation boundary can be expressed in terms of the Helmholtz energy

$$A_v + p_s V_v = A_L + p_s V_L \tag{3.13}$$

The data are included in the fit by rewriting eqn. (3.13) as

$$P_s\ (V_L - V_V) = A_V - A_L \tag{3.14}$$

where left hand side is assumed to be known. The quantities P_s, V_L, V_V are obtained from the ancillary correlations of the saturation boundary.

Table 3.1 : Summary of available R134a PVT Data

Source	No of data points	Pressure Range Bars	Temperature Range (K)
Zhu *et al.*[78]	42	1.4 - 1.27	283-353
Tillner-Roth Baehr[79] *	411	0.9 - 16.4	293-453
Tillner-Roth Baehr[80]*	432	7.3 - 158	243-413
Hou *et al.*[81] *	429	7.5 - 709	180-380
Weber[66]*	69	2.2 - 53.3	321-423
Wilson and Basu[67]	51	10.6 - 66.7	317-448
Piao *et al.*[68]	157	8.0 - 118	313-423
Magee[71]	150	25.8 - 349	187-343
Morrison and Ward[72]	131	7.0 - 57.8	279-367
Maezawa *et al.*[74]	10	5.1 - 20.0	280-340
Baroncini *et al.*[75]	46	1.8 - 19.4	263-359
Fukushima[76]	63	5.5 - 57.3	294-424
Qian *et al.*[77]	21	1.3 - 19.3	320-340

* Indicates Data Sets Used for Fitting

The vapor pressure equation is given in eqn (3.15). This correlation was developed by regressing experimental data. The vapor pressure data used in developing the correlation are summarized in Table 3.2.

$$\ln\left(\frac{P_\sigma}{P_c}\right) = c_1\chi^{1.66}+c_2\frac{\chi}{(1-\chi)}+c_3\chi+c_4\chi^3 \qquad (3.15)$$

where $\chi = T/T_c$.

3.4.3 Saturated Liquid and Vapor Density

The stepwise regression fitting form for the saturation densities is identical to that of the PVT data,

$$\frac{p_{sat}}{\rho_{sat}\,RT} = 1+\delta\Phi_\delta^r \qquad (3.18)$$

The saturated liquid density was obtained for a functional form developed by Ely *et al*[18].

$$\rho_{\sigma L} = \rho_c\left(1+\frac{d_1\chi^{.35}+d_3\chi^2 + d_4\chi^3}{1+d_2\chi^{0.65}}\right) \qquad (3.16)$$

The available data for the saturated liquid density are summarized in Table 3.4. The saturated vapor densities are correlated as

$$\rho_{\sigma V} = \frac{P_\sigma}{RT}[1+(\frac{P_\sigma}{P_c})(\frac{T_c}{T})^4(Z_c-1)F(\chi)]^{-1} \qquad (3.17)$$
$$\text{where} \quad F(\chi)=(1+f_1\chi^{0.35}+f_2\chi+f_3\chi^2+f_4\chi^3)$$

The available saturated vapor density data are present in Table 3.5. The coefficients c_i, d_i and f_i for eqns. (3.15-3.17) are shown in Table 3.6.

For purposes of the fit saturation densities were generated using eqns (3.17) and (3.18) at evenly spaced temperatures from the triple point to the critical point.

3.4.4 C_v Data

The C_v data are fitted in the form of residual heat capacity ΔC_v, where ΔC_v is defined by,

$$\Delta C_v = C_v^{id} - C_v \qquad (3.20)$$

The fitting form is given by the following equation

$$\frac{\Delta c_v(\delta,\tau)}{R} = \tau^2\phi^r_{\tau\tau} \qquad (3.21)$$

The data used for fitting the C_v data were reported by Magee[79] and cover the temperature range of 187-343 K.

Table 3.2 Data Used for Saturation Vapor Pressure Comparisons

Source	No of data Points	Pressure Range (MPa)	Temperature Range (K)
Goodwin *et al.*[62]	79	0.02-3.97	214-373
Arita *et al.*[63]	3	0.29-1.32	273-323
Zhu *et al.*[64]	43	0.36-3.23	279-363
Baehr and Tillner-Roth[65]	37	0.77-4.05	303-374
Weber[66]	22	1.02-3.97	313-373
Wilson & Basu[67]	32	0.01-3.66	211-369
Piao *et al.*[68]	51	0.88-4.06	308-374
Kubota *et al.*[69]	25	0.13-3.97	253-373
Nishiumi & Yokoyama[70]	16	0.10-4.03	247-373
Magee and Howley[71]	19	0.0001-2.46	180-350
Morrison and Ward[72]	12	0.24-4.07	268-374
Niesen *et al.*[73]	14	0.99-3.77	312-370
Maezawa *et al.*[74]	14	0.38-2.46	280-350
Baroncini *et al.*[75]	64	0.08-2.98	243-359
Fukushima[76]	41	0.19-3.87	262-372

Table 3.3 Data Used for Saturated Liquid Density Comparisons.

Source	No of data points	Density Range mol/dm^3	Temperature Range K
Fukushima[87]	7	12.08-13.58	244-292
Fukushima *et al.*[85]	8	5.04-7.31	369-374
Yokoyama & Takahashi[88]	21	7.72-13.34	252-367
Hou *et al.*[81]	10	8 .49-15.29	180-360
Wilson & Basu[82]	9	6.86-13.73	239-372
Piao *et al.*[68]	7	6.79-11.22	313-372
Kabata *et al.*[86]	12	5.05-9.79	343-374
Morrison and Ward[72]	27	5.05-12.80	268-374
Niesen *et al.*[73]	14	7.18-11.26	312-370
Maezawa *et al.*[75]	25	7.32-12.44	280-370
Fukushima[76]	3	8.79-10.80	323-357

Table 3.4 Data Sets Used for Saturated Vapor Density Comparisons.

Source	No of data points	Density Range mol/dm^3	Temperature Range K
Fukushima[85]	9	3.15 - 4.91	371-374
Weber[66]	5	0.60 - 2.33	320-365
Kabata *et al.*[86]	15	2.04 - 4.97	361-374
Morrison and Ward[72]	8	0.37 - 2.62	298-368
Niesen *et al.*[73]	14	0.49 - 2.95	312-370
Fukushima[76]	6	0.28 - 3.07	293-371

Table 3.5 Coefficients of the Ancillary Equations

i	1	2	3	4
c_i	3.946984	-11.313271	3.693108	5.566337
d_i	2.081196	-0.413003	-1.177335	1.116197
f_i	-0.894650	-0.023273	1.042141	-1.268685

3.4.5 C_p Data

Fitting the C_p data is different from fitting other data as the C_p fitting form involves as non-linear terms. As explained earlier, the way to get around this problem is to linearize the equation by calculating the value of the nonlinear term using a previous fit. In way the whole form becomes linear and can be fitted easily. The form is represented by the following equation.,

$$\frac{C_p(\delta,\tau)}{R} = -\tau^2(\Phi^0_{\tau\tau}+\Phi^r_{\tau\tau}) + \frac{(1+\delta\Phi^r_{\delta}-\delta\tau\Phi^r_{\delta\tau})^2}{1+2\delta\Phi^r_{\delta}+\delta^2\Phi^r_{\delta\delta}} \qquad (3.22)$$

The data sets used for C_p comparisons are shown in Table 3.6.

3.4.6 Second Virial Data

The second virial data are incorporated in the fit by using the form shown below.,

$$B(\tau)\ \rho_c = \lim_{\delta\to 0} \Phi^r_{\delta}(\delta,\tau) \qquad (3.23)$$

The data sets used for fitting the second virial data are shown in Table 3.7.

Table 3.6 Data Sets Used for C_p Comparisons

Source	Temperature Range, K	Pressure Range, MPa
Saitoh *et al.*[89]	275 - 356	2.6 - 34.9
Nakagawa *et al.*[90]	273 - 356	1.0 - 3.0

Table 3.7 Data Sets Used for Second Virial Comparisons

Source	Temperature Range, K
Goodwin & Moldover[92]	235 - 440
Weber[66]	323 - 423
Tillner-Roth and Baehr[78]	293 - 453

3.4.7 Sound Velocity Data

The sound velocity data cannot be fitted directly as it is nonlinear in the Helmholtz free energy. As described in the introduction, $(\partial p/\partial \rho)$ data is generated from the sound velocity data and used in the fit. The sound velocities are calculated from the obtained fit using the equation 3.24.

$$w^2(\rho,T) = \frac{RTC_p(\rho,T)}{M_w C_v(\rho,T)}\left(1 + 2\delta\Phi_\delta^r + \delta^2\Phi_{\delta\delta}^r\right) \qquad (3.24)$$

The ($\partial p/d\rho$) data are fitted in the form shown in eqn. (3.25).

$$\frac{\partial p}{\partial \rho} = RT(1 + 2\delta\Phi_\delta^r + \delta^2\Phi_{\delta\delta}^r) \qquad (3.25)$$

The available sound velocity data are summarized in Table 3.8.

3.5 FIXED POINTS

The values for triple points, critical point and the boiling points used by various equations for R134a are listed

Table 3.8 Data Used for Comparisons of Sound Velocity

Source	Temperature Range, K	Pressure Range,MPa
Guedes *et al.*[91]	179-380	0.1-71
Goodwin and Moldover[92]	231-340	0.01-0.6
Takagi[93]	290-379	sat-75

in the Table 3.9. In this table, the HM equation is the Huber-McLinden[18] equation of state. The HE equation is the Huber-Ely[19] equation of state and the SRM equation is the equation developed using the stepwise regression method.

Table 3.9 Fixed Points for R134a

EOS	T_t/K	T_b/K	T_c/K	P_c/(Bar)	$\rho_c$mol/dm^3
HM	169.85	247.08	374.179	40.56	5.0308
HE			374.179	40.5859	5.0308
SRM-20	169.85	247.08	374.179	40.5859	5.0308
SRM-29	169.85	247.08	374.18	40.582	4.9798

4. RESULTS

Two different equations of state SRM-20 and SRM-29 for R134a were determined using the stepwise regression algorithm. They differ in that SRM-20 equation was developed using the data available to Huber, McLinden and Ely when they developed their equations while the SRM-29 equation incorporated a new set of data that was published by Tillner-Roth in 1993. The first equation contains 20 terms while the second equation has 29 terms in it. These equations are referred to SRM-20 and SRM-29 in the text.

4.1 TERMS AND COEFFICIENTS

The fitting form for the 20-term equation of state developed using the stepwise regression method is shown in equations 4.1a.

$$\frac{A^r}{RT} = \sum_{n=1}^{8} a_n \, \delta^{j_n}\tau^{i_n} + \sum_{n=9}^{20} a_n \, \delta^{j_n}\tau^{i_n} Exp(-\delta^{k_n}) \qquad (4.1a)$$

The coefficients of the terms and the terms are shown in Table 4.1a.

The 29-term equation of state developed using the stepwise regression algorithm is given in eqn. (4.1b). The coefficients of the terms are given in Table 4.1b.

$$\frac{A^r}{RT} = \sum_{n=1}^{8} a_n \, \delta^{j_n}\tau^{i_n} + \sum_{n=9}^{29} a_n \, \delta^{j_n}\tau^{i_n} Exp(-\delta^{k_n}) \qquad (4.1b)$$

The equations developed using the stepwise regression method were compared with the 32-term Schmidt-Wagner equation developed by Huber and Ely[18], and with the 32-term MBWR equation published by Huber and McLinden[19]. The Huber-Ely equation terms are shown in Table 4.2 and the equation is given in eqn (4.2). The i, j and k correspond to the exponents in eqn (4.2). The functional form of the 32-term MBWR equation developed by Huber and McLinden is given in eqn. (4.3) and the coefficients are given in Table 4.3. It is important to note that the 32-term MBWR equation has 40 terms in the Helmholtz representation. The Schmidt-Wagner equation is a 32-term Helmholtz equation.

Table 4.1a Coefficients of the terms in the 20 terms equation developed using Stepwise Regression Method.

No	Coefficient	k	j	i
1	0.515642224879	0	1	0
2	-0.193917011718 10^{01}	0	1	1.5
3	0.537938165912	0	1	2
4	0.401426659968 10^{-01}	0	2	-0.5
5	0.305273248580 10^{-01}	0	4	0
6	-0.297810890688 10^{-02}	0	6	0
7	0.226854465512 10^{-03}	0	8	0
8	-0.483748925968 10^{-06}	0	11	0
9	-0.332238253273	1	1	3
10	-0.854539536926 10^{-01}	1	4	2
11	-0.148963202736	2	1	5
12	0.237064187194 10^{-01}	2	2	5.5
13	-0.473360114117 10^{-01}	2	2	8
14	-0.237764417898 10^{-02}	2	10	2
15	0.164691052754 10^{-02}	2	10	4
16	0.348274034129 10^{-04}	2	13	4
17	-0.492250920960 10^{-04}	2	14	4
18	-0.449774416003 10^{-02}	3	3	24
19	0.191748141792 10^{-02}	4	4	0
20	-0.152032540488 10^{-02}	5	4	19

Table 4.1b Coefficients of the terms for the 29 term equation developed using the stepwise regression method.

No.	Coefficient	k	i	j
1	0.841109200503	0	1	0
2	-0.125757148057 10^{01}	0	1	1
3	-0.491616856151	0	1	1.5
4	0.743368672405 10^{-01}	0	2	1.5
5	0.143508418211 10^{-01}	0	4	-0.5
6	-0.149900959363 10^{-04}	0	8	2
7	0.139111200775 10^{-05}	0	11	0
8	0.215271872748	1	1	0
9	-0.514743386174	1	1	1
10	-0.721264439476 10^{-01}	1	2	5
11	-0.897033194387 10^{-01}	1	4	3
12	-0.197723096804 10^{-03}	1	5	6
13	0.364194630679 10^{-02}	1	6	4
14	-0.654581678091	2	1	9
15	0.709809808513	2	1	10
16	-0.199287296713	2	1	11
17	0.250141817331 10^{-01}	2	5	2
18	0.687163601800 10^{-03}	2	6	12
19	-0.258945722826 10^{-02}	2	7	10
20	0.435245157915 10^{-02}	2	8	6.5
21	-0.477747611374 10^{-03}	2	10	4
22	0.353515856584 10^{-03}	2	11	7
23	-0.364671395761 10^{-04}	2	14	4
24	0.194518835486 10^{-01}	3	2	15
25	-0.299639515710 10^{-01}	3	3	24
26	0.942077507052 10^{-02}	3	5	23
27	0.416267736322 10^{-02}	4	2	8
28	0.169876569322 10^{-02}	4	4	38
29	-0.108950129049 10^{-02}	5	4	19

Table 4.2 : Huber-Ely Equation of state

No.	Coefficient	k	j	i
1	6.9850186888 10^{-01}	0	1	0
2	-2.4178317436	0	1	1.5
3	7.3378948102 10^{-01}	0	1	2.5
4	-4.1495768445 10^{-02}	0	2	-0.5
5	4.0532419921 10^{-01}	0	2	1.5
6	-1.5898815083 10^{-01}	0	2	2
7	7.2731621002 10^{-02}	0	3	0
8	-6.1352048664 10^{-02}	0	3	1
9	-6.4311629531 10^{-02}	0	3	2.5
10	6.2851751422 10^{-04}	0	6	0
11	1.1155870005 10^{-03}	0	7	2
12	8.0357317942 10^{-07}	0	7	5
13	-2.0918379083 10^{-04}	0	8	2
14	-5.1830999804 10^{-01}	2	1	5
15	1.3728130727 10^{-01}	2	1	6
16	2.5590649449 10^{-01}	2	2	3.5
17	-1.4462862070 10^{-01}	2	2	5.5
18	-2.3839665361 10^{-01}	2	3	3
19	-1.9663126342 10^{-02}	2	3	7
20	-1.0498227490 10^{-02}	2	5	6
21	-1.8993873817 10^{-03}	2	6	8.5
22	-2.7991913410 10^{-02}	2	7	4
23	4.8422141429 10^{-03}	2	8	6.5
24	-1.1647772450 10^{-03}	2	10	5.5
25	1.3054121301 10^{-03}	4	2	22
26	5.3318443331 10^{-02}	4	3	11
27	-4.1079326809 10^{-02}	4	3	18
28	-7.6728830119 10^{-02}	4	4	11
29	5.7908973713 10^{-02}	4	4	23
30	1.2353764109	4	5	17
31	-1.3769795888	4	5	18
32	1.5018380896 10^{-01}	4	5	23

The Huber-Mclinden equation is shown table 4.3. The form of the equation is shown in equation 4.3.

$$\frac{A^r}{RT} = \sum_{n=1}^{13} b_n\, \delta^j \tau^i + \sum_{n=14}^{32} b_n\, \delta^j \tau^i Exp(-\delta^k) \tag{4.2}$$

$$P-\rho RT = \sum_{n=1}^{19} b_n\, \rho^j T^i + \sum_{n=19}^{32} b_n\, \rho^j T^i Exp\left(-\left(\frac{\rho}{\rho_c}\right)^k\right) \tag{4.3}$$

4.2 PVT DATA

The 1992 data set published by Tillner-Roth Baehr (TRB) was used for fitting all the equations. The comparisons for the TRB data set are shown in Figures 4.1 and 4.2. The Huber-McLinden, Huber-Ely and SRM-20 equation fit the data well away from the critical region. Near the critical region, these equations show higher deviations. The SRM-29 term equation fits the data well over the whole range of PVT measurements including the critical region. Most of the deviations are within -0.5 and +0.5%. In this case, the SRM-29 equation is markedly better in reproducing the PVT results, especially near the critical region at pressures around 40 bar. Statistical summaries of the deviations are presented in Table 4.4. For the statistical test, AADP is the absolute average deviation for pressure, AAD is absolute average deviation for density, RMS

Table 4.3 Huber-Mclinden Equation: Coefficients and terms

No.	Coefficients	k	j	i
1	$0.965209362220 \ 10^{-01}$	0	2	1
2	$-0.401824768890 \ 10^{01}$	0	2	0.5
3	$0.395239532860 \ 10^{02}$	0	2	0
4	$0.134532868960 \ 10^{04}$	0	2	-1
5	$-0.139439741350 \ 10^{07}$	0	2	-2
6	$-0.309281355180 \ 10^{-02}$	0	3	1
7	$0.292381512280 \ 10^{01}$	0	3	0
8	$-0.165146613560 \ 10^{04}$	0	3	-1
9	$0.150706003120 \ 10^{07}$	0	3	-2
10	$0.534973948310 \ 10^{-04}$	0	4	1
11	0.543933317620	0	4	0
12	$-0.211326049760 \ 10^{03}$	0	4	-1
13	$-0.268191203850 \ 10^{-01}$	0	5	0
14	-0.541067125950	0	6	-1
15	$-0.851731779400 \ 10^{03}$	0	6	-2
16	0.205188253650	0	7	-1
17	$-0.733050188090 \ 10^{-02}$	0	8	-1
18	$0.380655963860 \ 10^{01}$	0	8	-2
19	-0.105832087590	0	9	-2
20	$-0.679243084420 \ 10^{06}$	2	3	-2
21	$-0.126998378600 \ 10^{09}$	2	3	-3
22	$-0.426234431830 \ 10^{05}$	2	5	-2
23	$0.101973338230 \ 10^{10}$	2	5	-4
24	$-0.186699526780 \ 10^{03}$	2	7	-2
25	$-0.933426323420 \ 10^{05}$	2	7	-3
26	$-0.571735208960 \ 10^{01}$	2	9	-2
27	$-0.176762738790 \ 10^{06}$	2	9	-4
28	$-0.397282752310 \ 10^{-01}$	2	11	-2
29	$0.143016844800 \ 10^{02}$	2	11	-3
30	$0.803085294260 \ 10^{-04}$	2	13	-2
31	-0.171959073550	2	13	-3
32	$0.226238385660 \ 10^{01}$	2	13	-4

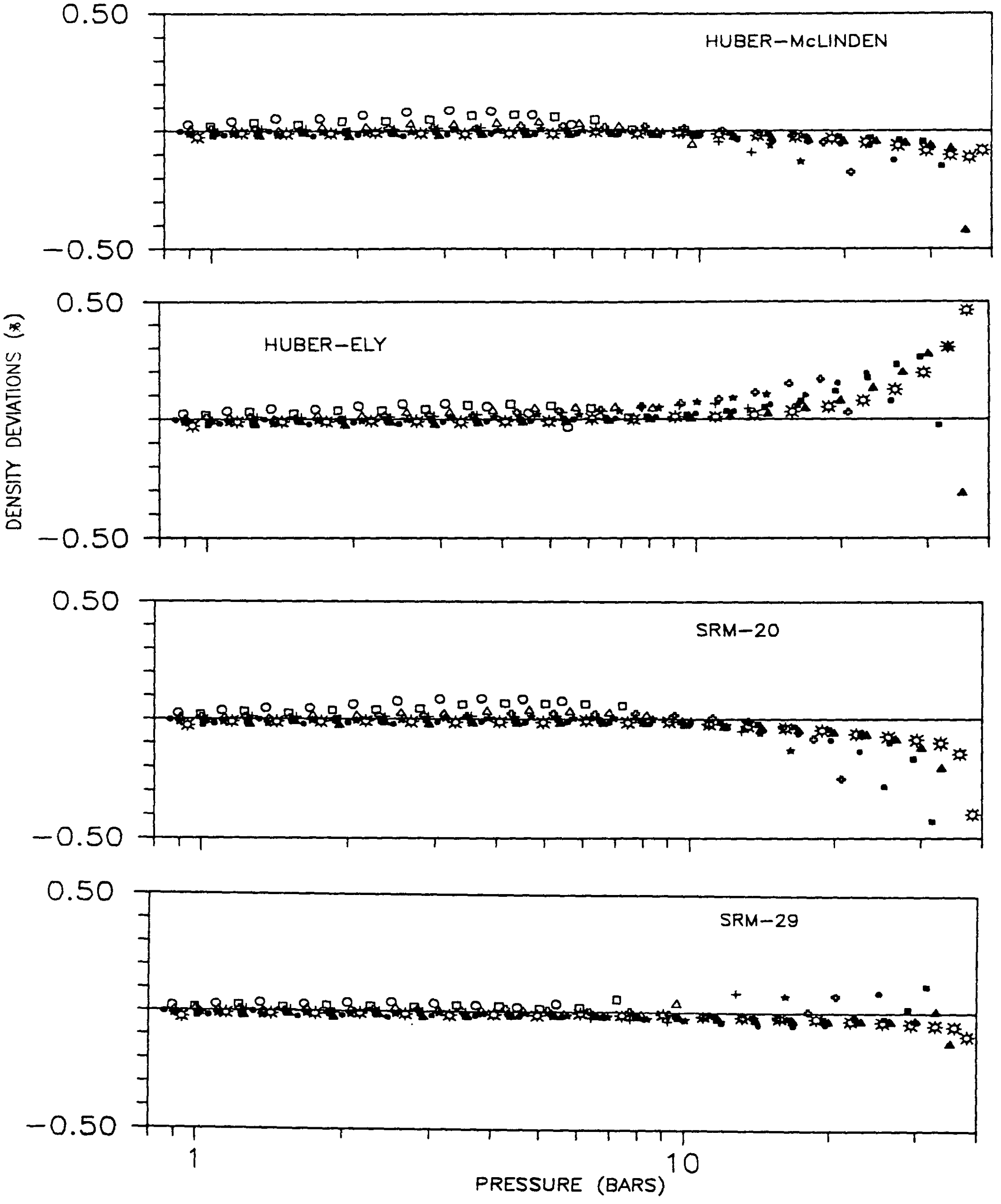


FIG 4.1 Comparison of experimental densities of Tillner-Roth and Baehr (1992) with calculated densities.
(○) 293 K; (□) 303 K; (Δ) 313 K ; (+) 323 K; (★) 333 K; (╬) 343 K; (•) 353 K; (■) 363 K; (▲)368 K; (✲) 373 K.

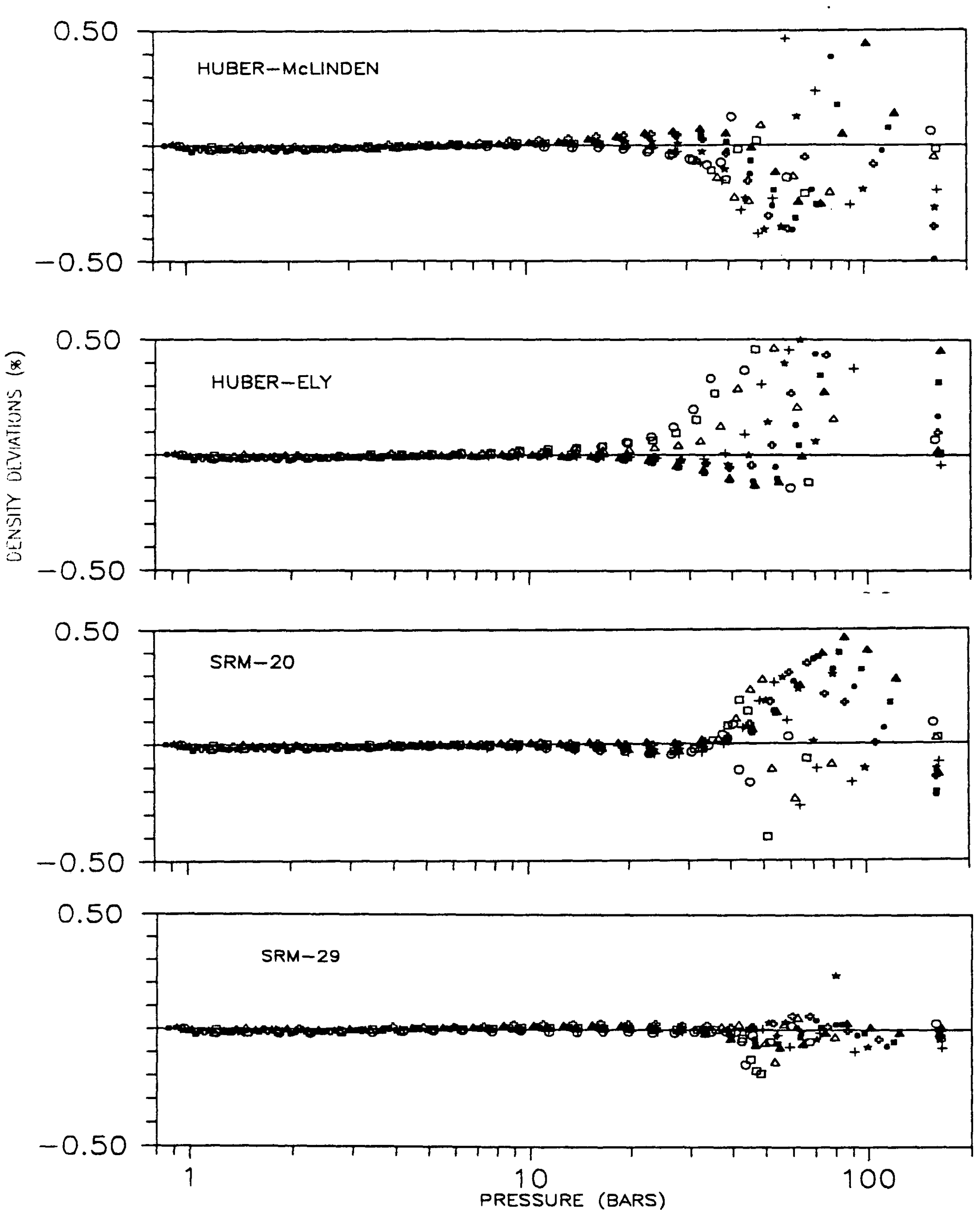


FIG 4.2 Comparison of experimental densities of Tillner-Roth and Baehr (1992) with calculated densities.
(○) 378 K; (□) 383 K; (Δ) 393 K ; (+) 403 K; (★) 413 K; (╬) 423 K; (•) 433 K; (■) 443 K; (▲)453 K.

Table 4.4 : Statistical comparisons for Tillner-Roth Baehr(1992) data

EQUATION	NO OF PTS	AAD % AADP %	BIAS % BIASP %	RMS % RMSP %
HE	411	0.134 0.077	0.048 -0.045	0.426 0.172
HM	411	0.086 0.061	0.000 0.018	0.227 0.155
SRM-20	411	0.074 0.044	-0.017 0.001	0.265 0.088
SRM-29	411	0.023 0.017	-0.009 0.006	0.055 0.031

is the root mean square error and RMSP is for the pressure.

The data set by Weber was used for fitting by HM, HE and the SRM-20. The comparisons for this data set are shown in Fig 4.3. All the equations fit this data to within an accuracy of ±1%. The experimental accuracy of this data set was reported to be ±0.02% in density. The statistical comparisons for the Weber data set are shown in Table 4.5. We note in this case that the SRM-20 and HM equations represent the data slightly better than the SRM-29 equations which did not incorporate this data set.

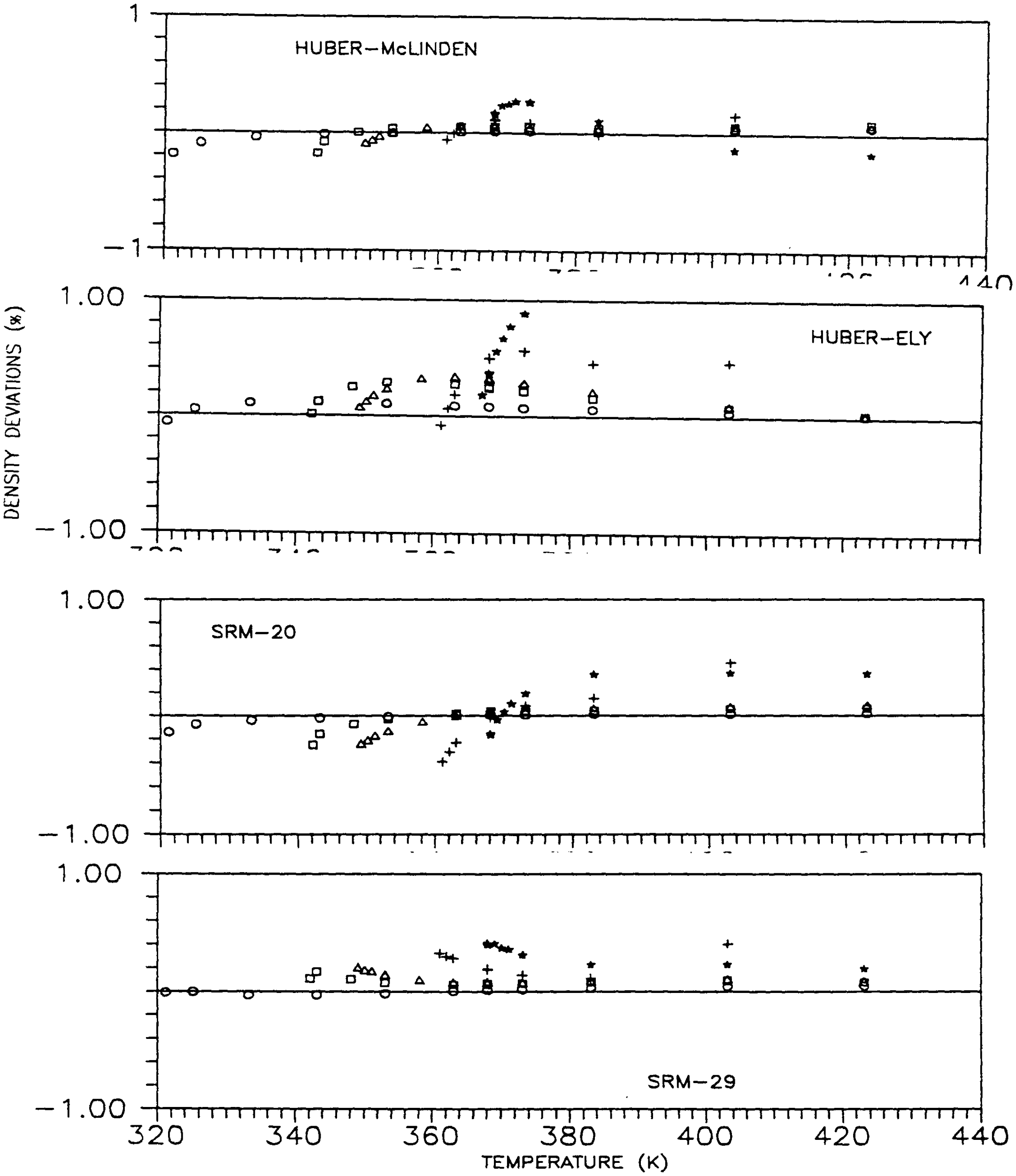


FIG 4.3 Comparison of experimental densities of Weber (1989) with calculated densities. Isochores in kg m^{-3}
(○) 61; (□) 110; (Δ) 133; (+) 196; (★) 237;

Table 4.5 : Statistical comparisons for Weber data

EQUATION	NO OF PTS	AAD % AADP %	BIAS % BIASP %	RMS % RMSP %
HE	69	0.241 0.114	0.237 -0.112	0.218 0.082
HM	69	0.076 0.036	0.042 0.017	0.093 0.042
SRM-20	69	0.092 0.044	-0.007 -0.003	0.145 0.069
SRM-29	69	0.137 0.059	0.135 -0.057	0.132 0.045

particularly in the 370-380 K region. This implies a slight inconsistency between the Weber and TRB(1993) data sets.

The (1993) TRB data set was used only by the SRM-29 equation for fitting purposes. The comparisons for this data set are shown in Fig 4.4, 4.5 and 4.6. For the low temperature isotherm, all the equations predict the data within ±0.2%. For isotherms close to the critical temperature, higher deviations are observed. The SRM-29 term equation fits this data best. The experimental accuracy of this data set was

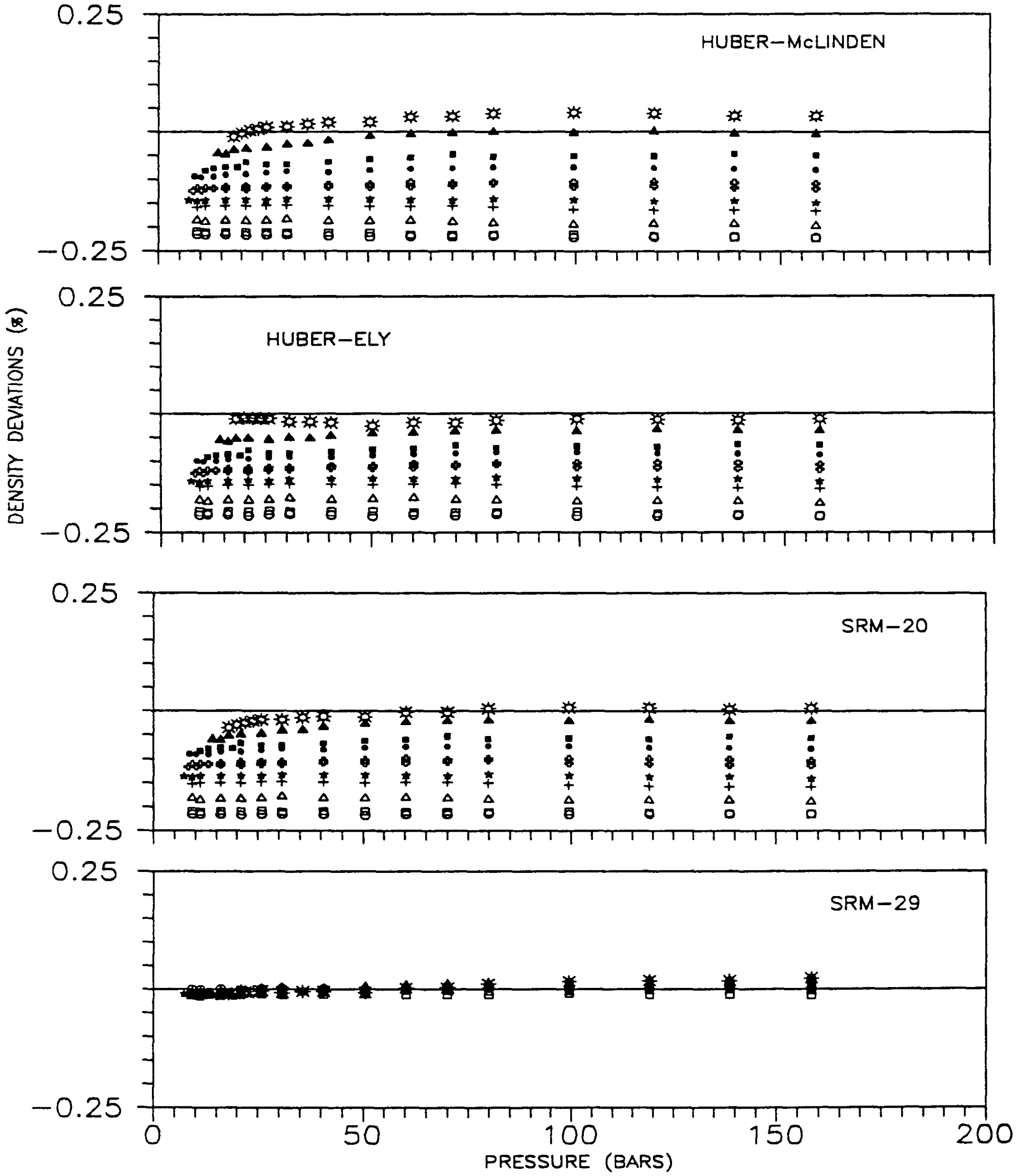


FIG 4.4 Comparison of experimental densitites of Tillner-Roth and Baehr (1993) with calculated densities.
(○) 243 K; (□) 253 K; (Δ) 263 K ; (+) 273 K; (★) 283 K; (╬) 293 K; (●) 303 K; (■) 313 K; (▲)323 K; (✲) 333 K.

