A NUMERICAL MODELING OF THREE PHASE DIRECT CONTACT HEAT EXCHANGERS

by

Mustafa Turhan Çoban

A dissertation submitted to the faculty of
The University of Utah
in partial fulfillment of the requirements for the degree of

Doctor of Philosophy

in

Mechanical Engineering

Department of Mechanical and Industrial Engineering

University of Utah

August 1986

THE UNIVERSITY OF UTAH GRADUATE SCHOOL

SUPERVISORY COMMITTEE APPROVAL

of a dissertation submitted by

Mustafa Turhan Coban

This dissertation has been read by each member of the following supervisory committee and by majority vote has been found to be satisfactory.

June 27, 1986
Chairman: Dr. R. F. Boehm

June 27, 1986
M. Q. Brewster
Dr. M. Q. Brewster

June 27, 1986
Dr. K. Chen

June 27, 1986
L. K. Isaacson
Dr. L. K. Isaacson

June 27, 1986
J. D. Seader
Dr. J. D. Seader

ABSTRACT

A one-dimensional numerical model is developed for motion and heat transfer in a three-phase, spray-column, direct-contact heat exchanger. General equations for fluid motion and heat transfer are defined for distance up the column using a physically based model. A new formulation is given for a mixed time-averaged temperature that may be representative of measurements taken with temperature transducers in direct-contact heat exchangers. The resultant temperature profiles are compared to experimental data found from the operation of a 0.6-m diameter, 6-m high spray column heat exchanger using pentane and water, (University of Utah data performed in conjuction with this project) and a 1.016-m diameter, 9-m high spray column heat exchanger using isobutane and water (a DOE East Mesa project). Good agreement is shown between the predictions and the data. Eventhough component fluid temperatures generally increase with height, it is shown here that a decreasing temperature level with height may represent good operation.

The influence of various parameters upon total heat transfer and these parameters' behavior during the process of direct contact evaporation have been investigated. It is found that the total heat transfer in the column decreases when initial drop radius and initial temperature of the dispersed fluid increase. Meanwhile the total heat transfer increases when the mass flow rates of the dispersed and continuous phases increase. It is shown that insufficient height is found to be an important factor causing incomplete boiling. But after the completion of boiling, additional column height has only small effect on the total heat transfer. The model is checked for the accuracy of predicting total heat transfer for several experimental tests. Results of the model give good agreement with the experimental results. All these comparisons proved that this model can be used as a design tool to predict physical dimensions of the next generation of direct contact boilers.

TABLE OF CONTENTS

Page

Chapter

Appendices

LIST OF FIGURES

NOMENCLATURE

A	Bubble surface area
A	Cross sectional area of the Column; also flow coefficient
Bo	Liquid Bond number
C_D	Drag coefficient
C_p	Specific heat under constant pressure
d	Diameter of orifice
D	Diameter of droplet, also diameter of pipe
D	An averaged diameter, $(D^2 + D^2{}_0)/(2DD_0)$
E	Orifice coefficient
F	Orifice coefficient
g	Acceleration due to gravity
h	Heat transfer coefficient
h	Enthalpy
i	Counter index
k	Thermal conductivity
k	Correlation factor for Sideman and Taitel equation
K	Orifice coefficient
LMTD	logarithmic mean temperature difference
m	Mass flow rate
M	Ratio of liquid density to vapor density; also molecular weight

Nu	Nusselt number
n	Number of drops per unit volume
P	Pressure
Pe	Peclet number
Pe'	Modified Peclet number
Pr	Prandtl number
Re	Reynolds Number
t	Time
T	Temperature
U	Velocity
V	Volume
x	Quality
z	Vertical distance along column from bottom
α	Thermal diffusivity
α	A vapor angle dependent of surface tension
β	Vapor half opening angle, also area ratio of orifice
η	Ratio: heat transfer to the bubble to heat transfer from the continuous phase
σ	Surface tension
γ	Constant rate of increase of surface temperature with time
ϕ	Holdup ratio
μ	Viscosity

ρ	Density

Subscripts and superscripts

ave	Average value
c	Continuous fluid
cd	Between continuous and dispersed fluids
d	Dispersed phase
D	Drag
l	Liquid
mix	Mixed value
p	Pentane
o	Initial also flow coefficient of orifice
v	Vapor
W	Water

ACKNOWLEDGEMENTS

A special thanks is extended to my committee chairman Dr. Robert F. Boehm for his guidance, support and patience throughout my graduate work.

Gratitude is also expressed to my parents Osman and Hatice for making my education possible.

The partial support of this work by the Solar Energy Research Institute and Utah Power and Light Company is gratefully noted.

CHAPTER I

INTRODUCTION

Three-phase direct-contact heat exchangers have been proposed for a variety of applications. One of the first applications was for water desalination. More recent applications are usually concerned with power generation. The decreasing availability of conventional energy resources has increased the need for developing renewable energy resources. Geothermal and solar energies are two such alternatives.

Geothermal energy is available in limited amounts in the form of dry steam which can be utilized directly by a turbine- condenser system. Most of the geothermal resources are found as wet steam or in a hot water phase. The economical utilization of wet steam or hot water by using classical methods such as flashing cycles has introduced many practical difficulties. The use of binary cycles where the heat of the primary fluid is transferred to a secondary fluid with a lower boiling point holds significant potential. Transferring of heat from the primary to the secondary fluid requires heat exchangers with a high effectiveness. Unfortunately the impurities in the geothermal brine cause corrosion and fouling, and the effectiveness of classical heat exchangers decreases with time. Cleaning of the heat exchangers is expensive and is a time-consuming job. Furthermore it can dramatically reduce total operational time of the power unit. One solution to this problem is to build more heat exchangers than required and establish a continuous cleaning process for the system. This certainly will cause a high operational and capital cost. All these conditions have made direct contact heat exchangers an attractive alternative to conventional heat transfer units.

The advantages of using direct contact heat exchangers over closed heat exchangers with metallic surfaces can be summarized as:

1. Simple and relatively inexpensive equipment,
2. Large effective heat transfer areas per unit volume, higher heat transfer coefficients and closer approach temperatures, and
3. Fewer corrosion and fouling problems.

Possible disadvantages of direct contact heat exchangers can also be pointed out as:

1. End effects, backmixing and flooding,
2. Less choice in secondary fluids, and
3. More complex heat transfer process that is not always well understood.

More research and better understanding should eliminate many of these disadvantages.

Direct contact heat exchangers can be grouped from several different perspectives. One classification is according to the heat exchanger design. This includes spray towers, packed towers, baffle towers, perforated plate towers and wetted wall towers. The device which has received most attention is the spray column. The advantage of the spray column is that it is free of heat transfer enchancement devices. A second classification can be made according to the heat transfer processes in the column. This class combines freezers, defrosters, preheaters, boilers and superheaters. In some designs more than one of these heat transfer units can be used together, such as combined preheaters and boilers. The third classification is according to the flow direction of the primary and secondary fluids. If the two flows are in the same direction, the heat exchanger is referred to as parallel flow; if it is in the opposite direction it referred to as counter flow. The other definition is also related to the flow conditions. The fluid that flows mostly in bubble form and rises or falls by buoyancy forces is called the dispersed phase. The fluid that occupies most of the volume through the body of the tower and flows as a continuous flow is called the continuous phase.

In contrast to the large amount of experimental and theoretical work on physical behavior for conventional closed heat exchangers, very little information is available on physical behavior of direct contact heat exchangers. Thus very little is known regarding the basic mechanism associated with evaporation in immiscible liquids.

The scope of the present work includes further investigation into the characteristics of boiling in direct contact three phase heat exchangers and the development of a computer model that closely represents a real direct contact heat exchanger. Investigation of a three phase, counter current, spray column heat exchanger is discussed in this dissertation. Three-phase heat transfer here refers to a single continuous phase fluid and a working fluid which undergoes a change of phase by boiling. The primary fluid is taken as the continuous phase and the secondary fluid is taken as the dispersed phase. A cross sectional view of a typical spray column , counter flow direct contact heat exchanger is shown in Figure 1. It consists of a vertical column, an injection plate with several injection nozzles mounted on it for dispersed fluid entrance, another injection nozzle for the continuous phase and exit ports for each fluid. Dispersed fluid enters the column through the injection nozzles. Meanwhile continuous fluid enters from the top of the column. The dispersed liquid bubbles up through the falling continuous fluid. Heat, absorbed by the dispersed phase, causes the liquid to undergo a phase transition. Thus, by the time the dispersed fluid reaches the active fluid level it is completely vaporized and leaves the column from the dispersed phase exit pipe.

The key to a better understanding of direct contact

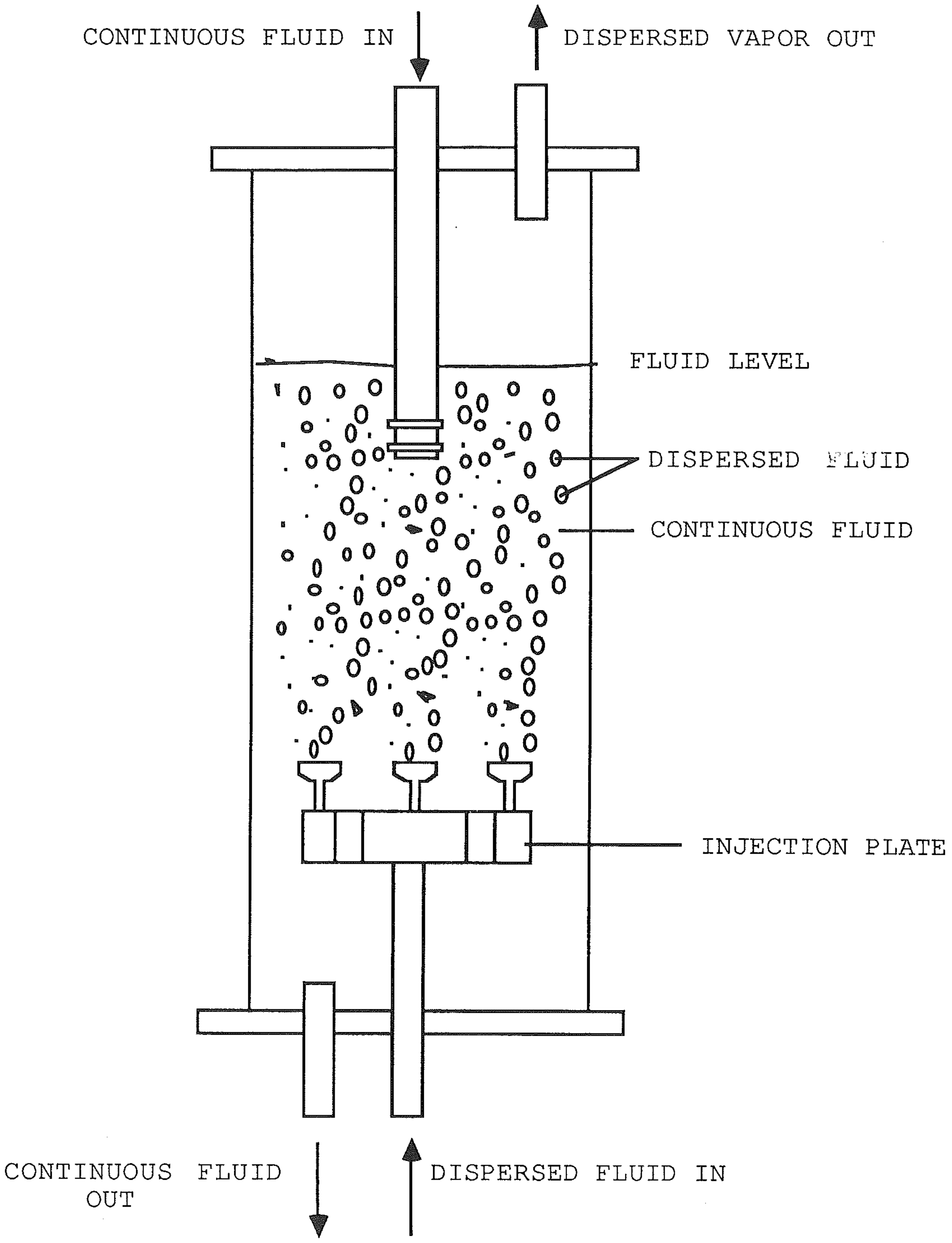

Figure 1. Schematic of a counter-flow spray-column direct-contact heat exchanger.

heat exchangers will almost certainly be the development of numerical models that accurately reflect the heat transfer and fluid mechanics phenomena that are present. Several problems exist in developing an accurate model. One of the most critical areas of the modeling is the representation of the heat transfer coefficient. While several predictive equations exist in the literature, none of them is able to explain the heat transfer to the droplet completely. One of the earliest reports of work in this area is given by Sideman and Taitel [6]. They studied evaporation of pentane drops in a continuous water phase. Their study included theoretical work to attempt to explain the heat transfer mechanism present. It was assumed that an evaporating liquid drop at constant boiling temperature was moving in a potential flow field and the resulting convection outside of the drop controlled the total heat transfer. Heat transfer to the vapor phase of the bubble was ignored. Their equation is presented in Appendix A-1. In this equation β is defined as half opening angle. The relation between half opening angle, β, and quality of the vapor phase is given by Equation A-1-1 in Appendix A. The authors indicate that the maximum values of heat transfer may be obtained when β is taken to be 135 degrees. For liquid/liquid heat transfer (no phase change), β is zero, and the equation reduces to the Boussinesq and

Higbie equations for potential flow passing over a sphere.

Sideman and Hirsch [33] investigated the importance of the external and internal resistance during the evaporation of a liquid drop. The authors indicated that at the very beginning of the evaporation the external resistance of the heat transfer controls the total heat transfer process. They claimed that, in the beginning period, overcoming internal resistance is the basic heat transfer mechanism. The outside heat transfer coefficient becomes the controlling mechanism over more than 70 percent of the evaporation process due to increasing vapor phase volume. Their investigation led them to suggest that the internal resistance may also be the controlling mechanism for the preheating region. However, they did not mention which resistance is in effect at the end of evaporation and in the superheating region.

Sideman and Isenberg [5] compared the theoretical heat transfer coefficient of Sideman and Taitel [4] to experimentally determined heat transfer coefficents. The experimental data were taken as a function of vapor weight percent in the bubble from the pentane-water system. Good agreement between the data and the theoretical heat transfer coefficient was only found in the 5 to 10 percent vapor range. They suggested a correction factor to the Sideman and Taitel equation [A-1-1] as a form of Prandtl number dependent function [A-2-1]. These investigators reported that the modified form of the equation yields closer results with the experimental cases [A-2-2].

Tochitani et al. [14] derived a theoretical relationship for the heat transfer coefficient assuming that the surface of a two phase droplet is rigid and that the surrounding flow obeys Stoke's law. An expression for the heat transfer coefficient developed by their model is given by [A-3-1] in Appendix A. Their equation for the heat transfer coefficient is similar to that for heat transfer from a rigid sphere and also contains a multiplication factor to represent the surface area change of the dispersed phase liquid in the spherical bubble. Definition of the opening angle β is the same as the Sideman and Taitel equation [A-1-2].

Raina and Grover [9] developed a theoretical model for three phase heat exchangers based on the liquid-liquid transfer interface by considering the effect of viscous shear on the spreading of dispersed liquid over the bubble surface [A-4-1]. The half opening angle, β, still has the same definition as used by Sideman but a second angle, α, was also defined to describe the geometrical formation of the bubble in the equation [A-4-2]. The configuration of angles α and β is shown in Figure A-2. Raina and Grover reported that their model provides very close results to the experimental values of Sideman and Taitel for pentane-water data as compared to the theoretical models of Tochitani, et al. and Sideman and Isenberg.

In all these models the following assumptions were made:

1. The droplet is spherical and symmetrical with respect to the vertical axis. The liquid phase accumulates at the bottom of the droplet, forming a sickle-shaped cross-section.
2. Within the vapor phase there is no temperature gradient and heat transfer.
3. The continuous phase is infinite and has a constant temperature and velocity with the exception of the zone around the droplet.
4. The flow pattern around the droplet is defined as potential flow or creeping flow.
5. The droplet rises vertically without oscillation.
6. The viscous dissipation term in the energy equation is disregarded.
7. Boundary conditions are fixed in such a way that starting problems such as retarded nucleation and effects caused by the detachment from the nozzle need not be taken into consideration.

Relations describing heat transfer to a freely moving fluid drop in an immiscible continuous medium are reviewed by Sideman and Shabtai [18]. Theoretical and experimental equations for determining the inside and outside transfer coefficients were presented for three cases: a rigid drop, a completely mixed drop and a drop with internal circulation. These authors also tabulated the important equations in a convenient tabular form. One of these equations which is used in our model is given in Appendix A.

A theoretical analysis of the evaporation of a single droplet rising in a column was performed by Mokhtarzadeh and El-Shirbini [4]. The authors assumed a spherical coordinate system moving along with the bubble droplet. The origin was located at the center of the bubble. They formulated the energy and momentum equations analytically according to this coordinate system and developed a set of equations which are solvable with numerical techniques. The researchers compared several variables calculated by their model to the experimental measurements of Sideman.

There are few experimental models for the heat transfer coefficient during the direct evaporation. One of the earliest studies was done by Adams and Pinder [34]. In this study a dilatometric method was combined with a cine photographic technique to obtain average heat transfer coefficients for the evaporation of the droplets. These average heat transfer coefficients were correlated to obtain a best curve fit for the data. The Adams and Pinder equations are given by Eqn. [A-5-1] in Appendix A. A similar work was done by Parakash and Pinder [12]. In this study furan, isopentane and cyclopentane were used as working fluids. Measurements were possible over a range of zero to ten percent of quality. Two correlation equations were given in this experimental model over the measurable range [A-6-1],[A-6-2].

Simpson, Beggs and Nazir [7] performed both theoretical and experimental investigations on the evaporation of butane droplets in a vertical column of both water and geothermal brine. For bubbles growing up to 2.0 times the initial diameter for brine and 2.7 times for water, a constant velocity of 25 cm/sec was found. The authors observed that the residual liquid butane sloshed from side to side inside of the bubble surface. Taking into account the effect of the sloshing liquid on the heat transfer, a theoretical value for the overall heat transfer was obtained [A-7-1]. The authors reported that values for the heat transfer coefficients based on their sloshing method are in better agreement with the Sideman butane- water data then Sideman's own theory.

An analytical model was developed for calculating volumetric heat transfer coefficients by Smith, Rohsenow and Kazimi [27]. In this work, heat transfer was modeled using single droplet correlations for the Nusselt number, while the fluid dynamics are described by a drift model. The importance of this work is due to its nature of considering coalescence and breakup effects on the calculations. They also conducted experiments with cyclopentane as the dispersed fluid and stagnant water as the continuous fluid. Their analytical and experimental results showed a good agreement with the data obtained from the experimental direct contact evaporator using cyclopentane and water.