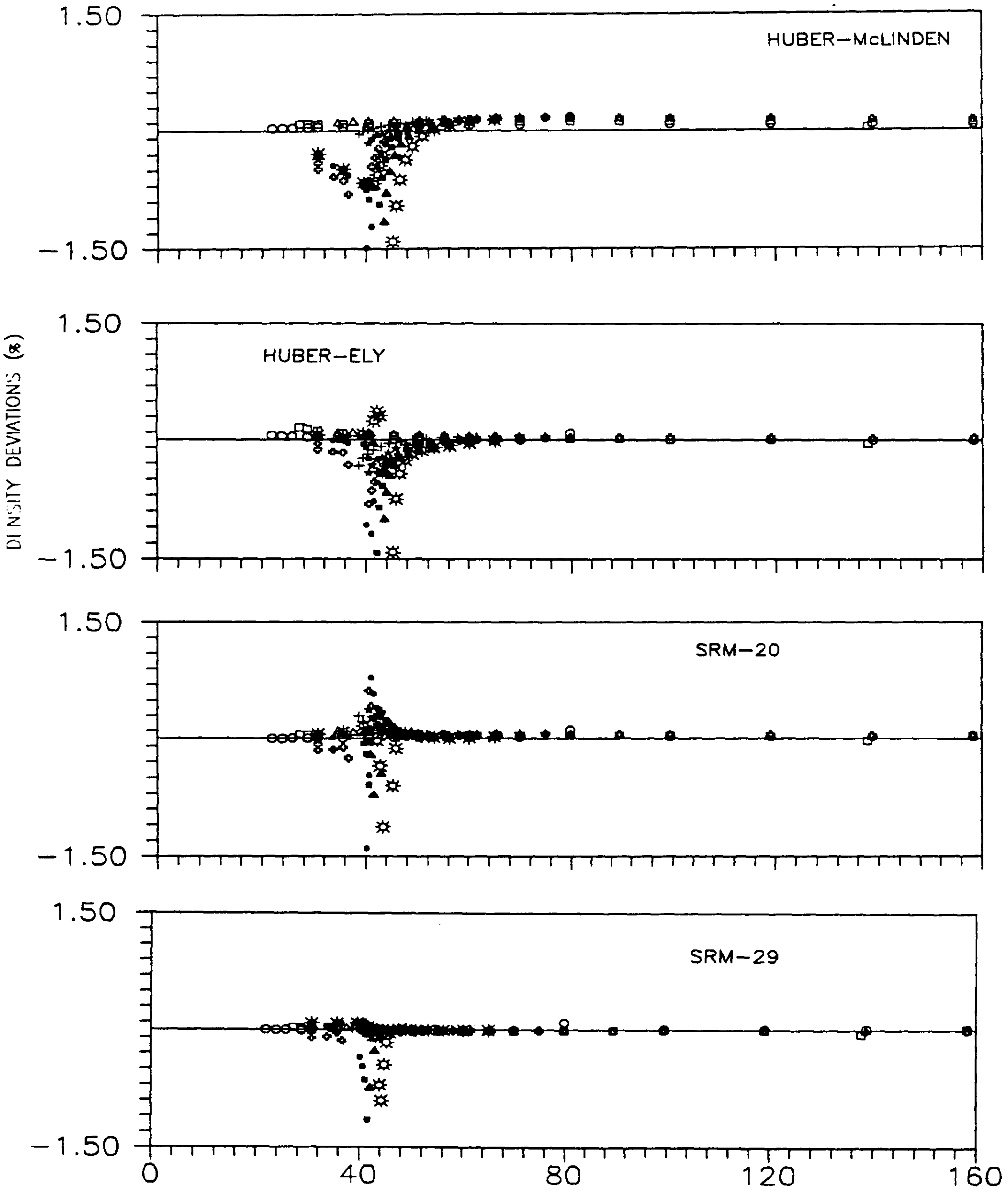


FIG 4.5 Comparison of experimental densitites of Tillner-Roth and Baehr (1993) with calculated densities.
(○) 343 K; (□) 353 K; (Δ) 363 K ; (+) 370 K; (★) 372 K; (╬) 373 K; (●) 374 K; (■) 375 K; (▲) 376 K; (✲) 378 K.

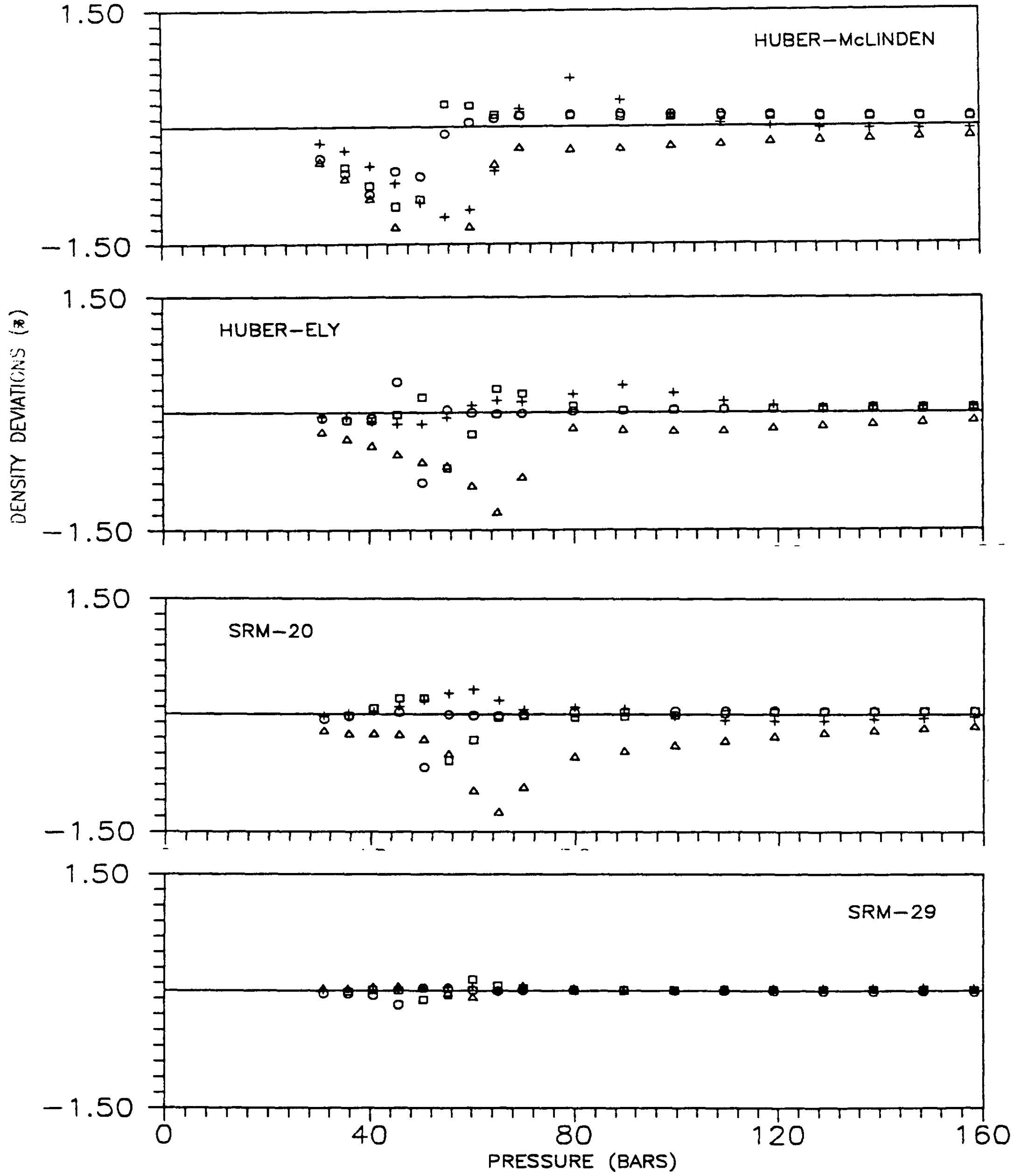


FIG 4.6 Comparison of experimental densitites of Tillner-Roth and Baehr (1993) with calculated densities.
(○) 383 K; (□) 393 K; (Δ) 403 K ; (+) 413 K.

Table 4.6 : Statistical comparisions for (1993) TRB data

EQUATION	NO OF PTS	AAD % AADP %	BIAS % BIASP %	RMS % RMSP %
HE	432	0.266 6.066	-0.181 5.902	0.515 14.991
HM	432	0.141 6.263	-0.037 5.801	0.401 15.392
SRM-20	432	0.168 5.967	-0.087 5.761	0.414 15.10
SRM-29	432	0.030 0.272	-0.010 0.157	0.103 0.702

reported to be ±0.1%. The statistical comparisons for the TRB data set are shown in Table 4.6. The SRM-29 equation which uses this data set in the fitting process, is better by almost an order of magnitude of the data set.

The data set by Hou *et al.* and Tillner-Roth and Baehr(1993) overlap. All the equations used either one of the data sets. The Hou data set was used for fitting by HM, HE and the SRM-20. The Tillner-Roth Baehr (1993) data set was used by the SRM-29 equation.

The comparisons for the Hou *et al.* are shown in Fig. 4.7 and 4.8. The lower temperature isotherms are fitted by both equations to within an accuracy of ±0.2%, whereas the isotherms near the critical temperature are fitted to within an accuracy of ±0.6%. The accuracy of this data set was reported to be ±0.1% in density. The statistical comparisons for the Hou et.al. data set are shown in Table 4.7.

Table 4.7 : Statistical comparisons for Hou et.al. data

EQUATION	NO OF PTS	AAD % AADP %	BIAS % BIASP %	RMS % RMSP %
HE	429	0.144 2.983	0.104 -0.437	0.163 5.829
HM	429	0.134 2.829	0.099 -0.454	0.158 5.708
SRM-20	429	0.149 2.770	0.105 0.473	0.215 5.446
SRM-29	429	0.174 4.799	0.171 -4.020	0.174 12.306

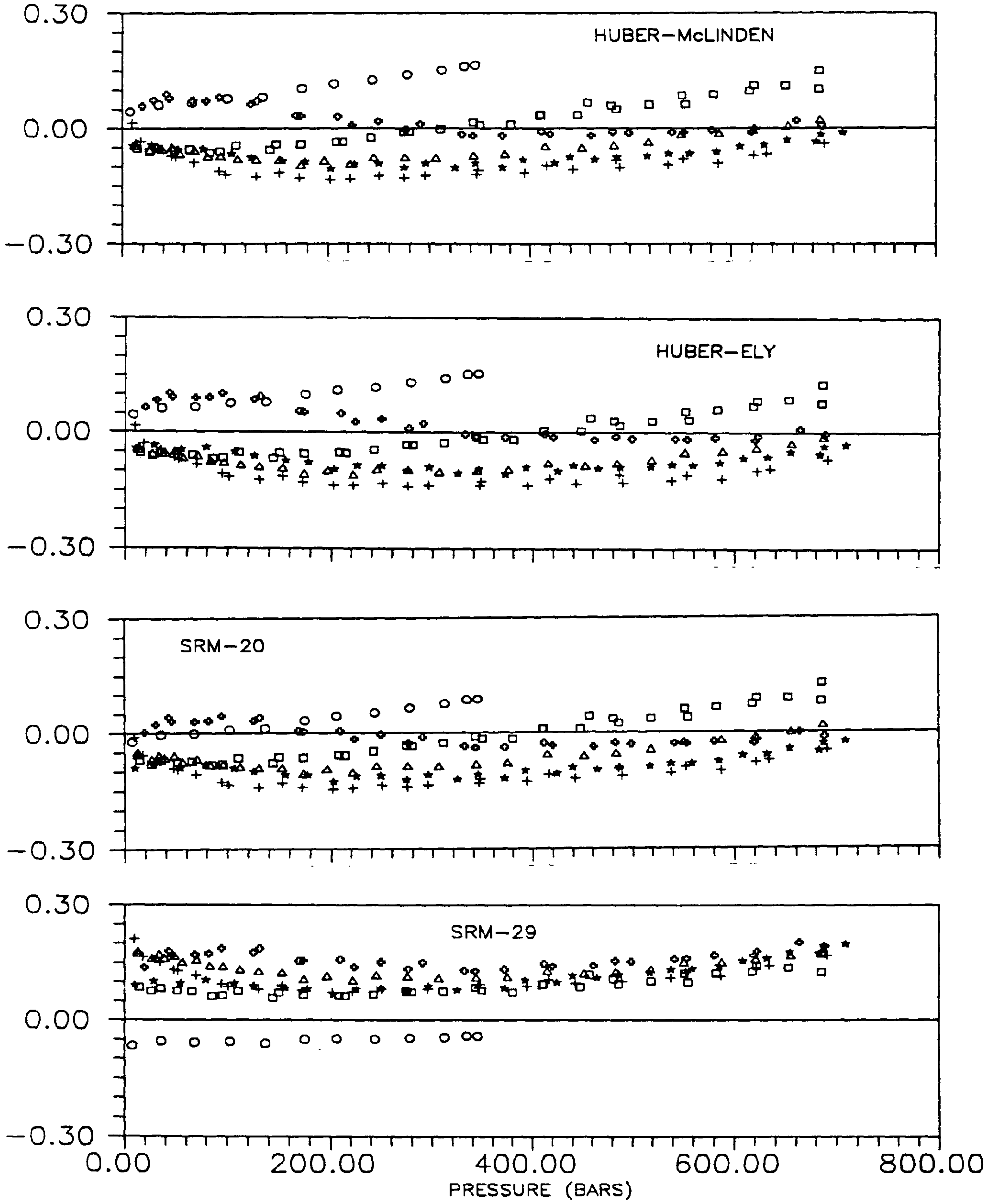


FIG 4.7 Comparison of experimental densitites of Hou et. al. (1993) with calculated densities.
(○) 180 K; (□) 200 K; (Δ) 220 K ; (+) 240 K; (★) 260 K; (╬) 280 K.

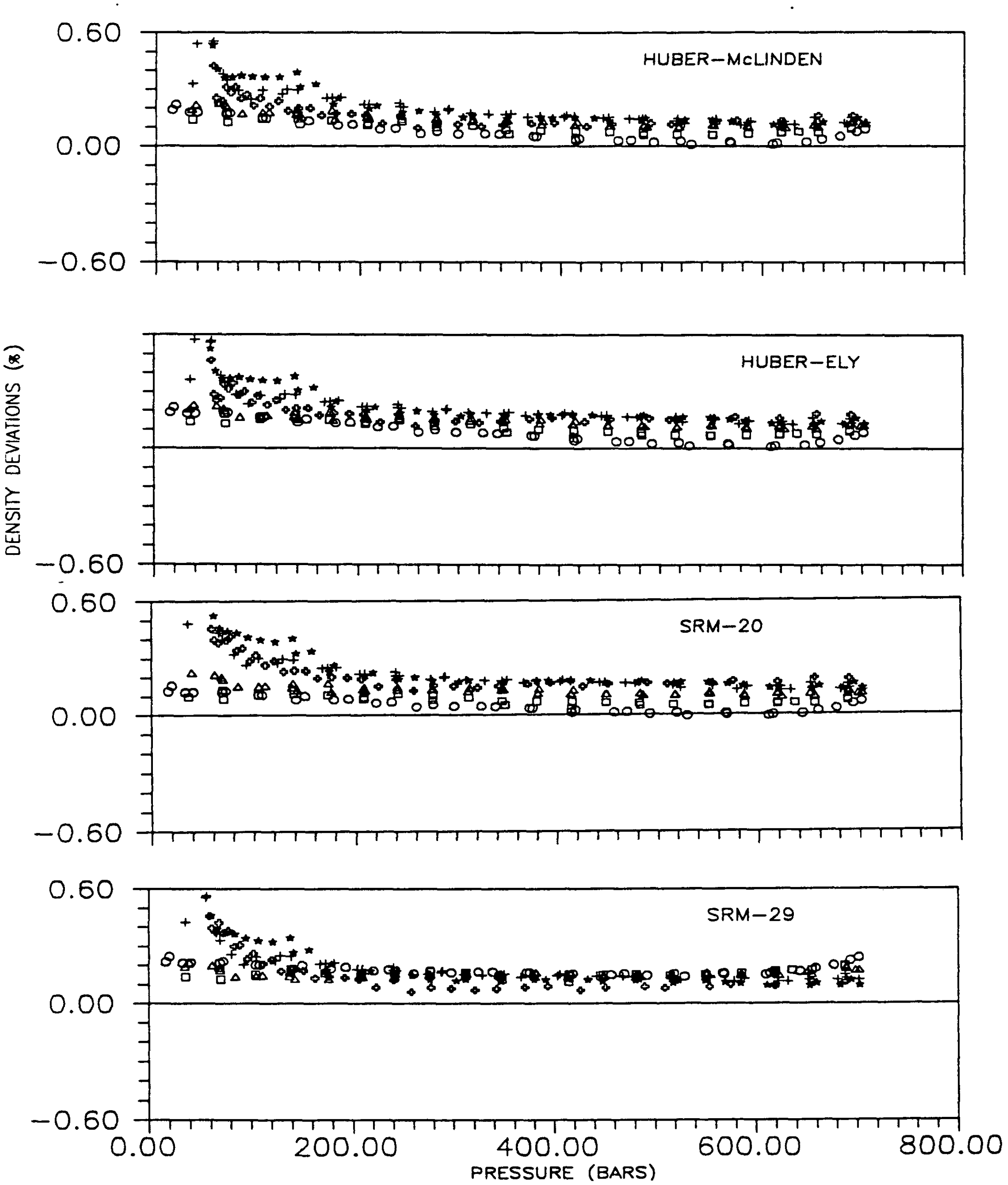


FIG 4.8 Comparison of experimental densitites of Hou et. al. (1993) with calculated densities.
(○) 300 K; (□) 320 K; (Δ) 340 K ; (+) 360 K; (★) 370 K; (╬) 380 K.

The Morrison and Ward data set was used by HM, HE and the SRM-20 equation. This data set overlaps both the Tillner-Roth Baehr (1993) and the Hou *et al*. data set. The comparisons for this data set are shown in Fig 4.9 and 4.10. This data set is represented well by all equations. The experimental accuracy of this data set was reported to be within ±0.3% in density.The statistical comparisons for the Morrison and Ward data set are shown in Table 4.8.

Table 4.8 : Statistical comparisons for Morrison and Ward data

EQUATION	NO OF PTS	AAD % AADP %	BIAS % BIASP %	RMS % RMSP %
HE	128	0.058 1.665	0.025 0.205	0.101 2.377
HM	128	0.054 1.575	0.015 0.450	0.097 2.260
SRM-20	128	0.101 4.532	0.002 3.553	0.140 5.636
SRM-29	128	0.073 3.077	0.043 -1.919	0.109 5.174

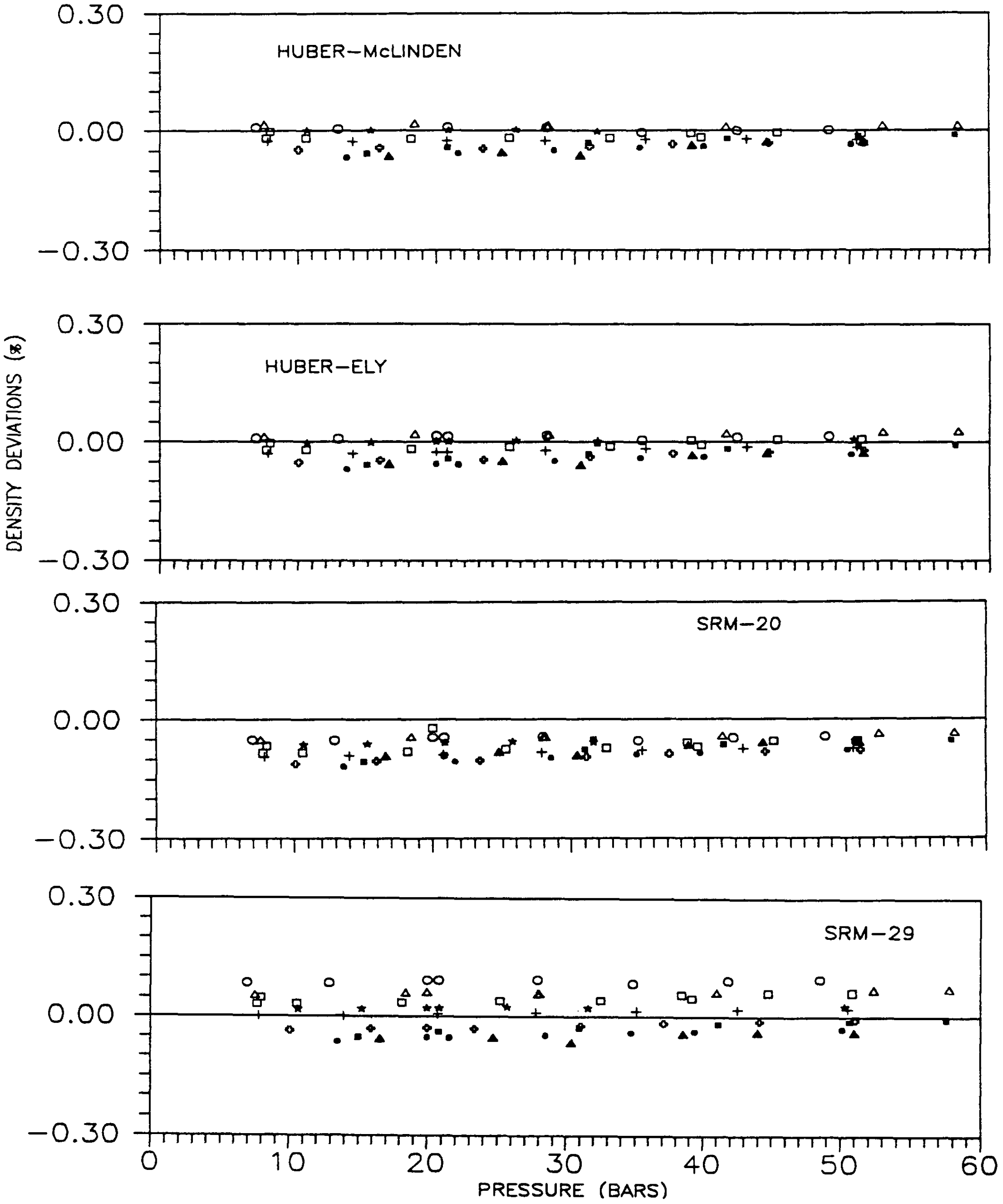


FIG 4.9 Comparison of experimental densities of Morrison and Ward (1989) with calculated densities.
(○) 278 K; (□) 288 K; (Δ) 293 K ; (+) 298 K; (★) 304 K; (╬) 307 K; (•) 317 K; (■) 318 K; (▲)327 K.

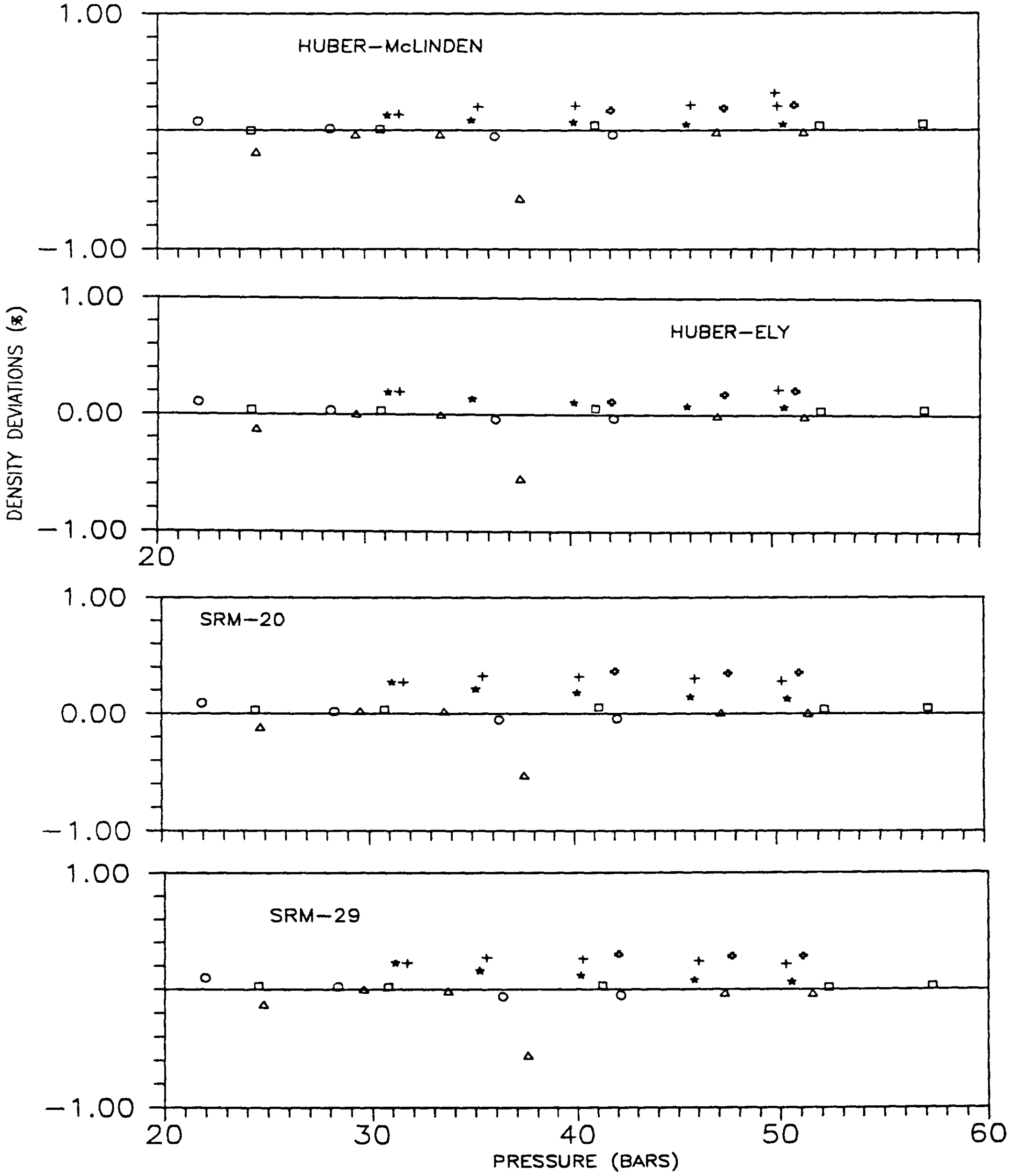


FIG 4.10 Comparison of experimental densities of Morrison and Ward (1989) with calculated densities.
(○) 337 K; (□) 341 K; (Δ) 347 K ; (+) 356 K; (★) 357 K;
(╬) 366 K.

The overall comparisons for all of the PVT data sets are shown in Table 4.9.

Table 4.9 : Overall comparisons for PVT data

EQUATION	NO OF PTS	AAD % AADP %	BIAS % BIASP %	RMS % RMSP %
HE	1341	0.141 2.848	-0.037 2.105	0.401 9.419
HM	1341	0.195 3.002	-0.070 2.052	0.381 9.657
SRM-20	1341	0.129 2.824	0.00 2.007	0.314 9.473
SRM-29	1341	0.079 1.631	0.056 -1.237	0.149 7.229

4.3 Second Virial Coefficients

The comparisons for all the virial data sets are shown in Fig. 4.11. The virial data is represented well by all equations to a temperature of 300 K. Below this temperature,

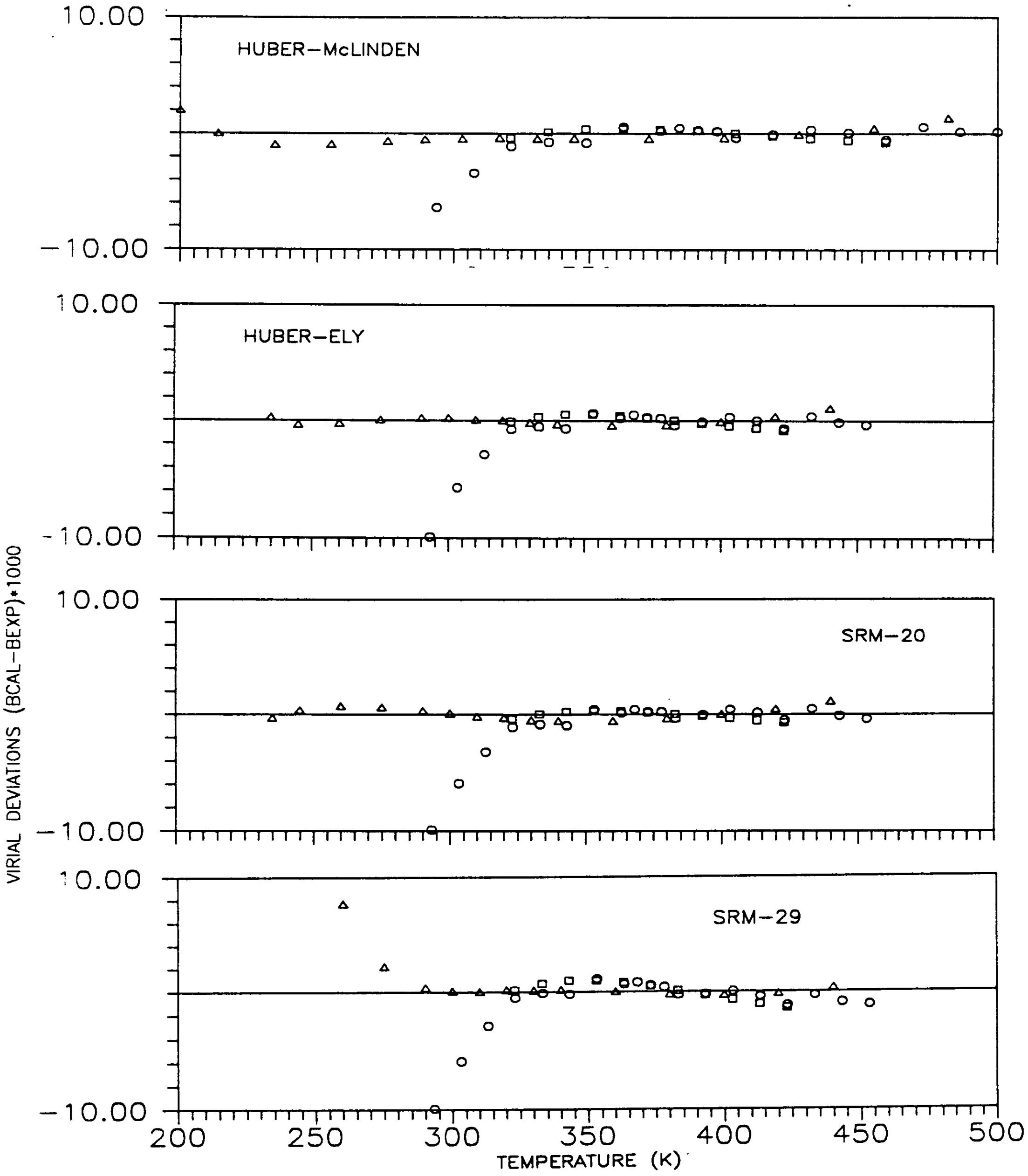


FIG 4.11 Comparison of experimental Second Virial Coefficients with calculated densities.
(○) Tillner-Roth Baehr; (□) Weber; (Δ) Goodwin and Moldover.

the deviations increase rapidly for the Weber data set for all equations. The overall statistical comparisons for the second virial data are shown in Table 4.10. It is interesting to note that the inclusion of the TRB (1993) data set which goes to 243 K, causes a major discrepancy with Goodwin and Moldover's low temperature virials.

Table 4.10 : Overall comparisons for second virial data

EQUATION	NO OF PTS	AAD cm^3/mol	BIAS cm^3/mol	RMS cm^3/mol
HE	42	0.346	0.013	0.418
HM	42	0.473	-0.050	1.025
SRM-20	42	0.392	-0.013	0.418
SRM-29	40	0.719	0.246	1.396

4.4 Isochoric Heat Capacity

There is only one data for C_v data by Magee (1992). This data set was used by all equations. The comparisons for this

Table 4.11 : Overall comparisons for ischoric heat capacity data

EQUATION	NO OF PTS	AAD %	BIAS %	RMS %
HE	150	0.282	-0.050	0.363
HM	150	0.367	-0.179	0.398
SRM-20	150	0.356	-0.017	0.439
SRM-29	150	0.305	-0.059	0.369

data set are shown in Fig. 4.12. All the equations show good prediction of the experimental values for this data set. The overall statistical comparisons for isochoric heat capacity data are shown in Table 4.11.

4.5 Isobaric Heat Capacity

The comparisons for C_p data set by Saitoh *et al.* are shown in Fig 4.13. All the equations fit this data set to within ±1.5%. The overall statistical comparisons for isobaric heat capacity data are shown in Table 4.12.