Another analytical model came from Battya, Raghavan and Secthamaramu [37] who developed an analysis of direct contact latent heat transfer between two immiscible liquids in a counterflow spray column. The derivation of the equations was approached by solving the energy and continuity equations in nondimensional form. The number of unknowns in these equations was reduced by defining nondimensional groups. The equations were solved by numerical techniques, and dimensionless parameters were used to investigate the effect of several parameters. Comparison of the resultant parameters to the results of Sideman yielded a fair data fitting. This model was not very effective in predicting actual physical parameters due to calculations limited to some dimensionless groups with more than one physical variable in each.

There are only a very few experimental studies on full-size direct-contact heat exchangers. A major thrust on this subject has come from the U.S. Department of Energy. A 500 KW Power plant at East Mesa, California has been built [14]. This unit combined counter current preheater and boiler utilizing geothermal brine to vaporize isobutane. There are several studies related to the East Mesa unit. Rapier [35] discussed the performance of the East Mesa Plant. He stated that the criterion for performance of the unit was the classical steam distillation efficiency. This efficiency was defined as ratio of the partial pressure actually exerted by the working fluid to its saturation pressure at the vapor exit temperature. Rapier also reported that the best boiler configuration was obtained by a conventional mixing nozzle and flash drum design with the heat transfer occurring only as a result of adiabatic mixing and isothermal separation.

As a portion of the present work a full scale spray column heat exchanger was investigated [23]. In this study, commercial grade n-pentane was used as the dispersed fluid undergoing the phase change. Water was utilized as the continuous fluid. The effect of the variations of mass flow rates and pressures on the heat transfer process were investigated when the inlet temperature of water was kept constant. The details of this work are given in Appendix B. Due to detailed information given in this study, it will be our basic source of information for experimental comparison.

A glass direct contact heat exchanger was operated by A. V. Simms [40]. The column was run at atmospheric pressure using hot water and normal hexane. Variations of temperature differences from 1 °C for packing to 13 °C for spray column were used. This research investigated optimum cycle efficiencies for the case of operating the heat exchanger as a part of binary power unit.

W. B. Suratt, et al. [36] tested and investigated direct contact heat exchangers in which an aqueous synthetic brine (21 percent NaCl) and isobutane were utilized as the dispersed and continuous fluids respectively. The pilot-plant apparatus included both a spray column boiler and a direct contact preheater. Brine entered from the top of each vessel and left from the bottom while isobutane entered from the bottom of the columns and left from the top. The isobutane passed through a distributor plate in the preheater to produce 0.15 inch diameter drops which did not coalesce and left the preheater to enter the boiler as a saturated liquid. Vapor leaving the boiler ranged up to 25 °F of superheat. Volumetric heat transfer coefficients of 4000 BTU/hr °F ft^3 in the preheater and 17000 BTU/hr °F ft^3 in the boiler were measured during the experiments. The authors reported that continuous and stable operation of the direct contact preheater and boiler could be accomplished. However the effects of backmixing and resultant direct contact heat exchanger performance due to variations in tower diameter and height were unclear.

A numerical model of a simulation of a liquid-liquid spray column was presented by Jacobs and Golafshani [3]. In this work, a steady-state one-dimensional multiphase model was developed to describe the characteristics of a spray column direct contact heat exchanger. The authors used a total flow concept in order to develop their model. Basic mass, momentum and energy equations were solved by using finite-difference equations. Furthermore assumptions of constant drag coefficient and change of radius as a function of density were made. A fair agreement was shown between the results of their numerical model and the experimental results of the East Mesa plant.

Another critical aspect of the modeling is the functional form of the velocity of the bubbles. Raina and coworkers [11] dealt with the analysis of the instantaneous velocity. They started from the basic force balance equation and derived an equation for terminal velocity [A-8-1]. It was found that the experimental values of the heat transfer for a given initial diameter of the dispersed liquid drop decreased with an increase of temperature driving force. From the observation they developed a correlation. They included this as a correlation to the velocity equations obtained from basic force balance. The resultant equation was given by [A-8-2]. Finally, a Prandtl number dependent correlation equation was added to the Equation [A-8-2] in order to take into account the effect of viscosity [A-8-3]. Raina and Wancho [13] also developed an analytical expression [A-9-1] for terminal velocity. These investigators claimed that their expression [A-9-1] agrees very well with the experimental data for n-pentane and furan drops evaporating through high viscosity aqueous glycerol.

Mokhtarzadeh and El-Shirbini [26] worked on the dynamics of the two-phase bubble-droplets of butane in distilled water. Experimental data was taken from a square column made of Perspex by photograpy with a cine-camera which was moved parallel to the droplet. The dimension of the bubble, position in the column and the time were obtained from the analysis of the pictures and the knowledge of the camera speed. They found that for a given droplet size and fluid properties the behavior of the two-phase bubble droplet system depends upon the mass flow ratio of the vapor to the liquid in the system. They also listed new experimental data for the ascent velocity of the butane bubble droplets in distilled water.

Note that the terms droplet and bubble will be used interchangeably in what follows. Often "droplet" is used to denote liquid and "bubble" to denote vapor, but the distinction will not be used here since some statements may apply to both (and a combination) of the two phases.

CHAPTER II

MATHEMATICAL MODELING

Development of the numerical model will be introduced in this chapter. As was mentioned earlier, it is a model of one- dimensional, three-phase, spray-column direct-contact heat exchanger. The difficulty of calculating multiphase flows is due to the fact that many physical parameters are involved in the process. More than one set of velocities, temperatures, masses per unit volume, etc. have to be calculated for each location of the flow. This increases the number of dependent variables in the physical phenomenon. In this model, it is assumed that mass flow rates, temperatures of the continuous and dispersed phases, initial pressure at the bottom of the column, and initial radius of the drops are known variables. Basic variables required to solve with the model are the following;

1. ϕ, Holdup Ratio : This is the ratio of volume of the dispersed phase to the total volume,
2. U_c , Absolute velocity of the continuous fluid,
3. U_d , Absolute velocity of the dispersed fluid,

4. P , Total pressure of the column at a given height,
5. h_c , Enthalpy of the continuous fluid,
6. h_d , Enthalpy of the dispersed fluid,
7. x , Quality of the dispersed fluid,
8. r , Radius of the droplet.

From this list of unknown variables, pressure,enthalpy and quality are the thermodynamic properties. Therefore knowing two out of three is enough to determine all of the thermodynamic properties. For preheater and superheater region of the column, pressure and enthalpy were taken as the basic thermodynamic properties . In the boiling region, pressure and quality were the chosen thermodynamic properties. By doing so, this total number of unknown variables reduced to seven.

A. Formulation of Governing Equations

The governing equations for one dimensional three phase flow are presented in this section. The basic assumptions to derive the governing equations can be summarized as follows:

1. The governing equations will hold for each phase separately, and for the total flow.

2. Momentum diffusion terms representing viscous

force will be neglected.

3. Conversion of mechanical and potential energy to thermal energy is neglected due to their small magnitude.

4. All of the equations are assumed to be in steady state.

5. Mass flow rates both for continuous and dispersed phases, remain constant. (Stripping of water due to evaporation into the dispersed phase is neglected.)

By using assumption 5, the continuity equations can be written as

$$m_d = \rho_d A \phi U_d \qquad \text{II-1}$$

$$m_c = \rho_c A (1-\phi) U_c \qquad \text{II-2}$$

where m_d, ρ_d and U_d are the mass flow rate, density and velocity of the dispersed fluid respectively. A subscript c refers to the continuous fluid. The holdup ratio, ϕ, is the volume of the dispersed fluid for a unit volume of the total fluid.

A momentum equation for the total flow can be written as

$$\frac{d}{dz}[\rho_d U_d^2 \phi] + \frac{d}{dz}[\rho_c U_c^2 (1-\phi)] = -\frac{dP}{dz} - [\rho_c (1-\phi) + \rho_d \phi] g \qquad \text{II-3}$$

where z is the vertical coordinate measured from the bottom of the column. The variables, P and g are pressure and gravitational constants, respectively. All of the other variables are the same as the continuity equations. Substituting continuity equations II-1 and II-2 into II-3, the momentum equation becomes

II-4 $$\frac{dP}{dz} = -\frac{m_d}{A}\frac{dU_d}{dz} - \frac{m_c}{A}\frac{dU_c}{dz} - [\rho_c(1-\phi) + \rho_d\phi]g$$

Note that the third term on the right-hand side of the equation shows the hydrostatic pressure difference. The first two terms in the right-hand side indicate the effect of the velocity changes on the pressure change.

Similarly the energy equations governing one-dimensional steady-state flow for dispersed and continuous fluids can be written as;

II-5 $$\frac{d}{dz}[\rho_d \phi U_d h_d] = \frac{Q_d}{V}$$

II-6 $$\frac{d}{dz}[\rho_c (1-\phi) U_c h_c] = -\frac{Q_c}{V}$$

In these equations Q_d indicates the heat transferred to the dispersed fluid and Q_c indicates the heat transferred from the continuous phase. The total heat transfer for the volume V is

II-7 $$Q_c = Q_d + Q_L$$

Q_L, in the equation II-7 is the heat loss. If all the terms are divided by Q_d then

II-8 $$\frac{Q_c}{Q_d} = 1 + \frac{Q_L}{Q_d} = \eta$$

II-9 $$\frac{Q_L}{Q_d} = \eta - 1$$

where η defines a heat loss coefficient. When equation II-8 is substituted into equation II-6 , it becomes

II-10 $$\frac{d}{dz}\left[\rho_c\,(1-\phi)\,U_c\,h_c\right] = -\,\eta\,\frac{Q_d}{V}$$

For the case of heat loss to the surroundings, η will be greater than unity as it was seen from the definition. Substituting the continuity equations into equations II-10 and II-5 results in the following relationship.

II-11 $$\frac{dh_d}{dz} = \frac{A}{m_d}\,\frac{Q_d}{V}$$

II-12 $$\frac{dh_c}{dz} = \frac{A}{m_c}\,\frac{\eta\,Q_d}{V}$$

The heat transfer to the bubble, Q_d, is one of the major issues to be addressed in order to solve the model. While several equations exist in the literature, none of them gives a good physical definition of the heat transfer to the bubble. Some of these equations are discussed in the introduction section, and these equations are also listed in Appendix A. It should be noted that all these approaches represent only the resistance to the heat transfer between the liquid phase of the dispersed fluid and the continuous fluid. Among all these equations the equation of Tochitani et al. [A-1-1] was found to be best representation for this resistance. To allow a more realistic representation of the heat transfer, the resistance coefficient for the vapor phase of the bubble must be incorporated. In this study, it was assumed that convection in the gas phase of a small bubble was negligible. Therefore a conduction model may represent the energy flow in the vapor phase of the droplet.

Carlslaw and Jaeger [3] give a conductive heat transfer solution for a sphere where the surface temperature varies linearly with time. This solution was used as a representation of the bubble rising through the continuous phase which itself is flowing in the opposite direction. Because the entrance of the continuous phase is at the top of the column, the temperature of the fluid surrounding the bubble increases with height. Even though the increase of the continuous liquid temperature with height is not exactly linear, this may be an acceptable approximation. The solution for the temperature of the bubble is given by these authors as;

$$\text{II-13} \qquad T = \gamma\left[t - \frac{R^2 - r^2}{6\alpha}\right] - \frac{2\gamma R^3}{\alpha\pi^3 r}\sum_{i=1}^{\infty}\frac{(-1)^i}{i^3}\, e^{-\alpha i^2\pi^2 t/R^2}\sin\frac{i\pi r}{R}$$

Where γ is the proportionality coefficient of the linear variation of the bubble surface temperature, α is the thermal diffusivity, R radius of the bubble and r is the radial coordinate measured from the center of the bubble. The average temperature of the droplet given by the authors as

$$\text{II-14} \qquad T_{avg} = \gamma\left(t - \frac{R^2}{15\alpha}\right) + \frac{6\gamma R^2}{\alpha\pi^4}\sum_{i=1}^{\infty}\frac{1}{i^4}\, e^{-\alpha i^2\pi^2 t/R^2}$$

From the Fourier's conduction law:

$$\text{II-15} \qquad Q = -k\mathbf{A}\frac{\partial T}{\partial r}$$

By taking only the steady-state portion of the temperature profile Fourier's equation yields the following.

$$\text{II-16} \qquad Q = -k\mathbf{A}\frac{\gamma R}{3\alpha}$$

Meanwhile heat transfer by the convection can be expressed

$$\text{II-17} \qquad Q = h\,\mathbf{A}\,(T_{avg} - T_c) = h\,\mathbf{A}\,(\gamma t - \frac{\gamma R^2}{15\alpha} - \gamma t) = -\,h\,\mathbf{A}\,\frac{\gamma R^2}{15\alpha}$$

by equating equations II-16 and II-17, a relation for the convective heat transfer coefficient results.

$$\text{II-18} \qquad Nu_v = \frac{h_v\,D}{k_v} = 10$$

Subscript v in this equation stands for the vapor phase. The combination of the liquid and vapor components of the heat transfer can be written as an area weighted basis.

$$\text{II-19} \qquad h = h_l\,\frac{\mathbf{A}_l}{\mathbf{A}} + h_v\,\frac{\mathbf{A}_v}{\mathbf{A}}$$

In this equation $\mathbf{A}$ represents the total surface area of the bubble, and $\mathbf{A}_l$ and $\mathbf{A}_v$ indicate the area of the droplet contacted with the bubble phase and the area of the droplet in contact with the liquid phase, respectively. As it was mentioned earlier, Tochitani et al. [35] used the following relation

$$\text{A-3-1} \qquad Nu_c = (Pr_c\,Re_c)^{1/3}\,\{0.466\,(\pi - \beta + 0.5\,\sin 2\beta)^{2/3}\}$$

The terms in the brackets, {}, actually represent the area ratio, $\mathbf{A}_l/\mathbf{A}$. Considering the total area ratio to be equal to one, the total heat transfer coefficient for the evaporation region of the column can be written as

$$\text{II-20} \quad h = (k_l/2R)(Pr_c\ Re_c)^{1/3}\{0.466\ (\pi - \beta + 0.5\ \sin 2\beta)^{2/3}\}$$

$$+\ (5\ k_v/R)\ \{1 - 0.466\ (\pi - \beta + 0.5\ \sin 2\beta)^{2/3}\}$$

The only aspect remaining to be completed is the description of the heat transfer in the preheating section of the column. The important part here is that the internal resistance of the droplet is not negligible. The heat transfer coefficient is given by Sideman and Shabtai [27] as follows:

$$\text{A-10-1} \qquad Nu_d = 0.00375\ (\ Re_d\ Pr_d\ )\ \frac{1}{1 + \frac{\mu_c}{\mu_d}}$$

The outside heat transfer coefficient is calculated with the Tochitani equation, and then the inside and outside heat transfer coefficients are combined as series resistances to yield the overall resistance.

The basic governing equations that have been described so far are suitable to determine five unknowns. Two more equations are still needed to solve all seven dependent equations that describe the physical phenomenon in the direct contact boiler. Empirical relations are used to solve for these additional unknowns. One of the relationships for the velocity variation was given by Raina and coworkers [18] in the following form:

$$\text{A-8-3} \qquad U = \frac{1.1547\left[\left\{ 1 - \frac{\rho_d}{\rho_c} \left(\frac{D_0}{D} \right)^3 \right\} \left(\frac{D\, g}{C_D} \right) \right]^{1/2} D^{\left(\frac{5}{6} \; \frac{1}{T_c} D \right)}}{\left[\frac{T_c^2 + T_d^2}{2\, T_c T_d} \right]^{D} \left[\frac{Cp_c \mu_c}{k_c} \right]^{\frac{D_0}{1.6D}}}$$

More detailed information about this equation is given in Appendix A. The variable C_D in equation A-8-3 is the drag coefficient. To calculate the drag coefficient, an experimentally determined curve-fitting of data for a solid sphere is combined with a correlation for Stokes formulation for a drag force [36]. The resultant equation is given by

$$\text{II-22} \qquad C_D = \left[\frac{24}{Re} + \frac{6}{1 + Re^{1/2}} + 0.4 \right] \left[\frac{1 + \frac{2\, \mu_c}{3\, \mu_d}}{1 + \frac{\mu_c}{\mu_d}} \right]$$

Equation II-A-22 was compared with several other empirical drag coefficient equations and the results are found to be satisfactory. Some other drag coefficient relationships are also included in the model and the results of these models are compared. One of them is the following, II-23 $$C_D = \frac{8}{3} \frac{\rho_d g}{1.82 \sigma} R^2$$

σ is the surface tension at the bubble- surrounding liquid boundary.

The last relation needed is derived from the continuity equation. The radius of the droplets is given as

II-24 $$R = R_0 \left[\frac{n_b \rho_c U_d}{n_{b0} \rho_{c0} U_{d0}}\right]^{1/3}$$

where o indicates the initial conditions and n_b is the number of droplets per unit volume for a given distance. n_b is given by

II-25 $$n_b = \frac{\phi}{4/3 R^3}$$

If there is no coalescence or breakup of the bubbles, n_b is constant throughout in the column. Therefore equation II-A-24 reduces to

II-26 $$R = R_0 \left[\frac{\rho_c U_d}{\rho_{c0} U_{d0}}\right]^{1/3}$$

With these additional relationships, the appropriate number of equations is obtained to solve the system of equations in order to examine the behavior of direct contact boiling.

B. Formulation of Combined Temperature

In order to compare the experimental temperature profiles to the model that computes separate temperatures for both continuous and dispersed fluids throughout the column, one final item is necessary. This is a special formulation to approximate the mix temperature when the temperature and physical conditions of the two phases are known. This is necassary because any experimentally used temperature measuring device will necessarily sense some time- averaged value of the combined effects of the dispersed and continuous fluid temperatures.

Consider a volume region in the column passing by the thermocouple at the statistical average temperature reading time Δt. If the total volume of each fluid passing by the thermocouple during this time and heat transfer coefficients of each phase are considered as the determining factors of the final thermocouple reading, the mixed temperature reading will be proportional to these properties. The total distance covered for each fluid during this time interval along with the total volume can be written as shown in the equations below regardless of their sign,

II-27 $$\text{Volume(dispersed)} = A_d \, \Delta x_d = A \, \phi \, \Delta x_d = A \, \phi \, U_d \, \Delta t$$

II-28 $$\text{Volume(continuous)} = A_c\,\Delta x_c = A(1-\phi)\Delta x_c = A(1-\phi)U_c\,\Delta t$$

Here $U_d\Delta t$ and $U_c\Delta t$ are the distances both fluids travel during the time, assuming that the flows are one dimensional. By considering Equation II-B-1 and II-B-2 the mixed temperature reading should be as follows.

II-29 $$T_{mix} = \frac{h_{dt}\,\phi\,U_d\,T_d + h_{ct}\,(1-\phi)\,U_c\,T_c}{h_{dt}\,\phi\,U_d + h_{ct}\,(1-\phi)\,U_c}$$

h_{dt} and h_{ct} are the heat transfer coefficients for the dispersed and continuous fluids. The heat transfer coefficients are calculated by using the Churchill and Bernstein [4] equation. This equation had been developed for circular cylinders in cross flow which covers the entire range of Reynolds number for which the data is available, as well as for a wide range of Prandtl number. The equation is in the following form

II-30 $$Nu = 0.3 + \frac{0.62\,Re^{1/2}\,Pr^{1/3}}{[1+(0.4/Pr)^{2/3}]^{1/4}}\left[1+\left(\frac{Re}{28200}\right)^{5/8}\right]^{4/5}$$

C. Computational Methods

In carrying out the numerical solution, differential equations are solved by using the Euler method.