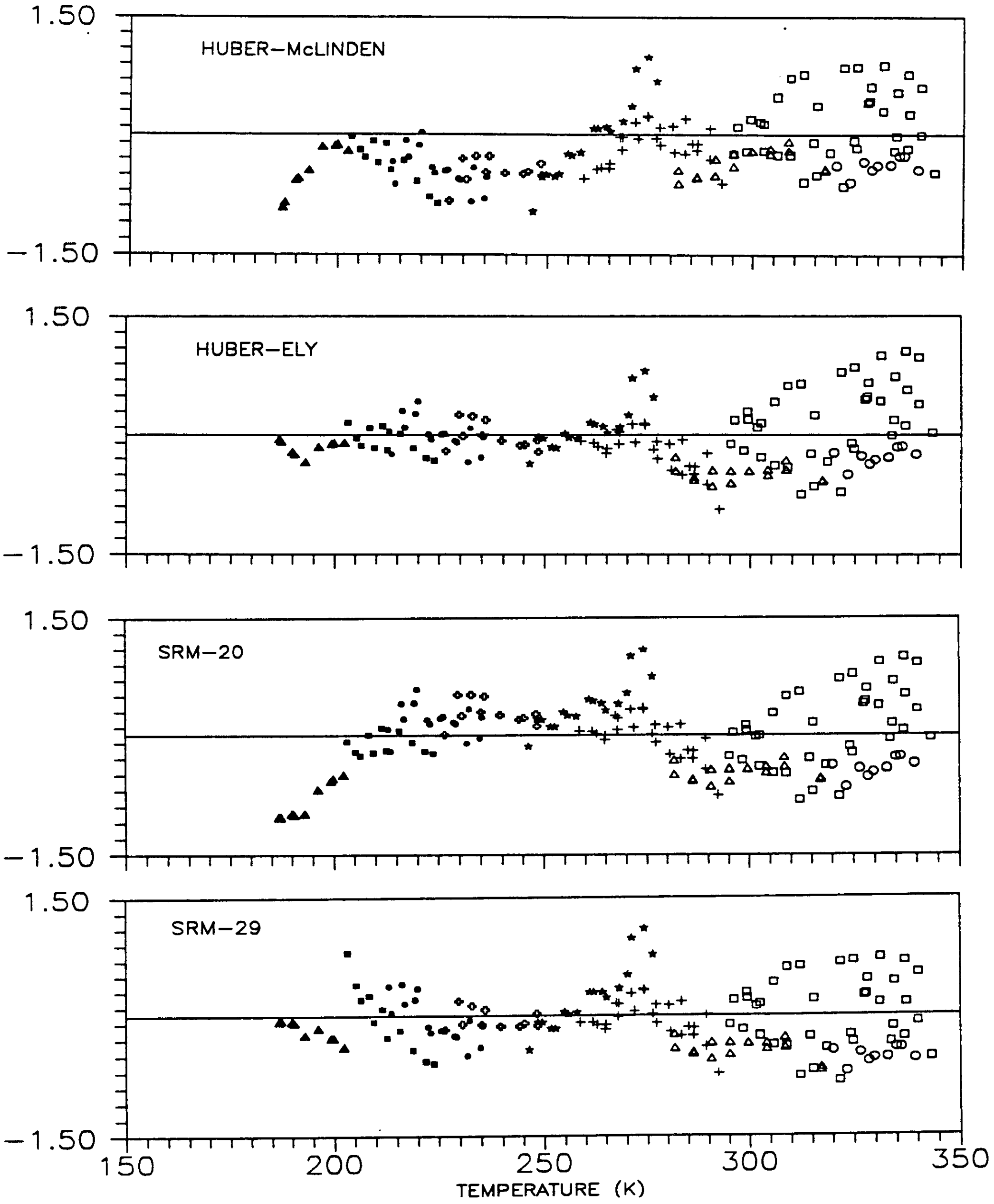


FIG 4.12 Comparison of experimental C_v measurements by Magee (1992) with calculated C_v values.
(○) 1127 kg m^{-3}; (□) 1223 kg m^{-3}; (Δ) 1283 kg m^{-3} ; (+) 1343 kg m^{-3}; (★) 1387 kg m^{-3}; (╬) 1439 kg m^{-3}; (●) 1480 kg m^{-3}; (■) 1505 kg m^{-3}; (▲) 1505 kg m^{-3}; (✲) 1545 kg m^{-3}.

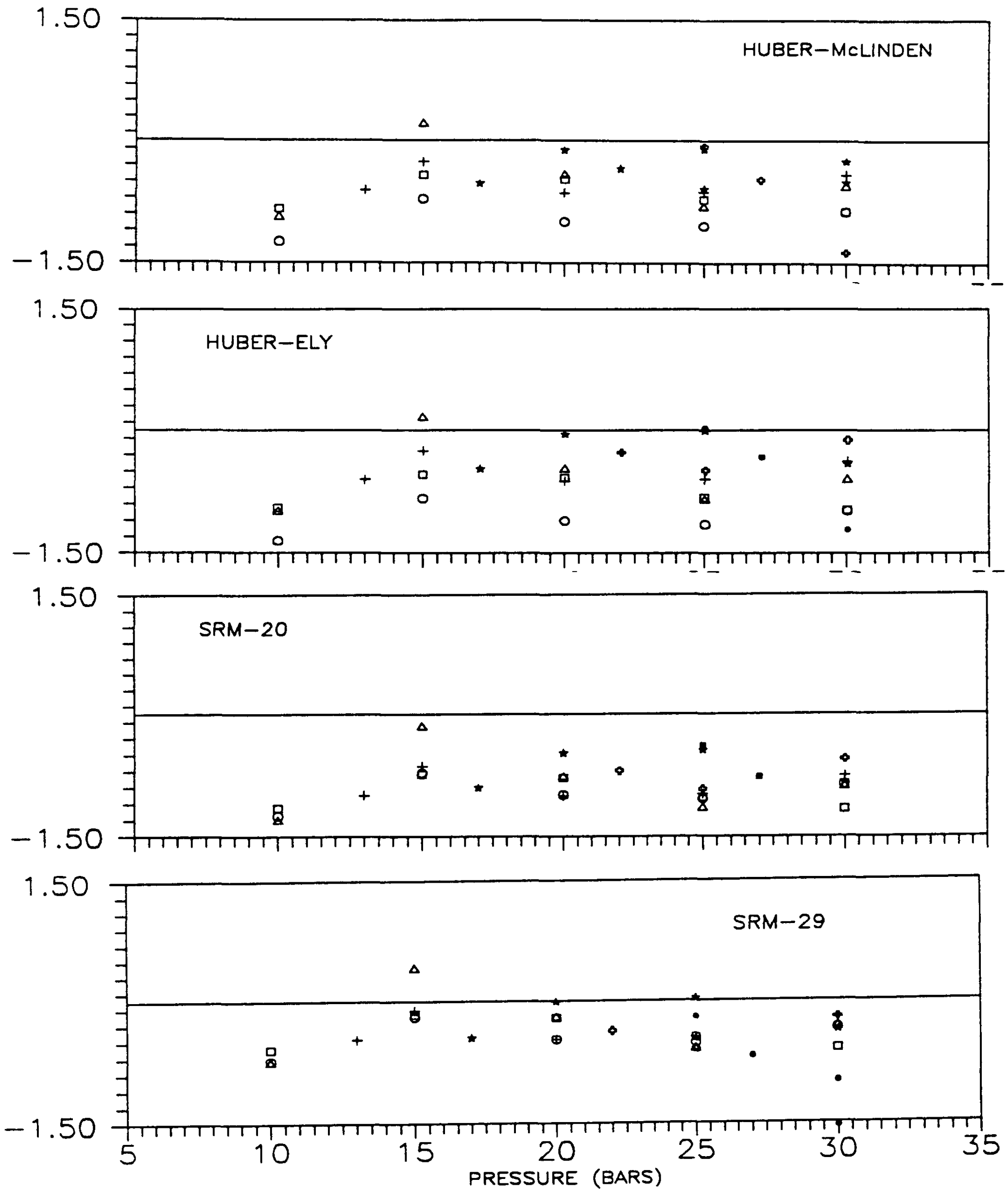


FIG 4.13 Comparison of experimental C_p measurements by Saitoh et al. (1990) with calculated C_p values.
(○) 275 K; (□) 300 K; (Δ) 310 K ; (+) 320 K; (★) 330 K; (╬) 340 K; (●) 350 K.

Table 4.12 : Statistical tests for isobaric heat capacity data

EQUATION	NO OF PTS	AAD %	BIAS %	RMS %
HE	31	0.594	-0.581	0.386
HM	31	0.602	-0.588	0.341
SRM-20	31	0.863	-0.863	0.291
SRM-29	31	0.438	-0.408	0.336

4.6 Speed of Sound

The Guedes and Zollweg data set was used for fitting sound velocity data by HM, HE and SRM-20 equation. The sound velocity data was used to generate $\partial P/\partial \rho$ data which was then fitted. Almost all the equations represent the isotherms very well above the critical pressure, whereas around the critical point, all equations fit the data within ±3.0% as shown by Fig 4.14. The experimental accuracy of this data set was reported to be 1.0 m/s in sound velocity. The overall statistical comparisons for the sound velocity data are shown in table 4.13.

Table 4.13 : Overall statistical tests for sound velocity data

EQUATION	NO OF PTS	AAD %	BIAS %	RMS %
HE	304	0.15	-0.05	0.27
HM	304	0.20	0.01	0.38
SRM-20	304	0.27	-0.12	0.48
SRM-29	304	0.36	-0.10	0.59

We note that inspite of the SRM-29 equation superior performance for PVT behavior, it does not represent the sound velocity as well as the other equations. The comparisons for the Goodwin and Moldover data are shown in Figs. 4.14 and 4.15. HM and SRM-20 equations represent this data well, while the HE equation shows systematic deviations for the lower temperature isotherms. The experimental accuracy of this data set was reported to be ±0.01% in sound velocity.

4.7 Saturation Boundary

4.7.1 Saturated Vapor Pressure

Comparisons for the vapor pressure experimental data

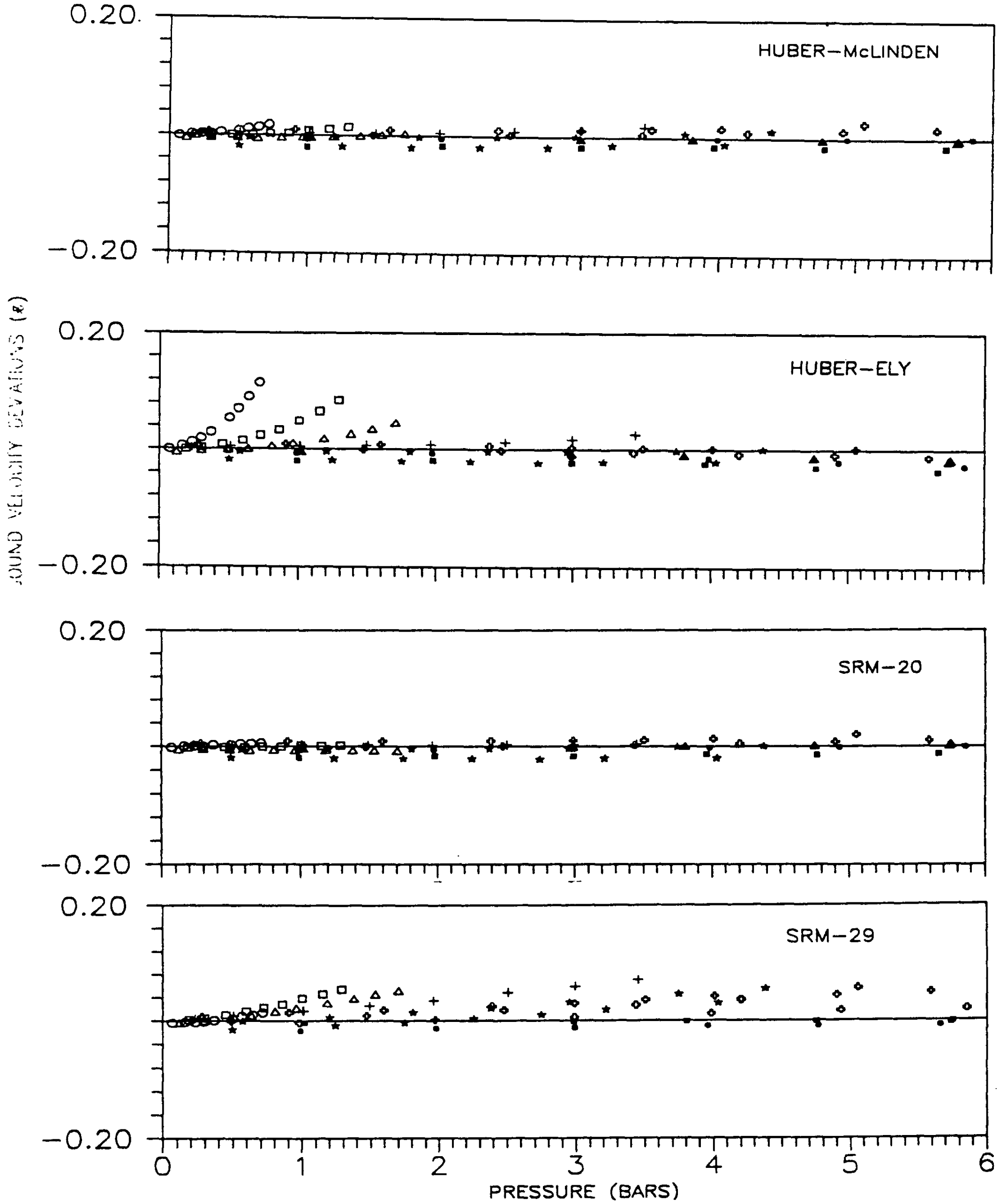


FIG 4.14 Comparison of experimental speed of sound measurements of Goodwin and Moldover with calculated values. (○) 245 K; (□) 260 K; (Δ) 270 K ; (+) 290 K; (★) 300 K; (╬) 310 K; (•) 320 K; (■) 330 K; (▲)340 K.

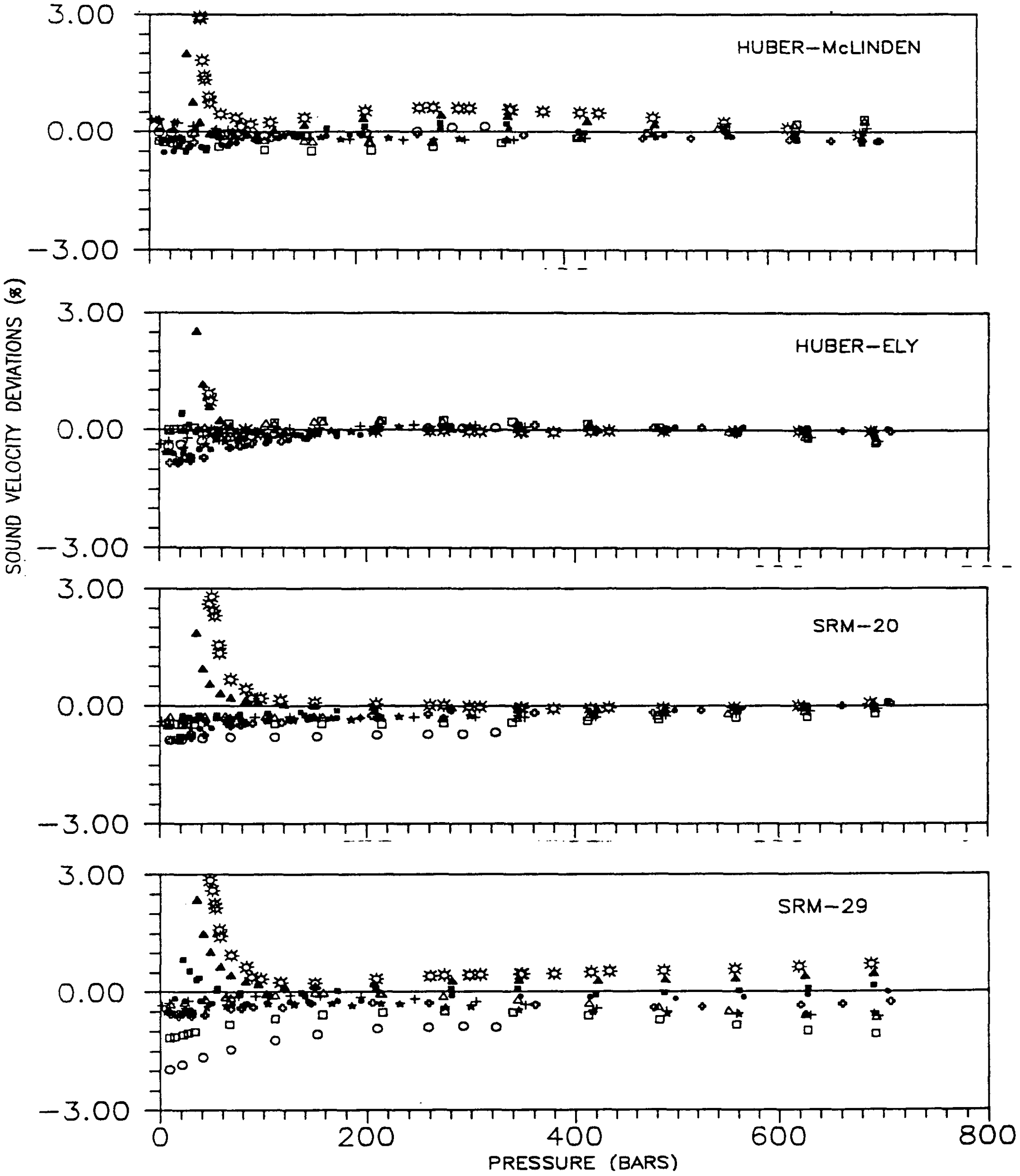


FIG 4.15 Comparison of experimental speed of sound measurements of Guedes and Zollweg (1992) with calculated values.
(○) 180 K; (□) 200 K; (Δ) 220 K ; (+) 240 K; (★) 260 K; (╬) 300 K; (●) 320 K; (■) 340 K; (▲) 360 K; (✲) 380 K.

are shown in Fig 4.16. These comparisons were made with the data sets used in developing the ancillary correlation described in the previous section. All the equations represent the data well within an accuracy of ±1%. The statistical comparisons for vapor pressure data are shown in Table 4.14.

Table 4.14 Statistical comparisons for vapor pressure data

EQUATIONS	AAD %	BIAS %	RMS %
HE	0.983	-0.983	0.372
HM	0.461	-0.441	0.474
SRM-20	1.087	-1.087	0.349
SRM-29	0.184	-0.010	0.346

4.7.2 Saturated Liquid Density

The saturated liquid density data was generated using a correlation developed from experimental data. The comparisons for the data used while developing the correlation are shown in Fig 4.17a & b. All the equations represent the data well within an accuracy of ±1%. The statistical comparisons for saturated liquid density data are shown in Table 4.15.

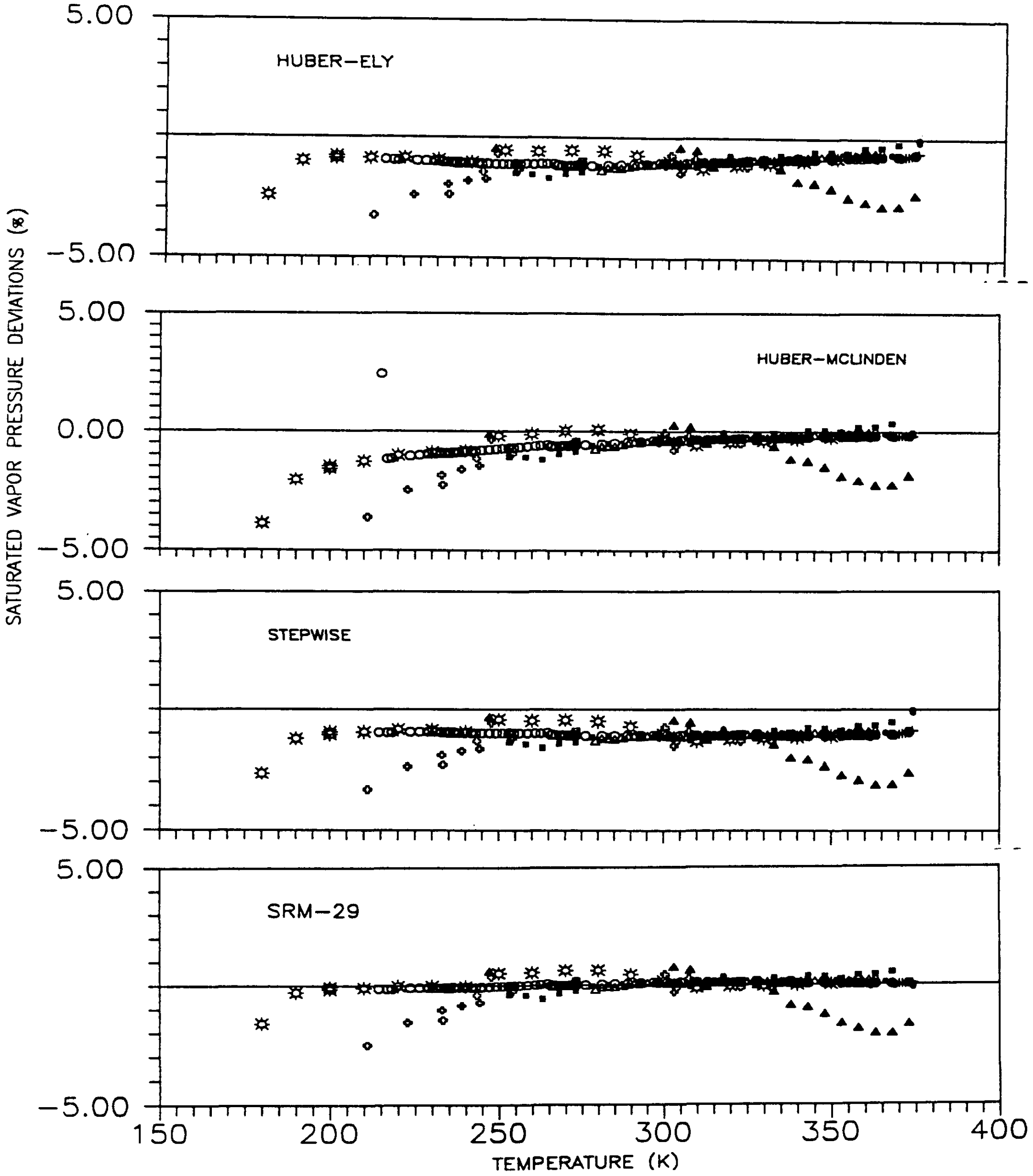


FIG 4.16a Comparison of experimental vapor pressures for R134a.
(○) Goodwin and Weber; (□) Arita et al; (Δ) Zhu and Wu; (+) Baehr and Tillner-Roth; (★) Weber; (╬) Wilson and Basu; (●) Piao et al; (■) Kubota et al; (▲) Nishiumi; (✲) Magee and Howley.

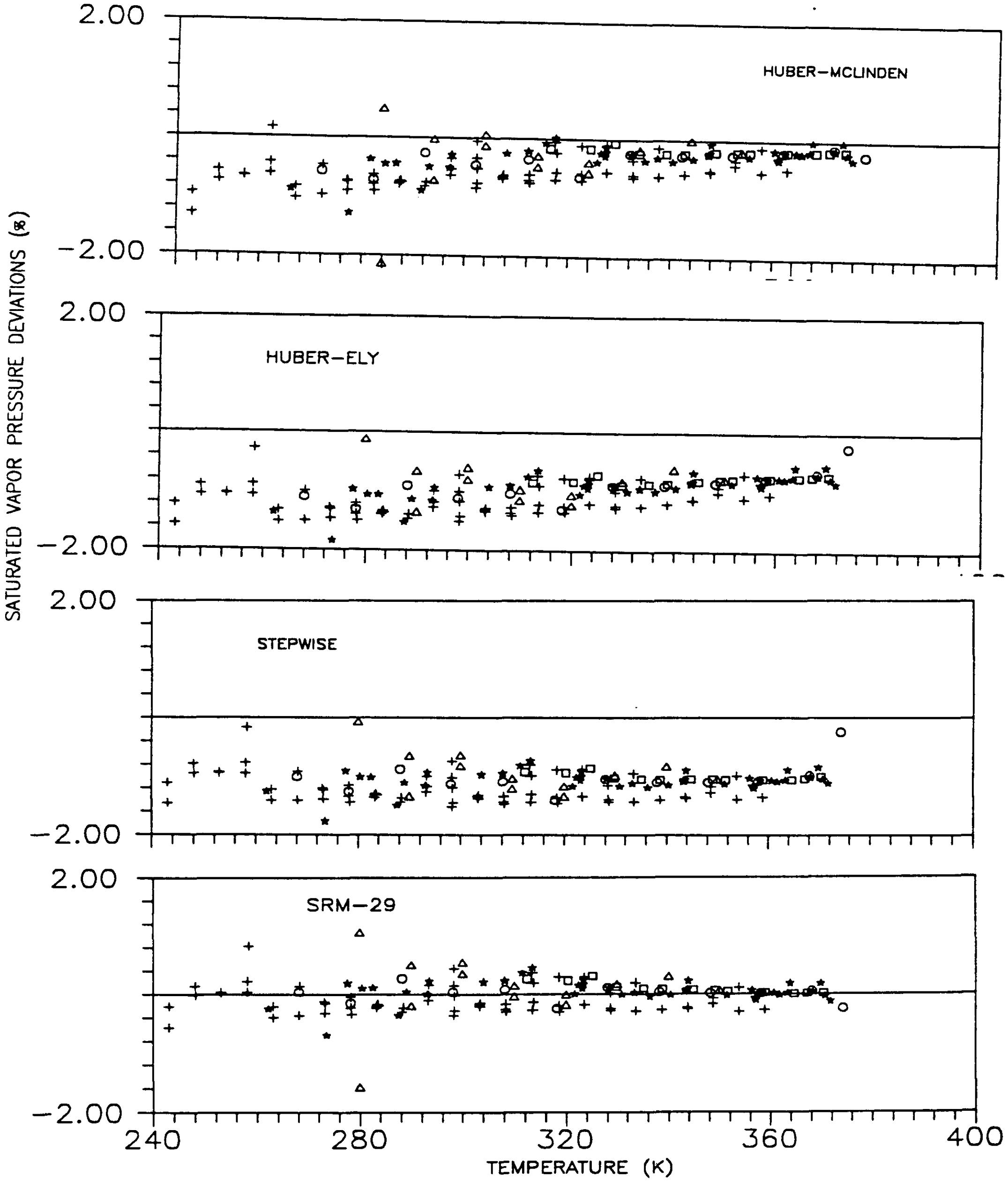


FIG 4.16b Comparison of experimental vapor pressures for R134a.
(○) Morrison and Ward; (□) Niesen; (Δ) Maezawa et al; (+)Baroncini et al; (★) Fukushima;

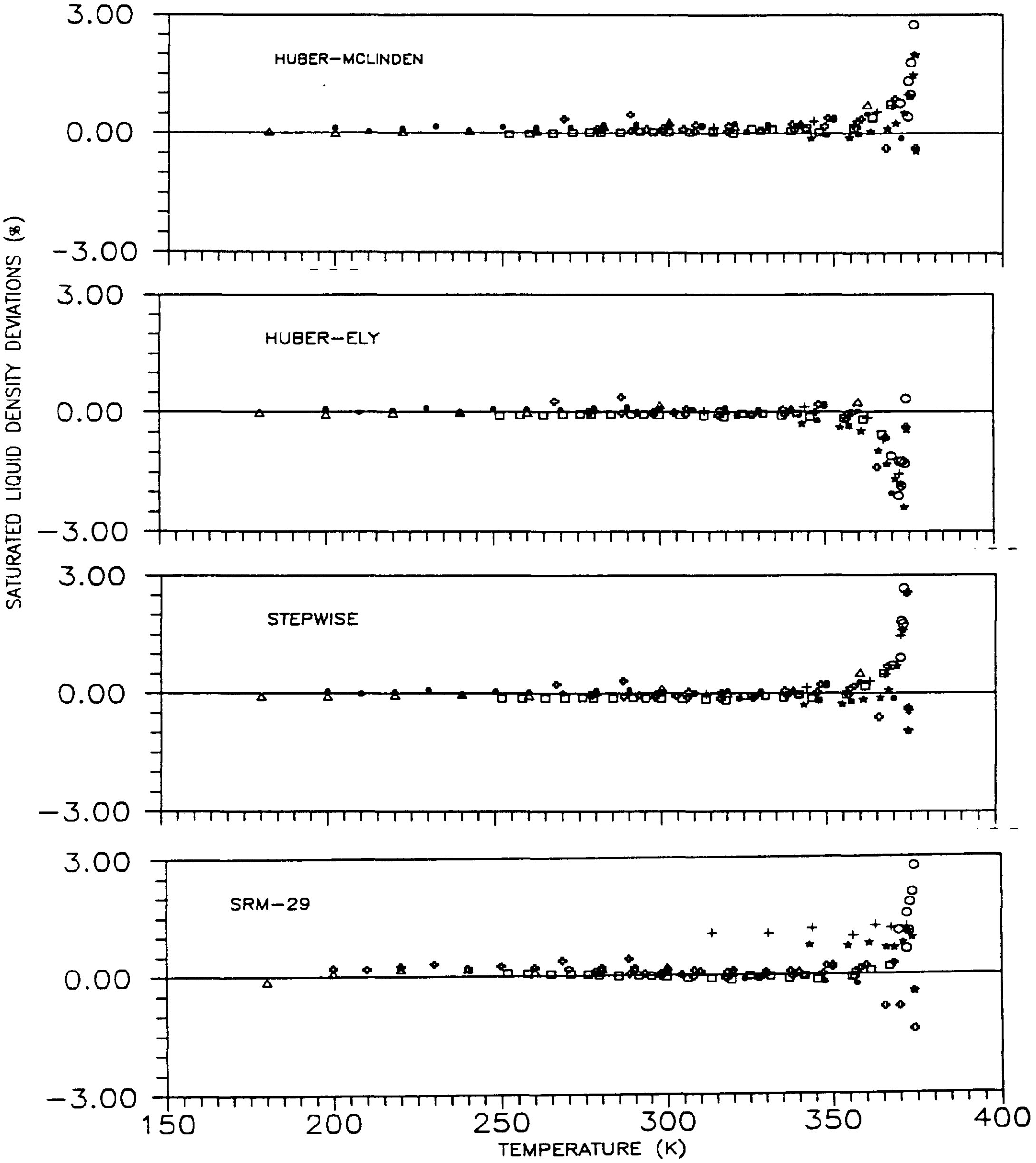


FIG 4.17a Comparison of experimental liquid densities for R134a.
(○) Fukushima; (□) Yokoyama and Takahashi; (Δ)Hou et al; (+) Piao et al; (★) Kabata et al.; (╬) Morrison and Ward; (●)Maezawa et al.; (■)Fukushima M.

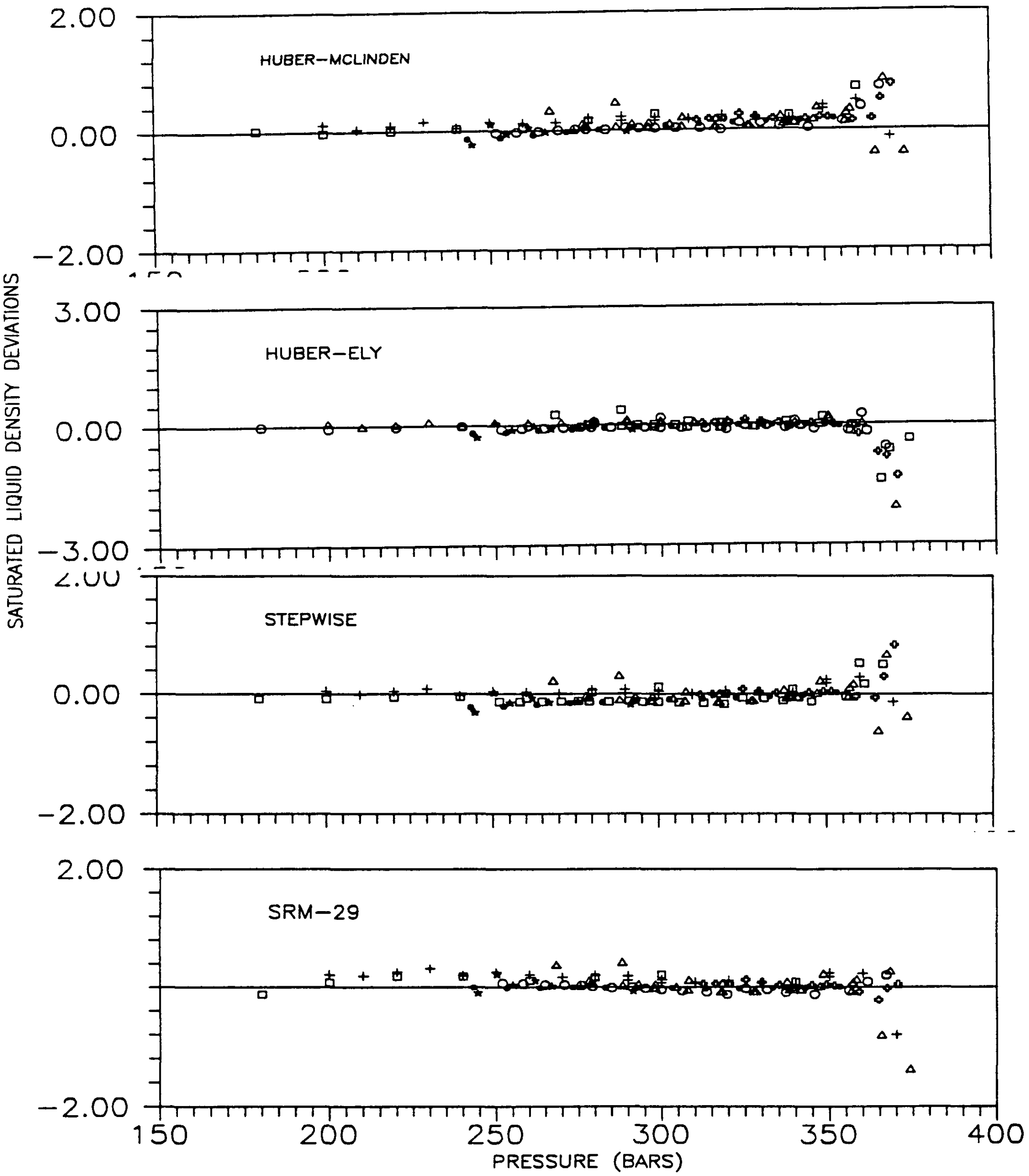


FIG 4.17b Comparison of experimental liquid densities used for fitting the equation of state for R134a. (○)Yokoyama and Takahashi; (□)Hou et al; (Δ)Morrison and Ward;(+) Maezawa et al.; (★) Fukushima M (╬) Niesen (●) Tillner-Roth and Baehr.

Table 4.15 Statistical comparisons for saturated liquid density data

EQUATIONS	AAD %	BIAS %	RMS %
HE	0.433	-0.300	1.181
HM	0.496	0.369	1.782
SRM-20	0.465	0.231	1.612
SRM-29	1.20	0.982	1.162

Table 4.16 Statistical comparisons for saturated liquid density data used for fitting purposes.

EQUATIONS	AAD %	BIAS %	RMS %
HE	0.141	-0.076	0.293
HM	0.154	0.127	0.174
SRM-20	0.113	-0.019	0.169
SRM-29	0.118	0.027	0.205

4.7.3 Saturated Vapor Density

The saturated vapor density data used in the fit was generated using a correlation developed from experimental data described previously. The comparisons for the data used to develop the correlation are shown in Fig 4.18. All the equations represent the data within an accuracy of ±1%. The statistical comparisons for saturated vapor density data are shown in Table 4.17 and 4.18.

Table 4.17 Statistical comparisons for saturated vapor density data

EQUATIONS	AAD %	BIAS %	RMS %
HE	1.748	-0.172	3.372
HM	1.891	-1.281	3.107
SRM-20	1.519	-0.777	3.096
SRM-29	2.067	0.915	3.117

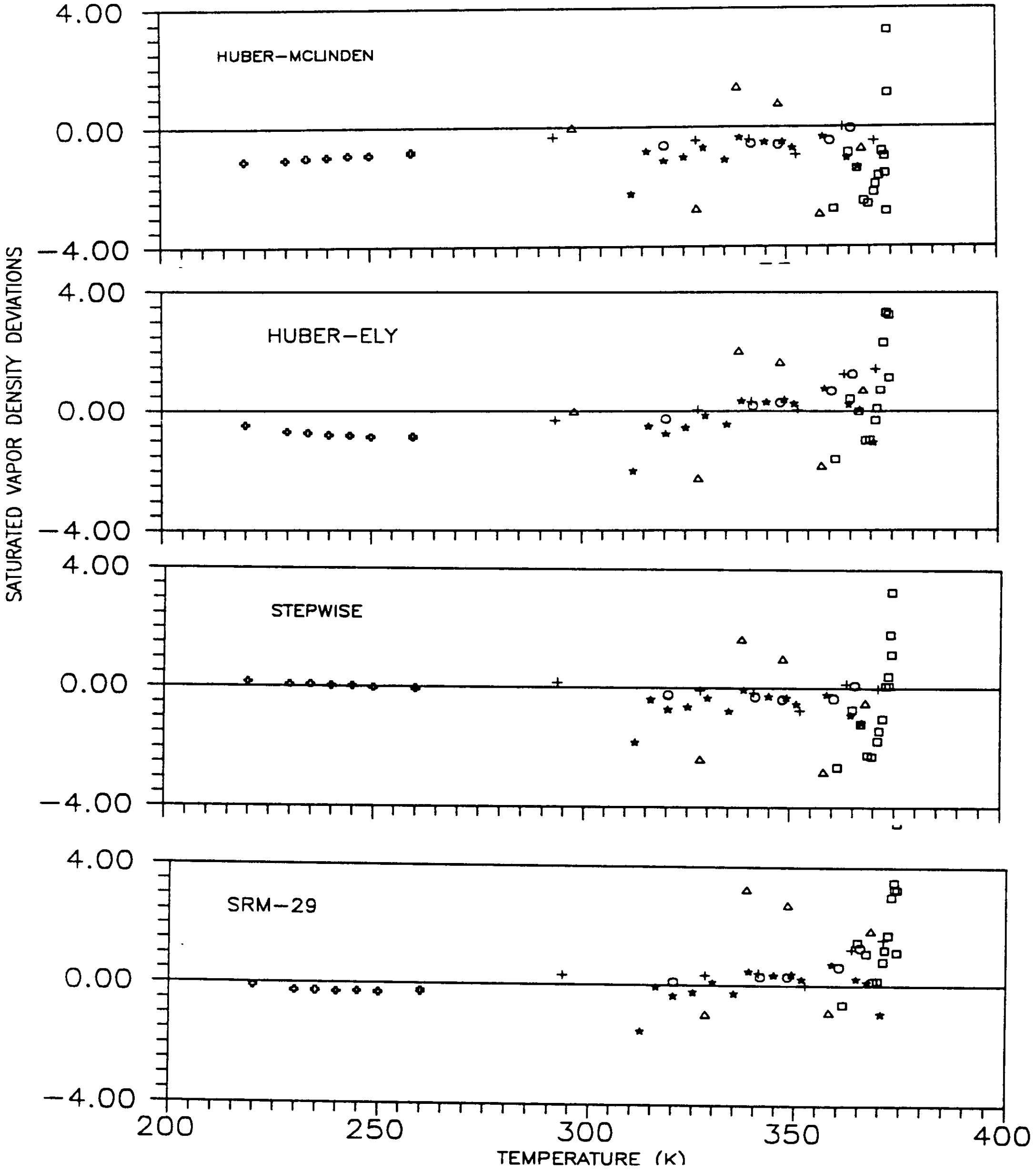


FIG 4.18a Comparison of experimental vapor densities for R134a.
(○) Weber; (□) Kabata et al; (Δ)Morrison and Ward; (+) Fukushima; (★) Niesen,(╬) Virial Intersection;

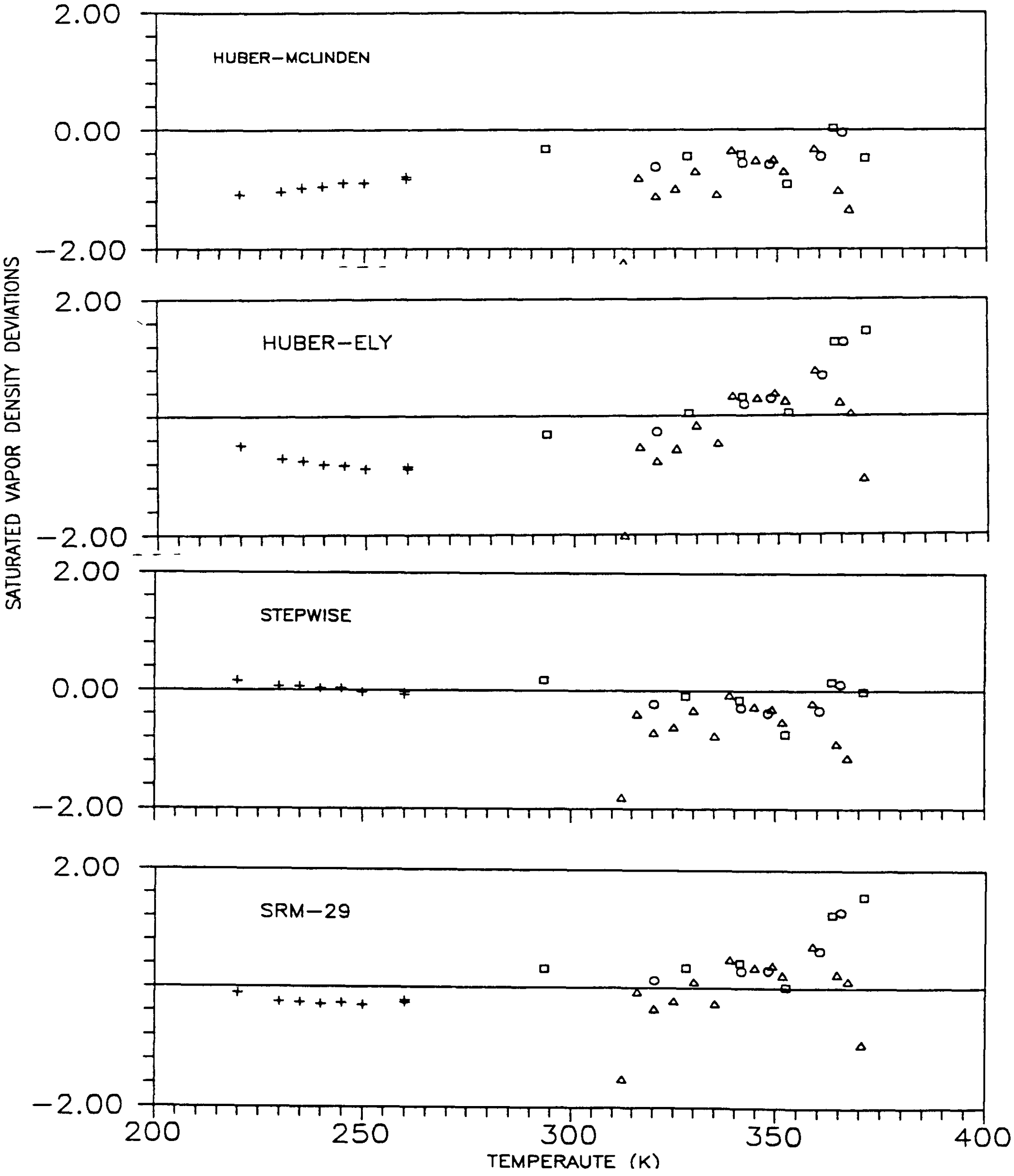


FIG 4.18b Comparison of experimental vapor densities used for fitting for R134a.
(○) Weber; (□) Fukushima ; (Δ) Niesen; (+) Virial Intersection.

Table 4.18 Statistical comparisons for saturated vapor density data used for fitting

EQUATIONS	AAD %	BIAS %	RMS %
HE	0.613	-0.133	0.741
HM	0.810	-0.810	0.536
SRM-20	0.411	-0.362	0.553
SRM-29	0.443	0.123	0.587

The overall comparisons for all equations are shown in Table 4.19.

Table 4.19 AAD AND RMS FOR ALL THE DIFFERENT DATA SETS CALCULATED ALL EQUATIONS

DATA	POINTS	HUBER-ELY		HUBER-MCLINDEN		SRM-20		SRM-29	
		AAD %	RMS %	AAD %	RMS %	AAD %	RMS %	AAD %	RMS %
PVT	1341	0.141	0.401	0.195	0.381	0.168	0.314	0.079	0.149
VIRIAL	42	0.346	0.418	0.473	1.025	0.392	0.418	0.719	1.396
C_v	150	0.282	0.363	0.367	0.398	0.356	0.439	0.305	0.369
C_p	31	0.594	0.386	0.602	0.341	0.863	0.291	0.438	0.336
SOUND	304	0.14	0.270	0.20	0.380	0.27	0.48	0.36	0.59
SAT P	374	0.983	0.372	0.461	0.474	1.087	0.346	0.184	0.346
SAT ρ_L	119	0.141	0.293	0.154	0.174	0.113	0.169	0.118	0.205
SAT ρ_V	33	0.613	0.741	0.810	0.536	0.411	0.553	0.443	0.587

AAD = ABSOLUTE AVERAGE DEVIATION, RMS = ROOT MEAN SQUARE ERROR.

5. CONCLUSIONS AND RECOMMENDATIONS

The equation of state for R134a recommended as a norm by the report published by Center of Thermodynamics Studies[94] is the Huber-McLinden equation. One of the primary reasons stated by this report for recommending the HM equation is the fact that accurate low pressure vapor phase properties are more important for most refrigeration systems. The HM equation predicts these properties very well. The SRM-20 equation for R134 proposed in work compares very favorably with the HM equation in the low pressure vapor phase properties. One of the advantages of the SRM-20 is the fact that the SRE has only 20 terms in it as compared to the HM equation which has 32 terms in it and in the Helmholtz free energy form, it would have 40 terms. In addition to this the SRM-20 equation is in the form of Helmholtz free energy and this form makes it more convenient to derive other properties. The HM equation is in the form of pressure, and thus deriving some of the properties involves complicated integrals.

The SRM-29 equation does well in the critical region as compared to the HM equation. One of the problems with the SRM-29 equation is in the sound velocity data. This problem could possibly rectified by modifying the weighing scheme.

One of the recommendations for this work would be to perform more extensive thermodynamic tests on the SRM-20 and SRM-29 terms to examine their validity. Also work should be done to evaluate the impacts of weighing schemes for the fit. Since the stepwise regression method works for pure fluids, its scope should be extended to mixtures. Also, work should be done to incorporate the simulated annealing method in the stepwise regression method.

REFERENCES

1. Robinson E.A., Least Squares Regression Analysis in terms of Linear Algebra, Goose Pond Press, Houston (1981).

2. Leland O.M., Practical Least Squares, McGraw Hill, New York, (1921).

3. Press W.H.,B.P.Flannery, S.A.Teukolsky, W.T.Vetterling, NUMERICAL RECIPES, pg. 501-515 (Cambridge University Press, 1987).

4. Wagner W.,"A New Correlation Method for Thermodynamic Data Applied to the Vapor Pressure curve of Argon, Nitrogen and Water", PC/T15 IUPAC Thermodynamic Tables Project Centre (1979).

5. McCarty R.D., and J.G.Hust, "Curve Fitting Techniques and Applications to Thermodynamics", Cryogenics Vol 7, Pg 200 (1967).

6. Hildebrand F.B., Introduction to Numerical Analysis, Chap. 7 McGraw-Hill, New York, (1956).

7. Hildebrand F.B., Methods of Applied Mathematics, Chap 3-19 Dover Publications Inc., New York, (1992).

8. de Reuck K.M., and B.Armstrong, " A Method of correlation using a search procedure based on a stepwise least squares technique and application to an equation of state for propylene", Cryogenics, Vol 25, pg 505 (1979).

9. Draper N.R., and H.Smith, Applied Regression Analysis (John Wiley, New York, 1966) pp 163-195.

10. Efroymson M.A., Mathematical Methods for Digital Computers, Chap 7, A.Ralston and H.S.Wilt, eds (John Wiley, New York, 1960).

11. Fraser D.A.S., Probability and Statistics : Theory and Applications, Duxbury Press (North Scituate, MA, 1976).

12. Wagner W., and U.Setzmann, "A New Method for Optimizing the Structure of Thermodynamic Correlation Equations", Int. J. Thermo., Vol 10, 6 (1989). International Journal of Thermophysics

13. Ewers J., and W. Wagner, Proc. 8th Symp. Thermophys. Prop,

pp 78-87, J.V. Sengers, ed (American Society of Mechanical Engineers, New York, 1982).

14. Saul A., and W. Wagner, "A Fundamental Equation for Water Covering the Range form Melting Line to 1273 K at Pressures up to 25000 MPa", J.Phys.Chem.Ref.Data, Vol 18, No.4, (1989).

15. Jacobsen, R.T., S.G.Penoncello and W.P.Clarke, " A Thermodynamic property formulation for air. I. Single-Phase Equation of State from 60 to 873 K at pressures upto 70 MPa", Intl.J.Thermo., vol 11, pg 169, 1990.

16. Robinson E.A., Least Squares Regression Analysis in terms of Linear Algebra, Goose Pond Press, Houston (1981).

17. McCarty R.D., Experimental Thermodynamics, Vol II - Experimental Thermodynamics of non-reacting fluids, Chap. 10, B. Le Neindre and B. Vodar Editors, Butterworth and Co., Ltd., London, (1975).

18. Huber M.L., and J.F.Ely, "An Equation of State Formulation for Thermodynamic Properties of R134a (1,1,1,2-tetrafluoroethane)", Int. J. Refrigeration (1992), 15, 393-400.

19. Huber M.L., and M.O.McLinden, "Thermodynamics Properties of R134a (1,1,1,2-Tetrafluoroethane)". Proceedings Intl. Refrigeration Conference (1992), Purdue University July 14-17.

20. Kirkpatrick S., C.D.Gelatt and M.P.Vecchi, Optimization by Simulated Annealing. Science 220, 671(1983).

21. Karlin S., A First Course in Stochastic Processes, Academic Press, New York (1973).

22. Metropolis N., A.Rosenbluth, M.Rosenbluth, A.Teller, and E.Teller,"Equation of State Calculations by Fast Computing Machines", J.Chem.Phys., 21, 1087(1953).

23. Schrodinger E., Statistical Thermodynamics (Cambridge Univ Press, London, 1946).

24. Hammersley J.M., and D.C.Handscomb, Monte Carlo Methods, Methuen, London (1964).

25. Dolan, W.B., P.T.Cummings, and M.D.LeVan, "Process

Optimization via Simulated Annealing : Application to Network Design". AIChE Journal, 35, 725(1989).

26. Dolan W., P.T.Cummings and M.D.LeVan, "Heat Exchanger Network Design by Simulated Annealing", Sci., 220, 671 (1983).

27. Bounds David.G., "New Optimization Methods from Physics and Biology", Nature, 329, 215 (1987).

28. Corey Ellen., and David Young., " Optimization of physical data tables by simulated annealing", Computers in Physics, 33-37 May/Jun 1989.

29. Meyers C.H., and M.S.Van Dusen., "The vapor pressure of liquid and solid carbon dioxide", J. Nat. Bur. Stand. 10, 381, 1933.

30. Schmidt, E and W. Thomas., " Precision determination of the critical points of carbon dioxide and ethane by measurement of light refraction", Forschung Geb. Ingenieurw, 20, 161, 1954.

31. Cook. D., " The carbon-dioxide-nitrous-oxide system in the critical region", Proc. Roy. Soc., A219:245, 1953.

32. Michels. A., T.Wassenaar, TH.Zwietering and P.Smits, "The vapour pressure of liquid carbon dioxide", Physica 16:501, 1950.

33. Jenkin C.F., and D.R.Pye, "Thermal Properties of Carbonic acid at low temperatures", Phil. Trans. Roy. Soc., A213:67,(1919).

34. Cook. D., "The vapor pressure equation and orthobaric density of nitrous oxide", Trans. Farad. Soc., 49:716, 1953.

35. Levelt-Sengers, J.H. and Chen, W.T.,"Vapor Pressure, Critical Isochore and some metastable states of CO_2" J. Chem. Phys. 56:595 (1972).

36. Michels, A., B. Blaise, and C. Michels.," The Isotherms of CO_2 in the Neighbourhood of the Critical Point and Round the Coexistence Line", Proc. Roy Soc, A160:358 (1937).

37. Vukalovich, M.P., N.I.Timoshenko and V.P.Kobelev, "Experimental investigation of CO2 density at temperatures

below 0° C." Thermal Engineering 17:82 (1970).

38. Kirillin, V. A., S.A.Ulybin, E.P.Zherdev, "Experimental determination of the density of carbon dioxide at temperatures from 0 to -50° C and pressures up to 500 bar" Thermal Engineering 16:133, (1969).

39. Amagat, E.H., "Sur la détermination de la densité des gaz liquefies et de leurs vapeurs saturées.-Éléments du point critique de l'acide carbonique",Comput. Rend. Acad. Sci. (Paris) 114:1093, (1982).

40. Holste, J.C., K.R.Hall, P.T.Eubank, K.N.Marsh, " Properties of CO_2-Rich Mixtures Literature Search and Pure CO_2 Phase I". Research Report RR-86, GPA, GRI, June 1985.

41. Fernandez-Fassnacht E., and F. Del Rio., "The vapor pressure of CO_2 from 194 to 243 K", J. Chem. Thermo. 16:469 (1984).

42. Ely, J.F., W.M.Haynes, and B.C.Bain., " Isochoric (p,V_m,T) measurements on CO_2 and on $(0.982CO_2+0.018N_2)$ from 250 to 330K at pressures to 35 MPa", J.Chem.Thermo, 1989, 21, 879-894.

43. Tickner, A.W., and F.P.Lossing, "The measurement of low vapor pressures by means of a mass spectrometer, J Phys. and Colloid Chem. 55, 733 (1951).

44. Carruth, G.F., and R.Kobayashi, " Vapor pressure of normal paraffins ethane through n-decane from their triple points to about 10 mm Hg, J. Chem. Eng. Data 18, 115 (1973).

45. Maass, O., and C.H. Wright, " Some physical properties of hydrocarbons containing two and three atoms, J. Am. Chem. Soc., 43, 1098 (1921).

46. Dana, L.I., A.C.Jenkins, J.N.Burdick and R.C.Timm, " Thermodynamic Properties of butane, isobutane, and propane", Refrigeration Engineering, 12, (12), 387 (June 1926).

47. Sage, B.H., J.G.Schaafsma, and W.N.Lacey, "Phase equilibria in hydrocarbon systems. V. Pressure-Volume-Temperature relations and thermal properties of propane", Ind. Eng. Chem. 26 (11), 1218 (Nov. 1934).

48. Beattie, J.A., N. Poffenberger, and C. Hadlock, " The critical constants of propane", J. Chem. Phys. 3, 96 (Feb 1935).

49. Kemp. J.D. and C.J.Egan, " Hindered rotation of the methyl groups in propane. The heat capacity, vapor pressure, heats of fusion and vaporization of propane. Entropy and density of the gas", J. Am Chem. Soc. 60 (7), 1521 (Jul 1938).

50. Deschner, W.W and G.G.Brown, "P-V-T relations for propane", Ind. Eng. Chem. 32 (6), 836 (Jun 1940).

51. Reamer, H.H., B.H.Sage and W.N.Lacey, " Phase equilibria in hydrocarbon systems : Volumetric behavior of propane", Ind. Eng. Chem. 41 (3), 482, (1949).

52. Cherney, B.J., H. Marchman and R.York Jr.,"Equipment for compressibility measurements", Ind. Eng. Chem. 41, 2653 (Nov 1949).

53. Clegg, H.P., and J.S.Rowlinson, "The physical properties of some fluorine compunds and their solutions. Part 2. The system sulfur hexafluoride + propane", Trans. Faraday Soc. 51, 1333 (1955).

54. Helgeson, N.L., and B.H.Sage, " Latent heat of vaporization of propane", J. Chem. Eng. Data. 12 (1), 47 (1967).

55. Gilliland, E.R., and H.W.Scheeline, "High-pressure vapor-liquid equilibrium", Ind. Eng. Chem. 32 (1), 48 (1940).

56. Burrell, G.A., and I.W.Robertson, "Vapor pressures of various compounds at low temperatures", U. S. Bur. Mines Tech. Paper No. 142 (1916).

57. Thomas, R.H.P., and R.H.Harrison, "Pressure, volume, temperature relations of propane", J. Chem. Eng. Data.

58. Mousa, A.H.N., "The physical properties of highly purified samples of propane and n-hexane, J.Chem.Thermodynam. 9, 1063 (1977).

59. Kratze, H., " Thermodynamic quantities for propane. 1. The vapor pressure of liquid propane", J.Chem.Thermodynam. 12, 305 (1980).

60. Teichmann, J., "Pressure-density-temperature measurements of liquid propane and benzene, Ph.D. dissertation, Ruhr University, Bochum (1978).

61. Goodwin, R.D., "Provisional thermodynamic functions of propane, from 85 K to 700 K at pressures to 700 bar", Nat.Bur. Stan. (U.S.) NBSIR 77-860 (Jul 1977).

62. Goodwin A.R.H., D.R.Defibaugh and L.A.Weber., "The vapor pressure of 1,1,1,2-tetrfluoroethane (R134a) and chlorodifluoromethane (R22)", Int J. Thermophys., 13:837-854, 1990.

63. Arita K., T.Tomizawa, Y. Nagakawa and Y. Yoshida., " Vapor-liquid equilibrium of the non-azeotropic refrigerant mixture formed by chloro-fluoromethane and 1,1,1,2-tetrafluoroethane ", Fluid Phase Equilibria (1991), 63:151-156.

64. Zhu. M., and J.Wu, "New experimental vapor pressure data and a new vapor pressure equation for HFC 134a", Fluid Phase Equilibria, 80:99-105, 1992.

65. Baehr. H.D., and R.Tillner-Roth, " Measurement and correlation of the vapor pressures of 1,1,1,2-tetrfluoroethane (R134a) and of 1,1-difluoroethane (R152a). J.Chem.Thermodynamics (1991) 23:1063-1068.

66. Weber, L.A., "Vapor Pressures and gas-phase PVT data for 1,1,1,2-tetrafluoroethane", Int. J. Thermophysics (1989) 10(3):617-627.

67. Wilson & Basu, Intl. J. Thermophys, 10(3),(591-603).

68. Piao, C., H.Sato and K. Watanabe., " An experimental study for PVT properties of CFC alternative refrigerant 1,1,1,2-tetrafluoroethane (R134a). ASHRAE Trans 96 ptl (1990) 132-140.

69. Kubota H., T. Yamashita, Y. Tanaka and T. Makita, "Vapor pressures of new fluorocarbons", Int. J. Thermophysics (1989) 10:629-637.

70. Nishiumi, H. and T.Yokoyama, " Vapor-Liquid equilibrium of the system of R134a-R22". The 11th Japan Symposium on Thermophysical Properties, 1990 paper B105, 95-98.

71. Magee. J.W., and J.B.Howley "Vapor pressure measurements on 1,1,1,2-tetrfluoroethane (R134a) form 180-350K. Int. J. Refrigeration., 15:362-364, 1992.

72. Morrison. G., and D.K.Ward, "Properties of two alternative refrigerants, 1,1-dichloro-2,2,2-trifluoroethane (R123) and 1,1,1,2-tetrafluoroethane (R134a)", Fluid Phase Equilibria (1991) 62:65-85.

73. Niesen, V.G., L.J.Van Poolen and S.L.Outcalt, "Coexisting densities and vapor pressures of refrigerants R22, R134a and R134 at 300 to 395 K", Submitted of Fluid Phase Equilibria, 1992.

74. Maezawa Y., H.Sato and K.Watanabe, "Saturated liquid densities fo HCFC 123 and HFC 134a. J.Chem.Eng.Data (1990) 35:225-228.

75. Baroncini C., G. Giuliani, M. Pacetti and F. Polonara., "Experimental study of thermodynamic properties of 1,1,1,2-tetrfluoroethane (R134a). I.I.F.-I.I.R. Commission B1, Herzlia, Israel (1990) 83-89.

76. Fukushima M., " Measurements of the PVT Properties of HCFC 123 and HFC 134a", Trans. of the JAR. (1990) 7:243-256.

77. Qian, Z., H.Sato and K.Watanabe, "Compressibility measurements of R134a by the Burnett techniquie", Fluid Phase Equilibria, 78:323-329, 1992.

78. Zhu, M.S., Y.D. Fu and L.Z.Han., "An Experimental Study of PVT Properties of CFC Alternative HFC134a", Fluid Phase Equilibria, 80:149-156, 1992.

79. Tillner-Roth, R., and H.D.Baehr., "H.D.Burnett measurements and correlation of gas phase (P,ρ,T) data of 1,1,1,2- tetrafluoroethane (R134a) and 1,1-difluoroethane (R152a)., J. Chem. Thermodynamics (1992) 24:413-424.

80. Tillner-Roth R., and H.D.Haehr, "Measurements of Liquid, Near-Critical and Supercritical (P,ρ,T)-Properties of 1,1,1,2-Tetrafluoroethane (R134a) and of 1,1-Difluoroethane (R152a)", J. Chem. Thermodynamics, 25:277-292, 1993.

81. Hou, H., B.E.Gammon, K.N.Marsh and J.C.Holste,

"Experimental densities for compressed R-134a.", Int.J.Refrig., 15:365-371, 1990.

82. Wilson D.P., and R.S.Basu, "Thermodynamic properties of a new stratosphereically safe working fluid- refrigerant 134a", ASHRAE Trans (1988) 94, part 2:2095-2118.

83. Magee, J.W., "Measurements for Molar Heat Capacity at Constant Volume (Cv) for 1,1,1,2-Tetrafluoroethane R134a", Int. J. Refrigeration, 15:372-380, 1992.

84. McLinden, M.O., J.S. Gallagher, L.A.Weber, G. Morrison, D. Ward, A.R.H.Goodwin, M.R.Moldover, J.W.Schmidt, H.B.Chae, T.J.Bruno, J.F.Ely, M.L.Huber, " Measurement and formulation of the thermodynamic properties of refrigerants 134a (1,1,1,2-tetrafluoroethane) and 123 (1,1-dichloro-2,2,2-trifluoroethane). ASHRAE Trans. (1989) 95 pt 2:263-283.

85. Fukushima M., N. Watanabe and T. Kamimura., "Measurements of the vapor-liquid coexistence curves and the critical parameters of HCFC123 and HFC134a", Trans of the JAR (1990) 7:85:95.

86. Kabata Y., S. Tanikawa, M. Uematsu, and K. Watanabe, " Measurements of the vapor-liquid coexistence curve and the ciritical parameters for 1,1,1,2-tetrafluoroethane. Int. J. Thermophysics (1989) 10:605:613.

87. Fukushima M., "Saturated liquid densities of HCFC 123, HFC 134a, CFC11 and CFC12", Trans. of the JAR, (1991) 8:65;75.

88. Yokoyama C., and S. Takahashi, "Saturated liquid densities of 2,2-dichloro-1,1,1-trifluoroethane (HCFC-123), 1,2-dichloro-1,2,2-trifluoroethane (HCFC-123a), 1,1,1,2-tetrafluoroethane (HFC-134a) and 1,1,1-trifluoroethane (HFC-143a)", Fluid Phase Equilibria (1991) 63:151-156.

89. Saitoh, A., S. Nakagaw, H.Sato and K.Watanabe, "Isobaric heat capacity data for liquid HFC-134a. J.Chem.Eng.Data (1990) 35:107-110.

90. Nakagawa, S., H.Sato and K.Watanabe, "Specific heat at constant pressure for liquid new refrigerants", The 27th National Heat Transfer Symposium of Japan, Nagoya (1990) 421-423.

91. Guedes, H.J.R, T.Takagi and J.Zollweg, "Speed of Sound in

Liquid R134a", Int.J.Regrig, 15:381-385, 1992.

92. Goodwin, A.R.H., and M.R.Moldover, "Thermophysical properties of gaseous regrigerants from speed of sound measurements", J.Chem.Phys. (1990), 93:2741-2753.

93. Takagi, T., Data reported in Thermophysical properties of Environmentally Acceptable Fluorocarbons HFC-134a and HCFC-123. (1991). Japanese Association Regrigeration and Japan Flon gas Association. p. 56.

94 Penoncello S.G., R.T.Jacobsen, R.C.Williams, E.W.Lemmon., "Equation of State Comparisons for HFC-134a and HCFC-123", International Energy Agency-Annex 18, 1993.

APPENDIX I

This example is about developing a vapor pressure equation for CO_2. There were 15 terms in the bank of terms in this case. Some of terms included were $\tau^{1.5}$, τ^5, τ^6 etc. The fitting form used for the equation was.,

$$\ln\left(\frac{P}{P_c}\right) = \sum_{m=1}^{M} a_m f_m(\tau) \qquad (1.24)$$

For this example.,

COEF = Value of the coefficient

SDCO = Standard Deviation of the Coefficient

STUDT = Student-t Value

PROBAL = Probability of Student-t Distribution

PROBF = Probability of Fisher-F Distribution

FISH = Value of Fisher-F Test

Table 1.4 Table with the functional forms (exponents of τ)

1	1
2	1.66
3	6.5
4	2.5
5	1.5
6	7.5
7	3.0
8	4.0
9	5.5
10	7.0
11	8.5
12	6.0
13	3.5
14	8.0
15	-1.0

Terms 1 and 3 were randomly chosen terms which were added initially to the equation.,

```
SUM OF SQUARES =       0.19080269D+04
THE TERM  1 HAS BEEN ADDED TO THE EQUATION
SUM OF SQUARES=,       0.12938696D+02
THE TERM    3HAS BEEN ADDED TO THE EQUATION
SUM OF SQUARES=,       0.18679742D+01
KRF =    0
NGO =    0
THE TERM WITH MINIMUM Q-VALUE IS    4
    QMIN= 0.11361918D-01
THE TERM  4 HAS BEEN ADDED TO THE EQUATION
SUM OF SQUARES=,       0.11361918D-01
STD =  0.654791E-02
```

	SDCO	COEF	STUDT	PROB
1	0.44572E-02	-0.68420E+01	0.15350E+04	0.10000E+01
3	0.80621E+01	-0.83852E+02	0.10401E+02	0.10000E+01
4	0.70925E-01	-0.14759E+02	0.20809E+03	0.10000E+01

```
FISH=,0.140292E+01  PROBAL =0.997015E+00
KRF =    0
NGO =    0
```

```
THE TERM WITH MINIMUM Q-VALUE IS    5

    QMIN= 0.10187966D-01

THE TERM    5HAS BEEN ADDED TO THE EQUATION

SUM OF SQUARES=,      0.10187966D-01

STD =  0.621215E-02
```

	SDCO	COEF	STUDT	PROB
1	0.25325E-01	-0.67043E+01	0.26473E+03	0.10000E+01
3	0.13020E+02	-0.14197E+03	0.10904E+02	0.10000E+01
4	0.30115E+00	-0.13140E+02	0.43632E+02	0.10000E+01
5	0.11132E+00	-0.61397E+00	0.55155E+01	0.10000E+01

```
FISH=,0.111102E+01  PROBAL =0.803780E+00

KRF =     0

NGO =     0

THE TERM WITH MINIMUM Q-VALUE IS   15

    QMIN= 0.96592751D-02

THE TERM   15HAS BEEN ADDED TO THE EQUATION

SUM OF SQUARES=,      0.96592751D-02

STD =  0.606030E-02
```

	SDCO	COEF	STUDT	PROB
1	0.24711E-01	-0.67062E+01	0.27139E+03	0.10000E+01
3	0.12703E+02	-0.14134E+03	0.11127E+02	0.10000E+01
4	0.29384E+00	-0.13159E+02	0.44784E+02	0.10000E+01
5	0.10862E+00	-0.60609E+00	0.55801E+01	0.10000E+01
15	0.44921E-07	0.17043E-06	0.37941E+01	0.99982E+00

LAST TERM ADDED FAILED T-TEST,TRY EXCHANGE

THE TERM 3 HAS BEEN ELIMINATED FROM THE EQN ,/,SUM OF

SQUARES = 0.14206252D-01

THE TERM 9HAS BEEN ADDED TO THE EQUATION

SUM OF SQUARES=, 0.96398609D-02

KRF = 2

STD = 0.605421E-02

	SDCO	COEF	STUDT	PROB
1	0.26085E-01	-0.66848E+01	0.25627E+03	0.10000E+01
4	0.34347E+00	-0.12549E+02	0.36537E+02	0.10000E+01
5	0.11780E+00	-0.73383E+00	0.62293E+01	0.10000E+01
9	0.47666E+01	-0.53203E+02	0.11162E+02	0.10000E+01
15	0.44877E-07	0.16984E-06	0.37847E+01	0.99981E+00

THE TERM 15HAS BEEN ELIMINATED FROM THE EQN ,/,SUM OF

SQUARES = 0.10164880D-01

KRF = 1

STD = 0.620511E-02

	SDCO	COEF	STUDT	PROB
1	0.26729E-01	-0.66828E+01	0.25002E+03	0.10000E+01
4	0.35197E+00	-0.12526E+02	0.35588E+02	0.10000E+01
5	0.12072E+00	-0.74266E+00	0.61522E+01	0.10000E+01
9	0.48849E+01	-0.53457E+02	0.10943E+02	0.10000E+01

KRF = 0

NGO = 0

THE TERM WITH MINIMUM Q-VALUE IS 15

QMIN= 0.96398609D-02

THE TERM 15HAS BEEN ADDED TO THE EQUATION SUM OF

SQUARES=, 0.96398609D-02, STD = 0.605421E-02

	SDCO	COEF	STUDT	PROB
1	0.26085E-01	-0.66848E+01	0.25627E+03	0.10000E+01
4	0.34347E+00	-0.12549E+02	0.36537E+02	0.10000E+01
5	0.11780E+00	-0.73383E+00	0.62293E+01	0.10000E+01
9	0.47666E+01	-0.53203E+02	0.11162E+02	0.10000E+01
15	0.44877E-07	0.16984E-06	0.37847E+01	0.99981E+00

```
LAST TERM ADDED FAILED T-TEST,TRY EXCHANGE

NO FURTHER REDUCTION INTER   CORRELATION IS POSSIBLE

KRF =    3

FISH=,0.105047E+01  PROBAL =0.655167E+00

THE TERM   15HAS BEEN ELIMINATED FROM THE EQN  ,/,SUM OF

SQUARES =      0.10164880D-01

NGO =    1

THE TERM    9HAS BEEN ELIMINATED FROM THE EQN  ,/,SUM OF

SQUARES =      0.14775974D-01

THE TERM    2HAS BEEN ADDED TO THE EQUATION

SUM OF SQUARES=,      0.10164851D-01

KRF =    2

STD =  0.620510E-02
```

```
          SDCO          COEF          STUDT         PROB

  1   0.76290E-01 -0.60926E+01 0.79861E+02 0.10000E+01

  2   0.17172E+01  0.18792E+02 0.10943E+02 0.10000E+01

  4   0.64678E+00 -0.23070E+02 0.35669E+02 0.10000E+01

  5   0.13659E+01 -0.14562E+02 0.10661E+02 0.10000E+01
```

```
STD =  0.620510E-02


FINAL COEFFICIENTS SELECTED ARE
```

```
  -0.609258555255E+01         1
   0.187921755958E+02         2
  -0.230700890763E+02         4
  -0.145623928886E+02         5
 THERE ARE  4 COEFFICIENTS IN THE EQUATION.

  STD = 0.62051E-02
```

In the above shown example, there are no constraints added to the equation, but they can be easily added using the method of Lagrange Multipliers.