The Euler method is one of the simpler approximation methods for solving the initial value problems. It is also called the tangent method. One divides the interval [a,b] into subintervals of length Δz , $z_i = a + i\ \Delta z$ sets $y_0 = C$, and seeks approximate values y_1, y_2... to the exact values $y(z_1)$, $y(z_2)$... by approximating the derivative at the point (z_n, y_n) with the difference quotient $(y_{n+1} - y_n)/\Delta z$. This gives the equation

$$\text{II-31} \qquad \frac{y_{n+1} - y_n}{\Delta z} = f(z_n, y_n)$$

or, the recursion formula

$$\text{II-32} \qquad y_{n+1} = y_n + \Delta z\ f(z_n, y_n)$$

with the boundary condition of $y_0 = C$.

In our case two out of three differential equations are given with initial values. However the boundary condition of the continuous phase energy equation is given as an end condition. In order to determine the initial condition for this equation a shooting method is used. In order to find the initial value, a first guess of an initial value range should be given as an input. This initial value range should be chosen so that the possibility of having the real initial value in this range will be high. Five equidistant points in this region are taken as initial values and the equation is solved. Each solution results with an end condition $y_n = C_n$. A third degree polynomial least square curve fit is applied to solve for the initial value as a function of the end boundary condition. Finally by using this polynomial function, the initial value for the given end boundary value is determined. A Gauss elimination method with pivoting is applied in order to solve the least square matrix.

The computational order progresses in the following manner. First the velocity of the dispersed phase is calculated. Then the holdup ratio and the continuous velocity are found from the continuity equation. Then the droplet radius is found. Initial derivatives of the velocities are assumed to be zero and then the derivatives of the velocities are calculated from the stepwise change.

II-33 $$\frac{\Delta U}{\Delta z} = \frac{U_i - U_{i-1}}{\Delta z}$$

The momentum and energy equations are solved by using

II-34 $$P_i = P_{i-1} + \Delta z \left(\frac{dP}{dz}\right)_i$$

II-35 $$h_{c\,i} = h_{c\,i-1} + \Delta z \left(\frac{dh_c}{dz}\right)_i$$

II-36 $$h_{d\,i} = h_{d\,i-1} + \Delta z \left(\frac{dh_d}{dz}\right)_i$$

Functional values of pressure and enthalpies are calculated from these equations. In the boiling region the quality of the dispersed phase is calculated from the following equation

II-37 $$x_i = x_{i-1} + \Delta z \left(\frac{dx}{dz}\right)_i$$

Then the actual enthalpy of the point is found from the thermodynamic relations of the saturation state. The thermodynamic properties are calculated by using subroutines developed at the University of Utah [21-23]. Additional subroutines were added to convert existing subroutines from English units to SI Units. Details of accessing these subroutines are given in Appendix C and the complete listing of the model program including the thermodynamic subroutines is given in Appendix D. In the subroutines for water, the basic equation used in deriving the thermodynamic properties is the Helmholtz free energy equation. All the other properties are derived from the Helmholtz free energy equation. These sets of equations are the same ones that Keenan and Keyes Steam Tables originally used. The Starling-Benedict-Webb-Rubin equation of state is the basic equation used to calculate thermodynamic properties of various types of freons and hydrocarbons. Transport properties such as surface tension and conductive heat transfer coefficients are supplied to the model as temperature dependent curve fit from published values [37]. Viscosities are taken as temperature and pressure dependent curve fits. Specific heats at constant pressure are evaluated by using thermodynamic property subroutines. For the preheater region

$$\text{II-38} \qquad C_p = \frac{h_i(T_i) - h_i(T_i - \Delta T_i)}{\Delta T_i} \Bigg|_{p\,=\,\text{constant}}$$

and for the superheater region

$$\text{II-39} \qquad C_p = \frac{h_i(T_i + \Delta T_i) - h_i(T_i)}{\Delta T_i} \Bigg|_{p\,=\,\text{constant}}$$

The reason for using two different equations is due to numerical difficulties of calculating specific heats close to the boiling points. In the boiling region saturated liquid and gas specific heats are determined separately. For water, specific heat is calculated by equation II-38 for the given range. Transport properties of pentane, butane and isobutane are built into the model. The properties of additional fluids can also be easily added to the existing list of the fluids.

CHAPTER III

RESULTS AND DISCUSSION

The influence of various parameters and their behavior during the process of direct contact evaporation has been investigated. It is necessary to ascertain that the heat transfer model developed here describes the actual performance of such a device in the best way. Steady state data are available for the temperature profiles from the data taken as a part of this project [6] and from the 500 KW Direct contact pilot plant at East Mesa [14]. Figures 2 through 10 show the comparison of the temperature profiles obtained by running the University of Utah direct contact boiler and the calculations of this model. In this thesis, a mixed temperature concept is developed for the first time to approximate analytically the thermocouple readings from the combined dispersed and continuous fluid temperatures. The calculated continuous fluid, dispersed fluid, and mixed temperatures, and experimental thermocouple readings are shown in these figures. Temperature profiles for different mass flow rates and pressures have been examined through these plots. These independent variables were chosen because of the availability of the experimental values and due to their relatively large effect on the heat transfer. The

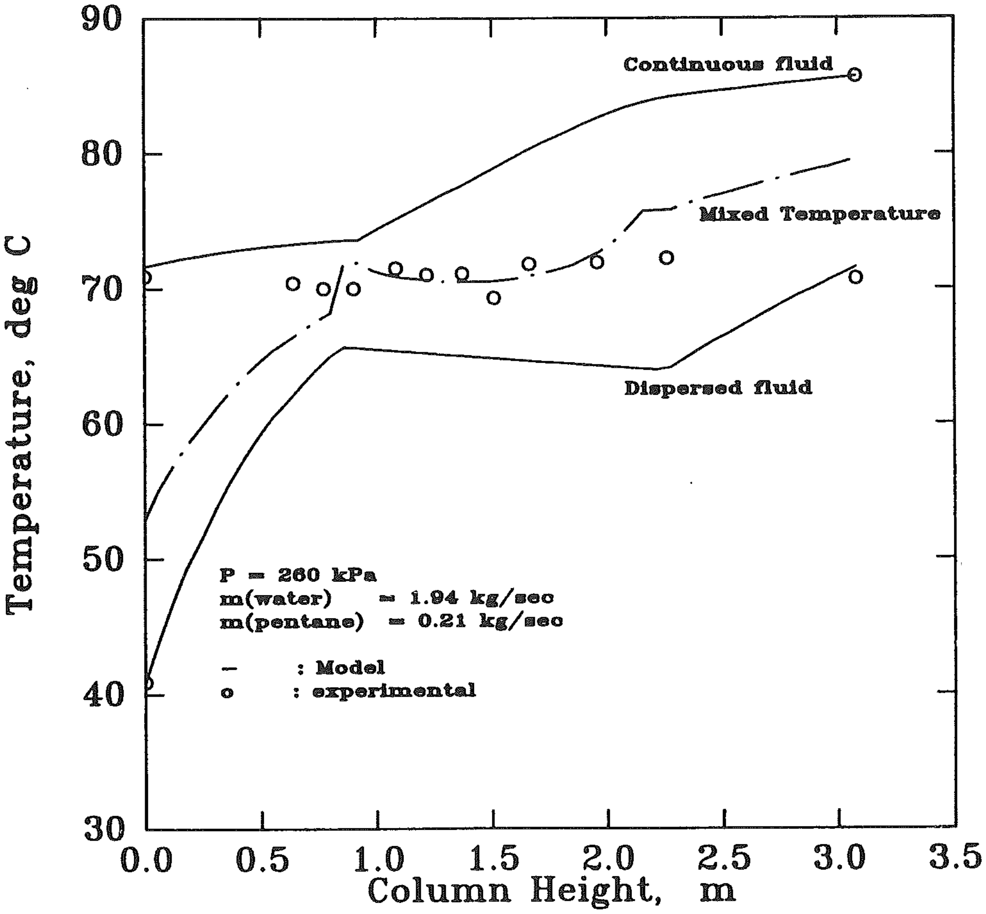

Figure 2. Temperature profile of dispersed and continuous fluid, mixed temperature and experimental temperature profile from Goodwin, Coban and Boehm [6] data along the length of column, m(pentane) = 0.21 kg/sec, m(water) = 1.94 kg/sec, P = 260 kPa, initial drop radius = 2 mm.

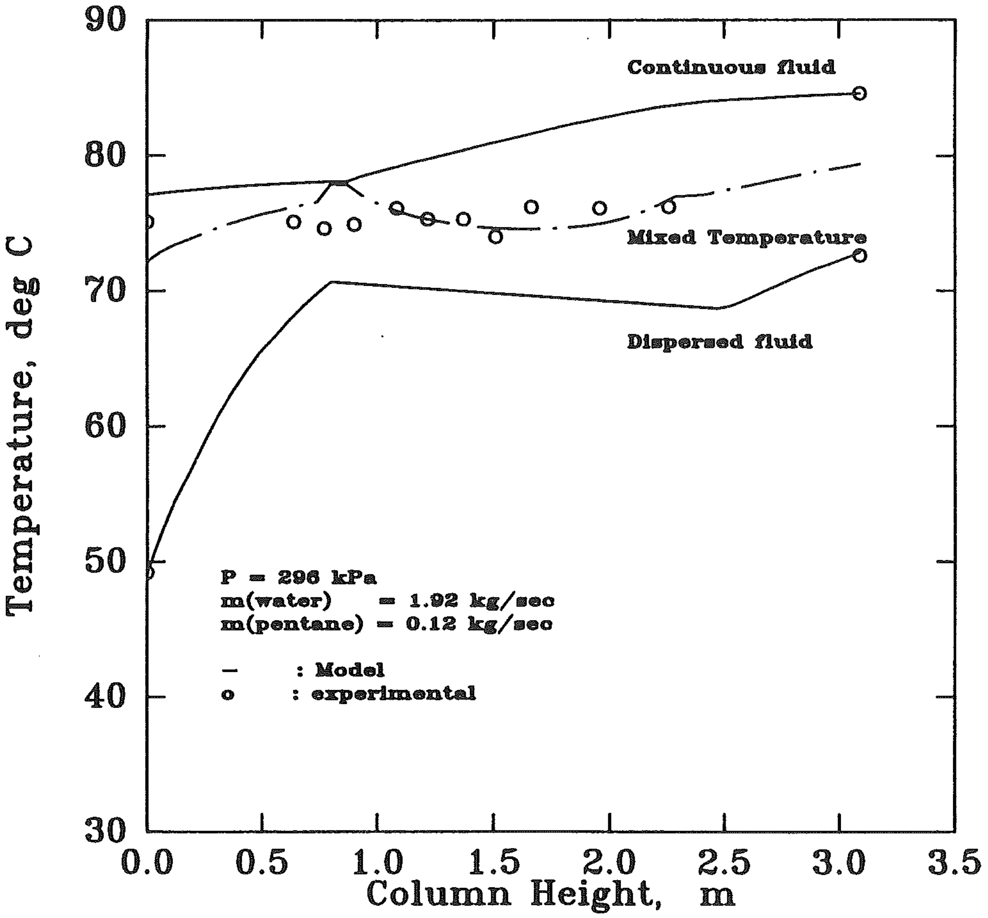

Figure 5. Temperature profile of dispersed and continuous fluid, mixed temperature and experimental temperature profile from Goodwin, Coban and Boehm [6] data along the length of column, m(pentane) = 0.12 kg/sec, m(water) = 1.92 kg/sec, P = 296 kPa, initial drop radius = 2 mm.

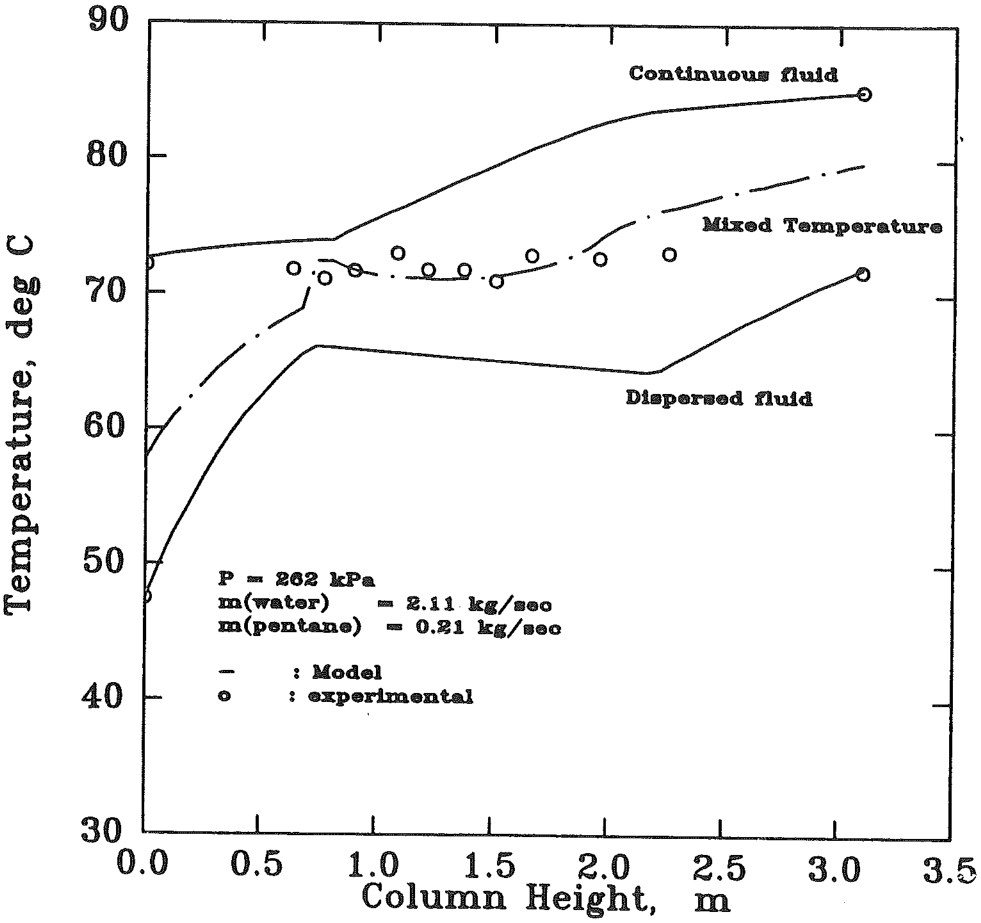

Figure 7. Temperature profile of dispersed and continuous fluid, mixed temperature and experimental temperature profile from Goodwin, Coban and Boehm [6] data along the length of column, m(pentane) = 0.21 kg/sec, m(water) = 2.11 kg/sec, P = 262 kPa, initial drop radius = 2 mm.

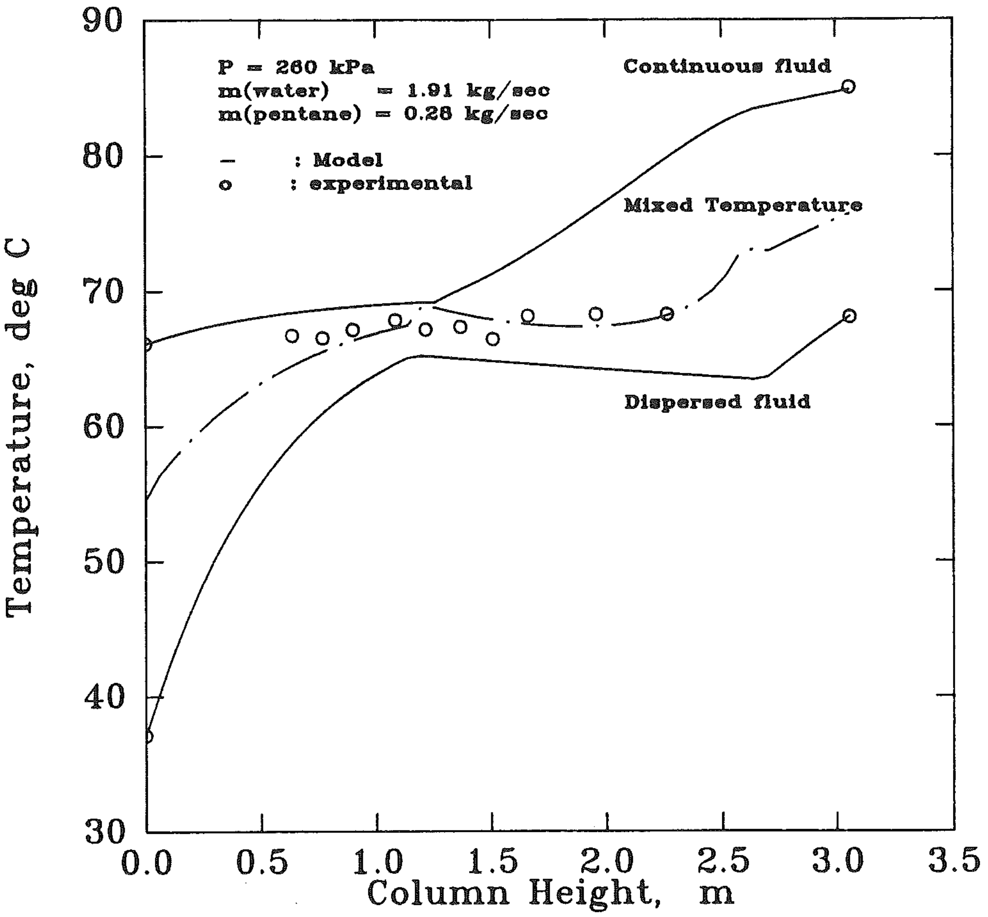

Figure 8. Temperature profile of dispersed and continuous fluid, mixed temperature and experimental temperature profile from Goodwin, Coban and Boehm [6] data along the length of column, m(pentane) = 0.28 kg/sec, m(water) = 2.11 kg/sec, P = 260 kPa, initial drop radius = 2 mm.

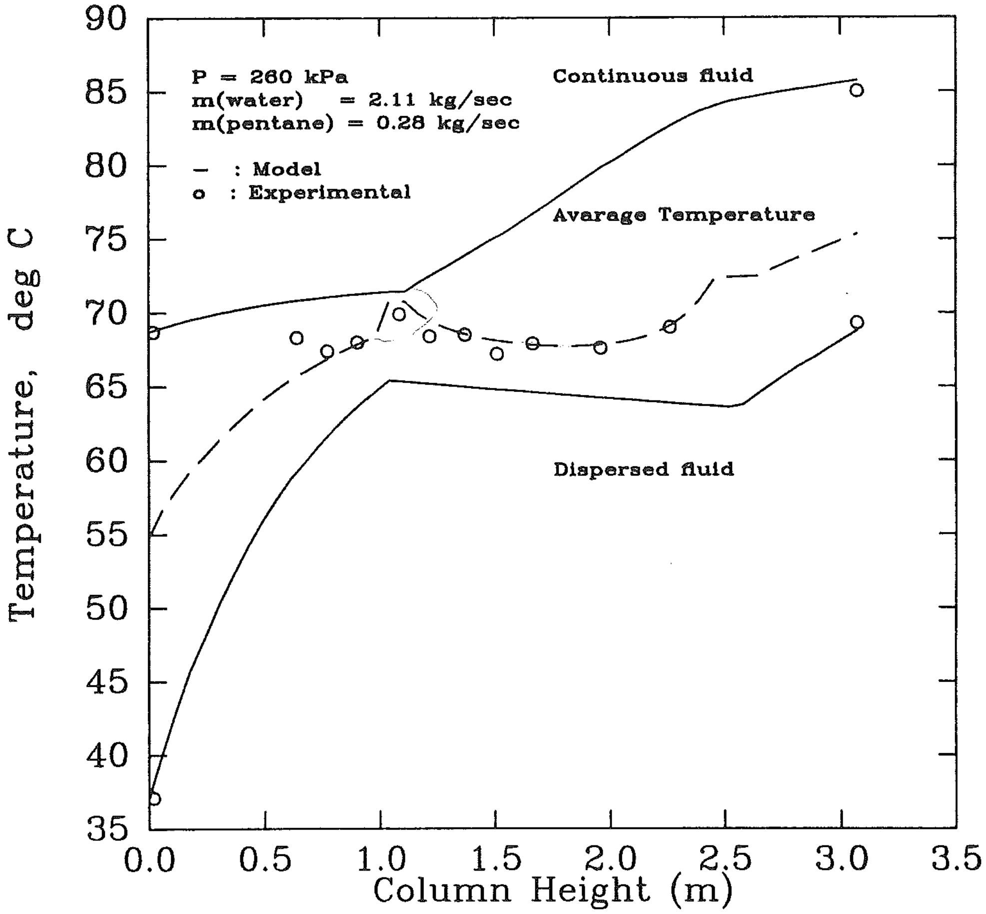

Figure 9. Temperature profile of dispersed and continuous fluid, mixed temperature and experimental temperature profile from Goodwin, Coban and Boehm [6] data along the length of column, m(pentane) = 0.12 kg/sec, m(water) = 2.11 kg/sec, P = 290 kPa, initial drop radius = 2 mm.

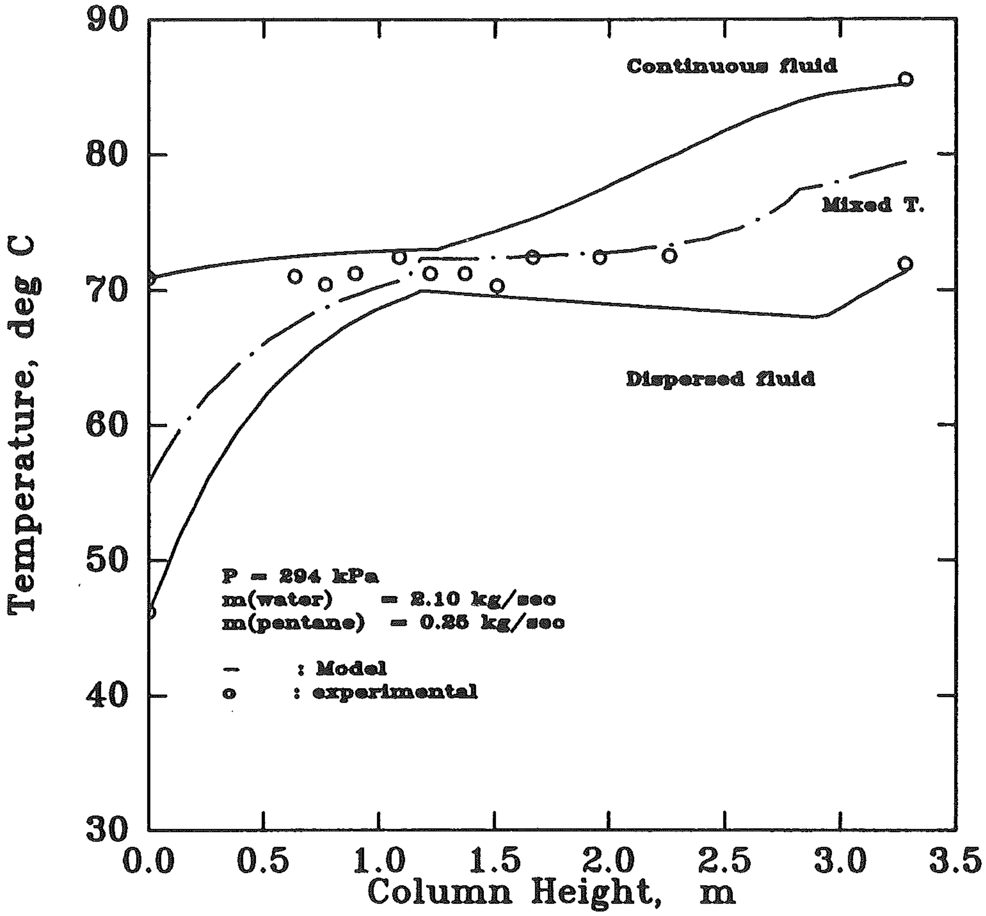

Figure 3. Temperature profile of dispersed and continuous fluid, mixed temperature and experimental temperature profile from Goodwin, Coban and Boehm [6] data along the length of column, m(pentane) = 0.25 kg/sec, m(water) = 2.10 kg/sec, P = 294 kPa, initial drop radius = 2 mm.

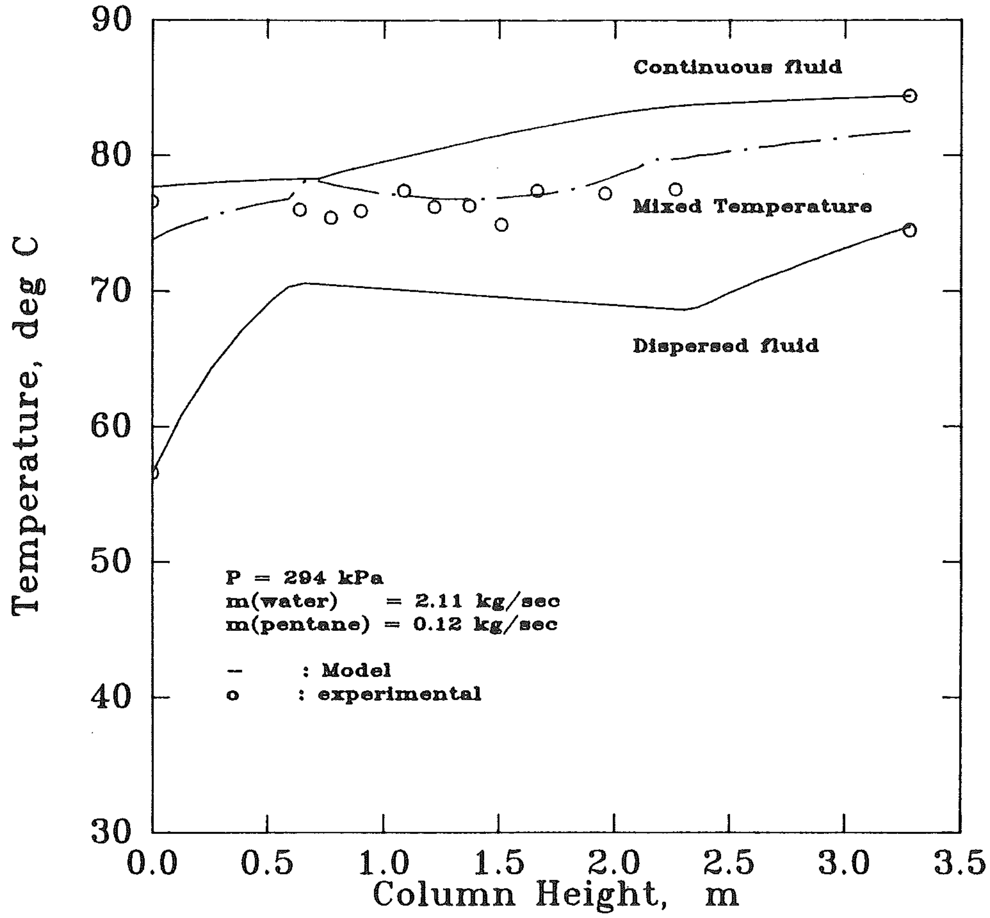

Figure 6. Temperature profile of dispersed and continuous fluid, mixed temperature and experimental temperature profile from Goodwin, Coban and Boehm [6] data along the length of column, m(pentane) = 0.12 kg/sec, m(water) = 2.11 kg/sec, P = 294 kPa, initial drop radius = 2 mm.

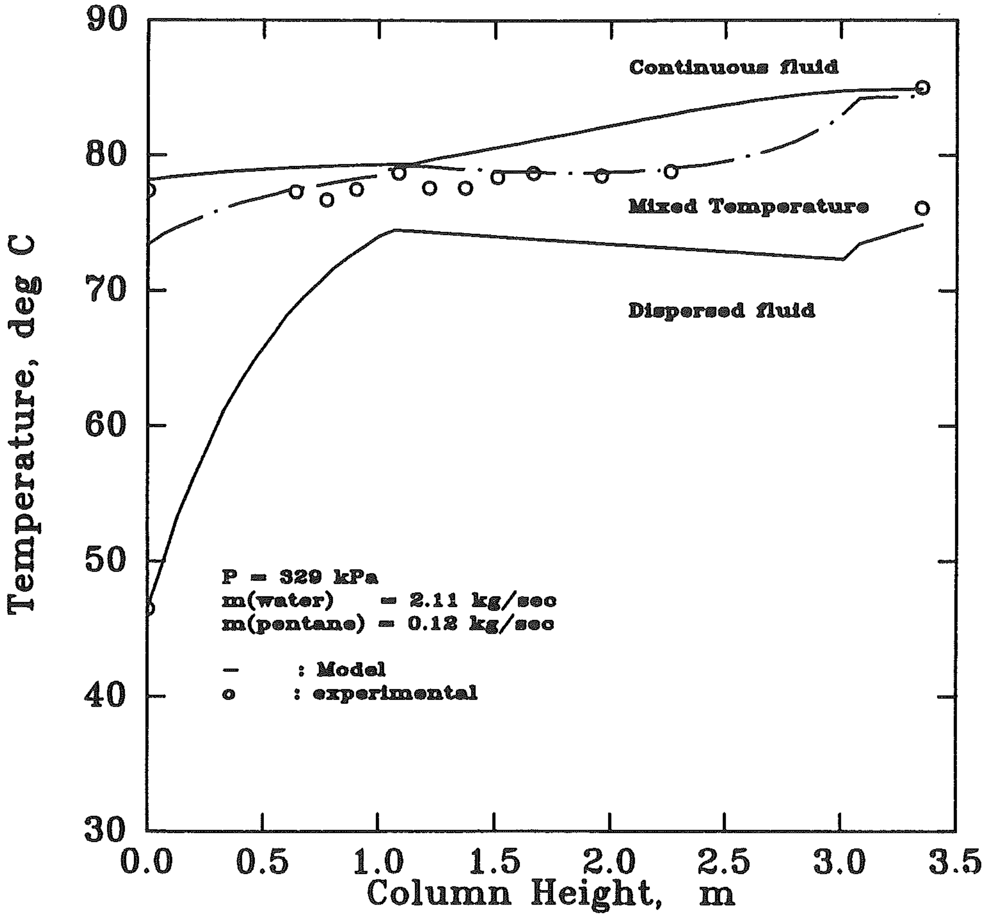

Figure 4. Temperature profile of dispersed and continuous fluid, mixed temperature and experimental temperature profile from Goodwin, Coban and Boehm [6] data along the length of column, m(pentane) = 0.12 kg/sec, m(water) = 2.11 kg/sec, P = 329 kPa, initial drop radius = 2 mm.

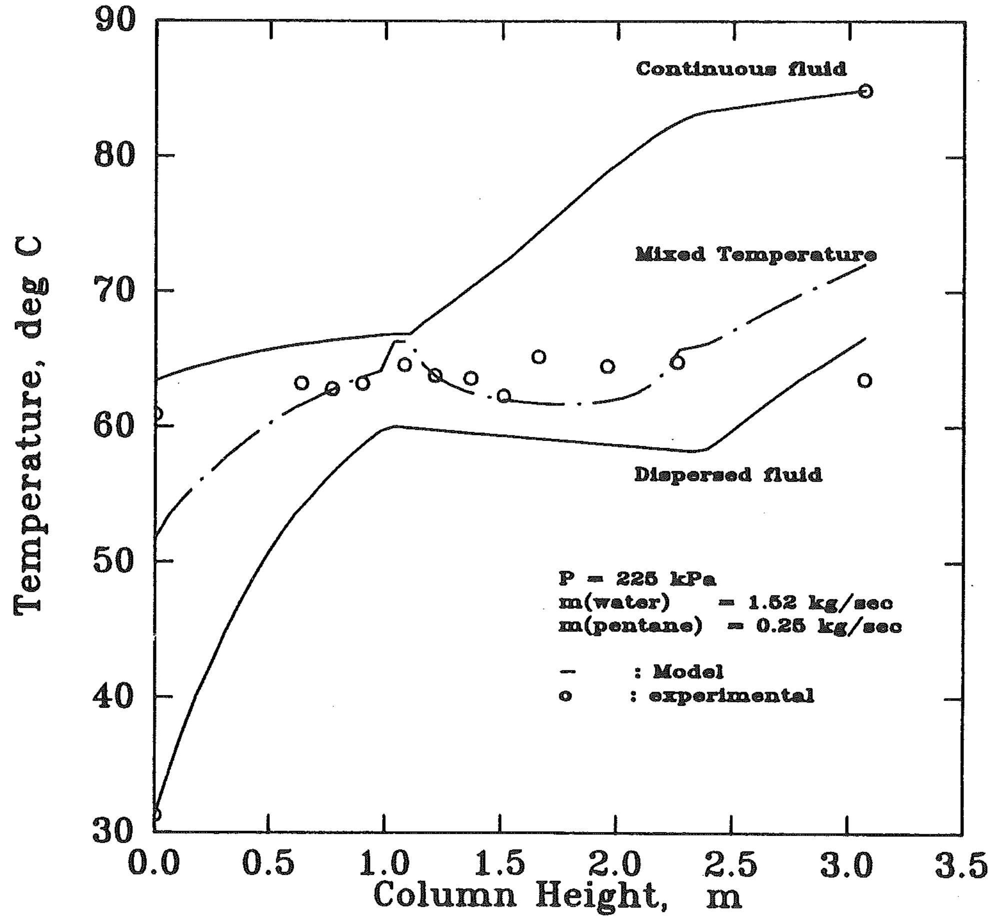

Figure 10. Temperature profile of dispersed and continuous fluid, mixed temperature and experimental temperature profile from Goodwin, Coban and Boehm [6] data along the length of column, m(pentane) = 0.25 kg/sec, m(water) = 1.52 kg/sec, P = 225 kPa, initial drop radius = 2 mm.

physical parameters might not follow this change as quickly and an unsteady region occurs or model prediction of the change of vapor phase might be quicker than the actual change.

Figures 11 through 15 show temperature profiles from the East Mesa plant experimental data. Figure 11 is actually the same data used by Jacobs and Golafshani [8]. Actual comparisons of both model predictions and experimental data points are tabulated. Even though it is hard to predict which temperature profile fits better due to strange heat transfer characteristics of the experimental profile, good fitting end conditions indicates that these profiles are fairly close to the actual temperatures. It is observed that the continuous fluid inlet temperatures are far off from the experimentaly measured temperature in Jacobs and Golafshani profile. The flat temperature region shown in the middle of the column might possibly be due to a mid-platform fin effect or an extra circulation which is impossible to predict by these kinds of models. Figures 12 to 15 also show the East Mesa profile. These profiles have an incomplete boiling section as well as the preheating region. Incomplete boiling indicates a liquid output from the column. This might be the main reason for the very high pressures obtained in this column.

Figures 16 through 20 show the variation of calculated values of quality and holdup ratio. These figures

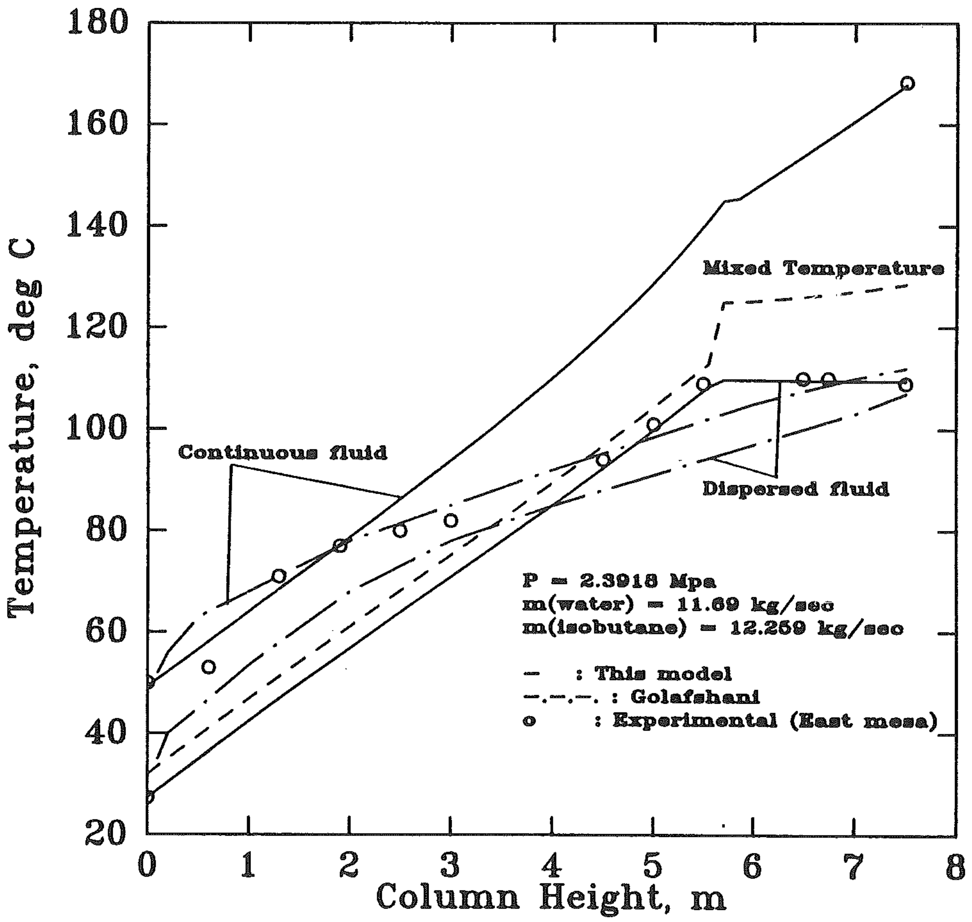

Figure 11. Temperature profile of dispersed and continuous fluid, of this model and the model of Jacobs and Golafshani [8], and mixed temperature, and experimental temperature from East Mesa data [14], along the length of the column, P = 2.39 MPa, m(water) = 11.69 kg/sec, m(isobutane) = 12.26 kg/sec, initial drop radius = 2 mm.