APPENDIX II.

```
C***********************************************************
C
C     A SIMULATED ANNEALING PROGRAM TO FIND A VAPOR PRESSURE
C     EQUATION FOR  CARBON DIOXIDE
C
C
C***********************************************************
     IMPLICIT DOUBLE PRECISION (A-G,P-Z)
     LOGICAL MOJO
     COMMON/INPUT/P(1000),T(1000),WT(1000)
     DIMENSION PCAL(1000),VAR(20),R(20),F(20),PCT(1000)
     DIMENSION Q(20),M(2,20),E(20),G(20),NOSE(20),R1(20)
     IDUM=-1
     JDUM=1
C
C    READING THE DATA FILE
C
     CALL DREAD(N)
     OPEN(2,FILE='RESULT1.DAT',STATUS='OLD')
C
C    CALL A ROUTINE TO GENERATE RANDOM NUMBERS
C
     J=1
     JUMP=1
     NUM=0
     COOL=.5
     COOLF=.9
     PRINT*,'NO OF RUNS ?'
     READ*,NOR
C
C   SET A RANDOM CRITERION FOR REJECTION OR ACCEPTANCE OF A
C   TERM
C
     PRINT*,'RAN ?'
     READ*,RAN
     PRINT*,'ORDER ?'
     READ*,NOT
     CALL NEW(NOT,NOSE)
     WRITE(2,95)
60        IF (NUM.LT.NOR) THEN
     CALL JCS(M,N,NVAR,J,IDUM,RAN,R,NOSE)
C
C    FIND THE COEFFICIENTS OF EACH OF THE TERMS SELECTED
C
     CALL COEFF(G,SSY,NDF)
```

```
     COUNT=1
C
C   INITIALIZE AN ARRAY TO STORE THE COEFFS
C
     DO 44 II=1,16
       Q(II)=0
44   CONTINUE
       DO 45 II=1,16
        IF (M(J,II)) THEN
                Q(II)=G(COUNT)
           COUNT=COUNT+1
        ENDIF
        IF (J.EQ.1) E(II)=Q(II)
45     CONTINUE
C
C    CALL THE FUNCTION THE CALCULATE THE OBJECTIVE FUNCTION
C
     CALL OBJ(N,PCAL,M,Q,J,COST,NOSE)
     IF (JUMP) C1=COST
     IF (JUMP.EQ.0) C2=COST
     J=2
C
C    CHECK IF IT IS NOT THE FIRST PASS AND THEN DO THE
C    COST COMPARISION
C
     IF (JUMP.EQ.0) THEN
      PRINT*,C1,C2
        DELC =C2-C1
        CALL METROP(DELC,COOL,JDUM,MOJO)
        IF (MOJO) THEN
         C1=C2
         COUNT=1
         DO 47 D=1,16
            M(1,D)=M(2,D)
               IF (M(1,D)) THEN
              E(D)=Q(D)
            ENDIF
            IF (M(1,D).EQ.0) E(D)=0
47       CONTINUE
        ENDIF
     ENDIF
C
     JUMP=0
     NUM=NUM+1
     COOL=COOL*COOLF
     GO TO 60
     ENDIF
     J=1
```

```
C
C    CALL THE OBJECTIVE FUNCITON TO OBTAIN THE
C    FINAL RESULTS
C
     CALL OBJ(N,PCAL,M,E,J,COST,NOSE)
     DO 50 I=1,16
       WRITE(2,100)NOSE(I),M(1,I),E(I)
50   CONTINUE
     WRITE(2,*)
     WRITE(2,*)
     WRITE(2,97)
     DO 70 I=1,N
       P(I)=P(I)*73.76462
       T(I)=(1-T(I))*304.25
       PCAL(I)=EXP(PCAL(I))*73.76462
       PCT(I)=100*(P(I)-PCAL(I))/P(I)
       WRITE(2,105)T(I),P(I),PCAL(I),PCT(I),WT(I)
70   CONTINUE
95   FORMAT(3X,'NO',7X,'MASK',7X,'COEFFICIENTS')
97   FORMAT(7X,'TEMP
F',11X,'P',11X,'PCAL',11X,'PCT',9X,'WT')
100     FORMAT(5X,I2,5X,I2,5X,1PE19.10)
105       FORMAT(F9.4,6X,F10.4,6X,F10.4,6X,F11.4,3X,F10.4)
     WRITE(2,*)
     WRITE(2,*)
     WRITE(2,*)'RAN=',RAN
     WRITE(2,*)'NO OF RUNS=',NOR
     WRITE(2,*)'COST=',COST
     WRITE(2,*)'FAC1=',FAC1,'FAC2=',FAC2,'FAC3=',FAC3
     PRINT*,COST
     END

C************************************************************
C
C    THIS FUNCTION RETURNS A RANDOM NUMBER
C
C************************************************************
     FUNCTION RAN3(IDUM)
     IMPLICIT DOUBLE PRECISION (A-H,M-Z)
     DIMENSION MA(55)
     DATA IFF /0/
     MBIG=1E09
     MSEED=161803398
     MZ=0
     FAC=1/MBIG
     IF(IDUM.LT.0.OR.IFF.EQ.0)THEN
        IFF=1
        MJ=MSEED-IABS(IDUM)
```

```
        MJ=MOD(MJ,MBIG)
        MA(55)=MJ
        MK=1
        DO 10 I=1,54
          II=MOD(21*I,55)
          MA(II)=MK
          MK=MJ-MK
          IF(MK.LT.MZ)MK=MK+MBIG
          MJ=MA(II)
10      CONTINUE
         DO 40 K=1,4
           DO 30 I=1,55
          MA(I)=MA(I)-MA(1+MOD(I+30,55))
          IF(MA(I).LT.MZ)MA(I)=MA(I)+MBIG
30         CONTINUE
40       CONTINUE
     INEXT=0
     INEXTP=31
     IDUM=1
     ENDIF
     INEXT=INEXT+1
     IF(INEXT.EQ.56)INEXT=1
     INEXTP=INEXTP+1
     IF(INEXTP.EQ.56)INEXTP=1
     MJ=MA(INEXT)-MA(INEXTP)
     IF(MJ.LT.MZ)MJ=MJ+MBIG
     MA(INEXT)=MJ
     RAN3=MJ*FAC
     RETURN
     END

C************************************************************
C
C   THIS SUBROUTINE IS USED TO GENERATE A MATRIX USING THE
C    LINEAR LEAST SQUARES METHOD FROM THE GIVEN SET OF DATA
C    POINTS.
C************************************************************

      SUBROUTINE FITTER(F,Y,NFUN)
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      DIMENSION E(40), F(40), A(40,41)
      DOUBLE PRECISION A, SY, SYY, RES, DET
      EQUIVALENCE (CCC,RES),(NC,FNC)
      DATA NTR/-1/
      ENTRY FIT(F,Y,NFUN)
      IF(NTR) 010,030,030
  010 NP=0
      NF=NFUN
```

```
      IF(NF.GT.40)  GO TO 250
      NC=0
      SY=0.
      SYY=0.
      NY=NF+1
      DO 020 I=1,NY
      DO 020 J=1,NF
  020 A(J,I)=0.
      IF(NTR.EQ.0) GO TO 060
      NTR=0
  030 SY=Y+SY
      SYY=SYY+Y*Y
      DO 040 J=1,NF
      A(J,NY)=A(J,NY)+Y*F(J)
      DO 040 I=1,NF
  040 A(I,J)=A(I,J)+F(I)*F(J)
      NP=NP+1
      RETURN
C
C     SUBROUTINE CONSTR      F,Y,NFUN'
      ENTRY CONSTR(F,Y,NFUN)
      IF(NTR) 050,060,060
  050 NTR=0
      GO TO 010
  060 N=NY-1
      IF(NY.GT.40) GO TO 250
      DO 070 I=1,N
      A(I,NY+1)=A(I,NY)
      A(NY,I)=F(I)
  070 A(I,NY)=F(I)
      NC=NC+1
      DO 080 I=NF,N
      A(NY,I+1)=0.00
  080 A(I+1,NY)=0.00
      NY=NY+1
      A(NY-1,NY)=Y
      RETURN
C
C     SUBROUTINE COEFF      F,Y,NFUN'
      ENTRY COEFF(F,Y,NFUN)
      N=NY-1
      DO 090 I=1,NF
  090 F(I)=A(I,NY)
      IF(N.EQ.1) GO TO 120
      DO 110 I=2,N
      DO 100 J=I,NY
  100 A(I-1,J)=A(I-1,J)/A(I-1,I-1)
      DO 110 J=I,N
```

```
      DO 110 K=I,NY
  110 A(J,K)=A(J,K)-A(J,I-1)*A(I-1,K)
  120 A(N,NY)=A(N,NY)/A(N,N)
      IF(N.EQ.1) GO TO 140
      DO 130 I=2,N
      L=N-I+2
      DO 130 J=L,N
  130 A(L-1,NY)=A(L-1,NY)-A(L-1,J)*A(J,NY)
  140 NTR=-1
      RES=SYY
      DO 150 I=1,NF
      RES=RES-A(I,NY)*F(I)
  150 F(I)=A(I,NY)
      NFUN=NP
      NDF=NP-NF+NC
      DF=NDF
      Y=FNC
      RETURN
C
C     SUBROUTINE STAT      F,Y,NFUN'
      ENTRY STAT(F,Y,NFUN)
      TOT=SYY-SY*SY/NP
      SYY=RES/DF
      REG=TOT-RES
      IF(TOT.NE.0.0D0) CORR=REG/TOT
      ST=1.96+(2.72+8.04/(DF*DF))/DF
      DET=1.0
      DO 160 I=1,NF
     DET=DET*A(I,I)
      IF(A(I,I).LE.0.0)GO TO 160
      A(I,I)=1./A(I,I)
  160 CONTINUE
      IF(NF.EQ.1) GO TO 190
      DO 180 I=2,NF
      DO 180 J=2,I
      SY=0.
      DO 170 K=J,I
  170 SY=SY-A(I,K-1)*A(K-1,J-1)
  180 A(I,J-1)=SY*A(I,I)
  190 WRITE(6,300)
      DO 240 I=1,NF
      IF(I.EQ.NF) GO TO 205
      L=NF-I
      DO 200 J=1,L
      K=NF-J
      DO 200 M=1,J
      N=NF-M+1
  200 A(K,I)=A(K,I)-A(K,N)*A(N,I)
```

p

```
  205 IF(I.EQ.1) GO TO 220
      DO 210 J=2,I
  210 A(J-1,I)=A(I,J-1)*SYY
  220 DO 230 J=1,I
  230 A(I,J)=A(I,J)*SYY
      E(I)=ST*DSQRT(DABS(A(I,I)))
      F(I)=A(I,NY)
  240 WRITE(6,310)F(I),E(I)
      BB= DSQRT(DABS(SYY))
      WRITE(6,320) RES, REG, TOT, BB, DET, CORR, NP
      NFUN=NDF
      RETURN
  250 WRITE(6,330)
      STOP
  300 FORMAT(' THE COEFFICIENTS AND THEIR ESTIMATED ERRORS
ARE '//)
  310 FORMAT(1PE19.10,' +OR-',1PE9.2)
  320 FORMAT(    /' ESTIMATED RESIDUAL SUM OF SQUARES
=',E17.9/
     1             ' ESTIMATED REGRESSION SUM OF SQUARES
=',E17.9/
     2                  ' ESTIMATED TOTAL SUM OF SQUARES
=',E17.9/
     1' VARIANCE OF FIT =',E17.9/' DETERMINANT OF THE MATRIX
=',E17.9/
     4' CORRELATION COEFFICIENT =',E17.9/' NUMBER OF POINTS
=',I5//)
  330 FORMAT(//'THE ARRAYS IN THE FITTING PROGRAM ARE TOO
SMALL TO HOLD T
     1HE NUMBER OF CONSTRAINTS AND FUNCTIONS ASKED FOR IN
THE CALLING PR
     2OGRAM')
      END
C
     SUBROUTINE OBJ(N,PCAL,M,G,J,COST,O)

C************************************************************
C
C    THIS SUBROUTINE CALCULATES THE OBJECTIVE FUNCTION
C
C************************************************************
     IMPLICIT DOUBLE PRECISION (A-G,P-Z)
     INTEGER O
     COMMON/INPUT/P(1000),T(1000),WT(1000)
     DIMENSION PCAL(N),M(2,20),G(20),O(20)
     SCOST=0
     DO 10 I=1,N
     T1=M(J,O(1))*G(O(1))*T(I)
```

```
     T2=M(J,O(2))*G(O(2))*T(I)**1.9
     T3=M(J,O(3))*G(O(3))*T(I)**6.5
     T4=M(J,O(4))*G(O(4))*T(I)**2.5
     T5=M(J,O(5))*G(O(5))*T(I)**1.5
     T6=M(J,O(6))*G(O(6))*T(I)**7.5
     T7=M(J,O(7))*G(O(7))*T(I)**3
     T8=M(J,O(8))*G(O(8))*T(I)**4
     T9=M(J,O(9))*G(O(9))*T(I)**5.5
     T10=M(J,O(10))*G(O(10))*T(I)**7
     T11=M(J,11)*G(11)*T(I)**8.5
     T12=M(J,12)*G(12)*T(I)**6
     T13=M(J,13)*G(13)*T(I)**3.5
     T14=M(J,14)*G(14)*T(I)**8
     T15=M(J,15)*G(15)*(1-T(I))**2
     T16=M(J,16)*G(16)/T(I)
     PCAL(I)=(T1+T2+T3+T4+T5+T6+T7+T8+T9+T10
     +  +T11+T12+T13+T14+T15+T16)/(1-T(I))
     SCOST=SCOST+WT(I)*(PCAL(I)-LOG(P(I)))**2
10   CONTINUE
     COST=SCOST**.5
     RETURN
     END
C
C***********************************************************
C
C      THIS SUBROUTINE SELECTS THE TERMS RANDOMLY TO DEVELOP
C      THE VAPOR PRESSURE EQUATION
C
C***********************************************************
C
     SUBROUTINE JCS(M,N,NVAR,J,IDUM,RAN,R,O)
     IMPLICIT DOUBLE PRECISION (A-H,P-Z)
     INTEGER O
     COMMON/INPUT/P(1000),T(1000),WT(1000)
     DIMENSION M(2,20),F(20),R(20),VAR(20),O(20)
C
C   GENERATING RANDOM NUMBERS FOR EACH OF THE TERMS
C
     DO 20 I=1,16
     R(I)=RAN3(IDUM)
     M(J,I)=0
20   CONTINUE
     COUNT=1
     DO 30 I=1,16
C
C   SPECIFYING THE CONDITION FOR ACCEPTING A TERM
C
     IF (R(I).GT.RAN) THEN
```

```
         VAR(COUNT)=I
         M(J,I)=1
         COUNT=COUNT+1
      ENDIF
30       CONTINUE
      NVAR=COUNT-1
C
C    CALCULATING THE VALUE OF EACH OF THE TERMS AT SPECIFIED
C     TEMPERATURES
C
         DO 40 I=1,N
        K=0
        IF (M(J,1).AND.K.LT.6) THEN
           K=K+1
           F(K)=T(I)
        ENDIF
        IF (M(J,2).AND.K.LT.6) THEN
           K=K+1
           F(K)=T(I)**1.9
        ENDIF
        IF (M(J,3).AND.K.LT.6) THEN
           K=K+1
           F(K)=T(I)**6.5
        ENDIF
        IF (M(J,4).AND.K.LT.6) THEN
           K=K+1
           F(K)=T(I)**2.5
        ENDIF
        IF (M(J,5).AND.K.LT.6) THEN
           K=K+1
           F(K)=T(I)**1.5
        ENDIF
        IF (M(J,6).AND.K.LT.6) THEN
           K=K+1
           F(K)=T(I)**7.5
        ENDIF
        IF (M(J,7).AND.K.LT.6) THEN
           K=K+1
           F(K)=T(I)**3
        ENDIF
        IF (M(J,8).AND.K.LT.6) THEN
           K=K+1
           F(K)=T(I)**4
        ENDIF
        IF (M(J,9).AND.K.LT.6) THEN
           K=K+1
           F(K)=T(I)**5.5
        ENDIF
```

```
          IF (M(J,10).AND.K.LT.6) THEN
             K=K+1
             F(K)=T(I)**7
          ENDIF
          IF (M(J,11).AND.K.LT.6) THEN
            K=K+1
            F(K)=T(I)**8.5
          ENDIF
          IF (M(J,12).AND.K.LT.6) THEN
            K=K+1
            F(K)=T(I)**6
          ENDIF
          IF (M(J,13).AND.K.LT.6) THEN
            K=K+1
            F(K)=T(I)**3.5
          ENDIF
          IF (M(J,14).AND.K.LT.6) THEN
            K=K+1
            F(K)=T(I)**8
          ENDIF
          IF (M(J,15).AND.K.LT.6) THEN
            K=K+1
            F(K)=(1-T(I))**2
          ENDIF
          IF (M(J,16).AND.K.LT.6) THEN
            K=K+1
            F(K)=1/T(I)
          ENDIF
      DO 50 II=1,K
      F(II)=F(II)*WT(I)/(1-T(I))
50    CONTINUE
      PL=WT(I)*LOG(P(I))
      CALL FIT(F,PL,K)
40    CONTINUE
      RETURN
      END
C
      SUBROUTINE METROP(DE,T,JDUM,ANS)
C************************************************************
C
C    THIS IS THE METROPOLIS ALGORITHM FOR THE SIMULATED
C    ANNEALING METHOD.
C
C************************************************************
      DOUBLE PRECISION DE,T,VAR
      LOGICAL ANS
      VAR=DE/T
      IF ((DE/T).LT.-300) VAR=-300
```

```
      ANS=(DE.LT.0).OR.(RAN(JDUM).LT.EXP(-VAR))
      RETURN
      END
C
      SUBROUTINE NEW(N,O)
      INTEGER N,O
      DIMENSION O(20)
      DO 10 I=1,20
      IF (N.EQ.21) N=1
      O(I)=N
      N=N+1
10    CONTINUE
      RETURN
      END
C
C***********************************************************
C
C      THIS SUBROUTINE READS IN THE DATA FILE
C
C***********************************************************
C
      SUBROUTINE DREAD(N)
      IMPLICIT DOUBLE PRECISION (A-H,P-Z)
      CHARACTER*4 NTYPE(9), ITYPE, REFER*80, RECORD*72
      COMMON/INPUT/P(1000),T(1000),WT(1000)
      DIMENSION REFER(100),IDREF(100)
         NREF = 0
      I=0
      PRINT*,'FACO ','  FAC1'
      READ*,FACO,FAC1
      OPEN(1,FILE='CO2PS.DAT',STATUS='OLD')
20    READ (1,'(A4,I4,A)') ITYPE, IDENT, RECORD
         IF (ITYPE.EQ.'EOF') GO TO 1000
         IF (ITYPE.EQ.'REM') GO TO 020
         IF (ITYPE.NE.'REF ') THEN
       I=I+1
          READ (RECORD,*) T(I), P(I)
      IF (IDENT.GT.0) THEN
           SIGP=3E-4*P(I)
        SIGT=10E-3*T(I)
        WT(I)=1/SQRT(SIGP**2+SIGT**2)
        WT(I)=FACO/WT(I)
      ENDIF
      IF (T(I).LT.300) WT(I)=FAC1*WT(I)
       T(I)=1-T(I)/304.25
       P(I)=P(I)/73.76462
          GO TO 020
         ELSE
```

```
         NREF = NREF + 1
         REFER(NREF) = RECORD
         IDREF(NREF) = IDENT
         GO TO 020
        ENDIF
1000 N=I
     RETURN
     END
```

APPENDIX III.

THESE ARE THE TERMS INCLUDED IN THE BANK OF TERMS FOR DEVELOPMENT OF THE EQUATION OF STATE FOR R134a.

$$\frac{A^r}{RT} = \sum f(\delta,\tau)$$

$$f(\delta,\tau) = \delta^i \tau^j Exp(-\delta^k) \qquad k \neq 0$$
$$f(\delta,\tau) = \delta^i \tau^j \qquad k = 0$$

No.	k	i	j
1	0	1	0
2	0	1	1
3	0	1	1.5
4	0	1	2
5	0	1	2.5
6	0	1	3
7	0	1	50
8	0	2	-1
9	0	2	-0.5
10	0	2	0
11	0	2	1
12	0	2	1.5
13	0	2	2
14	0	2	3
15	0	2	40
16	0	3	-1
17	0	3	-0.5
18	0	3	0
19	0	3	1
20	0	3	2
21	0	3	2.5
22	0	3	5
23	0	3	32
24	0	4	-1
25	0	4	0

26	0	4	1
27	0	4	2
28	0	4	3
29	0	4	26
30	0	5	-1
31	0	5	0
32	0	5	1
33	0	5	9
34	0	6	0
35	0	6	1
36	0	6	2
37	0	6	3
38	0	7	1
39	0	7	2
40	0	7	5
41	0	8	0
42	0	8	1
43	0	8	2
44	0	8	5
45	0	9	1
46	0	9	2
47	0	10	2
48	0	11	0
49	0	11	12
50	0	13	7
51	0	13	13
52	1	1	0
53	1	1	1
54	1	1	3
55	1	2	1
56	1	2	5
57	1	3	5
58	1	4	2
59	1	4	3
60	1	4	5
61	1	5	6
62	1	6	4
63	1	7	1
64	1	8	8
65	1	9	0
66	1	11	1
67	2	1	0
68	2	1	5
69	2	1	6
70	2	1	7
71	2	1	9
72	2	1	10
73	2	1	11

74	2	2	0
75	2	2	1
76	2	2	2
77	2	2	3.5
78	2	2	5
79	2	2	5.5
80	2	2	8
81	2	3	2
82	2	3	3
83	2	3	4
84	2	3	6
85	2	3	7
86	2	3	13
87	2	4	0
88	2	4	2
89	2	4	3
90	2	4	4
91	2	4	5
92	2	4	6
93	2	5	2
94	2	5	3
95	2	5	4
96	2	5	6
97	2	6	2
98	2	6	4
99	2	6	8.5
100	2	6	12
101	2	7	0
102	2	7	2
103	2	7	3
104	2	7	4
105	2	7	10
106	2	7	11
107	2	8	0
108	2	8	1
109	2	8	2
110	2	8	3
111	2	8	4
112	2	8	6.5
113	2	9	0
114	2	9	2
115	2	9	4
116	2	10	2
117	2	10	4
118	2	10	5.5
119	2	10	8
120	2	11	0
121	2	11	1

122	2	11	2
123	2	11	3
124	2	11	4
125	2	11	5
126	2	11	6
127	2	11	7
128	2	12	2
129	2	12	3
130	2	13	2
131	2	13	4
132	2	14	2
133	2	14	3
134	2	14	4
135	3	1	0
136	3	1	3
137	3	1	4
138	3	2	13
139	3	2	14
140	3	2	15
141	3	3	0
142	3	3	14
143	3	3	15
144	3	3	16
145	3	3	22
146	3	3	24
147	3	4	13
148	3	4	26
149	3	5	15
150	3	5	23
151	3	5	25
152	4	2	0
153	4	2	8
154	4	2	20
155	4	2	22
156	4	3	11
157	4	3	18
158	4	4	0
159	4	4	11
160	4	4	14
161	4	4	18
162	4	4	23
163	4	5	17
164	4	5	18
165	4	5	23
166	5	4	19
167	5	5	23