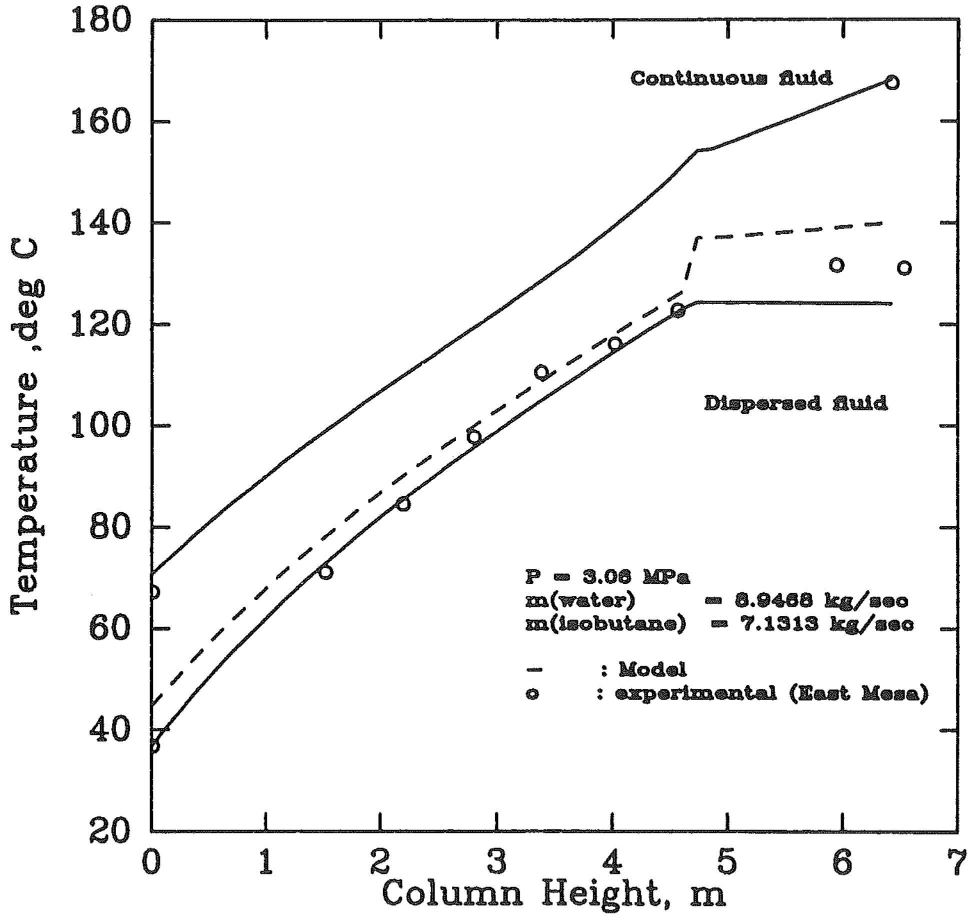

Figure 12. Temperature profile of dispersed and continuous fluid, of, and mixed temperature, and experimental temperature from East Mesa data [14], along the length of the column, P = 3.06 MPa, m(water) = 7.13 kg/sec, m(isobutane) = 8.94 kg/sec, initial drop radius = 2 mm.

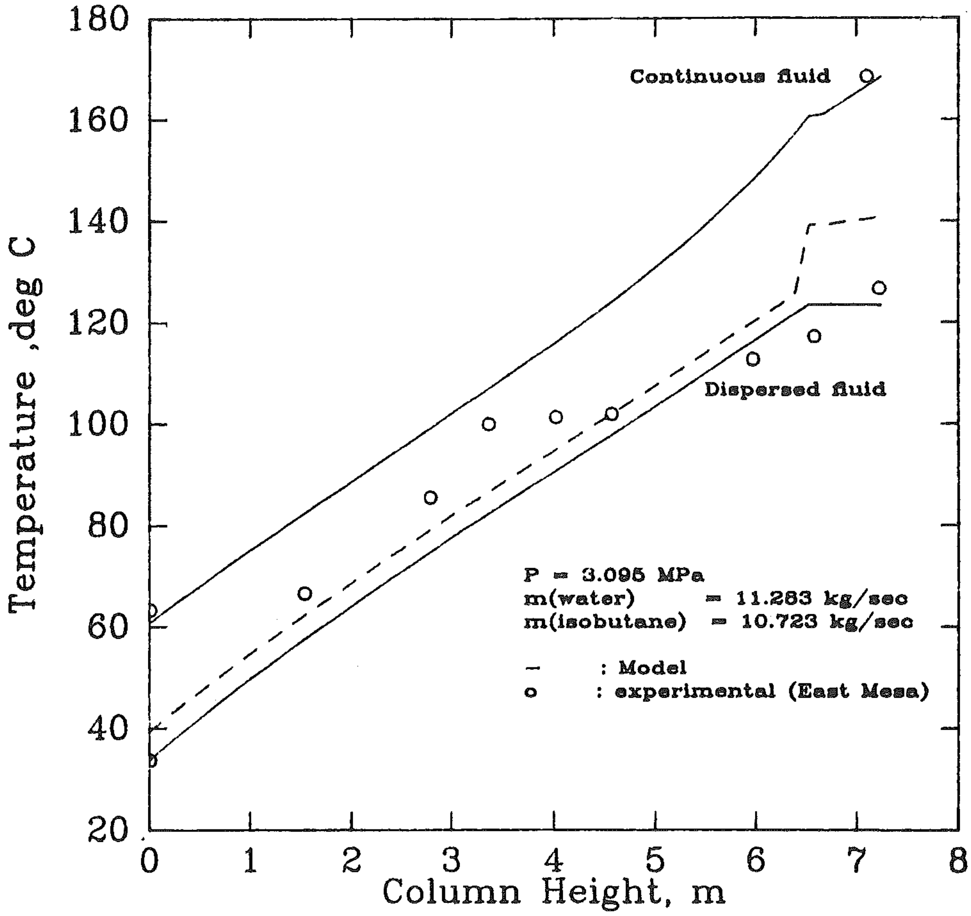

Figure 13. Temperature profile of dispersed and continuous fluid, of, and mixed temperature, and experimental temperature from East Mesa data [14], along the length of the column, P = 3.10 MPa, m(water) = 11.28 kg/sec, m(isobutane) = 10.72 kg/sec, initial drop radius = 2 mm.

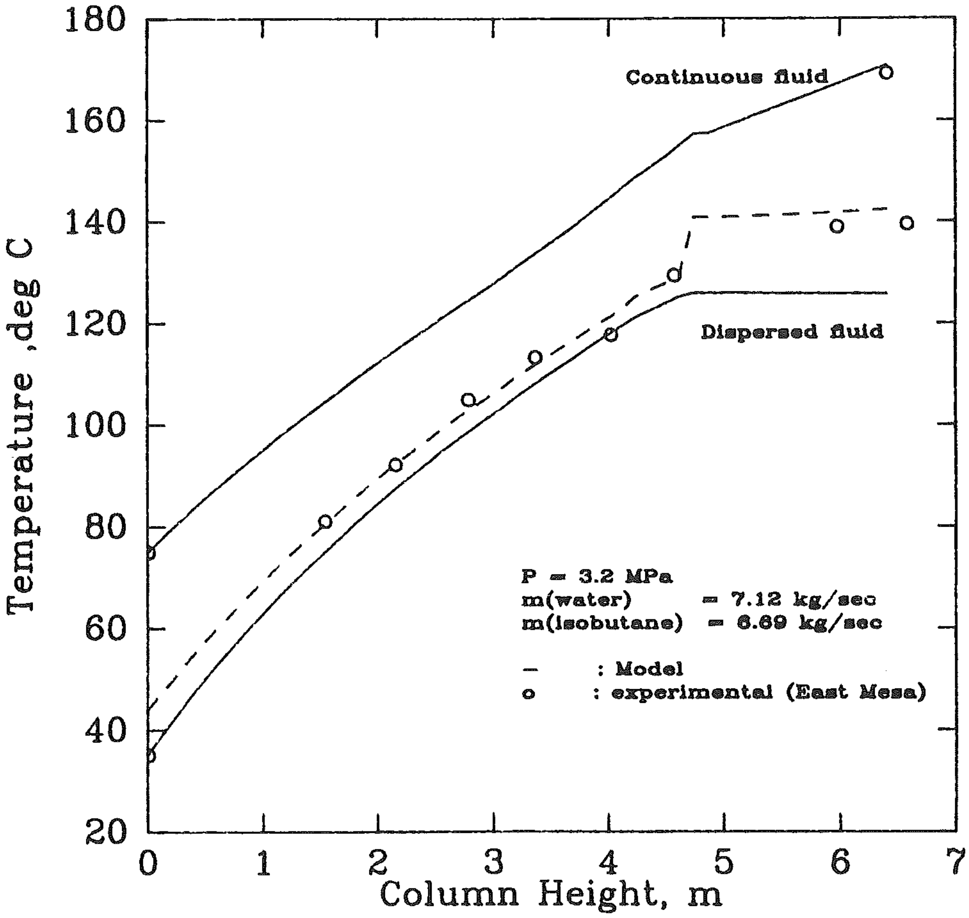

Figure 14. Temperature profile of dispersed and continuous fluid, of, and mixed temperature, and experimental temperature from East Mesa data [14], along the length of the column, P = 3.20 MPa, m(water) = 7.12 kg/sec, m(isobutane) = 6.69 kg/sec, initial drop radius = 2 mm.

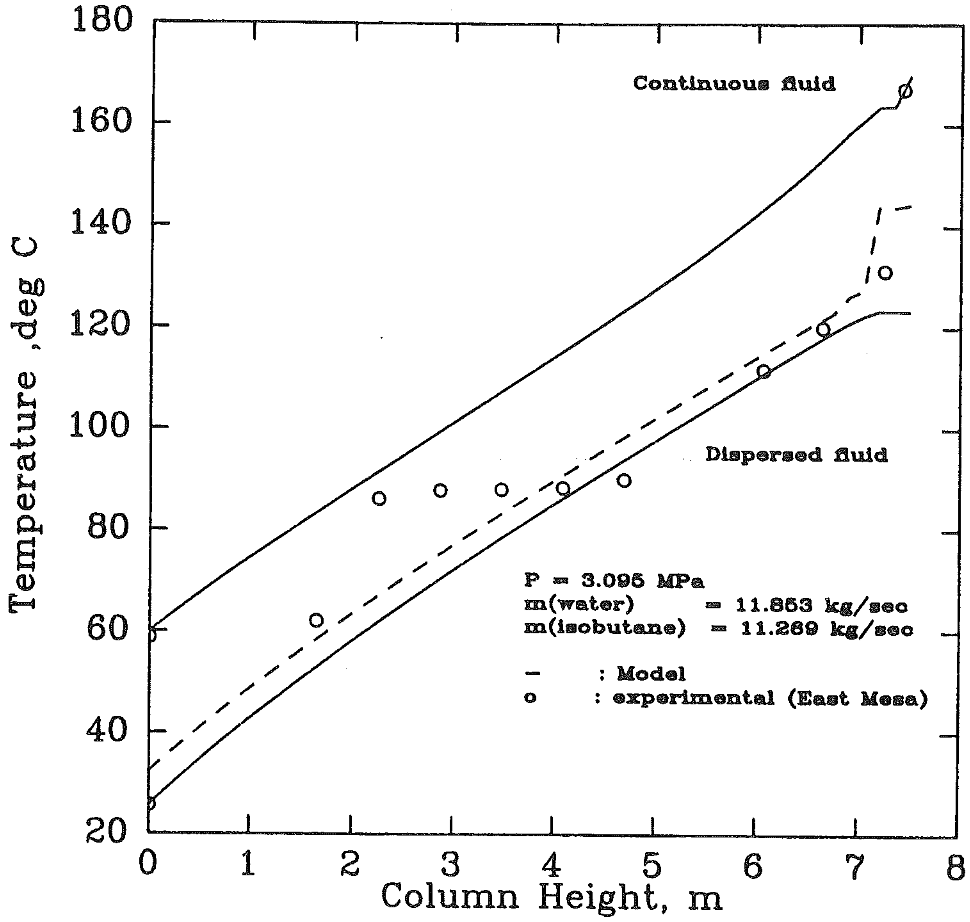

Figure 15. Temperature profile of dispersed and continuous fluid, of, and mixed temperature, and experimental temperature from East Mesa data [14], along the length of the column, P = 3.10 MPa, m(water) = 11.85 kg/sec, m(isobutane) = 11.27 kg/sec, initial drop radius = 2 mm.

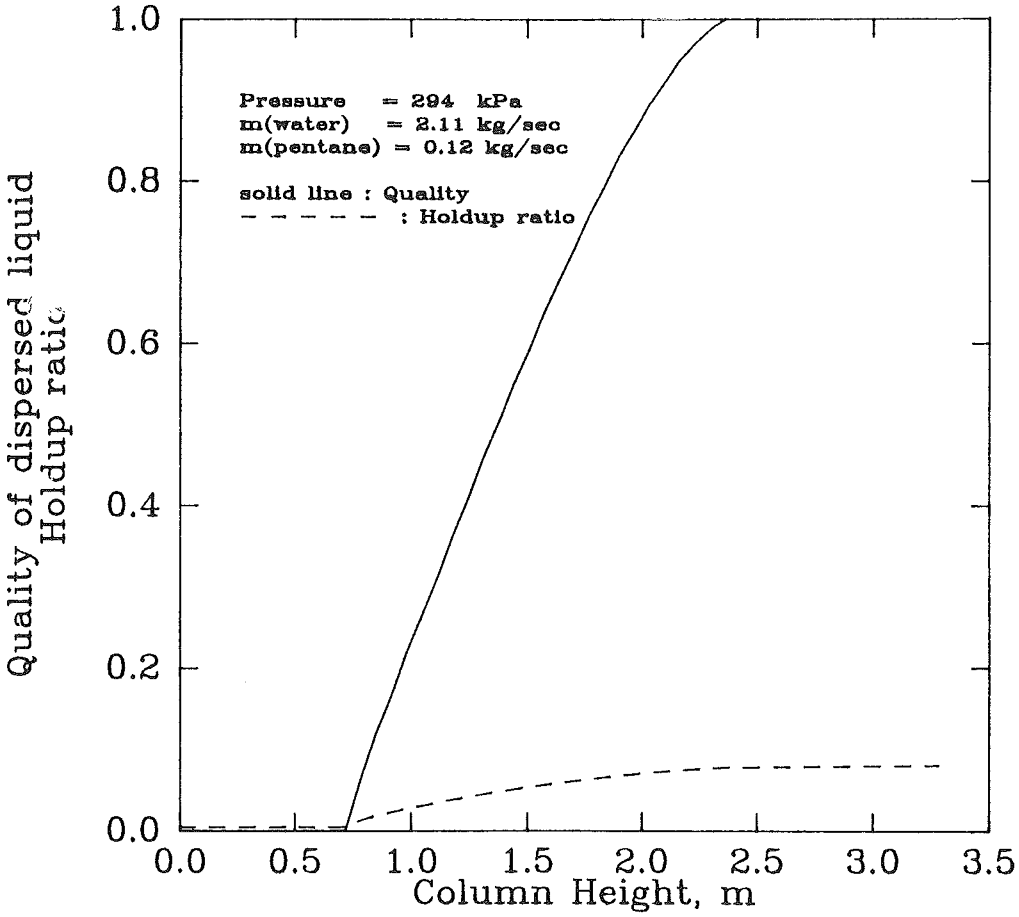

Figure 16. Quality of dispersed liquid and holdup ratio along the column height [6], m(pentane) = 0.12, m(water) = 2.11 kg/sec, P = 294 kPa.

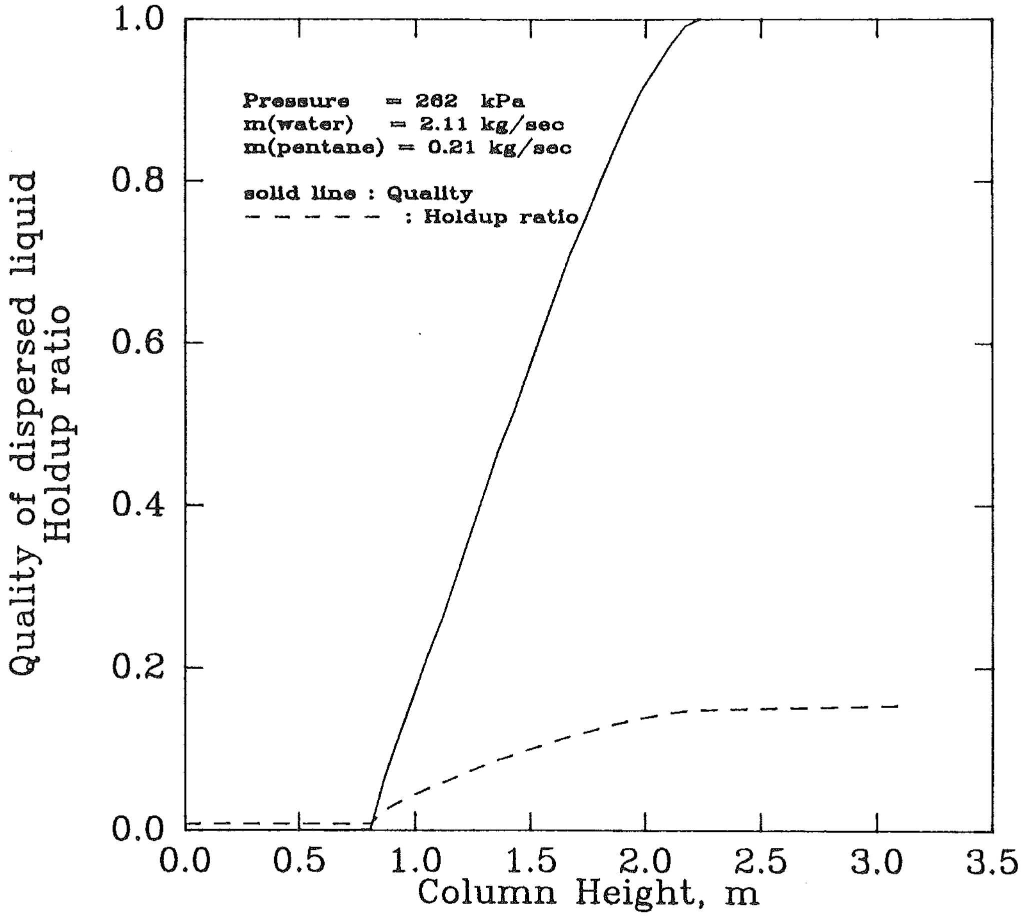

Figure 17. Quality of dispersed liquid and holdup ratio along the column height [6], m(pentane) = 0.21, m(water) = 2.11 kg/sec, P = 262 kPa.