APPENDIX IV.

```
C************************************************************
C
C    THIS PROGRAM USES THE STEPWISE REGRESSION METHOD TO
C    DEVELOP A NEW
C    EQUATION OF STATE FOR REFRIGERANT R134A.
C
C************************************************************
     IMPLICIT DOUBLE PRECISION (A-H,P-Z)
     COMMON/PVTD/NPVT,ID(2000),WT(2000),T(2000),
     P(2000),D(2000)
     COMMON /REFDAT/ZC, PTRP, DTRP, TTRP,CMW
     COMMON/CHECK/AA(604,604),KIR(604)
     COMMON/COE/COEF(600)
     COMMON/TERM/FACT(600)
     COMMON/CONST/TC,PC,DC,R
     COMMON/DAT/MOBS,NCO,NMAX,NCON
     COMMON/FEEDCO/NFEED,NINIT(32)
     COMMON/FISH/X(50,2),NUM
     COMMON/STAND/STD
     DIMENSION PCAL(2000),AB(604,604)
     CHARACTER*8 TIM
     OPEN(1,FILE='R134ADAT.T90',STATUS='OLD')
     OPEN(3,FILE='RESULT.DAT',STATUS='OLD')
     OPEN(2,FILE='ROLD.DAT',STATUS='OLD')
     OPEN(7,FILE='RNEW.DAT',STATUS='OLD')
     OPEN(11,FILE='DPDD.DAT',STATUS='OLD')
     OPEN(20,FILE='INIT.DAT',STATUS='OLD')
     OPEN(22,FILE='STD.DAT',STATUS='OLD')
        OPEN(21,FILE='TERMS.DAT',STATUS='OLD')
     OPEN(25,FILE='BWT.DAT',STATUS='OLD')
        OPEN(26,FILE='COEFS.DAT',STATUS='OLD')
     OPEN(30,FILE='BRUN.DAT',STATUS='OLD')
        OPEN(31,FILE='TRBC.DAT',STATUS='OLD')
     NCO=178
     NCON=3
     NFEED=19
     CALL INIT1()
     CALL JGE
C
C    INPUT THE THREE INITIAL TERMS FOR MINIMIZATION
C
C
     CALL TIME(TIM)
     WRITE(3,*)TIM
      READ(1,'(A)') IDUM
```

```
      READ(1,*) CRITC, TC, PC, DC, PTRP, DTRP, TTRP, CMW
      READ(1,'(A)') IDUM
      READ(1,*) NOFIT, FPVT, FCV, FDPD, FDSL, FDSV, FPS,
      FCP, FVIR, FDPT
        R=0.08314471D0
        CALL DREAD(NPVT,NCV,NCP,NDP,NSAT,NVIR,NWS,NPT)
        PCC=PC
        DCC=DC
        TCC=TC
     PROBC=0.9999
     PROBR=0.75
     NMAX=NCO+NCON+1
C
C    INITIALIZE THE MATRIX
C
     DO 10 I=1,NMAX
     DO 10 J=1,NMAX
10   AA(I,J)=0
     DO 13 I=1,NMAX
     DO 13 J=1,NMAX
13   AB(I,J)=0
C
C   FIT THE SATURATION BOUNDARY
C
     CALL FITSAT(FDSL,FDSV,FPS)
C
C  FIT THE SECOND VIRIAL DATA
C
        CALL FITVIR(FVIR)
C
C   FIT THE PVT DATA
C
     CALL FITPVT(FPVT)
C
C   FIT THE CV DATA
C
     CALL FITCV(FCV)
C
C   FIT THE DPDD DATA
C
     CALL FITDPD(FDPD)
C
C   ADD THE CONSTRAINTS
C
     CALL CONSTRAIN
     DO 50 I=1,NMAX
     DO 51 J=1,NMAX
51   AA(J,I)=AA(I,J)
```

```
50    CONTINUE
      DO 52 I=1,NMAX
      DO 53 J=1,NMAX
53    AB(I,J)=AA(I,J)
52    CONTINUE
C
C    CALL SEEQ FUNCTION TO BEGIN MINIMIZATION
C
      NRUN=1
58    IF (NRUN.NE.1) THEN
       DO 56 I=1,NMAX
       DO 57 J=1,NMAX
57     AA(I,J)=AB(I,J)
56     CONTINUE
      ENDIF
      PRINT*,'INPUT THE STARTING TERMS'
         DO 112 I=1,NFEED
      READ(31,*)NINIT(I)
112      CONTINUE
C
C   CALL THE STEPWISE REGRESSION SUBROUTINE
C
      CALL SEEQ(PROBC,PROBR,NEQ)
         WRITE(22,*)NRUN,STD,NEQ
      IF (NRUN.EQ.1) STDNEW=STD
      IF (NRUN.NE.1) THEN
        IF (STD.LT.STDNEW) THEN
          BRUN = NRUN
          STDNEW=STD
        ENDIF
      ENDIF
      NRUN=NRUN+1
      IF (NRUN.LT.1) GOTO 58
C
C    TAKE THE THREE  TERMS WITH HIGHEST SIGNIFICANCE AND
C    USING THEM AS STARTING POINTS FOR NEXT STEP
C
      WRITE(3,6000)
6000  FORMAT(1X,31HFINAL COEFFICIENTS SELECTED ARE)
      DO 70 II=1,NCO
      WRITE(7,*)COEF(II)
      IF (KIR(II).NE.1) GO TO 70
      WRITE(3,6001)COEF(II),II
900   FORMAT(E20.12)
6001       FORMAT(1X,E20.12,I10)
70    CONTINUE
C
C    CALCULATE THE CRITICAL POINT
```

```
C
        CALL CRITF(PCC,DCC,TCC)
C
C   DO THE SATURATION BOUNDARY COMPARISONS
C
        CALL SATCMP
C
C   DO  THE SECOND VIRIAL COMPARISONS
C
     CALL VIRCMP
C
C   DO THE PVT COMPARISONS
C
     CALL PVTCMP
C
C   DO THE CV COMPARISONS
C
     CALL CVCOMP
C
C   DO THE CP COMPARISONS
C
     CALL CPCOMP
C
C   DO THE DPDD COMPARISONS
C
     CALL DPDCMP
C
C   DO THE SOUND VELOCITY COMPARISONS
C
     CALL WSPCMP
     CALL TIME(TIM)
     WRITE(3,*)TIM
     WRITE(30,*)STDNEW
     WRITE(30,*)BRUN
     END
C
C
C
     SUBROUTINE SEEQ(PROBS,PROBF,NEQ)
C************************************************************
C
C   THIS PROGRAM USES THE STEPWISE REGRESSION METHOD
C   DESCRIBED BY DE REUCK AND ARMSTRONG
C
C************************************************************
     IMPLICIT DOUBLE PRECISION (A-H,P-Z)
     DIMENSION SDCO(600),STUDT(600),PROB(600),QQ(600)
     COMMON/COE/COEF(600)
```

```
      COMMON/DAT/MOBS,NCO,NMAX,NCON
      COMMON/CHECK/AA(604,604),KIR(604)
      COMMON/FEEDCO/NFEED,NINIT(32)
      COMMON/STAND/STD
      KRF=0
      NGO=0
      NOCIN=0
      IF (MOBS.GT.500) NAPROX=2
      ICON=NCON
      WRITE(6,6001) AA(NMAX,NMAX)
6001  FORMAT(1X,16HSUM OF SQUARES =,D20.8)
C
C KRF=0 BEFORE SEARCH PROCEDURE STARTS AND AFTER TERM HAS
C BEEN ADDED
C KRF=1 WHEN A TERM HAS BEEN ELIMINATED
C     KRF=2 WHEN AN EXCHANGE HAS TAKEN PLACE
C     KRF=3 WHEN ALL POSSIBLE EXCHANGES HAVE BEEN TRIED OR
C     THERE IS NO TERM AVAILABLE WHICH WILL REDUCE Q-VALUE
C
      DO 20 I=1,NMAX
20    KIR(I)=0
      IF (NFEED.EQ.0) GO TO 22
      DO 30 I=1,NFEED
      II=NINIT(I)
      NOCIN=NOCIN+1
      CALL TRADD(II)
30    KIR(II)=1
C
C
C
22    QREG=AA(NMAX,NMAX)
      QMIN=QREG
      KRF=0
      NGO=0
      WRITE(6,6002)KRF
      WRITE(6,6003) NGO
6002  FORMAT(1X,5HKRF =,I5)
6003  FORMAT(1X,5HNGO =,I5)
      DO 40 I=1,NCO
      IF (KIR(I).GT.0) GO TO 40
      QQ(I)=AA(NMAX,NMAX)-(AA(NMAX,I)*AA(I,NMAX)/AA(I,I))
      IF (QQ(I).LT.0) GO TO 42
      IF (QQ(I).LT.QMIN) GO  TO 41
      GO TO 40
41    QMIN=QQ(I)
      IQMIN=I
      GO TO 40
42    WRITE(6,6004) I
```

```
6004 FORMAT(1X,15HQ-VALUE OF TERM,I3,17HHAS GONE NEGATIVE)
40   CONTINUE
     DIV1=MOBS-NCON-NOCIN-1
     DIV2=MOBS-NCON-NOCIN
     IF (QMIN/DIV1.LT.QREG/DIV2) GO TO 50
     KRF=3
     WRITE(6,6002)KRF
     NGO=1
     WRITE(6,6003) NGO
     IF (NOCIN.NE.0) GO TO 85
     WRITE(6,6016)
6016 FORMAT(1X,33HSUM OF SQUARES CANNOT BE REDUCED
     +  ,/,30HTHERE ARE NO TERMS IN THE EQN.)
     STOP
50   MINQ=IQMIN
     WRITE(6,6005)IQMIN,QMIN
6005 FORMAT(1X,32HTHE TERM WITH MINIMUM Q-VALUE IS,I5,/,5X,
     +   5HQMIN=,D15.8)
C
     CALL TRADD(MINQ)
     NOCIN=NOCIN+1
     KIR(MINQ)=1
C
C
C
     IF (NOCIN.LT.2) GO TO 22
C
C
C
27   IF (ICON.EQ.0) GO TO 200
     IF (NOCIN.LT.NCON) GO TO 22
     DO 60 I=1,NCON
     I1=NCO+I
     CALL TRADD(I1)
60   KIR(I1)=1
     ICON=0
C
C
200  CALL STATS(VAR1,VAR2,SDCO,STUDT,PROB,NOCIN)
C
C
     MST=MINST(STUDT)
     DO 70 I=1,NCO
     IF (KIR(I).EQ.0) GO TO 70
     IF (PROB(I).GT.PROBS) GO  TO 70
     IF (KRF.EQ.2) GO TO 71
     IF (MST.EQ.MINQ) WRITE(6,6006)
6006 FORMAT(1X,42HLAST TERM ADDED FAILED T-TEST,TRY
```

```
      EXCHANGE)
      IF (MST.EQ.MINQ) GO TO 85
71    MELIM=MST
      CALL TRELIM(MELIM)
      KRF=1
      WRITE(6,6002) KRF
      NOCIN=NOCIN-1
      KIR(MELIM)=0
      GO TO 200
70    CONTINUE
      IF (KRF.EQ.1) GO TO 22
      IF (KRF.EQ.2) GO TO 85
      IF (KRF.EQ.3) GO TO 300
C
C
C
80    MST=MINST(STUDT)
      NFRE1=MOBS-NCON-NOCIN
      NFRE2=NFRE1+1
      IF (VAR2.LT.VAR1) WRITE(6,6007) VAR2,VAR1
      IF (VAR2.LT.VAR1) STOP
6007  FORMAT(1X,18HF-TEST IS INVERTED,2E12.5)
      PROBAL=FISHER(NFRE1,NFRE2,VAR1,VAR2,NAPROX)
      FISH=VAR2/VAR1
      WRITE(6,6008) FISH,PROBAL
6008  FORMAT(1X,6HFISH=,E12.6,2X,8HPROBAL =,E12.6)
      IF (PROBAL.GT.PROBF) GO TO 22
      IF (KRF.EQ.3) GO TO 95
C
C
      WRITE(6,6009)
6009  FORMAT(1X,29HINTERCORRELATION IS SUSPECTED)
85    QREG=AA(NMAX,NMAX)
      QMIN=QREG
      DO 90 NIN=1,NCO
      IF (KRF.NE.0) GO TO 93
      IF (NIN.EQ.MINQ) GO TO 90
93    IF (KIR(NIN).EQ.0) GO TO 90
      DO 91 NOUT=1,NCO
      IF (KIR(NOUT).EQ.1) GO TO 91
      QINOUT=QSWAP(NIN,NOUT)
      IF (QINOUT.LT.0) GO TO 96
      IF (QINOUT.LT.QMIN) GO TO 92
      GO TO 91
92    QMIN=QINOUT
      IIN=NIN
      IOUT=NOUT
      GO TO 91
```

```
96   WRITE(6,6010) NIN,NOUT,QINOUT
6010 FORMAT(1X,12HSWAP OF TERM,I3,9HWITH TERM
     +  ,I3,17HGIVES NEGATIVE Q-, E15.5)
91   CONTINUE
90   CONTINUE
     IF (QMIN.LT.QREG) GO TO 100
     IF (NGO.EQ.1) GO TO 300
     WRITE(6,6011)
6011 FORMAT(1X,50HNO FURTHER REDUCTION INTER
     +    CORRELATION IS POSSIBLE)
     KRF=3
     WRITE(6,6002)KRF
     GO TO 80
C
C
95   MST=MINST(STUDT)
     CALL TRELIM(MST)
     KIR(MST)=0
     NOCIN=NOCIN-1
     NGO=1
     WRITE(6,6003) NGO
     GO TO 85
C
C
100  CALL TRELIM(IIN)
     NOCIN=NOCIN-1
     KIR(IIN)=0
     CALL TRADD(IOUT)
     NOCIN=NOCIN+1
     KIR(IOUT)=1
     KRF=2
     WRITE(6,6002) KRF
     GO TO 200
300  CALL STATS(VAR1,VAR2,SDCO,STUDT,PROB,NOCIN)
     WRITE(6,6012)
6012 FORMAT(1X,31HFINAL COEFFICIENTS SELECTED ARE)
     NEQ=0
     DO 201 I=1,NCO
     IF (KIR(I).EQ.1) WRITE(6,6013) COEF(I),I
     IF (KIR(I).EQ.1) NEQ=NEQ+1
     IF (KIR(I).LT.1) COEF(I)=0
201  CONTINUE
     WRITE(6,6014) NEQ
6014 FORMAT(1X,9HTHERE ARE,I5,29HCOEFFICIENTS IN THE
     EQUATION.)
     STD=SQRT(VAR1)
     WRITE(6,6015) STD
6013 FORMAT(1X,E20.12,I10)
```

```
6015 FORMAT(1X,/,7H  STD =,G12.5//)
     RETURN
     END
C
C
     FUNCTION FISHER(NFRE1,NFRE2,VAR1,VAR2,NAPROX)
C************************************************************
C
C    THIS CALCULATES EQN(13) USING THE METHOD OF ALGORITHM
C    322 FOR DEGREES OF FREEDOM LESS THAN 500 AND
C
C************************************************************
     IMPLICIT DOUBLE PRECISION (A-H,P-Z)
     DATA PINV/0.3183098862/
     DATA RT2INV/0.70716781187/
     GO TO (10,100),NAPROX
10   M=NFRE1
     N=NFRE2
     FISH=VAR2/VAR1
     MA=2*(M/2)-M+2
     NB=2*(N/2)-N+2
     W=FISH*FLOAT(M)/FLOAT(N)
     Z=1/(1+W)
     IF (MA.NE.1) GO TO 22
     IF (NB.NE.1) GO TO 12
     P=SQRT(W)
     D=PINV*Z/P
     P=2*PINV*ATAN(P)
     GO TO 40
12        P=SQRT(W*Z)
     D=0.5*P*Z/W
     GO TO 40
22   IF (NB.NE.1) GO TO 30
     P=SQRT(Z)
     D=0.5*Z*P
     P=1-P
     GO TO 40
30   D=Z*Z
     P=W*Z
40   Y=2*W/Z
     JJ=NB+2
     IF (MA.NE.1) GO TO 51
     DO 50 J=JJ,N,2
     D=(1+FLOAT(MA)/FLOAT(J-2))*D*Z
     P=P+(D*Y/FLOAT(J-1))
50   CONTINUE
     GO TO 52
51   ZK=Z**(FLOAT((N-1)/2))
```

```
      D=D*ZK*FLOAT(N/NB)
      P=P*ZK+W*Z*(ZK-1)/(Z-1)
52    Y=W*Z
      Z=2/Z
      NB=N-2
      II=MA+2
      DO 60 I=II,M,2
      J=I+NB
      D=Y*D*FLOAT(J)/FLOAT(I-2)
      P=P-Z*D/FLOAT(J)
60    CONTINUE
      FISHER=P
      RETURN
100   XNUM=ABS(SQRT(VAR1)-SQRT(VAR2))
      XDEN=SQRT(VAR1/(2*FLOAT(NFRE1))+VAR2/(2*FLOAT(NFRE2)))
      Z=XNUM/XDEN
      FISHER=0.5*ERFCC(-RT2INV*Z)
      RETURN
      END
C
C
      FUNCTION ERFCC(X)
C************************************************************
C
C
C     THIS FUNCTION CALCULATES THE VALUE OF THE ERROR
C     FUNCTION.  IT IS USED WHILE CALCULATING THE VALUE OF
C     THE INTEGRAL.
C
C************************************************************
      IMPLICIT DOUBLE PRECISION (A-H,P-Z)
      Z=ABS(X)
      T=1/(1+0.5*Z)
      ERFCC=T*EXP(-Z*Z-1.26551223+T*(1.00002368+T*(
     *   0.37409196+T*(0.09678418+T*(-0.18628806+T*(
     *   0.27886807+T*(-1.13520398+T*(1.48851587+T*(
     *   -0.82215223+T*0.17087277)))))))))
      IF (X.LT.0) ERFCC=2-ERFCC
      RETURN
      END
C
C
      FUNCTION MINST(STUDT)
C************************************************************
C
C     THIS CALCULATES THE TERM WITH MIN STUDENT-T VALUE
C
C************************************************************
```

```
      IMPLICIT DOUBLE PRECISION (A-H,P-Z)
      COMMON/CHECK/AA(604,604),KIR(604)
      COMMON/DAT/MOBS,NCO,NMAX,NCON
      DIMENSION STUDT(600)
      STMIN=1E50
      DO 10 I=1,NCO
        IF (KIR(I).NE.1) GOTO 10
        IF (STUDT(I).LT.STMIN) GOTO 11
        GO TO 10
11      STMIN=STUDT(I)
        IMIN=I
10    CONTINUE
      MINST=IMIN
      RETURN
      END
C
C
      DOUBLE PRECISION FUNCTION QSWAP(NIN,NOUT)
C************************************************************
C
C
C     THIS CALCULATES THE Q-VALUE FOUND FOR A PROPOSED
C     EXCHANGE OF THE TERM NIN WITH NOUT
C
C************************************************************
      IMPLICIT DOUBLE PRECISION (A-H,P-Z)
      COMMON/CHECK/AA(604,604),KIR(604)
      COMMON/DAT/MOBS,NCO,NMAX,NCON
    BMAX=AA(NMAX,NMAX)-(AA(NIN,NMAX)*AA(NMAX,NIN)/AA(NIN,NIN))
      BI=AA(NOUT,NMAX)-(AA(NIN,NMAX)*AA(NOUT,NIN)/AA(NIN,NIN))
      DI=AA(NOUT,NOUT)-(AA(NIN,NOUT)*AA(NOUT,NIN)/AA(NIN,NIN))
      QSWAP=BMAX-(BI*BI)/DI
      RETURN
      END
C
C
      SUBROUTINE STATS(VAR1,VAR2,SDCO,STUDT,PROB,NOCIN)
C************************************************************
C
C
C     STD=STANDARD DEVIATION FO EQN (EQN 10)
C     SDCO(I)=STANDARD DEV OF COEEFF COEFF(I) EQN. (7)
C     STUDT(I)=STUDENT-T-VALUE (EQN 8) NEEDED FOR CALCULATING
C      THE PROB PROB(I)
C
C************************************************************
      IMPLICIT DOUBLE PRECISION (A-H,P-Z)
      COMMON/CHECK/AA(604,604),KIR(604)
```

```
      COMMON/COE/COEF(600)
      COMMON/DAT/MOBS,NCO,NMAX,NCON
      COMMON/FISH/X(50,2),NUM
C
C    THIS CALCULATES THE COEFFICIENTS AND THE STATISTICAL
C    QUANTITIES NEEDED FOR DETERMINING THE SIGNIFICANCE OF
C    EACH TERM AND OF THE EQUATION AS  A WHOLE
C
      DIMENSION SDCO(600),STUDT(600),PROB(600)
      DIV=MOBS-NCON-NOCIN
      STD=DSQRT(DABS(AA(NMAX,NMAX))/DBLE(DIV))
      VAR1=STD*STD
      WRITE(6,60) STD
      WRITE(6,61)
      DO 1 I=1,50
      DO 1 J=1,2
1     X(I,J)=0
      NUM=1
      DO 10 I=1,NCO
      IF (KIR(I).NE.1) GO TO 10
      SDCO(I)=DBLE(STD)*DSQRT(DABS(AA(I,I)))
      COEF(I)=AA(I,NMAX)
      STUDT(I)=ABS(COEF(I)/SDCO(I))
      TFRE=STUDT(I)
      NFRE=(MOBS-NCON-NOCIN)
      PROB(I)=STUDEP(NFRE,TFRE)
      X(NUM,1)=I
      X(NUM,2)=STUDT(I)
      NUM=NUM+1
      WRITE(6,62)I,SDCO(I),COEF(I),STUDT(I),PROB(I)
60    FORMAT(1X,7HSTD =   ,E12.6)
61    FORMAT(10X,4HSDCO,9X,4HCOEF,6X,5HSTUDT,8X,4HPROB)
62    FORMAT(1X,I5,4E12.5)
10    CONTINUE
      NUM=NUM-1
C     CALL SORT()
      MST=MINST(STUDT)

VAR2=(AA(NMAX,NMAX)-(AA(MST,NMAX)*AA(NMAX,MST)/AA(MST,MST)))
/
      +  DBLE(DIV+1.0)
      RETURN
      END
C
C
      FUNCTION STUDEP(NFRE,TFRE)
C************************************************************
C
```

```
C
C   THIS CALCULATES THE STUDENT-T PROB. FOR
C   EQN(9) USING ALGORITHM AS3 B
C
C************************************************************
      IMPLICIT DOUBLE PRECISION (A-H,P-Z)
      DATA PINV/0.3183098862/
      N=NFRE
      T=TFRE
      IF (N.GT.50.AND.TFRE.GT.6) GOTO 50
      FN=N
      IOE=N-2*(N/2)
      IN2=N-2
      A=T/SQRT(FN)
      B=FN/(FN+T**2)
      S=1
      C=1
      KS=2+IOE
      FK=KS
      IF (IN2-2) 6,7,7
7     DO 8 K=KS,IN2,2
      C=C*B*(FK-1)/FK
      S=S+C
8     FK=FK+2
6     IF (IOE) 1,1,2
1     STUDEP=A*SQRT(B)*S
      RETURN
2     IF (N-1)4,4,5
4     S=0
5     STUDEP=2*(A*B*S+ATAN(A))*PINV
      RETURN
50    STUDEP=1
      RETURN
      END
C
C
      SUBROUTINE TRADD(MINQ)
C************************************************************
C
C     THIS TRANSFORMS THE MATRIX AA BY ADDING A NEW TERM
C     MINQ TO THE EQUATION
C
C************************************************************
      IMPLICIT DOUBLE PRECISION (A-H,P-Z)
      COMMON/CHECK/AA(604,604),KIR(604)
      COMMON/DAT/MOBS,NCO,NMAX,NCON
      DIMENSION AQJ(604),AIQ(604)
      IQMIN=MINQ
```

```
     DO 5 I=1,NMAX
5    AIQ(I)=AA(I,IQMIN)
     DO 6 J=1,NMAX
6    AQJ(J)=AA(IQMIN,J)
     DO 10 I=1,NMAX
     DO 11 J=1,NMAX
     IF (I.EQ.IQMIN) GO TO 20
     IF (J.EQ.IQMIN) GO TO 30
     IF (KIR(J).EQ.1) GO TO 31
     AA(I,J)=AA(I,J)-(AQJ(J)*AIQ(I)/AIQ(IQMIN))
     GO TO 11
30   AA(I,J)=-(AIQ(I)/AIQ(IQMIN))
     GO TO 11
31   AA(I,J)=AA(I,J)-(AQJ(J)*AIQ(I)/AIQ(IQMIN))
     GO TO 11
20   IF (J.EQ.IQMIN) GO TO 21
     IF (KIR(J).EQ.1) GO TO 22
     AA(I,J)=AQJ(J)/AIQ(IQMIN)
     GO TO 11
21   AA(I,J)=1.0/AIQ(IQMIN)
     GO TO 11
22   AA(I,J)=AQJ(J)/AIQ(IQMIN)
11   CONTINUE
10   CONTINUE
     WRITE(6,60) IQMIN,AA(NMAX,NMAX)
60   FORMAT(1X,8HTHE TERM ,I5,30HHAS BEEN ADDED TO THE
     EQUATION
     +  ,/, 16HSUM OF SQUARES=,D20.8)
     RETURN
     END
C
C
     SUBROUTINE TRELIM(MINQ)
C************************************************************
C
C    THIS SUBROUTINE TRANSFORMS THE MATRIX AA BY ELIMINATING
C     THE TERM MINQ FROM THE EQUATION
C
C************************************************************
     IMPLICIT DOUBLE PRECISION (A-H,P-Z)
     COMMON/CHECK/AA(604,604),KIR(604)
     COMMON/DAT/MOBS,NCO,NMAX,NCON
     DIMENSION AQJ(604),AIQ(604)
     IQMIN=MINQ
     DO 5 I=1,NMAX
5    AIQ(I)=AA(I,IQMIN)
     DO 6 J=1,NMAX
6    AQJ(J)=AA(IQMIN,J)
```

```
      DO 10 I=1,NMAX
      DO 11 J=1,NMAX
      IF (I.EQ.IQMIN) GO TO 20
      IF (J.EQ.IQMIN) GO TO 30
      IF (KIR(J).NE.1) GO TO 31
      AA(I,J)=AA(I,J)-(AQJ(J)*AIQ(I)/AIQ(IQMIN))
      GO TO 11
30    AA(I,J)=-(AIQ(I)/AIQ(IQMIN))
      GO TO 11
31    AA(I,J)=AA(I,J)-(AQJ(J)*AIQ(I)/AIQ(IQMIN))
      GO TO 11
20    IF (J.EQ.IQMIN) GO TO 21
      IF (KIR(J).NE.1) GO TO 22
      AA(I,J)=AQJ(J)/AIQ(IQMIN)
      GO TO 11
21    AA(I,J)=1/AIQ(IQMIN)
      GO TO 11
22    AA(I,J)=AQJ(J)/AIQ(IQMIN)
11    CONTINUE
10    CONTINUE
      WRITE(6,60) IQMIN,AA(NMAX,NMAX)
60    FORMAT(1X,8HTHE TERM,I5,37HHAS BEEN ELIMINATED FROM THE
      EQN
      +  ,/,     16HSUM OF SQUARES =,D20.8)
      RETURN
      END
C
C
      SUBROUTINE SORT()
      IMPLICIT DOUBLE PRECISION (A-H,P-Z)
      COMMON/FISH/X(50,2),NUM
      LAST=NUM
      DO 20 J=1,NUM-1
       NTR=J
       IRST=J+1
         DO 5 K=IRST,LAST
         IF (X(K,2).LT.X(NTR,2)) NTR=K
5     CONTINUE
      HOLD=X(J,2)
      HOLD1=X(J,1)
      X(J,2)=X(NTR,2)
      X(J,1)=X(NTR,1)
      X(NTR,2)=HOLD
      X(NTR,1)=HOLD1
20    CONTINUE
      RETURN
      END
C
```

```
C
C
C
      SUBROUTINE FIT(RES,WGT)
C************************************************************
C
C     THIS SUBROUTINE BUILDS THE REGRESSION MATRIX
C
C************************************************************
      IMPLICIT DOUBLE PRECISION (A-H,P-Z)
      COMMON/TERM/FACT(600)
      COMMON/CHECK/AA(604,604),KIR(604)
      COMMON/DAT/MOBS,NCO,NMAX,NCON
      COMMON/PROBLEM/NCONT
      IF (NCONT.GT.0) GOTO 20
       DO 22 I=1,NCO
       DO 24 J=1,NCO
24    AA(I,J)=AA(I,J)+FACT(I)*FACT(J)
22    AA(I,NMAX)=AA(I,NMAX)+RES*FACT(I)
      AA(NMAX,NMAX)=AA(NMAX,NMAX)+RES*RES
      RETURN
20    DO 40 I=1,NCO
40    AA(I,NCO+NCONT)=FACT(I)
      AA(NCO+NCONT,NMAX)=RES
      RETURN
      END
C
C
C
       SUBROUTINE FITSAT(FDSL,FDSV,FPS)
C************************************************************
C
C      THIS SUBROUTINE FITS THE SATURATION BOUNDARY
C
C************************************************************
       IMPLICIT DOUBLE PRECISION (A-H,P-Z)
       PARAMETER (MSAT = 150)
       COMMON/DAT/MOBS,NCO,NMAX,NCON
C
       COMMON /SATD/ NSAT, WSL(MSAT), WSV(MSAT), TS(MSAT),
PS(MSAT),
      *              DSL(MSAT), DSV(MSAT)
C
       COMMON /REFDAT/ ZC, G(50), GAMMA, PTRP, DTRP, TTRP
      *               , CMW
C
       COMMON /DERIV/ DPSDT, DDSDT
C
```

```
      COMMON /TERM/FACT(600)
C
      COMMON/CONST/TC,PC,DC,R
C
      DIMENSION F(600)
C
      DATA EPS / 1.0E-12 /
C
      DO 290 J =1, NSAT
C
C                        SATURATED LIQUID DENSITY FIT
C
      PP = PSATF(TS(J))
      SIGP = 0.001 * PS(J)
      SIGD = 0.001 * DSL(J) * DPSDT / DDSDT
      SIGT = 0.002 * DPSDT
C     WSL(J) = FDSL / SQRT(SIGP**2 + SIGD**2 + SIGT**2)
      WSL(J) = FDSL
      if (ts(j).gt.358) wsl(j) = 2*wsl(j)
      PSAT = PS(J) + EPS
      CALL PRESSB(TS(J),PK,DSL(J),0)
      RES=PS(J)-R*DSL(J)*TS(J)
     DO 11 IJ=1,NCO
         FACT(IJ)=FACT(IJ)*WSL(J)
11      CONTINUE
      RES = RES*WSL(J)
      CALL FIT(RES,WSL(J))
C
C                        SATURATED VAPOR DENSITY
C
      PSAT = PSAT - 2.0 * EPS
      SIGD = 0.001 * DSV(J) * DPSDT / DDSDT
      WSV(J)=FDSV
C     IF(WSV(J).LT.2.0) WSV(J) = 2.0
C     IF(J.GT.12) WSV(J) = WSV(12)
      CALL PRESSB(TS(J),PK,DSV(J),0)
      RES=PS(J)-R*DSV(J)*TS(J)
      DO 12 IJ=1,NCO
      FACT(IJ)=FACT(IJ)*WSV(J)
12    CONTINUE
      RES = RES*WSV(J)
      CALL FIT(RES,WSV(J))
C
C                        GIBBS CONSTRAINT DATA
C
      WGIBB =   FPS
      IF (TS(J).GT.368.) WGIBB=2*WGIBB
      CALL AR(PP,DSL(J),TS(J),0)
```

```
      DO 270 K = 1, NCO
  270 F(K) = FACT(K)
      CALL AR(PP,DSV(J),TS(J),0)
      DO 280 K = 1, NCO
  280 FACT(K) = (FACT(K)-F(K))
      RES=PS(J) * (1./DSL(J)-1./DSV(J))
     * +(LOG(DSL(J)/DSV(J)))*R*TS(J)
      DO 13 IJ=1,NCO
       FACT(IJ)=FACT(IJ)*WGIBB
13    CONTINUE
      RES = RES*WGIBB
      CALL FIT(RES,WGIBB)
  290 CONTINUE
      MOBS=MOBS+NSAT
      return
      END

C
      SUBROUTINE FITCP(FCP)
C************************************************************
C
C      THIS SUBROUTINE FITS THE CP DATA
C
C************************************************************
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      PARAMETER (MCP = 100)
      PARAMETER (NC = 32, NF = NC + 1 )
C
      COMMON /CPTP/ NCP, IDCP(MCP), WCP(MCP), TCP(MCP),
      PCP(MCP), DCP(MCP), CP(MCP)
      COMMON /FITCOM/ A(600)
C      COMMON /REFDAT/ R, PC, DC, TC, ZC, G(NC), GAMMA,
       PTRP, DTRP, TTRP
C     *              , CMW
      COMMON /TERM/FACT(600)
      COMMON/CONST/TC,PC,DC,R
      COMMON/DAT/MOBS,NCO,NMAX,NCON
      DIMENSION F(600)
C
      IF (NCP.EQ.0) GO TO 120
      DO 100 I = 1, NCP
C
C         CALCULATE THE DENSITY OF THE CP POINT
C
  020 DCP(I) = RHOF(PCP(I),0.0D0,TCP(I))
C
C         FORM THE DERIVATIVE DS/DT AT CONSTANT P
C
```

```
C      CALL DPDD(DPD,DCP(I),TCP(I),2)
C      CALL DPDT(DPT,DCP(I),TCP(I),2)
C      DPT = DPT / DCP(I)
C      TERM = 100.0 * TCP(I) * DPT * DPT / DPD
       CALL NONLIN(PCP(I),DCP(I),TCP(I),CPNL)
       TERM=CPNL*100
       CALL IDEAL(TCP(I),CV0,S0)
       RES = 0.01 * (CP(I) - CV0 - TERM)
      CALL CVR(PP,DCP(I),TCP(I),0)
      DO 050 J =1, NCO
  050 F(J) = FACT(J)
       CALL CVR(PP,0.0D0,TCP(I),0)
      DO 060 J= 1, NCO
  060 FACT(J) = FACT(J) - F(J)
      TR = TCP(I) / TC
      DR = DCP(I) / DC
      SIGC = 0.02 * CP(I)
      SIGD = 0.001 * TCP(I) * DP2 / DCP(I)
      SIGT = 0.5 * CP(I) / TCP(I)
      WCP(I) = 25.0 * FCP / SQRT(SIGC**4 + SIGD**2 +
      SIGT**2) / TR**2
      IF(ABS(TR-1.0).LT.0.025 .AND. ABS(DR-1.0).LT.0.1)
     * WCP(I) = 0.0
      IF (IDCP(I).GT.0) GO TO 070
      IDCP(I) = -IDCP(I)
      WCP(I) = 0.0
  070 CALL FIT(RES,WCP(I))
      IF (WCP(I).GT.0) MOBS=MOBS+1
  100 CONTINUE
  120 RETURN
      END
C
C
C
      SUBROUTINE CPCOMP
C************************************************************
C
C    THIS SUBROUTINE DOES THE CP COMPARISONS.
C
C************************************************************
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      PARAMETER (MCP = 100)
C
      COMMON /CPTP/ NCP, IDCP(MCP), WCP(MCP), TCP(MCP),
      PCP(MCP), DCP(MCP), CP(MCP)
C
      WRITE(3,200)
      AAD = 0.0
```

```
      BIAS = 0.0
      RMS = 0.0
      NI = 0
      DO 100 I = 1, NCP
      D = RHOF(PCP(I),0.0D0,TCP(I))
      CVCAL = CVF(D,TCP(I))
      CALL DPDT(P1,D,TCP(I),1)
      CALL DPDD(P2,D,TCP(I),1)
      CPCAL = CVCAL + 100.0 * TCP(I) * P1 * P1 / (D * D *
      P2)
      PCT = -100.0 * (CPCAL - CP(I)) / CP(I)
      WRITE(3,220) IDCP(I), TCP(I), PCP(I), CPCAL, CP(I),
      PCT, WCP(I)
C     IF (WCP(I).LE.0.0) GO TO 100
      NI = NI + 1
      AAD = AAD + ABS(PCT)
      BIAS = BIAS + PCT
      RMS = RMS + PCT * PCT
  100 CONTINUE
      AAD = AAD / NI
      BIAS = BIAS / NI
      RMS = SQRT(RMS/NI-BIAS*BIAS)
      WRITE(3,240) NI, AAD, BIAS, RMS
  200 FORMAT('1CP(P,T) COMPARISONS'/
      1'0  ID     T,K     P,BAR    CP,CALC  CP,EXP   CP,%
      WT')
  220 FORMAT(I5,F9.3,F9.4,2F9.3,F8.3,F8.2)
  240 FORMAT('0N = ',I6,'  AAD = ',F7.3,'  BIAS = ',F7.3,'
      RMS =',F7.3)
      RETURN
      END

C
      FUNCTION PSATF(T)
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
C
C * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C
C
C  PURPOSE --- THIS ROUTINE CALCULATES THE SATURATION
C  PRESSURE AND ITS DERIVATIVE WITH RESPECT TO T, GIVEN T.
C
C
C            CODED BY--J. F. ELY
C                CHEMICAL ENGINEERING SCIENCE DIVISION 773.20
C                     NATIONAL BUREAU OF STANDARDS
C                     BOULDER, COLORADO   80303
C
```

```
C
C           VERSION 1.0 -- R134A 11/17/89
C
C
C * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C
C
      DIMENSION G(6)
      COMMON /DERIV/ DPSDT, DDSDT
C     DATA TC, EPP, G /     374.255D0,      1.90D0,
      0.449274293643D+00,
C    *      0.856704281132D+01,-0.763187819745D+01,
      0.116786832983D+02,
C    *     -0.148122815240D+02, 0.590114977172D+01/
C     DATA TC, EPP, G /     374.255D0,      1.90D0,
      0.372921742638D+02,
      .234940894419D+02,-0.171719531831D+02,
      -0.224297362353D+02,
C    *      0.465208401919D+02,-0.267091331530D+02/
      DATA TC, EPP, G /374.255D0,      1.90D0,
      0.136660830830D+02,
     *      0.130706099332D+02,-0.110674364581D+02,
      0.231467040853D+01,
     *      0.403653437325D+01,-0.465093928476D+01/
      X = T / TC
      IF (X.GT.1.0) GO TO 010
      PCAL = G(1)*(1.0-X)**EPP + G(2) + G(3)/X + G(4)*X +
      G(5)*X**2
      PCAL = EXP(PCAL + G(6)*X**3)
      DPSDT=PCAL*(-EPP*G(1)*(1.0-X)**(EPP-1.0) - G(3)/(X*X)
     * + G(4) + 2.0*G(5)*X + 3.0*X*X*G(6))/ TC
      PSATF = PCAL
      RETURN
  010 PSATF = 26.55
      DPSDT = 100.0
      RETURN
      END
      FUNCTION DSATL(T)
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
C * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C
C      PURPOSE --- THIS ROUTINE CALCULATES THE SATURATED
C      LIQUID DENSITY OF R134a
C
C     11/17/89 -- JFE
C
C * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C
```

```
      DIMENSION G(4)
C
      COMMON /DERIV/ DPSDT, DDSDT
C
      DATA DC / 5.0167D0 /
      DATA TC, BETA, G     / 374.255D0,     0.35D0,
      0.211351877568D+01,
     *     -0.355571666597D+00,-0.795726229378D+00,
      0.780289360803D+00/
C
      IF (T.GT.TC) GO TO 010
      X = 1.0 - T/TC
      DENOM = 1.0 + G(2) * X**(1.0-BETA)
      Y = (G(1) * X**BETA + G(3) * X * X + G(4) * X*X*X) /
      DENOM
      DYDX = BETA*G(1)*X**(BETA-1.0) + 2.0*G(3)*X +