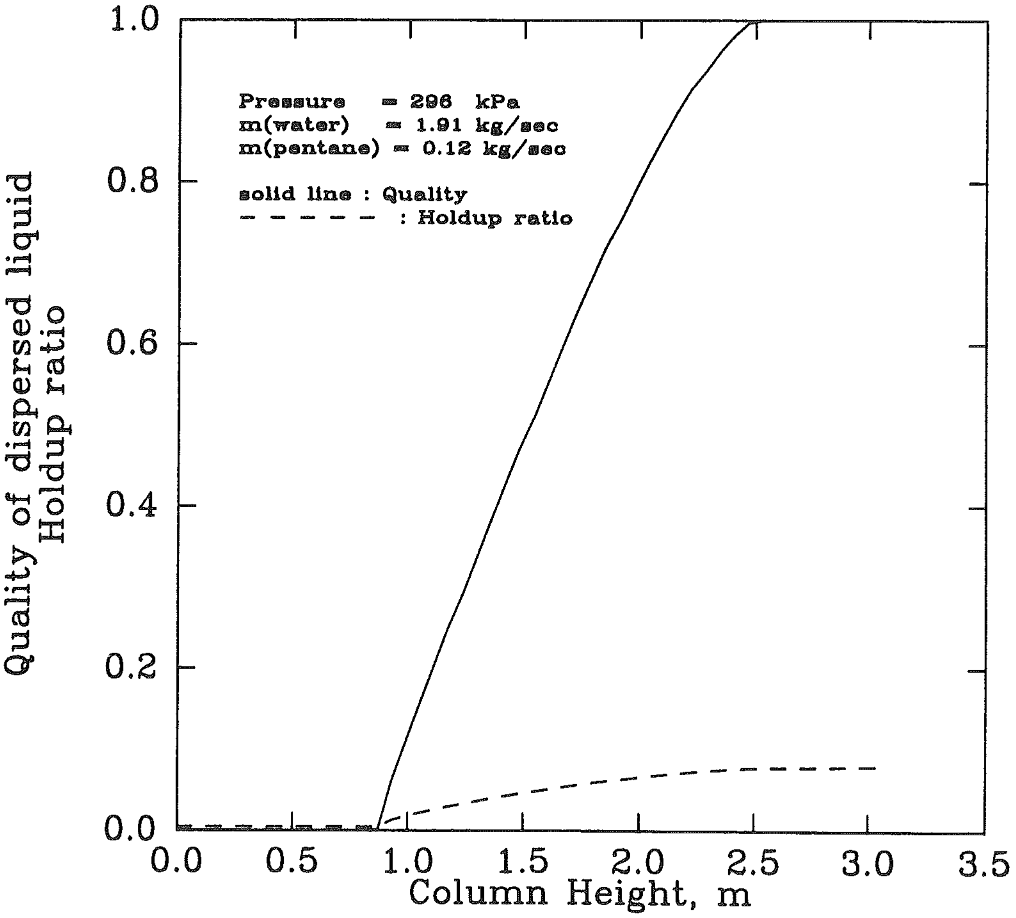

Figure 18. Quality of dispersed liquid and holdup ratio along the column height [6], m(pentane) = 0.12, m(water) = 1.91 kg/sec, P = 296 kPa.

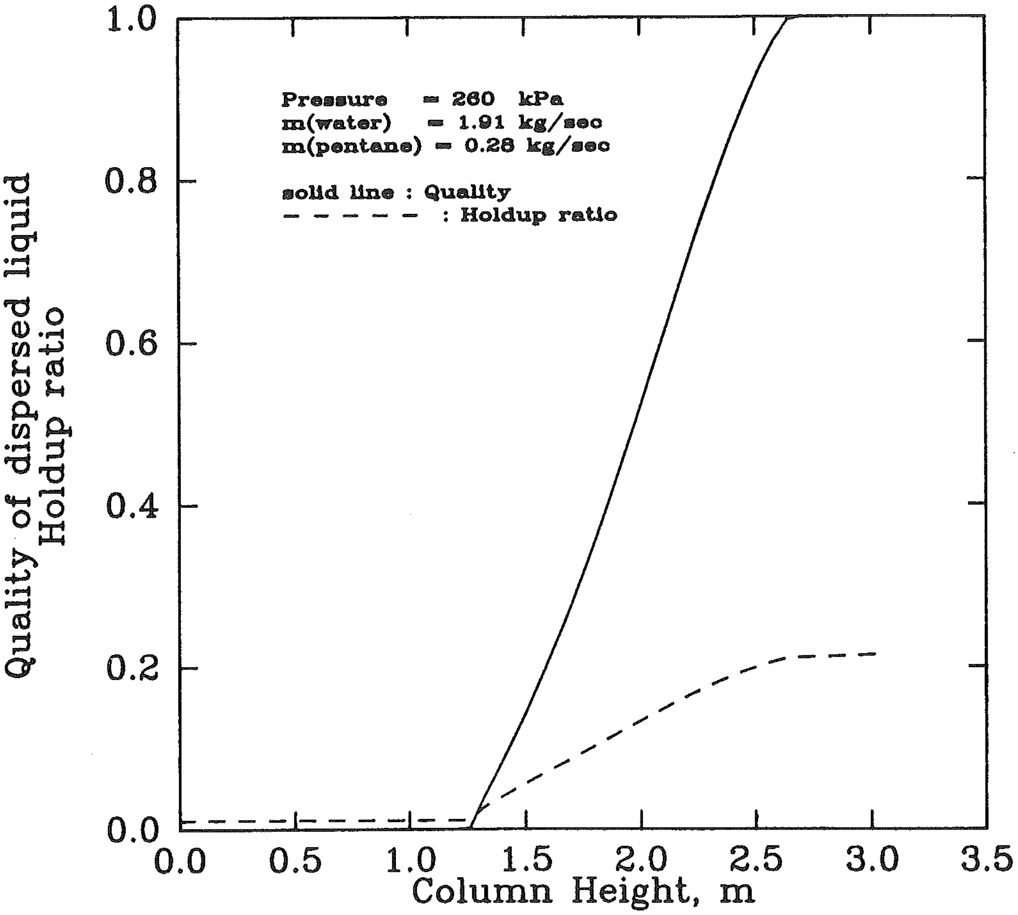

Figure 19. Quality of dispersed liquid and holdup ratio along the column height [6], m(pentane) = 0.28, m(water) = 21.91 kg/sec, P = 260 kPa.

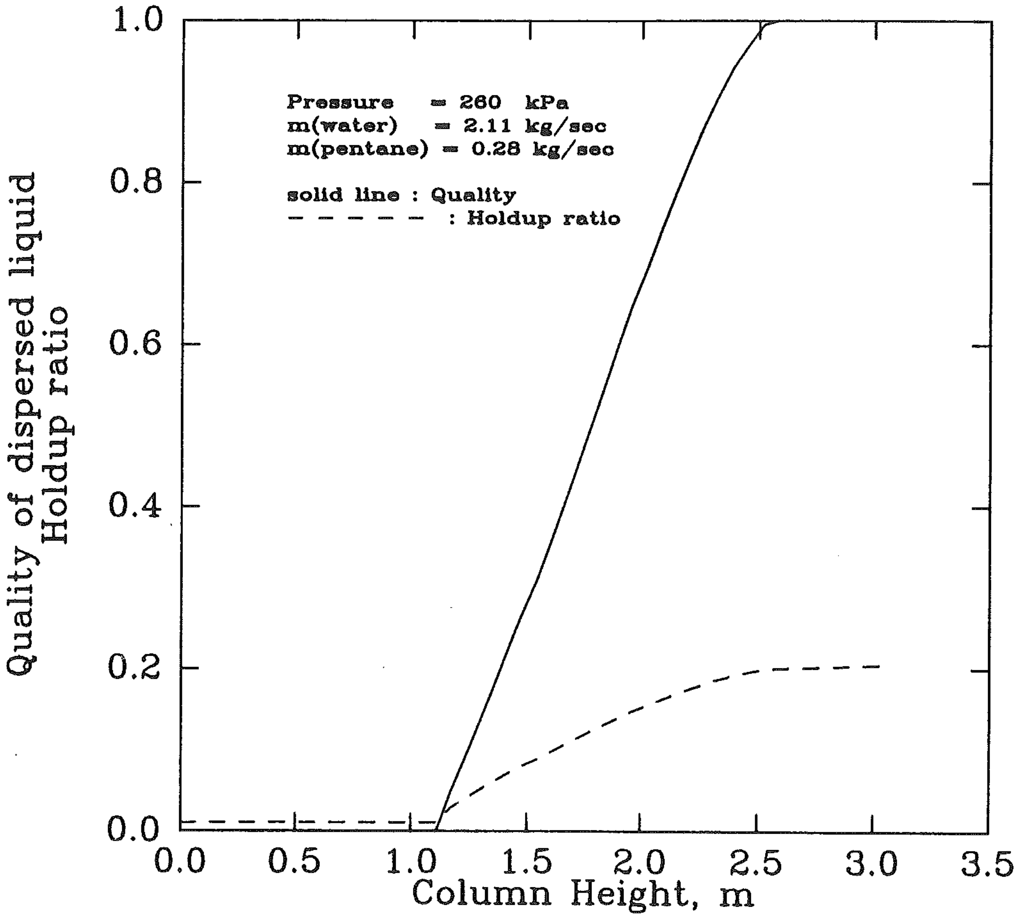

Figure 20. Quality of dispersed liquid and holdup ratio along the column height [6], m(pentane) = 0.12, m(water) = 2.11 kg/sec, P = 294 kPa.

evaporation. Unfortunately there are no experimental data taken to compare the validity of these curves. The holdup ratio curves have a flat profile for the preheating and the superheating regions, and an almost linear second degree curve for the boiling region. The quality profile also varies linearly. Although the bubble size becomes quite large near the end portion of the vaporization process, the bubble spacing widens, causing the holdup to remain moderate. If the holdup ratio grows too large it is almost certain that breakup and coalescence will be introduced to the actual physical process. In this case, model and experimental temperatures may be widely separated .

The effect of several physical parameters on the total heat transfer is the next thing to be investigated. The effect of initial radius of droplet on total heat transfer is shown in Figure 21. It is clearly seen from this figure that the total heat transfer is inversely proportional to the initial drop radius. The sudden change in the slope of the curves is because of incomplete boiling after some critical initial radius for the given column height. This figure also shows that the larger dispersed liquid mass flow rate causes the larger total heat transfer. This was an expected outcome.

In Figure 22 relations of initial dispersed phase and total heat transfer are given for two different initial pressures. As is seen from the figure, the total heat transfer is reduced when the temperature is increased and it

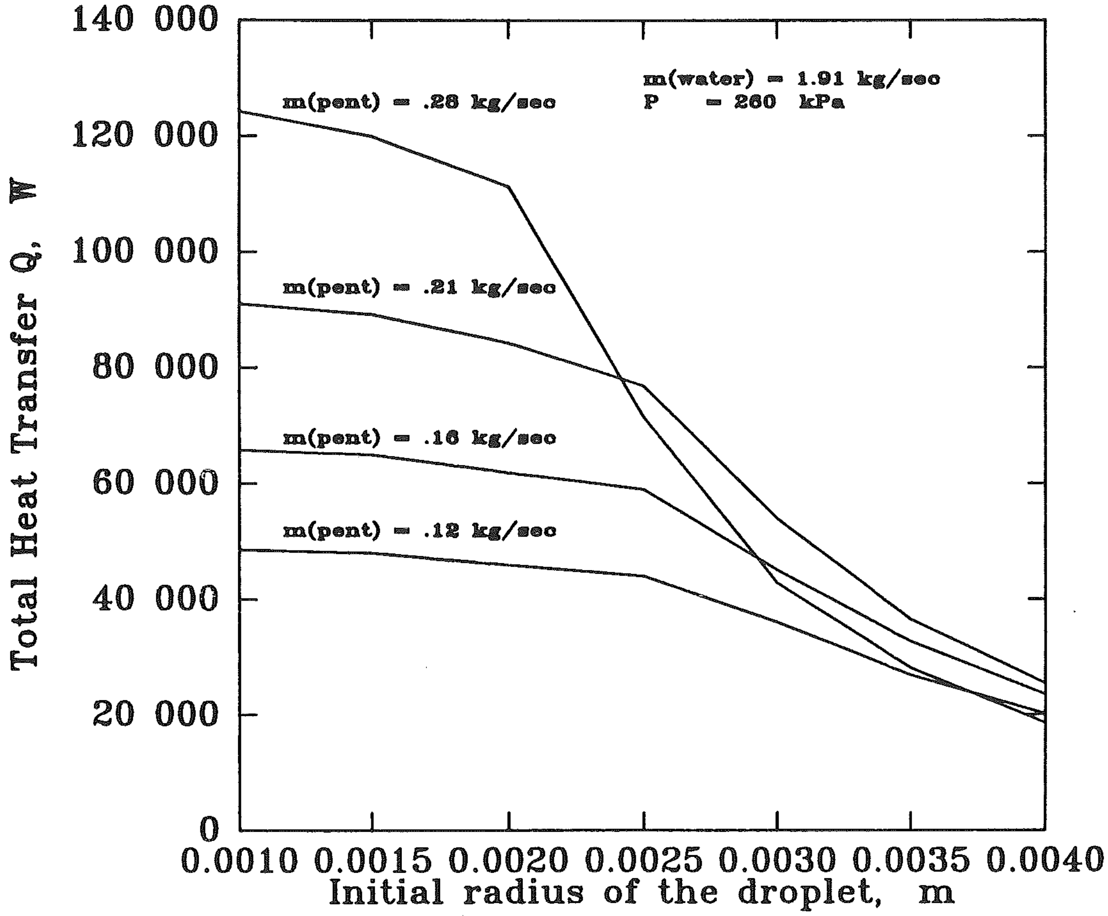

Figure 21. The effect of initial radius of the droplet on total heat transfer for several pentane mass flow rates.

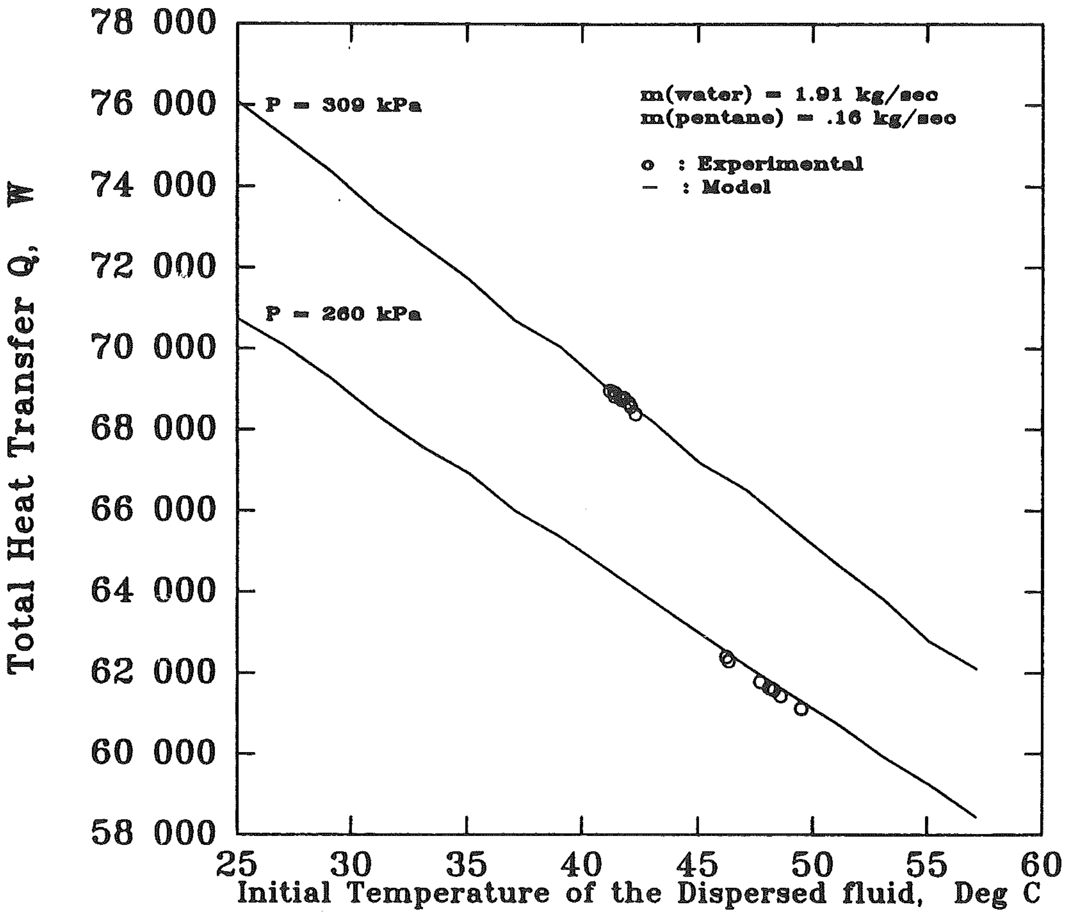

Figure 22. Relations of initial dispersed phase temperature on total heat transfer for 309 kPa and 260 kPa pressures.

increases when the pressure increases. Available experimental data are also shown on this plot. The data predicted by the model are nearly the same as those from the actual experimental situation.

Figure 23 shows the relation of total heat transfer and mass flow rate of the water. The data indicate an increasing dependence of total heat transfer on the water mass flow rate. Figures 24 and 25 give mass flow rates of the dispersed phase, initial pentane temperature and total heat transfer relations. The same kind of linearly decreasing relation for the total heat transfer relations with an increase of the initial temperature and a nonlinearly increasing relation of mass flow rate is observed. For these cases the agreement between the experimental data and the model prediction is very good.