      3.0*G(4)*X*X
      DYDX = (DYDX - Y * G(2) * (1.0-BETA)/ X**BETA) / DENOM
C
      DSATL = DC * (Y + 1.0)
      DDSDT = - DC * DYDX / TC
      RETURN
  010 DSATL = DC
      DDSDT = 100.
      RETURN
      END
      FUNCTION DSATV(T)
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
C
C * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C
C
C      PURPOSE --- THIS ROUTINE CALCULATES THE SATURATED
C      VAPOR DENSITY OF R134A
C
C      9/02/85 - JFE
C
C * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C
C
      DIMENSION G(4)
C
      LOGICAL ENTER
C
      COMMON /DERIV/ DPSDT, DDSDT
C
C
      DATA DC / 5.0167D0 /, R / 0.0831441D0 /
```

```
C     DATA TC, BETA, G /     374.255D0,     0.35D0,
      0.312475706814D+02,
C    *     -0.103250297083D+03,
      0.116419951787D+03,-0.452587571181D+02/
C     DATA TC, BETA, G /     374.255D0,     0.35D0,
      0.298596344915D+02,
C    *     -0.978891446364D+02,
      0.109675740236D+03,-0.424760053876D+02/
      DATA TC, BETA, G /     374.255D0,     0.35D0,
      0.289266947736D+02,
     *     -0.945769215843D+02,
      0.105792161455D+03,-0.409737577099D+02/
      DATA ENTER / .FALSE. /
C
      IF (ENTER) GO TO 010
      ENTER = .TRUE.
      PC = PSATF(TC)
      ZC = PC / (R*DC*TC)
C
  010 TR = T / TC
      IF (TR.GT.1.0) GO TO 020
      TAUB = (1.0 - TR)**BETA
      TRM = G(1) + TR * (G(2) + TR * (G(3) + TR * G(4)))
      PS = PSATF(T)
      PR = PS / PC
      FX = 1.0 + TAUB * TRM
      Z = 1.0 + (ZC-1.0) * PR * FX / TR
      DSATV = PS / (R * T * Z)
C
      TRMP = G(2) + TR * (2.0*G(3) + 3.0*G(4)*TR)
      DFDX = TAUB * (TRMP - BETA * TRM / (1.0 - TR))
      DZDX = ((TC * DPSDT / PC) - PR / TR) * FX
      DZDX = (ZC-1.0) * (DZDX + PR*DFDX) / TR
      DDSDT = (DPSDT/PS - 1.0/T - DZDX/(Z*TC)) * DSATV
      RETURN
  020 DSATV = DC
      DDSDT = 100.
      RETURN
      END
C

      SUBROUTINE DREAD(JPVT,JCV,JCP,JDP,JSAT,JVIR,JWS,JPT)
C***********************************************************
C
C
C    THIS SUBROUTINE READS IN THE DATA FILE FOR R134A.
C
C***********************************************************
```

```
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      PARAMETER (NC = 32, NF = NC + 1 )
      PARAMETER (MPVT = 2000, MVIR = 100, MSAT = 150, MCV =
    200)
      PARAMETER (MDPD = 500, MDPT = 001, MWSP = 450, MCP =
      100)
C
      COMMON /PVTD/ NPVT, ID(MPVT), WT(MPVT), T(MPVT),
      P(MPVT),
     *              D(MPVT)
C
      COMMON /CVDAT/ NCV, IDCV(MCV), WCV(MCV), TCV(MCV),
      PCV(MCV),
     *               DCV(MCV), CV(MCV)
C
      COMMON /CPTP/ NCP, IDCP(MCP), WCP(MCP), TCP(MCP),
      PCP(MCP),
     1              DCP(MCP), CP(MCP)
C
      COMMON /DPDAT/ NDP, IDP(MDPD), WDP(MDPD), TDP(MDPD),
      DDP(MDPD),
     1               DPDX(MDPD)
C
      COMMON /DPTDAT/ NPT, IDPT(MDPT), WDPT(MDPT),
      TPT(MDPT), DPT(MDPT),
     *                DPDTX(MDPT)
C
      COMMON /SATD/ NSAT, WSL(MSAT), WSV(MSAT), TS(MSAT),
      PS(MSAT),
     *              DSL(MSAT), DSV(MSAT)
C
      COMMON /VIRIAL/ NVIR, IDV(MVIR), WV(MVIR), TV(MVIR),
      BV(MVIR)
C
      COMMON /SOUND/ NWS, IDWS(MWSP), TWS(MWSP), PWS(MWSP),
      WSPT(MWSP)
C
      COMMON/CONST/TC,PC,DC,R
      COMMON /REFDAT/ZC,PTRP, DTRP, TTRP, CMW
C
      COMMON /REFN/ NREF, IDREF(100)
      COMMON /REFS/ REFER(100)
C
      CHARACTER*4 NTYPE(9), ITYPE, REFER*80, RECORD*72
C
      DATA NTYPE / 'PVTD', 'CVTD', 'CSLD', 'CPTP', 'DPDD',
      'BVIR',
     *             'WSPT', 'WSAT', 'DPDT'/
```

```
C
      NPVT = 0
      NCV  = 0
      NCP  = 0
      NDP  = 0
      NVIR = 0
      NSAT = 0
      NWS = 0
      NPT = 0
      NREF = 0
      NP=0
C

  020 READ (1,'(A4,I4,A)') ITYPE, IDENT, RECORD
      IF (ITYPE.EQ.'EOF') GO TO 1000
      IF (ITYPE.EQ.'REM') GO TO 020
      IF (ITYPE.NE.'REF ') THEN
         READ (RECORD,1100) TK, PROP1, PROP2, PROP3
         DO 040 J = 1, 9
         IF (ITYPE.EQ.NTYPE(J))
     *   GO TO (100,200,300,400,500,600,700,800,900),J
  040    CONTINUE
         GO TO 020
      ELSE
         NREF = NREF + 1
         REFER(NREF) = RECORD
         IDREF(NREF) = IDENT
         GO TO 020
      ENDIF
C                        PVT DATA
C
C         ELIMINATE DATA INSIDE 2-PHASE REGION
C
  100 CONTINUE
      NP=NP+1
      IF (TK.GE.TC) GO TO 110
      DL = DSATL(TK)
      DV = DSATV(TK)
      SP = PSATF(TK)
      IF (PROP1.GT.SP .AND. PROP2.LT.DL) GO TO 020
      IF (PROP1.LT.SP .AND. PROP2.GT.DV) GO TO 020
  110 CONTINUE
      NPVT = NPVT + 1
      ID(NPVT) = IDENT
      T(NPVT) = TK
      P(NPVT) = PROP1
      D(NPVT) = PROP2
C      READ(25,*)WT(NPVT)
```

```
      GO TO 020
C                               CV DATA
  200 NCV = NCV + 1
      IDCV(NCV) = IDENT
      TCV(NCV) = TK
      PCV(NCV) = PROP1
      DCV(NCV) = PROP2
      CV(NCV) = PROP3
      IF (PROP1.LE.0.0) CALL PRESSB(PCV(NCV),PROP2,TK,1)
C      WCV(NCV) = 0.0
C     DCV(NCV) = RHOF(PCV(NCV),PROP2,TCV(NCV))
      GO TO 020
C                               C(SAT) DATA
  300 NCV = NCV + 1
      IDCV(NCV) = IDENT
      TCV(NCV) = TK
      DCV(NCV) = DSATL(TK)
      CV(NCV) = -PROP1
      WCV(NCV) = 0.0
      GO TO 020
C                               CP DATA
  400 NCP = NCP + 1
      IDCP(NCP) = IDENT
      TCP(NCP) = TK
      PCP(NCP) = PROP1
      CP(NCP) = PROP2
      WCP(NCP) = 0.0
      GO TO 020
C                               DPDD DATA
  500 NDP = NDP + 1
      IDP(NDP) = IDENT
      TDP(NDP) = TK
      DDP(NDP) = PROP1
      DPDX(NDP) = PROP2
      GO TO 020
C                               SECOND VIRIAL COEFFICIENTS
  600 NVIR = NVIR + 1
      IDV(NVIR) = IDENT
      TV(NVIR) = TK
      BV(NVIR) = PROP1
      GO TO 020
C                               SOUND VELOCITY
  700 NWS = NWS + 1
      IDWS(NWS) = IDENT
      TWS(NWS) = TK
      PWS(NWS) = PROP1
      WSPT(NWS) = PROP2
      GO TO 020
```

```
C
  800 NWS = NWS + 1
      IDWS(NWS) = IDENT
      TWS(NWS) = TK
      WSPT(NWS) = - PROP1
      GO TO 020
C                             DPDT DATA
  900 NPT = NPT + 1
      IDPT(NPT) = IDENT
      TPT(NPT) = TK
      DPT(NPT) = PROP1
      DPDTX(NPT) = PROP2
      WDPT(NPT) = 0.0
      GO TO 020
C                             SATURATION BOUNDARY
 1000 ITRP = TTRP
      ITCR = TC
C
C************************************************************
C
C   R134A
C************************************************************
C
C     DO 1001 I = 100, 160, 20
C     TSAT = I
C     NSAT = NSAT + 1
C     PS(NSAT) = PSATF(TSAT)
C     DSL(NSAT) = DSATL(TSAT)
C     DSV(NSAT) = DSATV(TSAT)
C     TS(NSAT) = TSAT
C     WSL(NSAT) = 0.0
C     WSV(NSAT) = 0.0
C1001 CONTINUE
C
C************************************************************
************
C
C      DO 1010 I = ITRP, ITCR, 4
      DO 1010 I = ITRP, ITCR, 2
      TSAT = I
      NSAT = NSAT + 1
      PS(NSAT) = PSATF(TSAT)
      DSL(NSAT) = DSATL(TSAT)
      DSV(NSAT) = DSATV(TSAT)
      TS(NSAT) = TSAT
C      READ(25,*)WSL(NSAT),WSV(NSAT)
 1010 CONTINUE
C
```

```
C                 IF NO SATURATION DATA, SET NSAT = 0
C     NSAT=0
C
      JPVT = NPVT
      JCV = NCV
      JCP = NCP
      JDP = NDP
      JSAT = NSAT
      JVIR = NVIR
      JWS = NWS
      JPT = NPT
C
 1100 FORMAT(4F10.0)
      RETURN
      END
C
C
C
C
     SUBROUTINE FITPVT(FPVT)
C************************************************************
*******************
C
C     THIS SUBROUTINE FITS THE PVT DATA
C
C************************************************************
*******************
     IMPLICIT DOUBLE PRECISION (A-H,P-Z)
     COMMON /PVTD/
     NPVT,ID(2000),WT(2000),T(2000),P(2000),D(2000)
     COMMON/TERM/FACT(600)
     COMMON/DAT/MOBS,NCO,NMAX,NCON
     COMMON/CONST/TC,PC,DC,R
     DIMENSION PCAL(2000)
     DO 20 NP=1,NPVT
C         READ(25,*)WT(NP)
         CALL WEIGHT(D(NP),T(NP),DPD0,DPT0)
         SIGP=0.001*P(NP)
         SIGT=0.002*DPT0
         SIGD=0.001*D(NP)*DPD0
         WT(NP)=FPVT/(SIGP**2+SIGT**2+SIGD**2)**0.5
         IF (D(NP).GT.2.0.AND.D(NP).LT.10.0)
         WT(NP)=2.0*WT(NP)
         IF (ID(NP).LT.0) WT(NP)=0
         IF (WT(NP).GT.0)  MOBS = MOBS+1
     RES=P(NP)-(R*D(NP)*T(NP))
     CALL PRESSB(T(NP),PCAL(NP),D(NP),0)
C        IF (D(NP).GT.1.0.AND.D(NP).LT.8.0)
```

```
          WT(NP)=WT(NP)*FPVT
C          IF (P(NP).GT.0.5.AND.D(NP).LT.6.0)
          WT(NP)=WT(NP)*FPVT
C          IF (D(NP).GT.0.5.AND.ID(NP).EQ.7)
          WT(NP)=WT(NP)*FPVT
          DO 10 I=1,NCO
          FACT(I)=FACT(I)*WT(NP)
10     CONTINUE
          RES=RES*WT(NP)
          CALL FIT(RES,WT(NP))
20    CONTINUE
      RETURN
      END
C
C
C
C

      SUBROUTINE SATF(TS,PS,DSL,DSV)
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      PARAMETER (NC = 32, NF = NC + 1, NB = 5)
C * * * *  * * * *  * * * *  * * * *  * * * *  * * * *  * *
C
C
C      PURPOSE --- THIS ROUTINE CALCULATES THE SATURATION
C                  PRESSURE AND COEXISTING DENSITIES FROM
C                  AN EQUATION OF STATE.
C
C      VERSION 2.0  5/20/82
C
C      CODED BY -- J. F. ELY
C                  THERMOPHYSICAL PROPERTIES DIVISION
C                  NATIONAL BUREAU OF STANDARDS
C                  BOULDER, COLORADO  80303
C
C * * * *  * * * *  * * * *  * * * *  * * * *  * * * *  * *
C
C      COMMON /REFDAT/ R, PC, DC, TC, ZC, A(NC), GAMMA,
C      PTRP, DTRP, TTRP
C                , CMW
C
       COMMON /REFDAT/ZC, PTRP, DTRP, TTRP,CMW
       COMMON/CONST/TC,PC,DC,R
       LOGICAL ENTER
       DATA TOL, FTOL, ENTER / 1.0E-4, 1.0E-6, .FALSE. /
C
       ZC = PC / (R * TC * DC)
       IF(TS.LT.TC) GO TO 005
```

```
       PS = PC
       DSL = DC
       DSV = DC
       RETURN
C
  005 IF (ENTER) GO TO 010
      ENTER = .TRUE.
      BV = LOG(PTRP/PC) / (1.0/TTRP - 1.0/TC)
      AV = LOG(PC) - BV / TC
C
C          INITIAL GUESS AT THE VAPOR PRESSURE
C
  010 PS = EXP(AV+BV/TS)
C
C          INITAL GUESS AT THE VAPOR DENSITY
C
      DV = PS / (R*TS)
C
C          INITIAL GUESS AT LIQUID DENSITY
C
      TR = TS /TC
      EPS = (1.0-TR)**(2.0/7.0)
      DL = DC / ZC**EPS
      IF (DL.GT.DTRP) DL = DTRP
C
C                        IMPROVE VAPOR GUESS NEAR CRITICAL
C
      IF (TR.LT.0.85) GO TO 015
      DV = DL - 3.75 * DC * (1.0-TR)**0.333
C
C       NEWTON-RAPHSON ITERATION FOR DENSITIES
C
  015 DO 100 J = 1, 25
  020 CALL PVTF(PL,DL,TS,DPDL,D2PDD2,GL)
  030 IF (DPDL.GT.0.0.AND.PL.GT.0.0D0) GO TO 040
      DL = 1.02 * DL
      GO TO 020
  040 CALL PVTF(PV,DV,TS,DPDV,D2PDD2,GV)
      IF (DPDV.GT.0.0D0) GO TO 060
      DV = 0.98 * DV
      GO TO 040
  060 F1 = GL -GV
      F2 = PL - PV
      F2L = DPDL
      F2V = - DPDV
      F1L = F2L / DL
      F1V = F2V / DV
C     WRITE(6,300) J, DL, F1L, F2L, DV, F1V, F2V, F1, F2
```

```
      DENOM = F1L * F2V - F2L * F1V
      IF (ABS(DENOM).LE.1.0E-10) GO TO 120
      DDL = -(F1*F2V-F2*F1V) / DENOM
      DDV = -(F1+DDL*F1L) / F1V
      DL = DL + DDL
      IF(DL.LT.DC) DL = DC
      DVS = DV
      DV = DV + DDV
      IF(DV.GT.DC) DV = DC
      IF(DV.LE.0.0D0) DV=DVS/2.0
C     WRITE(6,310) J, DDL, DDV, DENOM
      IF (ABS(DDL/DL).LT.TOL .AND. ABS(DDV/DV).LE.TOL) GO TO
      110
      FNORM = F1*F1 + F2*F2
      IF(TR.LT.0.99 .AND. FNORM.LE.FTOL) GO TO 110
  100 CONTINUE
  110 PS = PV
      DSL = DL
      DSV = DV
C     WRITE(6,330) TS, PV, DL, DV, FNORM, GL
      RETURN
  120 WRITE(6,340) DENOM
      GO TO 110
  300 FORMAT(I3,8G10.4)
  310 FORMAT(I3,2F12.8,G13.6)
  330 FORMAT(F8.2,G13.6,2F10.6,2G13.6)
  340 FORMAT('DENOM IS TOO SMALL',G13.6)
      END
C
C
      SUBROUTINE SATCMP
C************************************************************
C
C
C    THIS SUBROUTINE PERFORMS COMPARISONS ALONG THE
C    SATURATION BOUNDARY
C
C************************************************************
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      PARAMETER (MSAT = 150)
      COMMON /SATD/ NSAT, WSL(MSAT), WSV(MSAT), TS(MSAT),
      SATP(MSAT,3)
C
      DIMENSION AAD(3), BIAS(3), RMS(3), PCT(3), PCAL(3)
C
      WRITE(3,200)
      DO 010 K = 1, 3
      AAD(K) = 0.0
```

```
      BIAS(K) = 0.0
  010 RMS(K) = 0.0
      DO 100 J = 1, NSAT
      CALL SATF(TS(J),PCAL(1),PCAL(2),PCAL(3))
      DO 040 K = 1, 3
      PCT(K) = PCAL(K) - SATP(J,K)
      PCT(K) = 100.0 * PCT(K) / SATP(J,K)
  020 AAD(K) = AAD(K) + ABS(PCT(K))
      BIAS(K) = BIAS(K) + PCT(K)
      RMS(K) = RMS(K) + PCT(K) * PCT(K)
  040 CONTINUE
      WRITE(25,*)WSL(J),WSV(J)
      WRITE(3,220)
      TS(J),(PCAL(K),PCT(K),K=1,3),WSL(J),WSV(J)
  100 CONTINUE
C
      DO 120 K = 1, 3
      AAD(K) = AAD(K) / NSAT
      BIAS(K) = BIAS(K) / NSAT
  120 RMS(K) = SQRT(RMS(K)/NSAT-BIAS(K)*BIAS(K))
      WRITE(3,240) NSAT, AAD, BIAS, RMS
C
  200 FORMAT('1      T,K                P , CAL          %
      DL,CAL
     *   %           DV, CAL            %            WL
      WV')
C 220 FORMAT(9E13.5)
  220 FORMAT(F13.3,  F13.5 ,F13.2,2X, F13.5,F13.2,2X,  F13.5
,
     *  F13.2,2D13.2)
C  220 FORMAT(D13.3,  1E13.5 ,D13.2,2X, D13.5,D13.2,2X,
       1E13.5 ,
C     *  D13.2,2D13.2)
  240 FORMAT('0NSAT =',I4/'
      AAD:',14X,F8.3,8X,F8.3,15X,F8.3/
     *' BIAS:',14X,F8.3,8X,F8.3,15X,F8.3/'
      RMS:',14X,F8.3,8X,F8.3,
     *15X,F8.3)
      RETURN
      END
C

      SUBROUTINE PVTF(PO,DO,TO,DPDO,DPDDO,GO)
C
C * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C
C
C PURPOSE --- THIS ROUTINE CALCULATES THE  PRESSURE, ITS
```

```
C DENSITY DERIVATIVE AND THE GIBBS ENERGY RELATIVE TO THE
C IDEAL GAS AT UNIT PRESSURE OF SAUL-WAGNER 38 TERM WATER
C                   BWR EQUATION
C
C
C      VERSION  1.1  -- 8/10/92
C
C      CODED BY -- J. F. ELY
C                   CHEMICAL ENGINEERING DEPARTMENT
C                   COLORADO SCHOOL OF MINES
C                   GOLDEN, COLORADO  80401
C
C * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C
          IMPLICIT DOUBLE PRECISION (A-H,O-Z)
        COMMON/TERM/FACT(600)
        COMMON/IDNO/PIDN(600,4)
        COMMON/COE/COEF(600)
        COMMON/CONST/TC,PC,DC,R
        COMMON/OLDAT/BCOEF(32),BD(32,4)
        COMMON/DAT/MOBS,NCO,NMAX,NCON
C
        T=TC/TO
        D=DO/DC
        SUM4=0
        SUM5=0
        SUM6=0
        CALL PRESSB(TO,PO,DO,1)
        CALL AR(PO,DO,TO,IDD)
        DO 11 I=1,NCO
        IF (COEF(I).EQ.0) GOTO 11
         PIT=PIDN(I,2)
         PID=PIDN(I,3)
         PIDRO=PIDN(I,4)
        CALL DIFF5(D,T,PIT,PID,PIDRO,DPD,DPDD,DPT)
        SUM4=SUM4+DPD*COEF(I)
        SUM5=SUM5+DPDD*COEF(I)
        SUM6=SUM6+FACT(I)*COEF(I)/(R*TO)
11      CONTINUE
        Z0=PO/(DO*R*TO)
        DPDO=(SUM5*R*TO+R*TO)
        DPDDO=SUM2/DC**2
        GO=SUM6+LOG(DO*R*TO)+Z0-1
        GO=GO*R*TO
           RETURN
        END

C
```

```
C
      SUBROUTINE DIFF5(D,T,I,J,K,DIFF1,DIFF2,DPT)
      IMPLICIT DOUBLE PRECISION (A-Z)
      DIFF1=(D**(J-1)*J*T**I-D**(J+K-1)*K*T**I)/EXP(D**K)
      IF (K.EQ.0) DIFF1=DIFF1*EXP(D**K)
         DIFF2=2*d*(d**(-1 + j)*j*t**i/EXP(d**k) -
      *       d**(-1 + j + k)*k*t**i/EXP(d**k)) +
      *       d**2*(d**(-2 + j)*(-1 + j)*j*t**i/EXP(d**k) -
      *       d**(-2 + j + k)*j*k*t**i/EXP(d**k) +
      *       d**(-2 + j + 2*k)*k**2*t**i/EXP(d**k) -
      *       d**(-2 + j + k)*k*(-1 + j + k)*t**i/EXP(d**k))
      IF (K.EQ.0) DIFF2=DIFF2*EXP(D**K)
         dpt=2*d*(d**(-1 + j)*i*j*t**(-1 + i)/Exp(d**k) -
      -      d**(-1 + j + k)*i*k*t**(-1 + i)/Exp(d**k)) +
      -  d**2*(d**(-2 + j)*i*(-1 + j)*j*t**(-1 + i)/Exp(d**k)
-
      -      d**(-2 + j + k)*i*j*k*t**(-1 + i)/Exp(d**k) +
      -      d**(-2 + j + 2*k)*i*k**2*t**(-1 + i)/Exp(d**k) -
      -      d**(-2 + j + k)*i*k*(-1 + j + k)*t**(-1 +
             i)/Exp(d**k))
      IF (K.EQ.0) DPT=DPT*EXP(D**K)
      RETURN
      END
C
C
       SUBROUTINE AR(PP,DD,TT,IDD)
       IMPLICIT DOUBLE PRECISION (A-H,P-Z)
       COMMON/CONST/TC,PC,DC,R
       COMMON/IDNO/PIDN(600,4)
       COMMON/DAT/MOBS,NCO,NMAX,NCON
       COMMON/TERM/FACT(600)
       D=DD/DC
       T=TC/TT
      DO 102 I=1,NCO
      PIT=PIDN(I,2)
      PID=PIDN(I,3)
      PIDR=PIDN(I,4)
      IF (PIDR.EQ.0) GOTO 105
      FACT(I)=T**PIT*D**PID*EXP(-D**PIDR)*TT*R
      GOTO 102
105      FACT(I)=T**PIT*D**PID*TT*R
102   CONTINUE
      RETURN
      END
C
C
       SUBROUTINE INIT()
       IMPLICIT DOUBLE PRECISION (A-H,P-Z)
```

```
        COMMON/IDNO/PIDN(600,4)
        COMMON/DAT/MOBS,NCO,NMAX,NCON
       KK=1
       DO 10 I=1,15
       DO 20 J=1,20
       DO 30 K=0,2,2
         PIDN(KK,1)=KK
         PIDN(KK,4)=K
         PIDN(KK,3)=J
         PIDN(KK,2)=I
         KK=KK+1
30     CONTINUE
20     CONTINUE
10     CONTINUE
       RETURN
       END
C
        SUBROUTINE INIT1()
        IMPLICIT DOUBLE PRECISION (A-H,P-Z)
        COMMON/IDNO/PIDN(600,4)
        COMMON/DAT/MOBS,NCO,NMAX,NCON
       OPEN(4,FILE='TEST1.DAT',STATUS='OLD')
       DO 10 I=1,NCO
       READ(4,*)PI1,PI4,PI3,PI2
       PIDN(I,1)=I
       PIDN(I,2)=PI2
       PIDN(I,3)=PI3
       PIDN(I,4)=PI4
10     CONTINUE
       RETURN
       END
C
       SUBROUTINE PRESSB(TEM,PCA,DEN,IND)
       IMPLICIT DOUBLE PRECISION (A-H,P-Z)
       COMMON/COE/COEF(600)
       COMMON/TERM/FACT(600)
       COMMON/DAT/MOBS,NCO,NMAX,NCON
       COMMON/CONST/TC,PC,ROC,R
       COMMON/IDNO/PIDN(600,4)
       D=DEN/ROC
       T=TC/TEM
       DO 10 KK=1,NCO
        PI=PIDN(KK,2)
        PJ=PIDN(KK,3)
        PK=PIDN(KK,4)
        IF (PK.EQ.0) GOTO 5
         FACT(KK)=(PJ*D**(PJ-1)*T**PI -
         PK*D**(PJ+PK-1)*T**PI)/EXP(D**PK)
```

```
        GOTO 11
5       FACT(KK)=PJ*D**(PJ-1)*T**PI
11      FACT(KK)=DEN*TEM*D*R*FACT(KK)
10    CONTINUE
      IF (IND.EQ.0) RETURN
      SUM=0
      DO 50 II=1,NCO
50    SUM=SUM+FACT(II)*COEF(II)
      PCA=DEN*TEM*R+SUM
      RETURN
      END
C
       SUBROUTINE DPDD(DPD,DD,TT,IDD)
       IMPLICIT DOUBLE PRECISION (A-H,P-Z)
       COMMON/CONST/TC,PC,DC,R
       COMMON/COE/COEF(600)
       COMMON/IDNO/PIDN(600,4)
       COMMON/DAT/MOBS,NCO,NMAX,NCON
       COMMON/TERM/FACT(600)
       COMMON/OLDAT/BCOEF(32),BD(32,4)
      D=DD/DC
      T=TC/TT
      SUM5=0
      IF (IDD.EQ.2) GOTO 12
      DO 11 I=1,NCO
      IF (COEF(I).EQ.0.AND.IDD.EQ.1) GOTO 11
       PIT=PIDN(I,2)
       PID=PIDN(I,3)
       PIDRO=PIDN(I,4)
      CALL DIFFP(D,T,PIT,PID,PIDRO,DPDO)
      IF (IDD.EQ.0) FACT(I)=DPDO
      IF (IDD.NE.0)SUM5=SUM5+DPDO*COEF(I)
11    CONTINUE
      GOTO 20
12    DO 15 I=1,19
       PIT=BD(I,2)
       PID=BD(I,3)
       PIDRO=BD(I,4)
       CALL DIFFP(D,T,PIT,PID,PIDRO,DPDO)
      SUM5=SUM5+DPDO*BCOEF(I)
15    CONTINUE
20    IF (IDD.NE.0) DPD=(SUM5*R*TT+R*TT)
      RETURN
      END
C
C
       FUNCTION RHOF(P,DEN,T)
       IMPLICIT DOUBLE PRECISION (A-H,O-Z)
```

```
C      PARAMETER (NC = 32, NF = NC + 1 )
C
C * * * *  * * * *  * * * *  * * * *  * * * *  * * * *  * *
C
C
C PURPOSE -- THIS ROUTINE CALCULATES THE DENSITY OF A FLUID
C   AT  T AND P GIVEN AN INITIAL GUESS IN FOP.  ON EXIT,
C   IT RETURNS THE FUGACITY COEFFICIENT IN FOP.  IT
C   REQUIRES A ROUTINE 'PVTF' WHICH CALCULATES P,
C                  DPDD, AND GR = G(T,P)-G*(T,1)
C
C       CODED BY--J. F. ELY
C                  THERMOPHYSICAL PROPERTIES DIVISION
C                  NATIONAL ENGINEERING LABORATORY
C                  NATIONAL BUREAU OF STANDARDS
C                  BOULDER, COLORADO 80302
C
C       VERSION 2.0 -- 5/23/82
C
C * * * *  * * * *  * * * *  * * * *  * * * *  * * * *  * *

     COMMON /REFDAT/ZC, PTRP, DTRP, TTRP,CMW
     COMMON/CONST/TC,PC,DC,R
      LOGICAL SAT
      DATA TOLERD, TOLERP, TOLERB, TLOWP/ 2 * 1.0D-7,
      1.0D-6, 1.0D-10/
C
C      ESTABLISH BOUNDS AND START NEWTON-RAPHSON
C
      SAT = .FALSE.
      D = DEN
      IF (D.GT.0.0D0) GO TO 030
  005 IF (T.LT.TC) GO TO 010
      D = 2.0D0 * DC
      IF (P.LT.PC) D = P / (R*T)
      GO TO 030
  010 SAT = .TRUE.
      CALL SATF(T,PS,DSL,DSV)
      D = DSV
  015 IF (P.LT.PS) GO TO 030
      D = (2.0D0*DSL + DTRP) / 3.0D0
C
C            ESTABLISH BOUNDS AND START NEWTON RAPHSON
C
  030 DLO=0.0
      DHI = 1.25D0 * DTRP
      D1 = D
      DMAX = DHI
```

```
      DO 100 LAP = 1, 20
      CALL PVTF(PX,D,T,DPDD,D2PDD2,GR)
C
C         IF DPDD IS ZERO OR NEGATIVE, TRY BISECTION
C
      IF(DPDD.LE.1.0D-3) GO TO 120
      IF (PX.LE.0.0) GO TO 120
      DP=P-PX
      DD=DP/DPDD
C             SAVE DENSITY FOR POSSIBLE BISECTION
C
      IF (DP) 040,300,060
  040 DHI=D
      GO TO 080
  060 DLO=D
  080 DN=D+DD
C                        KEEP D WITHIN BOUNDS OR GO TO
BISECTION
C
      IF (DN.LT.0.0D0 .OR. DN.GT.DMAX) GO TO 120
      D=DN
      IF(LAP.EQ.1) GO TO 100
      IF(ABS(DP/P).LE.TOLERP .AND. ABS(DD/D).LE.TOLERD) GO
      TO 300
      IF(ABS(DP).LE.TLOWP .AND. ABS(DD/D).LE.TOLERD) GO TO
       300
  100 CONTINUE
C         NEWTON-RAPHSON FAILURE.  TRY  BISECTION
C
  120 IF (T.GT.TC) GO TO 160
C
C   SUB-CRITICAL.  MAKE SURE THAT WE HAVE THE
C                        PROPER BOUNDS ON THE DENSITY.
C
  130 IF (.NOT.SAT) GO TO 010
C
      IF(D1.LT.DC) GO TO 140
      DLO = DSL
      IF(DHI.LE.DSL) DHI=DMAX
      GO TO 160
C
  140 IF(DLO.GE.DSV) DLO=0.0D0
      DHI = DSV
C
C                        START THE BISECTION
  160 D=0.50D0*(DLO+DHI)
      CALL PVTF(PX,D,T,DPDD,D2PDD2,GR)
      DP=PX-P
```

```
      IF(DP) 200,300,220
  200 DLO=D
      GO TO 240
  220 DHI=D
C
  240 IF(ABS(DP/P) .LE. TOLERB) GO TO 300
C
      IF(ABS(DLO/DHI-1.0D0).GT.TOLERD) GO TO 160
C
C                      BISECTION FAILED.  GIVE UP
  260 WRITE (5,400) T, P
C                      CONVERGENCE ! ! !
  300 RHOF=D
      RETURN
  400 FORMAT(' RHOF FAILED AT T =',F9.3,' P =',G14.7,' DP
=',G14.7)
      END
C
C
      SUBROUTINE PVTCMP
C************************************************************
C
C
C    THIS SUBROUTINE PERFORMS THE PVT COMPARISONS.
C
C************************************************************
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      PARAMETER (MPVT = 2000)
C
      COMMON /PVTD/ NPVT, ID(MPVT), WT(MPVT), T(MPVT),
      P(MPVT),
     *              D(MPVT)
C
      COMMON /REFN/ NREF, IDREF(100)
      COMMON /REFS/ REFER(100)
      CHARACTER*80 REFER
C
      PMIN = 1000000.
      PMAX = 0.
      TMIN = 1000000.
      TMAX = 0.
      DMIN = 1000000.
      DMAX = 0.
      IDN = -1
      ILT = 0
      AADT = 0.0
      BIAST = 0.0
      RMST = 0.0
```

```
    NT = 0
    BIASPT = 0.0
    AADPT = 0.0
    RMSPT = 0.0
    ITAB = 5
    DO 100 I=1,NPVT
    ID(I) = IABS(ID(I))
    IF(ID(I).EQ.IDN) GO TO 080
    IF (IDN.LT.0) GO TO 040
020 AADT = AADT + AAD
    BIAST = BIAST + BIAS
    RMST = RMST + RMS
    BIASPT = BIASPT + BIASP
    AADPT = AADPT + AADP
    RMSPT = RMSPT + RMSP
    IF (NI.EQ.0) GO TO 040
    NT = NT + NI
021 AAD = AAD / NI
    BIAS = BIAS / NI
    RMS = SQRT(RMS/NI-BIAS*BIAS)
    WRITE(3,200) IDREF(IR),NI,AAD,BIAS,RMS
    BIASP = BIASP / NI
    AADP = AADP / NI
    RMSP = SQRT(RMSP/NI-BIASP*BIASP)
    WRITE(3,220) IDREF(IR), NI, AADP, BIASP, RMSP
    WRITE(3,225) TMIN, TMAX, PMIN, PMAX, DMIN, DMAX
    TMIN = 1000000.
    TMAX = 0.
    PMIN = 1000000.
    PMAX = 0.
    DMIN = 1000000.
    DMAX = 0.
    IF (I.EQ.NPVT) GO TO 100
040 AAD = 0.0
    ITAB = ITAB + 1
    BIAS = 0.0
    RMS = 0.0
    BIASP = 0.0
    AADP = 0.0
    RMSP = 0.0
    NI = 0
    IDN = ID(I)
    DO 050 K = 1, NREF
    IF (IDN.NE.IDREF(K)) GO TO 050
    IR = K
    GO TO 060
050 CONTINUE
060 ILT = 0
```

```
      WRITE(3,239) ITAB
      WRITE(3,240) IDREF(IR), REFER(IR)
  080 ILT = ILT + 1
      IF (ILT.GE.55) GO TO 060
      CALL PRESSB(T(I),PCAL,D(I),1)
      DCAL = RHOF(P(I),D(I),T(I))
      DDIF = -(DCAL-D(I))*100./D(I)
      PDIF = -(PCAL-P(I))*100./P(I)
      CALL DPDD(DPD,D(I),T(I),1)
      DT1 = (DCAL-D(I)) / DPD
      IF (P(I).GT.PMAX) PMAX = P(I)
      IF (P(I).LT.PMIN) PMIN = P(I)
      IF (T(I).GT.TMAX) TMAX = T(I)
      IF (T(I).LT.TMIN) TMIN = T(I)
      IF (D(I).GT.DMAX) DMAX = D(I)
      IF (D(I).LT.DMIN) DMIN = D(I)
      WRITE(25,*)WT(I)
      WRITE(3,260) T(I), PCAL, P(I), PDIF, DCAL, D(I), DDIF
     *        , WT(I)  , DT1
C     IF (WT(I).LE.0.0) GO TO 100
      NI = NI + 1
      AAD = AAD + ABS(DDIF)
      BIAS = BIAS + DDIF
      RMS = RMS + DDIF * DDIF
      BIASP = BIASP + PDIF
      AADP = AADP + ABS(PDIF)
      RMSP = RMSP + PDIF * PDIF
      IF (I.EQ.NPVT) GO TO 020
  100 CONTINUE
      AADT = AADT / NT
      BIAST = BIAST / NT
      RMST = SQRT(RMST/NT-BIAST*BIAST)
      WRITE(3,280) NT, AADT, BIAST, RMST
      AADPT = AADPT / NT
      BIASPT = BIASPT / NT
      RMSPT = SQRT(RMSPT/NT - BIASPT*BIASPT)
      WRITE(3,300) NT, AADPT, BIASPT, RMSPT
C
  200 FORMAT('0[',I3,'] N = ',I4,'  AAD = ',F7.3,'  BIAS =
     ',F7.3,
     * ' RMS =',F7.3)
  220 FORMAT(' [',I3,'] N = ',I4,'  AADP= ',F7.3,'  BIASP=
     ',F7.3,
     * ' RMSP=',F7.3)
  225 FORMAT(' RANGES(T,P,D): ',6F8.3)
  239 FORMAT('1',6X,'TABLE',I3,' EQUATION OF STATE VS.
      EXPERIMENTAL PVT
     * DATA (CONTINUED)')
```

```
  240 FORMAT('0  REF',I3,2X,A//'        T,K         P, CAL
  P, EXP
     1  P, %','          D, CALC      D, EXP       D, %
     *   WT          DT')
  260
FORMAT(F12.4,2F12.3,F12.3,2F12.4,F12.3,1PE13.5,1PE13.5)
  280 FORMAT('0OVERALL PVT RESULTS: N=',I5,' AAD =',F7.3,'
BIAS =',
     1 F7.3,' RMS =', F7.3)
  300 FORMAT(22X,'N=',I5,' AADP= ',F6.3,' BIASP= ',F6.3,'
RMSP =',F6.3)
      RETURN
      END
C
C
C
C
      SUBROUTINE FITVIR(FVIR)
C************************************************************
C
C
C   THIS SUBROUTINE FITS THE VIRIAL DATA
C
C************************************************************
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      PARAMETER (MVIR = 100)
C      PARAMETER (NC = 32, NF = NC + 1 )
C
     COMMON /VIRIAL/ NVIR, IDV(MVIR), WV(MVIR), TV(MVIR),
     BV(MVIR)
     COMMON/COE/COEF(600)
     COMMON/TERM/FACT(600)
     COMMON/DAT/MOBS,NCO,NMAX,NCON
     COMMON/CONST/TC,PC,ROC,R
     COMMON/IDNO/PIDN(600,4)
C
     DO 10 II=1,NVIR
     DO 20 KK=1,NCO
      PI=PIDN(KK,2)
      PJ=PIDN(KK,3)
      PK=PIDN(KK,4)
     FACT(KK)=0
     if (Pj.gt.1) goto 20
     FACT(KK)=(PJ*(TC/TV(II))**PI)
20   CONTINUE
     TVIR=TC/TV(II)
     CALL VIRDIF(TVIR,DBDT)