The effect of the initial temperature of the dispersed phase on the total heat transfer coefficient is shown in Figure 26. The volumetric heat transfer coefficient is defined here as the total heat transfer per unit volume divided by logarithmic mean temperature difference. An exponential increase in the heat transfer coefficient is observed by increasing inlet dispersed fluid temperature. Considering decreasing total heat transfer for the same range of data concludes that this heat transfer coefficient is not a very good selection to represent heat transfer phenomenon of the direct contact column. Changes in slopes at discussed

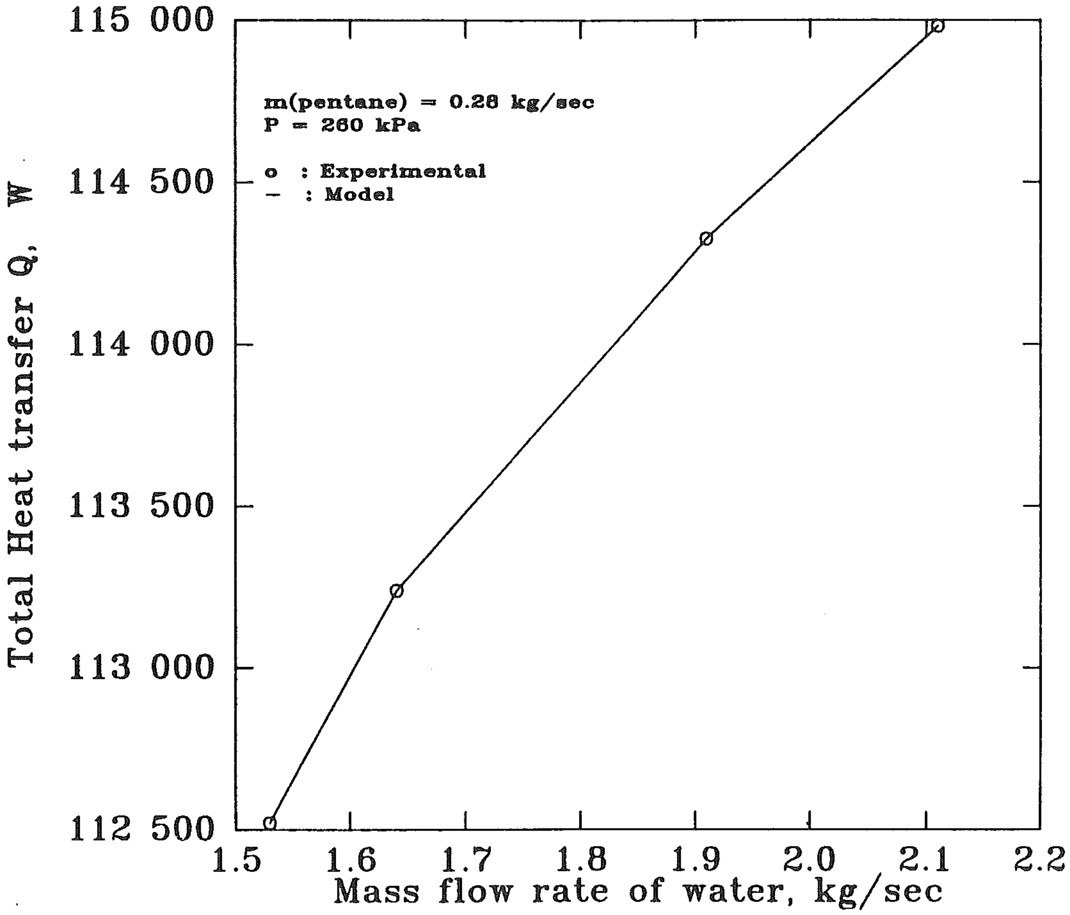

Figure 23. Relation of total heat transfer and mass flow rate of continuous phase (water) for m(pentane) = 0.28 kg/sec and P = 260 kPa.

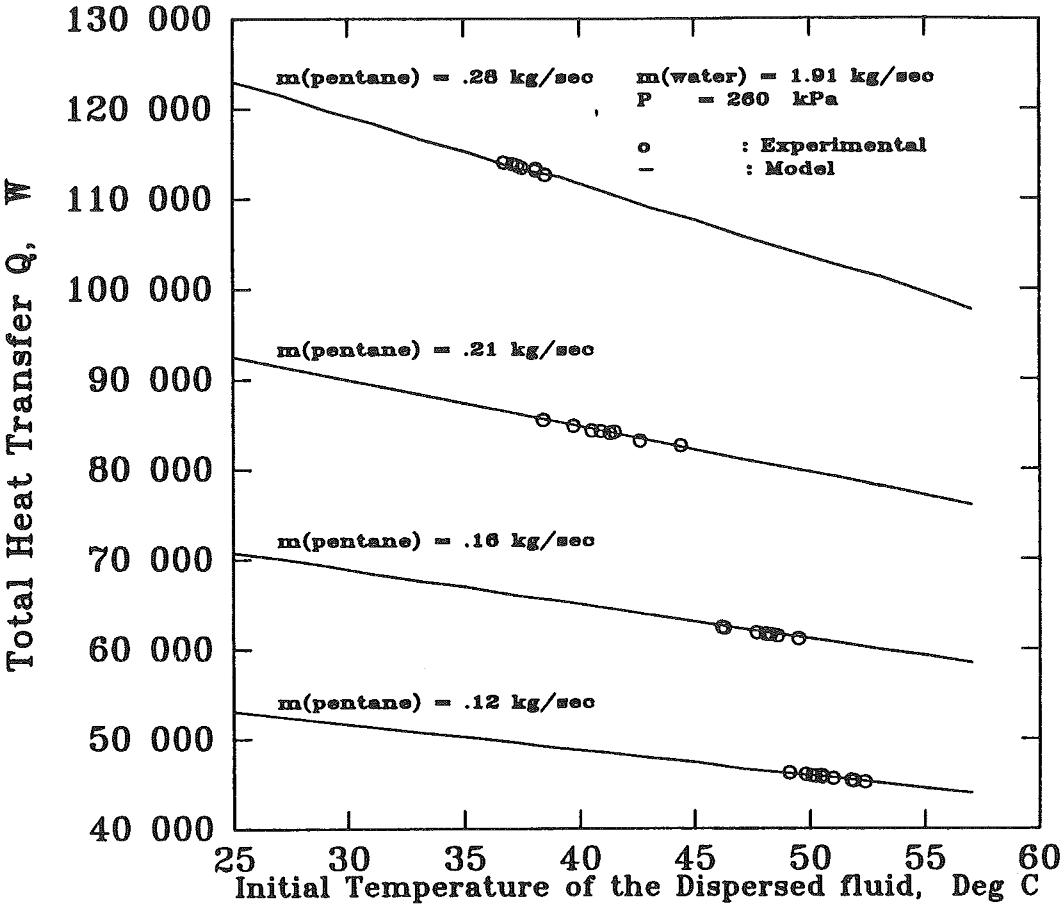

Figure 24. The effect of dispersed fluid (pentane) mass flow rates and initial dispersed fluid temperature on total heat transfer, m(water) = 1.91 kg/sec, P = 260 kg/sec.

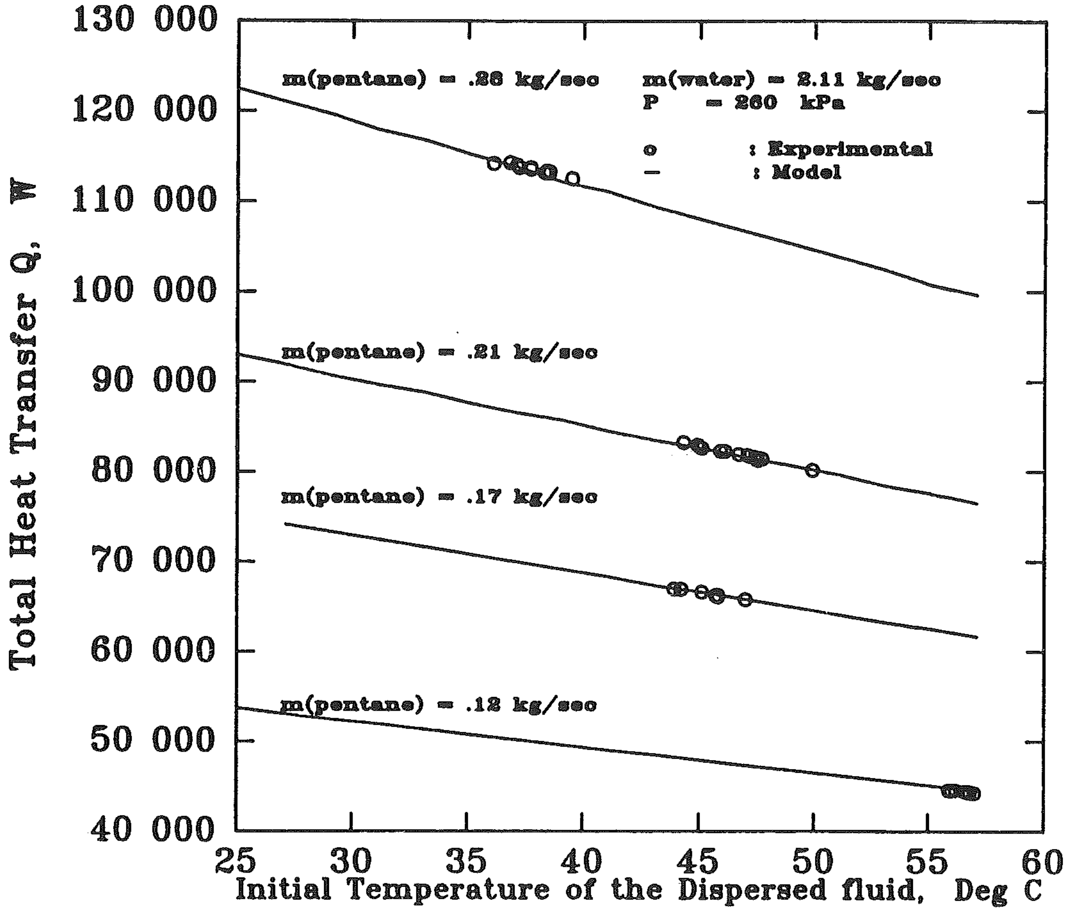

Figure 25. The effect of dispersed fluid (pentane) mass flow rates and initial dispersed fluid temperature an total heat transfer, m(water) = 2.11 kg/sec, P = 260 kPa.

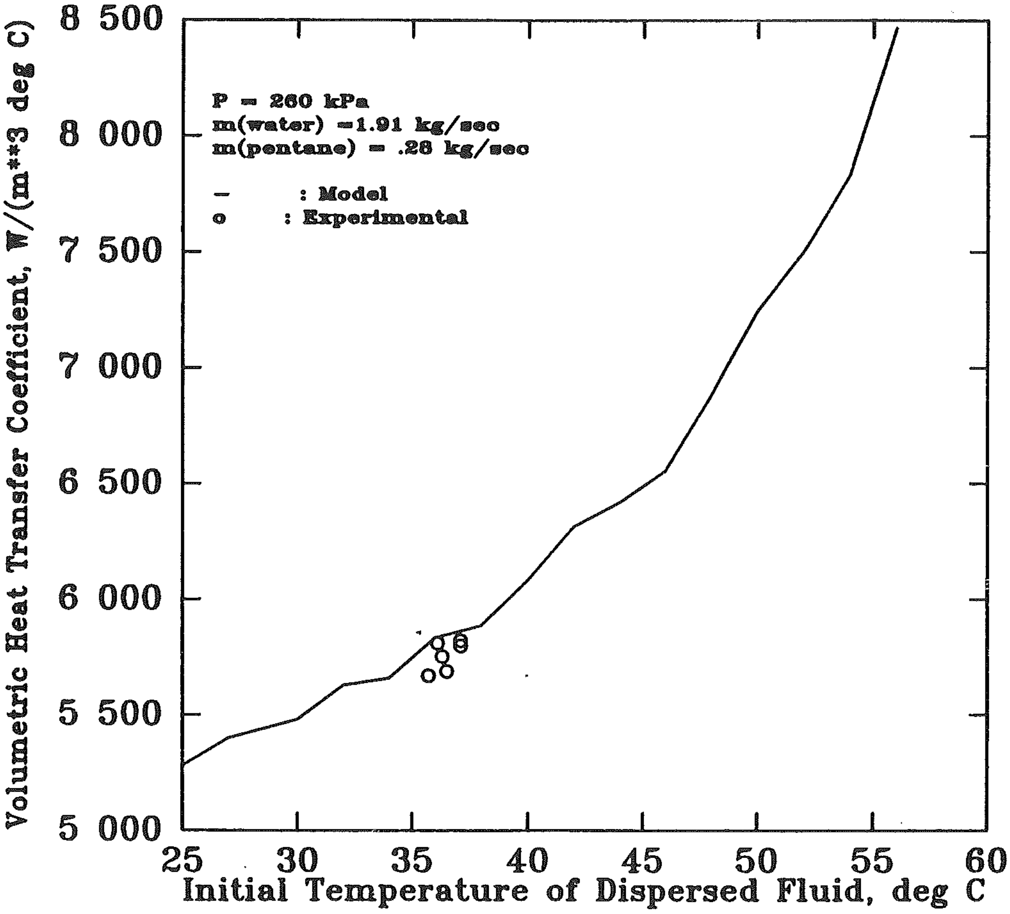

Figure 26. The effect of the initial temperature of dispersed fluid on volumetric heat transfer coefficient, m(water) = 1.91 kg/sec, P = 260 kPa.

location represent step-size effect.

In Figure 27 the effect of droplet radius on the droplet velocity is displayed. This figure also shows a typical velocity rate and radius change of the dispersed phase droplets in the column. Approximately a second degree polynomial relation exists between these two variables.

The heat transfer coefficient inside of the droplet for the liquid and superheated regions is shown in Figure 28. The solid line in this figure indicates the heat transfer to the liquid phase of the bubble and the dashed line represents the heat transfer to the vapor phase. When the drop radius gets larger the heat transfer to the liquid phase increases; meanwhile the heat transfer to the vapor phase decreases. The heat transfer coefficient on the phase decreases. The heat transfer coefficient on the outside of the droplet is shown in Figure 29. It decreases due to an increase in the droplet radius.

Holdup ratios tabulated by various workers are also compared to those predicted by the model developed here. Figures 30 through 32 are three such comparisons. Figure 30 is a comparison of Jacobs and Golafshani's model [8] results to those predicted in this work. Note that the experimental East Mesa data used in this holdup ratio profile have the largest mass flow rates ever reported in any experimental study. Holdup ratio relations follow very complex dependency relations with the other physical variables, and they are

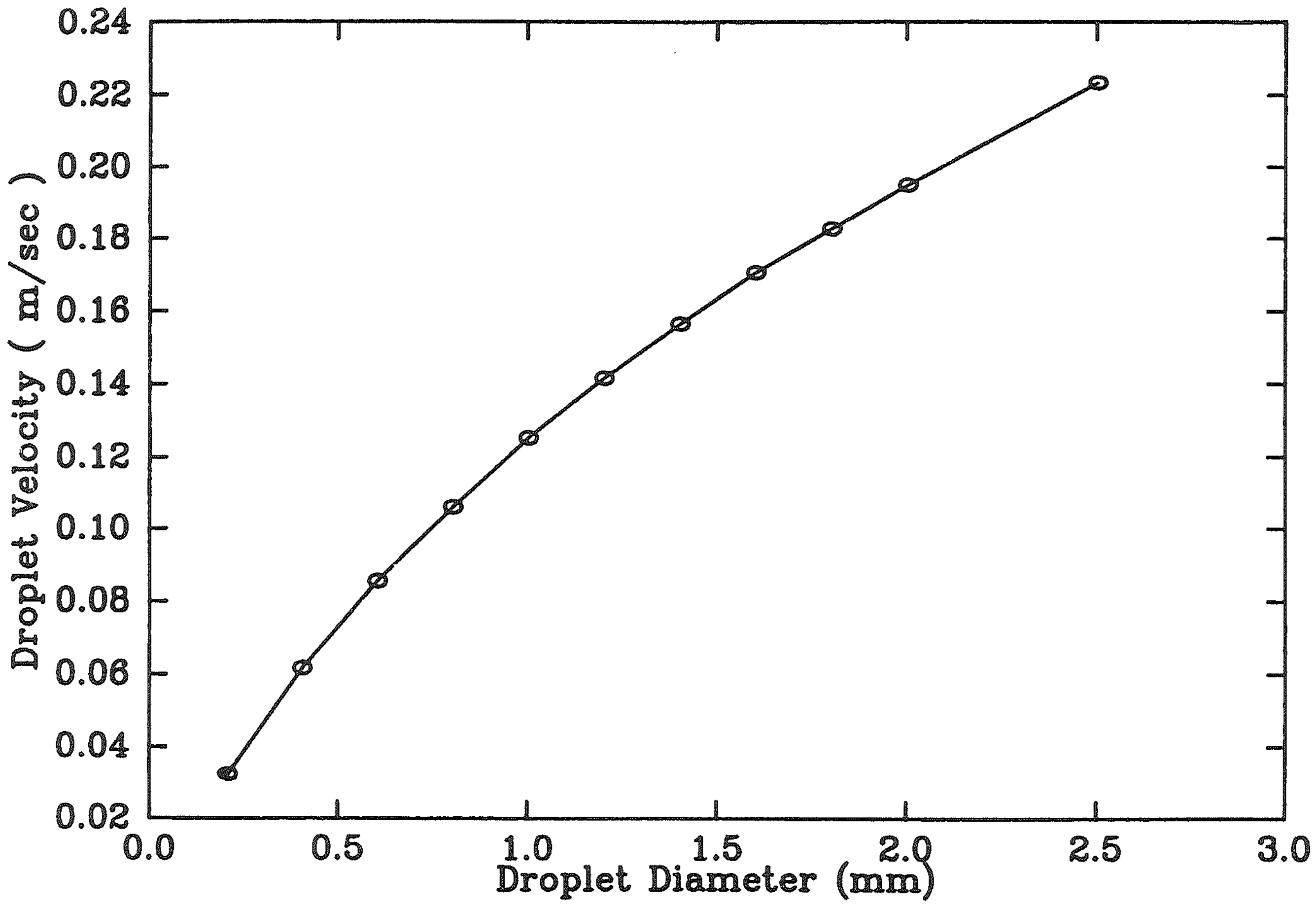

Figure 27. The effect of droplet radius on the droplet velocity.

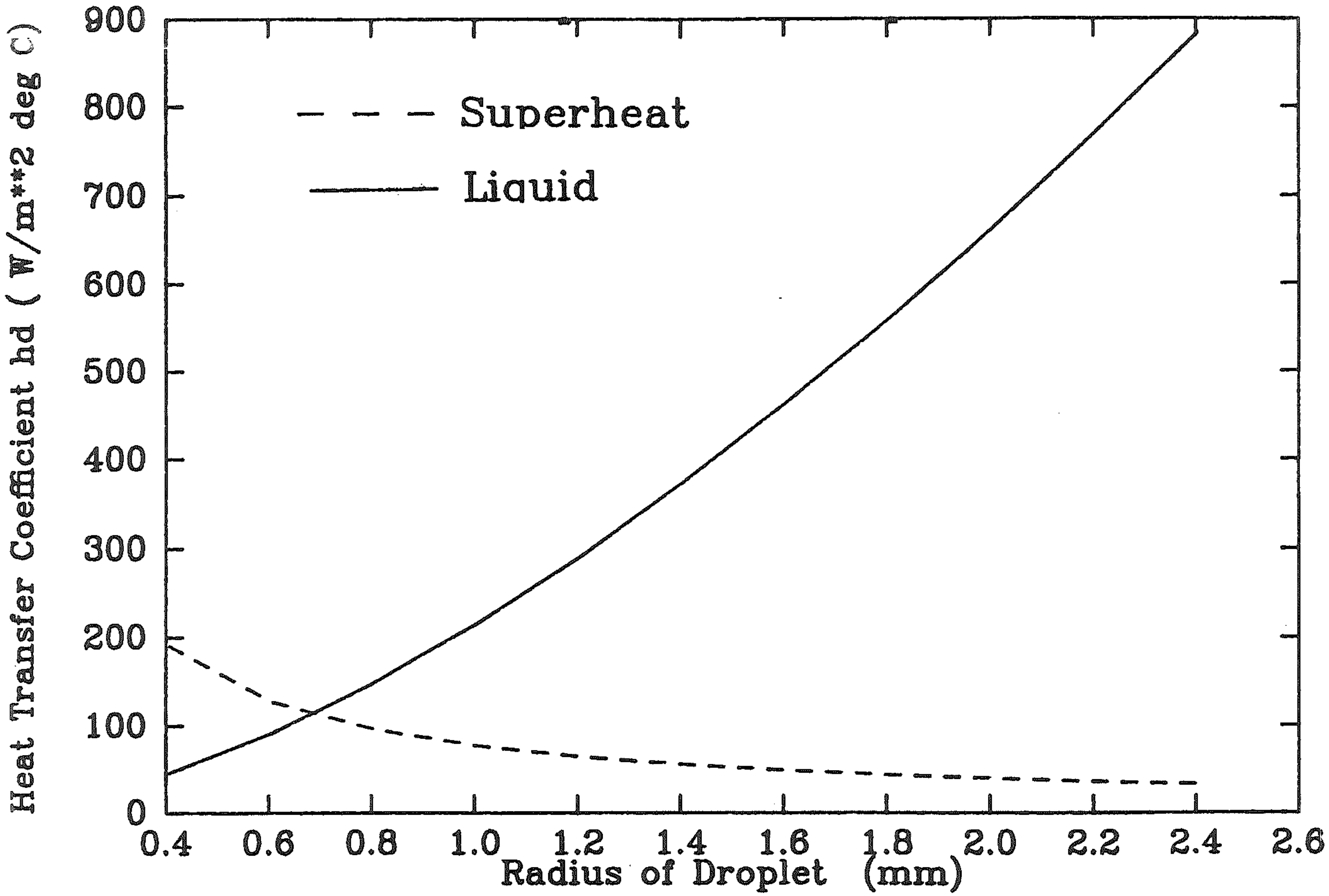

Figure 28. The total heat transfer coefficient inside the droplet in case of liquid and vapor droplets.

figures show a good agreement between the experimental data and the calculated mixed temperatures. It should be noted that at each end of the column there are a pair of experimentally measured temperatures shown. These points are the only information available for the experimental profile of the separate dispersed and continuous fluid temperatures. The exit and inlet temperatures are also very closely approximated by the model. One of the interesting results about the dispersed phase temperature is the decreasing characteristics of the profile during evaporation. This effect is due to hydrostatic and dynamic pressure difference throughout the column. Another interesting point is that the experimental temperature profile remains relatively constant throughout. Contrary to intuition, a fairly flat temperature profile in a spray column does not necessarily imply large amounts of back-mixing, premature boiling, or a number of other oft-cited possible shortcomings of spray columns.