        SB = 0.05 * BV(II)
```

```
          ST = 0.005 * DBDT
          WV(II) = FVIR / SQRT(SB*SB + ST*ST)
          IF (TV(II).LT.276.) WV(II) = .5*WV(II)
          IF (IDV(II).LE.0) WV(II) = 0.0
          RES=BV(II)
          DO 11 IJ=1,NCO
          FACT(IJ)=FACT(IJ)*WV(II)
11        CONTINUE
          RES = RES*WV(II)*ROC
          CALL FIT(RES,WV(II))
       IF (WV(II).GT.O) MOBS=MOBS+1
10     CONTINUE
       RETURN
       END
C
C
        SUBROUTINE VIRCMP
C***********************************************************
C
C
C    THIS SUBROUTINE PERFORMS THE SECOND VIRIAL COMPARISONS.
C
C***********************************************************
C
        IMPLICIT DOUBLE PRECISION (A-H,O-Z)
        PARAMETER (MVIR = 100)
        PARAMETER (NC = 32, NF = NC + 1 )
C
        COMMON /VIRIAL/ NVIR, IDV(MVIR), WV(MVIR), TV(MVIR),
        BV(MVIR)
C
       COMMON/COE/COEF(600)
       COMMON/TERM/FACT(600)
       COMMON/DAT/MOBS,NCO,NMAX,NCON
       COMMON/CONST/TC,PC,ROC,R
       COMMON/IDNO/PIDN(600,4)
C
        WRITE(3,200)
        NV = 0
        AAD = 0.0
        BIAS = 0.0
        DO 100 J = 1, NVIR
        B=0
        DO 1 JJ=1,NCO
        PI=PIDN(JJ,2)
        PJ1=PIDN(JJ,3)
        PK=PIDN(JJ,4)
        FACT(JJ)=0
```

```
      IF (PJ1.GT.1) GOTO 1
      FACT(JJ)=(PJ1*(TC/TV(J))**PI)/ROC
      B=B+FACT(JJ)*COEF(JJ)
1     CONTINUE
C     B = (G(1)*T + G(2)*TS + G(3) + G(4)/T + G(5)/T**2) /
     (R*T)
      DIF = -1000.0*(B - BV(J))
      IF (IDV(J).LE.0) GO TO 090
      AAD = AAD + ABS(DIF)
      NV = NV + 1
      BIAS = BIAS + DIF
  090 WRITE(3,210) IDV(J), TV(J), B, BV(J), DIF, WV(J)
      WRITE(25,*)WV(J)
  100 CONTINUE
      AAD = AAD / NV
      BIAS = BIAS / NV
      WRITE(3,220) NV, AAD, BIAS
C
  200 FORMAT('1SECOND VIRIAL COMPARISONS'/  '0 ID    T,K
      B(CALC)
     *B(EXP)    DIF,CC      WT')
  210 FORMAT(I5,F9.3,2F10.6,F8.3,F10.1)
  220 FORMAT('0N = ',I4,' AAD =',F8.3, ' BIAS =',F8.3)
      RETURN
      END

C
C
     SUBROUTINE WEIGHT(DO,TO,DPD0,DPT0)
     IMPLICIT DOUBLE PRECISION (A-H,P-Z)
     COMMON/CONST/TC,PC,ROC,R
     COMMON/OLDAT/BCOEF(32),BD(32,4)
     SUM4=0
     SUM5=0
     SUM6=0
     T=TC/TO
     D=DO/ROC
     DO 10 I=1,19
     PIT=BD(I,2)
     PID=BD(I,3)
     PIDRO=BD(I,4)
     CALL DIFF5(D,T,PIT,PID,PIDRO,DPD,DPDD,DPT)
     SUM4=SUM4+DPD*BCOEF(I)
     SUM5=SUM5+DPDD*BCOEF(I)
     SUM6=SUM6+DPT*BCOEF(I)
10   CONTINUE
     SUM4=SUM4*TO*DO*D*R
     PP=SUM4+DO*TO*R
```

```
     DPDO=(SUM5*R*TO+R*TO)
     DPTO=PP/TO-SUM6*DO*R*D*T
     RETURN
     END
C
     SUBROUTINE JGE
     IMPLICIT DOUBLE PRECISION (A-H,P-Z)
     COMMON/OLDAT/BCOEF(32),BD(32,4)
     OPEN(10,FILE='COEF.DAT',STATUS='OLD')
     OPEN(8,FILE='OLDTERM.DAT',STATUS='OLD')
     DO 10 I=1,19
     READ(10,*)BCOEF(I)
     READ(8,*)BD(I,1),BD(I,4),BD(I,3),BD(I,2)
10   CONTINUE
     RETURN
     END
C
C
      SUBROUTINE FITCV(FCV)
C************************************************************
C
C
C     THIS SUBROUTINE FITS THE CV DATA.
C
C************************************************************
C
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      PARAMETER (MCV = 200)
C
     COMMON/COE/COEF(600)
     COMMON/TERM/FACT(600)
     COMMON/DAT/MOBS,NCO,NMAX,NCON
     COMMON/CONST/TC,PC,DC,R
     COMMON/IDNO/PIDN(600,4)
     COMMON /REFDAT/ZC, PTRP, DTRP, TTRP,CMW
     COMMON /CVDAT/ NCV, IDCV(MCV), WCV(MCV), TCV(MCV),
     PCV(MCV),
     *               DCV(MCV), CV(MCV)
C
      DIMENSION F(600),A(200)
      DATA RJ, EPS /8.31434, 0.01/
C
      DO 140 I = 1, NCV
      IF(CV(I).LE.0.0) GO TO 050
      CALL IDEAL(TCV(I),CVO,SO)
      DELCV = 0.01 * (CV(I) - CVO)
      TR = TCV(I) / TC
      DR = DCV(I) / DC
```

```
      CALL DP2DT2(DP2,DCV(I),TCV(I),2)
      SIGC = 0.02 * CV(I)
      SIGD = 0.001 * TCV(I) * DP2 / DCV(I)
      SIGT = 0.5 * CV(I) / TCV(I)
      WCV(I) = 100*FCV / SQRT(SIGC**4 + SIGD**2 + SIGT**2) *
      TR**2
      IF(ABS(TR-1.0).LT.0.025 .AND. ABS(DR-1.0).LT.0.1)
     * WCV(I) = 0.0
      IF(IDCV(I).GT.0) GO TO 020
      IDCV(I) = -IDCV(I)
      WCV(I) = 0.0
      GO TO 140
  020 CALL CVR(PP,DCV(I),TCV(I),0)
      DO 030 J = 1, NCO
  030 F(J) = FACT(J)
      CALL CVR(PP,0.0D0,TCV(I),0)
      DO 040 J = 1, NCO
  040 FACT(J) =  (FACT(J) - F(J))
      RES=DELCV
      IF (WCV(I).GT.0) MOBS=MOBS+1
      DO 11 IJ=1,NCO
       FACT(IJ)=FACT(IJ)*WCV(I)
11    CONTINUE
      RES = RES*WCV(I)
      GO TO 120
C                            C(SAT) DATA
  050 T1 = TCV(I) - EPS
      D1 = DSATL(T1)
C      CALL SR(PP,D1,T1,0)
      DO 060 J = 1, NC
  060 F(J) = A(J)
C      CALL SR(PP,0.0D0,T1,0)
      DO 070 J = 1, NC
  070 F(J) = F(J) -A(J)
      T2 = T1 + 2.0*EPS
      D2 = DSATL(T2)
C     CALL SR(PP,D2,T2,0)
      CALL IDEAL(T1,CV1,S1)
      CALL IDEAL(T2,CV2,S2)
      TERM = - TCV(I) / (2.0 * EPS)
      Y = ABS(CV(I)) + (RJ*LOG(D1*T1/(D2*T2)) - S1 + S2) *
      TERM
      Y = 0.01 * Y
      TR = TCV(I) / TC
      WCV(I) = FCV / SQRT(0.02 * Y * TR**3)
      IF(IDCV(I).GT.0) GO TO 080
      IDCV(I) = -IDCV(I)
      WCV(I) = 0.0
```

```
  080 Y = Y * WCV(I)
      DO 090 J = 1, NC
  090 F(J) = F(J) - A(J)
C     CALL SR(PP,0.0D0,T2,0)
      DO 100 J = 1, NC
  100 F(J) = WCV(I) * (F(J) + A(J)) * TERM
  120 CALL FIT(RES,WCV(I))
  140 CONTINUE
      RETURN
      END
C
C
      SUBROUTINE IDEAL(T,CVZ,SZ)
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
C * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C
C     PURPOSE --- THIS ROUTINE CALCULATES THE IDEAL GAS
C                 PROPERTIES OF NEON USING THE SATISTICAL
C                 MECHANICAL EXPRESSION FOR CP
C
C     REFERENCE STATES ARE:   S(298.15) = 83.520 CAL/MOL-K
C                                  H(0) = 0
C
C     CODED BY: J. F. ELY
C               THRMOPHYSICS  DISIVISON 774.03
C               NATIONAL BUREAU OF STANDARDS
C               BOULDER, CO  80303
C
C     VERSION 1.0 -  3/26/88
C
C * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C
      DATA C1, C2, C3 / 1.94006D+01, 2.58531D-01,
      -1.29665D-04 /
      DATA HRF, SRF / 0.0D0, 0.0D0 /
C
      CPZ = C1 + C2*T + C3*T**2
      SZ = C1 * LOG(T) + T * (C2 + T * C3/2.0D0)  + SRF
      CVZ = CPZ - 8.31441D0
      RETURN
      END
C
C
      SUBROUTINE CVCOMP
C
C************************************************************
C
C
```

```
C     THIS SUBROUTINE PERFORMS THE CV COMPARISONS.
C
C************************************************************
C
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      PARAMETER (MCV = 200)
C
      COMMON /CVDAT/ NCV, IDCV(MCV), WCV(MCV), TCV(MCV),
      PCV(MCV),
     *               DCV(MCV), CV(MCV)
C
C      COMMON /DERIV/ DPSDT, DDSDT
C
      WRITE(3,200)
      AAD = 0.0
      BIAS = 0.0
      RMS = 0.0
      NI = 0
      DO 100 I = 1, NCV
      IF (CV(I).LT.0) GO TO 100
      CALL PRESSB(TCV(I),PCAL,DCV(I),1)
      DCAL = RHOF(PCV(I),DCV(I),TCV(I))
      PPCT = -100.0 * (PCAL - PCV(I)) / PCV(I)
      DPCT = -100.0 * (DCAL - DCV(I)) / DCV(I)
      CVCAL = CVF(DCV(I),TCV(I))
      PCT = -100.0 * (CVCAL-CV(I)) / CV(I)
      WRITE(3,240) IDCV(I), TCV(I), PCAL, PPCT, DCAL,
     *            DPCT, CVCAL, CV(I), PCT, WCV(I)
C     IF (WCV(I).LE.0.0) GO TO 100
      AAD = AAD + ABS(PCT)
      BIAS = BIAS + PCT
      RMS = RMS + PCT*PCT
      NI = NI + 1
  100 CONTINUE
      IF (NI.EQ.0) GO TO 120
      AAD = AAD / NI
      BIAS = BIAS / NI
      RMS = SQRT(RMS/NI - BIAS*BIAS)
      WRITE(3,260) NI, AAD, BIAS, RMS
C
C                       C(SAT) COMPARISONS
C
  120 NI = 0
      AAD = 0.0
      BIAS = 0.0
      RMS = 0.0
C      WRITE(3,220)
      DO 140 I = 1, NCV
```

```
      IF (CV(I).GT.0) GO TO 140
      CV(I) = -CV(I)
      CALL DPDT(DPT,DCV(I),TCV(I),1)
      DL = DSATL(TCV(I))
      CVCAL = CVF(DL,TCV(I))
      CVCAL = CVCAL - 100.0 * (TCV(I) * DPT * DDSDT /
      (DL*DL))
      PCT = 100.0 * (CVCAL - CV(I)) / CV(I)
      WRITE(3,250) IDCV(I), TCV(I), DCV(I), CVCAL, CV(I),
      PCT, WCV(I)
      WRITE(25,*)WCV(I)
C     IF (WCV(I).LE.0.0) GO TO 140
      AAD = AAD + ABS(PCT)
      RMS = RMS + PCT * PCT
      BIAS = BIAS + PCT
      NI = NI + 1
  140 CONTINUE
      IF (NI.EQ.0) GO TO 160
      AAD = AAD / NI
      BIAS = BIAS / NI
      RMS = SQRT(RMS/NI-BIAS*BIAS)
      WRITE(3,260) NI, AAD, BIAS, RMS
  160 RETURN
  200 FORMAT('1CV(D,T) COMPARISONS'/
     *'0  ID     T,K    P,CAL       P,%       D,CAL      D,%
     *     CV,CAL     CV,EXP   %           WT')
C  220 FORMAT('1C(SAT) COMPARISONS'/
C     *'0  ID     T,K     D,M/L    CS,CAL  CS,EXP    CS,%
      WT')
  240 FORMAT(I5,2F9.3,F9.2,1X,F9.4,F9.2,1X,2F9.3,F6.2,F9.1,
     *  2F9.3)
  250 FORMAT(I5,F9.3,F9.4,2F9.3,F8.3,F8.2)
  260 FORMAT('0N = ',I6,'  AAD = ',F7.3,'  BIAS = ',F7.3,'
      RMS =',F7.3)
      END
C
C
      FUNCTION CVF(D,T)
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      CALL CVR(CVD,D,T,1)
      CALL CVR(CV0,0.0D0,T,1)
      CALL IDEAL(T,CVI,SI)
      CVF = CVI - 100.0 * (CVD-CV0)
      RETURN
      END
C
C
     SUBROUTINE CVR(PP,DD,TT,L)
```

```
      IMPLICIT DOUBLE PRECISION (A-H,P-Z)
      COMMON/COE/COEF(600)
      COMMON/TERM/FACT(600)
      COMMON/CONST/TC,PC,DC,R
      COMMON/DAT/MOBS,NCO,NMAX,NCON
      COMMON/IDNO/PIDN(600,4)
      D=DD/DC
      T=TC/TT
      DO 10 II=1,NCO
      PI=PIDN(II,2)
      PJ=PIDN(II,3)
      PK=PIDN(II,4)
      IF (PK.EQ.0) GOTO 5
      FACT(II)=D**PJ*(-1+PI)*PI*T**(-2+PI)/EXP(D**PK)
      GOTO 8
5     FACT(II)=D**PJ*(-1+PI)*PI*T**(-2+PI)
8     FACT(II)=R*T**2*FACT(II)
10    CONTINUE
      IF (L.EQ.0) RETURN
      PP=0
      DO 20 JJ=1,NCO
      IF (COEF(JJ).EQ.0) GOTO 20
      PP=PP+COEF(JJ)*FACT(JJ)
20    CONTINUE
      RETURN
      END
C
C
C
      SUBROUTINE DIFFP0(D,T,I,J,K,DIFF1)
      IMPLICIT DOUBLE PRECISION (A-Z)
      DIFF1=(D**(J-1)*J*T**I-D**(J+K-1)*K*T**I)/EXP(D**K)
      IF (K.EQ.0) DIFF1=DIFF1*EXP(D**K)
      RETURN
      END
C
C
      SUBROUTINE VIRDIF(T,DBDT)
      IMPLICIT DOUBLE PRECISION (A-Z)
      COMMON/CONST/TC,PC,ROC,R
      COMMON/OLDAT/BCOEF(32),BD(32,4)
         SUM=0
         DO 10 II=1,19
      I=BD(II,2)
      J=BD(II,3)
      K=BD(II,4)
         IF (J.NE.1) GOTO 10
C         dddt=d**(-1 + j)*i*j*t**(-1 + i)/Exp(-d**k) -
```

```
C     -   d**(-1 + j + k)*i*k*t**(-1 + i)/Exp(-d**k)
        DDDT = I*T**(-1+I)
        SUM=SUM+DDDT*BCOEF(II)
10      CONTINUE
        DBDT=SUM*(-T**2)/(TC*ROC)
     RETURN
     END
c
     SUBROUTINE DIFFP(D,T,I,J,K,DIFF2)
     IMPLICIT DOUBLE PRECISION (A-Z)
        DIFF2=2*d*(d**(-1 + j)*j*t**i/EXP(d**k) -
     *      d**(-1 + j + k)*k*t**i/EXP(d**k)) +
     *      d**2*(d**(-2 + j)*(-1 + j)*j*t**i/EXP(d**k) -
     *      d**(-2 + j + k)*j*k*t**i/EXP(d**k) +
     *      d**(-2 + j + 2*k)*k**2*t**i/EXP(d**k) -
     *      d**(-2 + j + k)*k*(-1 + j + k)*t**i/EXP(d**k))
     IF (K.EQ.0) DIFF2=DIFF2*EXP(D**K)
     RETURN
     END
c
c
     SUBROUTINE DIFFT2(D,T,I,J,K,DPT2)
     IMPLICIT DOUBLE PRECISION (A-Z)
        dpT2=2*d*(d**(-1 + j)*i*j*t**(-1 + i)/Exp(d**k) -
     -     d**(-1 + j + k)*i*k*t**(-1 + i)/Exp(d**k)) +
     -     d**2*(d**(-2 + j)*i*(-1 + j)*j*t**(-1 +
           i)/Exp(d**k) -
     -     d**(-2 + j + k)*i*j*k*t**(-1 + i)/Exp(d**k) +
     -     d**(-2 + j + 2*k)*i*k**2*t**(-1 + i)/Exp(d**k) -
     -     d**(-2 + j + k)*i*k*(-1 + j + k)*t**(-1 +
           i)/Exp(d**k))

     IF (K.EQ.0) DPT2=DPT2*EXP(D**K)

     RETURN
     END
c
      SUBROUTINE DP2DD2(DPD,DD,TT,IDD)
      IMPLICIT DOUBLE PRECISION (A-H,P-Z)
      COMMON/CONST/TC,PC,DC,R
      COMMON/COE/COEF(600)
      COMMON/IDNO/PIDN(600,4)
      COMMON/DAT/MOBS,NCO,NMAX,NCON
      COMMON/TERM/FACT(600)
        D=DD/DC
        T=TC/TT
        SUM5=0
        DO 11 I=1,NCO
```

```
         PIT=PIDN(I,2)
         PID=PIDN(I,3)
         PIDRO=PIDN(I,4)
         CALL DIFFP2(D,T,PIT,PID,PIDRO,DPDO)
         FACT(I)=DPDO
11      CONTINUE
        RETURN
        END
C
     SUBROUTINE DIFFP2(D,T,I,J,K,DPD2)
     IMPLICIT DOUBLE PRECISION (A-Z)
        DPD2=2*d**(-1 + j)*j*t**i/Exp(d**k) -
     -     2*d**(-1 + j + k)*k*t**i/Exp(d**k) +
     -     2*d*(d**(-2 + j)*(-1 + j)*j*t**i/
     -     Exp(d**k) -
     -     d**(-2 + j + k)*j*k*t**i/Exp(d**k) +
     -     d**(-2 + j + 2*k)*k**2*t**i/Exp(d**k) -
     -     d**(-2 + j + k)*k*(-1 + j + k)*t**i/
     -     Exp(d**k)) +
     -     d*(2*d**(-2 + j)*(-1 + j)*j*t**i/
     -     Exp(d**k) -
     -     2*d**(-2 + j + k)*j*k*t**i/Exp(d**k) +
     -     2*d**(-2 + j + 2*k)*k**2*t**i/
     -     Exp(d**k) -
     -     2*d**(-2 + j + k)*k*(-1 + j + k)*t**i/
     -     Exp(d**k) +
     -     d*(d**(-3 + j)*(-2 + j)*(-1 + j)*j*
     -     t**i/Exp(d**k) -
     -     d**(-3 + j + k)*(-1 + j)*j*k*t**i/
     -     Exp(d**k) +
     -     d**(-3 + j + 2*k)*j*k**2*t**i/
     -     Exp(d**k) -
     -     d**(-3 + j + 3*k)*k**3*t**i/
     -     Exp(d**k) -
     -     d**(-3 + j + k)*j*k*(-2 + j + k)*
     -     t**i/Exp(d**k) +
     -     d**(-3 + j + 2*k)*k**2*(-1 + j + k)*
     -     t**i/Exp(d**k) -
     -     d**(-3 + j + k)*k*(-2 + j + k)*
     -     (-1 + j + k)*t**i/Exp(d**k) +
     -     d**(-3 + j + 2*k)*k**2*
     -     (-2 + j + 2*k)*t**i/Exp(d**k)))
     IF (K.EQ.0) DPD2=DPD2*EXP(D**K)
     RETURN
     END
C
C
     SUBROUTINE DIFFDD(D,T,I,J,K,DDDD)
```

```
      IMPLICIT DOUBLE PRECISION (A-Z)
         dddd=d**(-2 + j)*(-1 + j)*j*t**i/EXP(d**k) -
     -   d**(-2 + j + k)*j*k*t**i/EXP(d**k) +
     -   d**(-2 + j + 2*k)*k**2*t**i/EXP(d**k) -
     -   d**(-2 + j + k)*k*(-1 + j + k)*t**i/
     -    EXP(d**k)
      IF (K.EQ.0) DDDD=DDDD*EXP(D**K)
      RETURN
      END
C
C
      SUBROUTINE DIFFTD(D,T,I,J,K,DDDT)
      IMPLICIT DOUBLE PRECISION (A-Z)
         dddt=d**(-1 + j)*i*j*t**(-1 + i)/EXP(d**k) -
     -   d**(-1 + j + k)*i*k*t**(-1 + i)/EXP(d**k)
      IF (K.EQ.0) DDDT=DDDT*EXP(D**K)
      RETURN
      END

C
      SUBROUTINE NONLIN(PP,DD,TT,CPNL)
      IMPLICIT DOUBLE PRECISION (A-H,P-Z)
      COMMON/OLDAT/BCOEF(32),BD(32,4)
      COMMON/CONST/TC,PC,DC,R
      D=DD/DC
      T=TC/TT
      DO 10 II=1,19
      I=BD(II,2)
      J=BD(II,3)
      K=BD(II,4)
      CALL DIFFTD(D,T,I,J,K,DDDT)
      CALL DIFFP0(D,T,I,J,K,DADD)
      CALL DIFFDD(D,T,I,J,K,DDDD)
      SUMD=SUMD+BCOEF(II)*DADD
      SUMDD=SUMDD+BCOEF(II)*DDDD
      SUMDT=SUMDT+BCOEF(II)*DDDT
10    CONTINUE
      CPNL=(1+D*SUMD-D*T*SUMDT)**2/(1+2*D*SUMD+D**2*SUMDD)
      CPNL=CPNL*R
      RETURN
      END
C
       SUBROUTINE DPDT(DPT,DD,TT,IDD)
       IMPLICIT DOUBLE PRECISION (A-H,P-Z)
      COMMON/CONST/TC,PC,DC,R
      COMMON/COE/COEF(600)
         COMMON/IDNO/PIDN(600,4)
         COMMON/DAT/MOBS,NCO,NMAX,NCON
```

```
         COMMON/TERM/FACT(600)
      COMMON/OLDAT/BCOEF(32),BD(32,4)
      D=DD/DC
      T=TC/TT
      SUM4=0
      SUM5=0
      IF (IDD.EQ.2) GOTO 12
      DO 11 I=1,NCO
       PIT=PIDN(I,2)
       PID=PIDN(I,3)
       PIDRO=PIDN(I,4)
      CALL DIFFP0(D,T,PIT,PID,PIDRO,DADD)
      CALL DIFFTD(D,T,PIT,PID,PIDRO,DPDO)
      SUM4=SUM4+DADD*COEF(I)
      SUM5=SUM5+DPDO*COEF(I)
11    CONTINUE
      GOTO 20
12    DO 15 I=1,19
       BIT=BD(I,2)
       BID=BD(I,3)
       BIDRO=BD(I,4)
      CALL DIFFP0(D,T,BIT,BID,BIDRO,DADD)
      CALL DIFFTD(D,T,BIT,BID,BIDRO,DPDO)
      SUM4=SUM4+DADD*BCOEF(I)
      SUM5=SUM5+DPDO*BCOEF(I)
15    CONTINUE
20    DPT=R*(DD+DC*D**2*SUM4-DC*D**2*T*SUM5)
      RETURN
      END

C
C
       SUBROUTINE FITDPD(FDPD)
C************************************************************
C
C
C   THIS SUBROUTINE FITS THE DPD DATA.
C
C************************************************************
C
       IMPLICIT DOUBLE PRECISION (A-H,O-Z)
       PARAMETER (MDPD = 500)
C        PARAMETER (NC = 32, NF = NC + 1 )
C
       COMMON /DPDAT/ NDP, IDP(MDPD), WDP(MDPD), TDP(MDPD),
       DDP(MDPD),
      *               DPDX(MDPD)
C
```

```
      COMMON/DAT/MOBS,NCO,NMAX,NCON
      COMMON/TERM/FACT(600)
      COMMON/CONST/TC,PC,DC,R
C       COMMON /FITCOM/ F(NF)
C
C       COMMON /REFDAT/ R, PC, DC, TC, ZC, G(NC), GAMMA,
PTRP, DTRP, TTRP
C     *                , CMW
C
C
      DO 100 J=1, NDP
      CALL DPDD(DPDX(J),DDP(J),TDP(J),0)
      RES = DPDX(J)-R*TDP(J)
      WDP(J) = FDPD / SQRT(ABS(RES))
C       IF (TDP(J).GT. 370.) WDP(J) = 50.0*WDP(J)
C       IF (ABS(TDP(J)-TC).LT.20.0D0) WDP(J) = 2.0*WDP(J)
      IF(IDP(J).GT.0) GO TO 020
      IDP(J) = -IDP(J)
      WDP(J) = 0.0
  020 DO 040 K = 1, NCO
  040 FACT(K) = FACT(K) * TDP(J)*R
C       Y = WDP(J) * Y
C       WDP(J)=WDP(J)*FDPD
      IF (WDP(J).GT.0) MOBS=MOBS+1
      DO 11 IJ=1,NCO
       FACT(IJ)=FACT(IJ)*WDP(J)
11    CONTINUE
      RES=RES*WDP(J)
      CALL FIT(RES,WDP(J))
  100 CONTINUE
      RETURN
      END
C
C
C
      SUBROUTINE DPDCMP
C************************************************************
C
C
C    THIS SUBROUTINE PERFORMS THE DPDD COMPARISONS.
C
C************************************************************
C
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      PARAMETER (MDPD = 500)
C
      COMMON /DPDAT/ NDP, IDP(MDPD), WDP(MDPD), TDP(MDPD),
      DDP(MDPD),
```

```
     *                DPDX(MDPD)
C
      WRITE(3,200)
C
      AAD = 0.0
      BIAS = 0.0
      RMS = 0.0
      DO 100 J=1,NDP
      CALL DPDD(DPDC,DDP(J),TDP(J),1)
      PCT = -100.0*(DPDC-DPDX(J))/DPDX(J)
      WRITE(3,220) IDP(J), TDP(J), DDP(J), DPDX(J), DPDC,
      PCT, WDP(J)
      WRITE(25,*)WDP(J)
      IF (WDP(J).LE.0.0) GO TO 100
      AAD = AAD + ABS(PCT)
      BIAS = BIAS + PCT
      RMS = RMS + PCT*PCT
  100 CONTINUE
      AAD = AAD / NDP
      BIAS = BIAS /NDP
      RMS = SQRT(RMS/NDP-BIAS*BIAS)
      WRITE(3,240) NDP, AAD, BIAS, RMS
C
      RETURN
  200 FORMAT('1DPDD COMPARISONS'/'  ID    T        D
      DPDDX
      1DPDDC    PCT       WT')
  220 FORMAT(I5,F8.3,F10.5,2F10.3,F8.3,F10.4)
  240 FORMAT('0N = ',I6,'  AAD = ',F7.3,'  BIAS = ',F7.3,'
      RMS =',F7.3)
      END
C
C
C
      SUBROUTINE WSPCMP
C************************************************************
C
C
C   THIS SUBROUTINE PERFORMS THE SOUND VELOCITY COMPARISONS.
C
C************************************************************
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      PARAMETER (MWSP = 450)
C
      COMMON /SOUND/ NWS, IDWS(MWSP), TWS(MWSP), PWS(MWSP),
      WSPT(MWSP)
C
C      COMMON /REFDAT/ R, PC, DC, TC, ZC, G(NC), GAMMA,
```

```
       PTRP, DTRP, TTRP
C     *              , CMW
        COMMON /REFDAT/ZC, PTRP, DTRP, TTRP,CMW
       COMMON/CONST/TC,PC,DC,R
C
      COMMON /DPDCAL/ CP, CV, DPD, DPT
C
      AAD = 0.0
      BIAS = 0.0
      RMS = 0.0
      N = 0
      WRITE(3,400)
      DO 020 J = 1, NWS
      IF (WSPT(J).LE.0.0) GO TO 020
      D = RHOF(PWS(J),0.0D0,TWS(J))
      WCAL = WSF(D,TWS(J))
      N = N + 1
      PCT = -100.0 * (WCAL - WSPT(J)) / WSPT(J)
      AAD = AAD + ABS(PCT)
      BIAS = BIAS + PCT
      RMS = RMS + PCT * PCT
      WRITE(3,410) IDWS(J), TWS(J), PWS(J), WCAL, WSPT(J),
      PCT
      DPDX = 1.0E-5 * CMW * WSPT(J) * WSPT(J) * CV / CP
      DPDX = 0.25*DPD + 0.75*DPDX
      WRITE(11,460) IDWS(J), TWS(J), D, DPDX
  020 CONTINUE
      IF (N.GT.0) THEN
         AAD = AAD / N
         BIAS = BIAS / N
         RMS = SQRT(RMS/N - BIAS*BIAS)
         WRITE(3,450) N, AAD, BIAS, RMS
         IF (N.EQ.NWS) RETURN
      ENDIF
C
  025 AAD = 0.0
      BIAS = 0.0
      RMS = 0.0
      N = 0
      WRITE(3,420)
      DO 040 J = 1, NWS
      IF (WSPT(J).GT.0.0) GO TO 040
      WSPT(J) = - WSPT(J)
      CALL SATF(TWS(J),PS,DSL,DSV)
      WCAL = WSF(DSL,TWS(J))
      N = N + 1
      PCT = 100.0 * (WCAL - WSPT(J)) / WSPT(J)
      AAD = AAD + ABS(PCT)
```

```
      BIAS = BIAS + PCT
      RMS = RMS + PCT * PCT
      WRITE(3,440) IDWS(J), TWS(J), DSL, WCAL, WSPT(J), PCT
      DPDX = 1.0E-5 * CMW * WSPT(J) * WSPT(J) * CV / CP
      DPDX = 0.25*DPD + 0.75*DPDX
      WRITE(5,460) IDWS(J), TWS(J), DSL, DPDX
C     CALL WSD(TWS(J),DSL,WSPT(J))
  040 CONTINUE
      IF (N .GT. 0) THEN
         AAD = AAD / N
         BIAS = BIAS / N
         RMS = SQRT(RMS/N - BIAS*BIAS)
         WRITE(3,450) N, AAD, BIAS, RMS
      ENDIF
      RETURN
C
  400 FORMAT('1SINGLE PHASE SOUND VELOCITY COMPARISONS'//'
ID     T,K
     *       P,EXP    WS, CAL   WS,EXP      PCT')
  410 FORMAT(I5,2F10.3,2F10.2,F8.3)
  420 FORMAT('1SATURATED LIQUID SOUND VELOCITY
      COMPARISONS'//'   ID    T
     *,K      D,SAT     WS,CAL     WS,EXP     PCT')
  440 FORMAT(I5,F10.3,F10.5,2F10.2,F8.3)
  450 FORMAT('0N =',I4,' AAD =',F6.2,' BIAS =',F6.2,' RMS
      =',F6.2)
  460 FORMAT('DPDD',I4,F10.3,F10.5,F10.2)
      END
      SUBROUTINE WSD(T,D,WS)
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      DHI = 1.005*D
      DLO = 0.995*D
  010 DX = 0.5*(DLO+DHI)
      WCAL = WSF(DX,T)
      DIF = WCAL - WS
      IF (DIF) 030,050,020
  020 DHI = DX
      GO TO 040
  030 DLO = DX
  040 IF (ABS(DIF).LT.0.1) GO TO 050
      IF (ABS(DHI/DLO-1.0).LT.0.0001) GO TO 050
      GO TO 010
  050 PCT = 100.*(DX-D)/D
      WRITE(6,100) DX, D, PCT, WCAL, WS
      RETURN
  100 FORMAT(2F10.5,F6.3,2F10.2)
      END
C
```

```
C
      FUNCTION WSF(D,T)
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
C***********************************************************
C
C
C       THIS ROUTINE CALCULATES THE SOUND VELOCITY
C                        GIVEN THE DENSITY AND TEMPERATURE
C
C***********************************************************
C
C      COMMON /REFDAT/ R, PC, DC, TC, ZC, G(NC), GAMMA,
        PTRP, DTRP, TTRP
C     *               , CMW
        COMMON /REFDAT/ZC, PTRP, DTRP, TTRP,CMW
       COMMON/CONST/TC,PC,DC,R
C
      COMMON /DPDCAL/ CP, CV, DPD, DPT
C
      CV = CVF(D,T)
      CALL DPDT(DPT,D,T,1)
      CALL DPDD(DPD,D,T,1)
      DPT = DPT / D
      CP = CV + 100.0 * T * DPT * DPT / DPD
      WSF = SQRT(1.0E5 * CP * DPD / (CMW * CV))
      RETURN
      END
C
C
      SUBROUTINE DP2DT2(DPT,DD,TT,IDD)
      IMPLICIT DOUBLE PRECISION (A-H,P-Z)
     COMMON/CONST/TC,PC,DC,R
     COMMON/COE/COEF(600)
        COMMON/IDNO/PIDN(600,4)
        COMMON/DAT/MOBS,NCO,NMAX,NCON
        COMMON/TERM/FACT(600)
     COMMON/OLDAT/BCOEF(32),BD(32,4)
     D=DD/DC
     T=TC/TT
     SUM4=0
     SUM5=0
     IF (IDD.EQ.2) GOTO 12
     DO 11 I=1,NCO
      PIT=PIDN(I,2)
      PID=PIDN(I,3)
      PIDRO=PIDN(I,4)
     CALL DIFFP0(D,T,PIT,PID,PIDRO,DADD)
     CALL DIFFTD(D,T,PIT,PID,PIDRO,DPDO)
```

```
     SUM4=SUM4+DADD*COEF(I)
     SUM5=SUM5+DPDO*COEF(I)
11   CONTINUE
     GOTO 20
12   DO 15 I=1,19
      BIT=BD(I,2)
      BID=BD(I,3)
      BIDRO=BD(I,4)
     CALL DIFFP2T2(D,T,BIT,BID,BIDRO,DPDTV,DP2DT2V)
     SUM4=SUM4+DPDTV*BCOEF(I)
        SUM5=SUM5+DP2DT2V*BCOEF(I)
15   CONTINUE
20   DPT=2*T**3*SUM4+T**4*SUM5
        DPT=R*DC*D**2*DPT/TC
     RETURN
     END
C
C
     SUBROUTINE DIFFP2T2(D,T,I,J,K,DPDTV,DP2DT2V)
     IMPLICIT DOUBLE PRECISION (A-Z)
     COMMON/CONST/TC,PC,DC,R

           dpdtv = (d**(-1 + j)*i*j*t**(-1 + i)/Exp(d**k) -
     -     d**(-1 + j + k)*i*k*t**(-1 + i)/Exp(d**k))/t
     -   - (d**(-1 + j)*j*t**i/Exp(d**k) -
     -     d**(-1 + j + k)*k*t**i/Exp(d**k))/t**2
        dp2dt2v = (d**(-1 + j)*(-1 + i)*i*j*t**(-2 +
        i)/Exp(d**k) -
     -  d**(-1 + j + k)*(-1 + i)*i*k*t**(-2 + i)/
     -  Exp(d**k))/t -
     -  2*(d**(-1 + j)*i*j*t**(-1 + i)/Exp(d**k) -
     -  d**(-1 + j + k)*i*k*t**(-1 + i)/Exp(d**k))/
     -  t**2 + 2*(d**(-1 + j)*j*t**i/Exp(d**k) -
     -  d**(-1 + j + k)*k*t**i/Exp(d**k))/t**3

     IF (K.EQ.0) DPDTV=DPDTV*EXP(D**K)
        IF (K.EQ.0) DP2DT2V=DP2DT2V*EXP(D**K)
     RETURN
     END
C
C
C
     SUBROUTINE CONSTRAIN
C************************************************************
C
C
C   THIS SUBROUTINE ADDS THE CONSTRAINTS TO THE LEAST
C  SQUARES FIT
```

```
C
C***********************************************************
C
     IMPLICIT DOUBLE PRECISION (A-H,P-Z)
     COMMON/DAT/MOBS,NCO,NMAX,NCON
     COMMON/CONST/TC,PC,DC,R
     COMMON/TERM/FACT(600)
     COMMON/PROBLEM/NCONT
     RES=PC-(R*TC*DC)
     CALL PRESSB(TC,PC,DC,0)
     NCONT=1
     CALL FIT(RES,1.0)
        CALL DPDD(DPD,DC,TC,0)
        NCONT=2
        RES= -R*TC
        DO 10 I=1,NCO
10      FACT(I)=R*TC*FACT(I)
        CALL FIT(RES,1.0)
        RES=0.0
        NCONT=3
        CALL DP2DD2(DPD2,DC,TC,0)
        DO 20 I=1,NCO
20      FACT(I)=FACT(I)*R*TC/DC
        CALL FIT(RES,1.0)
     RETURN
     END
      SUBROUTINE CRITF(PCC,DCC,TCC)
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
C * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C
C
C      PROGRAMMER         JOLENE K. BAKER
C                        THERMOPHYSICAL PROPERTIES DIVISION
C                         NATIONAL BUREAU OF STANDARDS
C                         BOULDER, COLORADO   80303
C
C      PURPOSE   THIS PROGRAM CALCULATES THE CRITICAL
C             PROPERTIES (TEMPERATURE, PRESSURE, DENSITY, Z)
C             FOR A PURE FLUID.  THE CRITERIA USED FOR
C                         CONVERGENCE IS:
C
C                         (1)  FIRST DERIVATIVE OF PRESSURE
C                              WRT DENSITY = 0.
C
C                       (2)  SECOND DERIVATIVE OF PRESSURE
C                              WRT DENSITY = 0.
C
C
```

```
C * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C
C
C            SET BOUNDS ON CRITICAL TEMP./CRITICAL DENSITY
C
      J = 0
      THI = 1.05*TCC
      TLO = 0.95*TCC
  3   DHI = 1.15*DCC
      DLO = 0.85*DCC
      J = J + 1
      IF(J .GE. 50) GO TO 50
C
C               START TEMP. ITERATION
C
      T = (THI + TLO) * 0.5
C
C
      I = 0
C               START DENSITY ITERATION
C
  5   D = (DHI + DLO) * 0.5
      I = I + 1
      IF(I .GE. 50) GO TO 25
      CALL PVTF(P,D,T,DPDD,D2PDD2,G0)
      IF(ABS(D2PDD2) .LE. 1.0E-8) GO TO 20
      IF(D2PDD2) 10,20,15
 10   DLO = D
      GO TO 5
 15   DHI = D
      GO TO 5
 20   IF(ABS(DPDD) .LE. 1.0E-8) GO TO 50
 25   IF(DPDD) 30,50,35
 30   TLO = T
      GO TO 3
 35   THI = T
      GO TO 3
 50   IF(DPDD) 30,60,60
 60   TCC = T
      DCC = D
      PCC = P
      WRITE(3,100) PCC, DCC, TCC
  100 FORMAT('1CONSTRAINED RESULTS WITH PC=',F9.4,',
DC=',F6.3,' AND T
     *C=',F8.3)
      RETURN
      END
```