The mixed temperature relationship developed here yields results that are nearly the same as the experimentally determined values and could be quite close to the actual situation. In the starting region of the boiling, the mixed temperature profile gives a sharp change. The reason for this effect is investigated and it is found that the half opening angle, β, and the volume ratio of vapor phase show a rapid increase in this region. In the experimental case the other

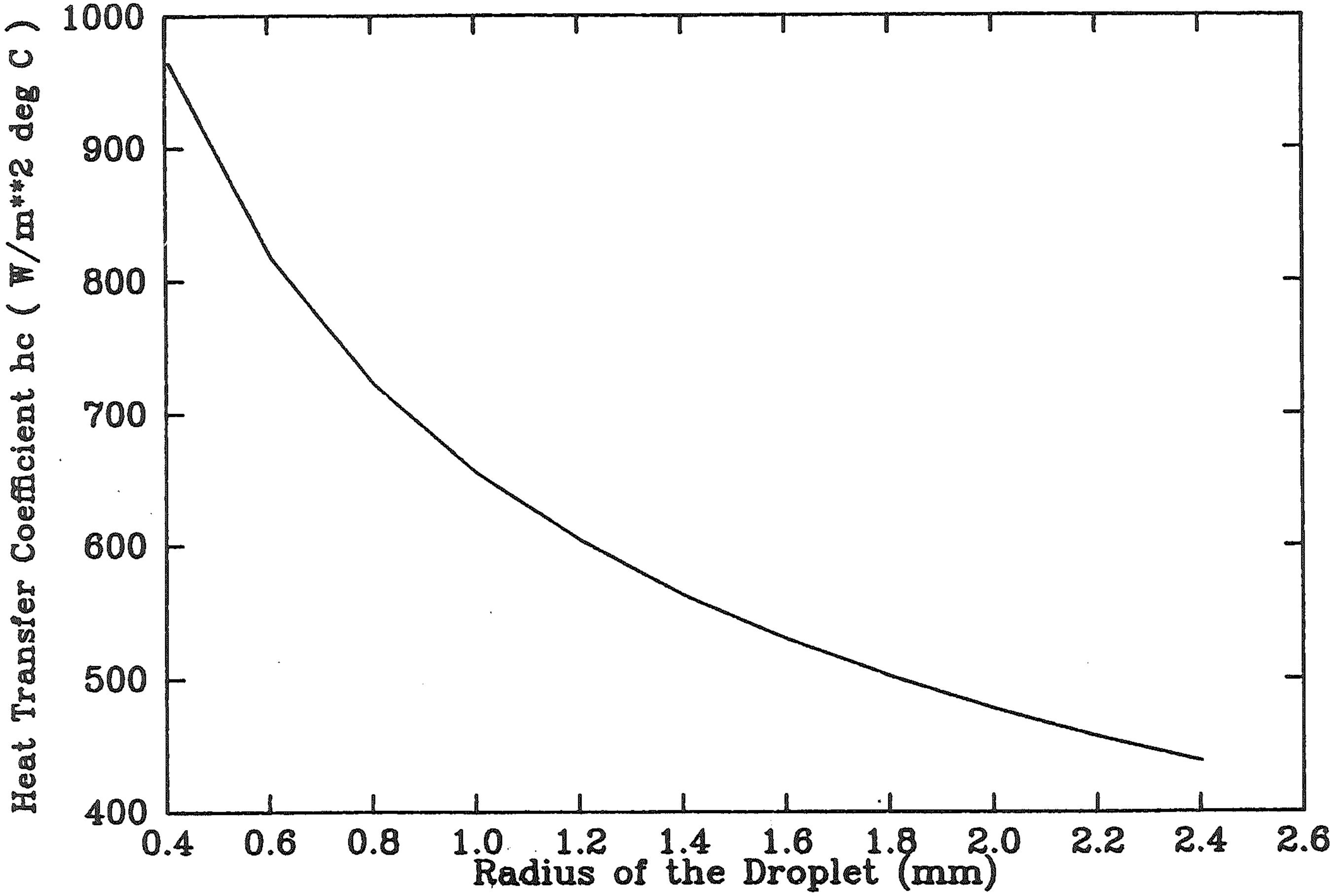

Figure 29. The heat transfer coefficient outside the droplets.

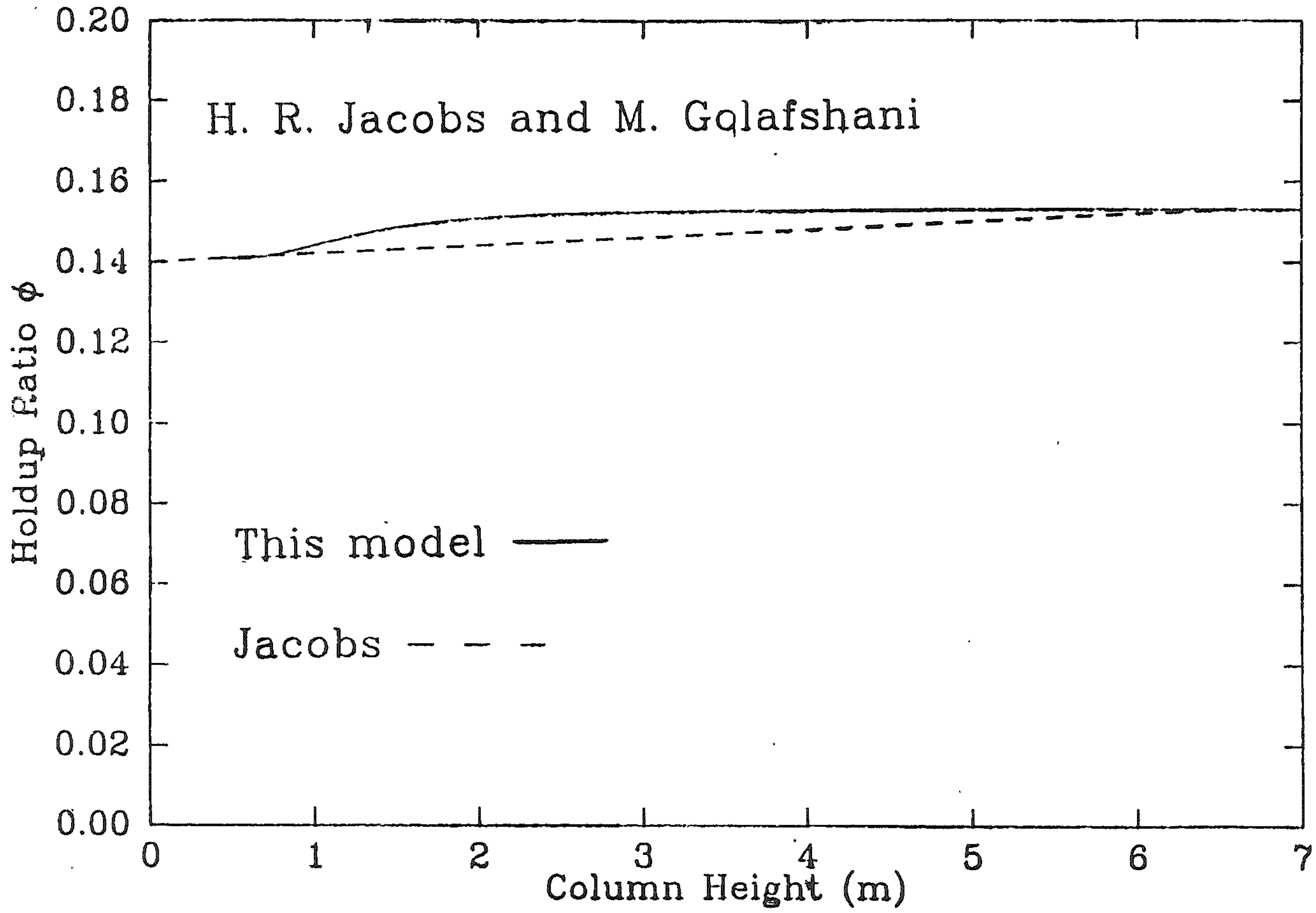

Figure 30. Holdup ratio comparision of H. Jacobs and M. Golafshani model [8] to those predicted in this work along the length of column.

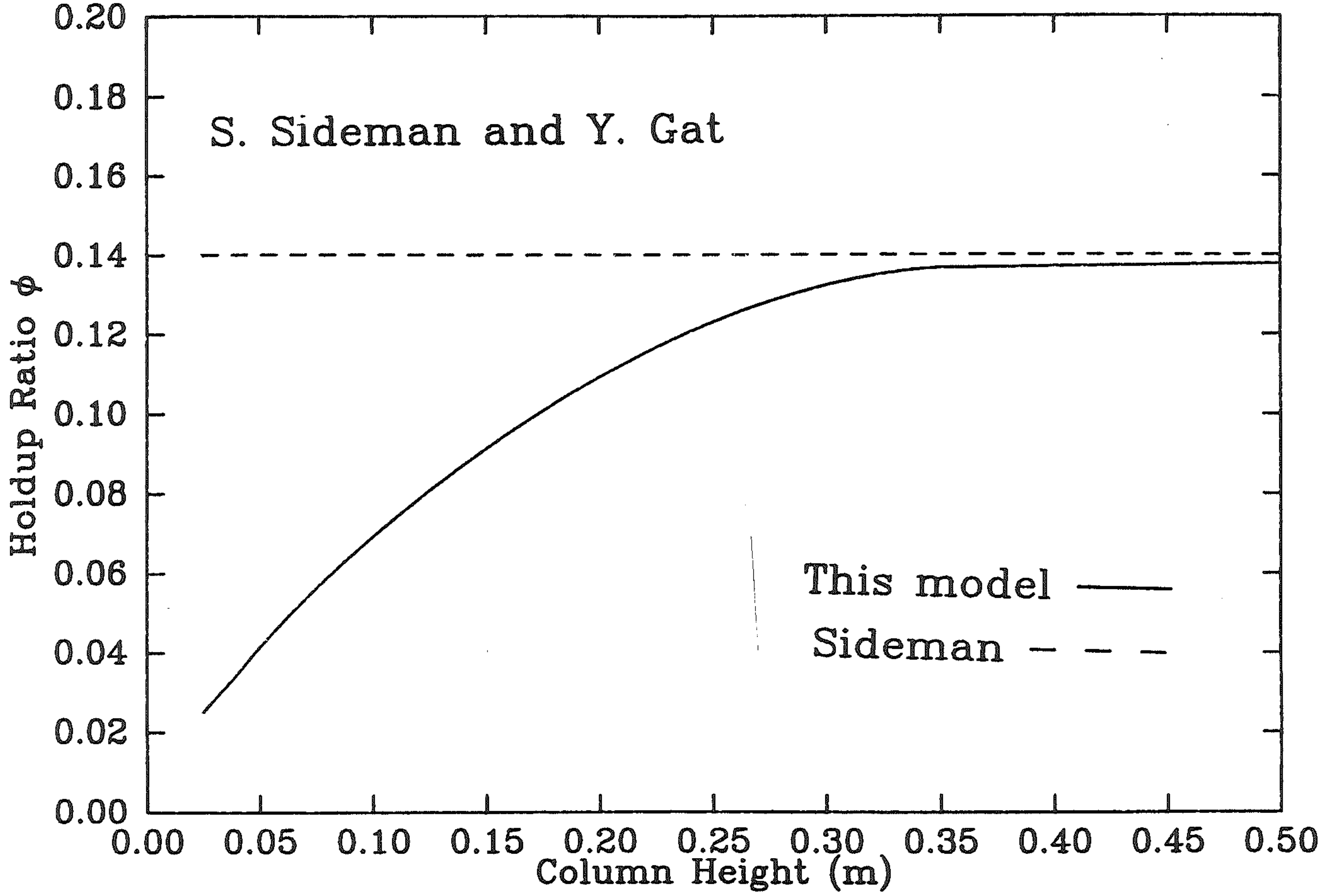

Figure 31. Comparison of experimental holdup ratio data of Sideman [28] to those predicted in this work, along the length of column.

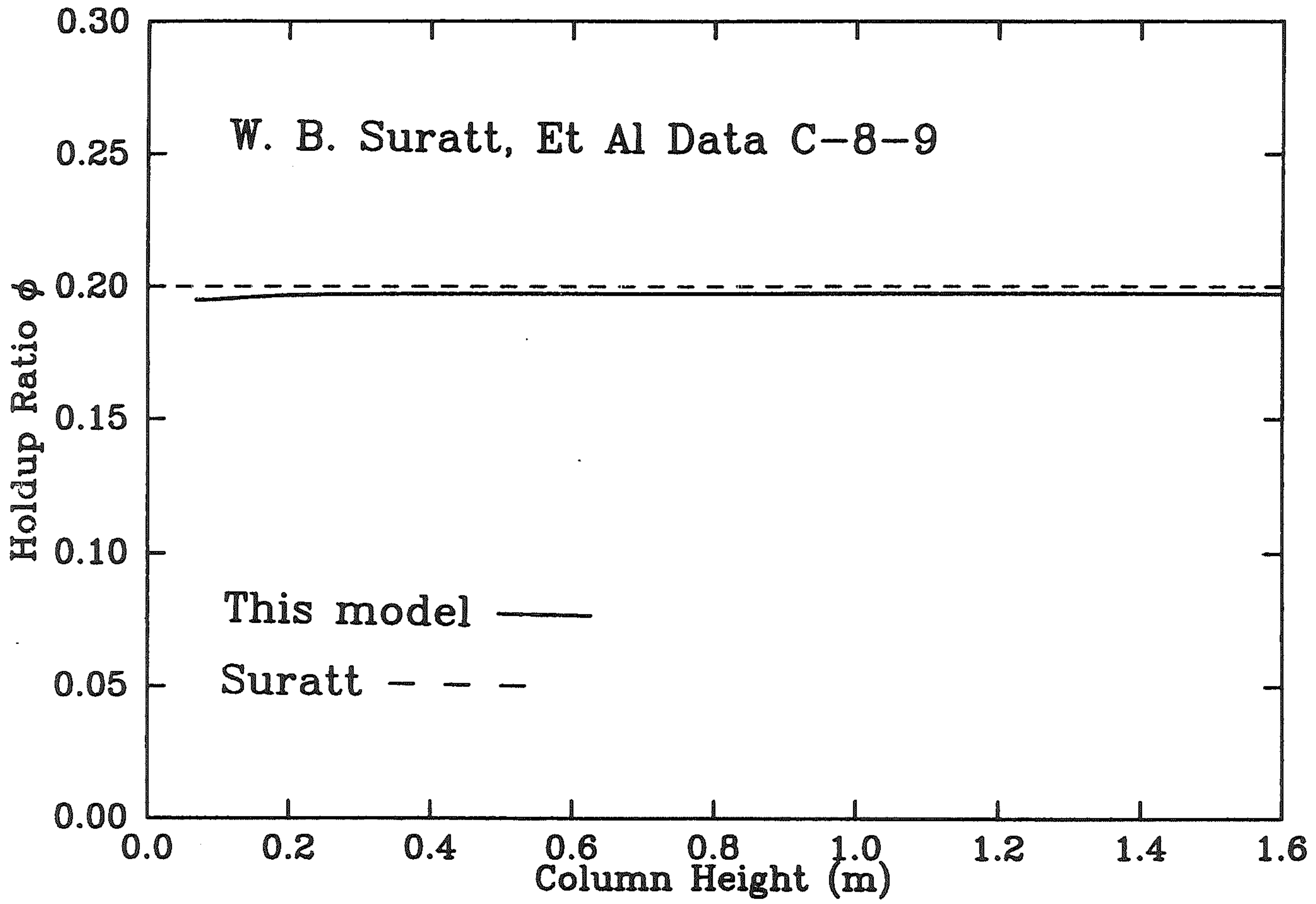

Figure 32. Comparison of experimental holdup ratio data of Suratt, et al. [33] to those predicted in this work, along the length of column.

affected by almost every variable. Figure 31 compares the experimental readings of Sideman [28] with the calculation of this model. Sideman calculated the total holdup of his column by measurements of column height without having any dispersed fluid present and having both phases present. It should be noted that these experimental values include only the boiling region since the inlet conditions of the dispersed phase were at a saturated liquid state. In our model, the beginning portion of the column is a preheating region, and boiling started at a point 0.28 meters up the column. The last figure (Figure 32) is a comparison with the experimental data of Suratt et al. [33] These data also fit very closely to our predicted results and indicate the validity of the model developed here as a design tool.

Various heat transfer equations are compared in Figure 33. It is observed that the Sideman and Taitel equation yields very large values for any possible experimental profile. Although the Sideman and Isenberg equation is much better than the Sideman and Taitel equation, the results are still too large. The Tochitani et al. equation represents the data taken by Goodwinn, Coban and Boehm [6] very well. The last one is the modified version of Tochitani equation in which the heat transfer to the vapor phase is also included. The small amount of heat transfer is necessary to complete the boiling process and a realistic superheating region heat transfer.

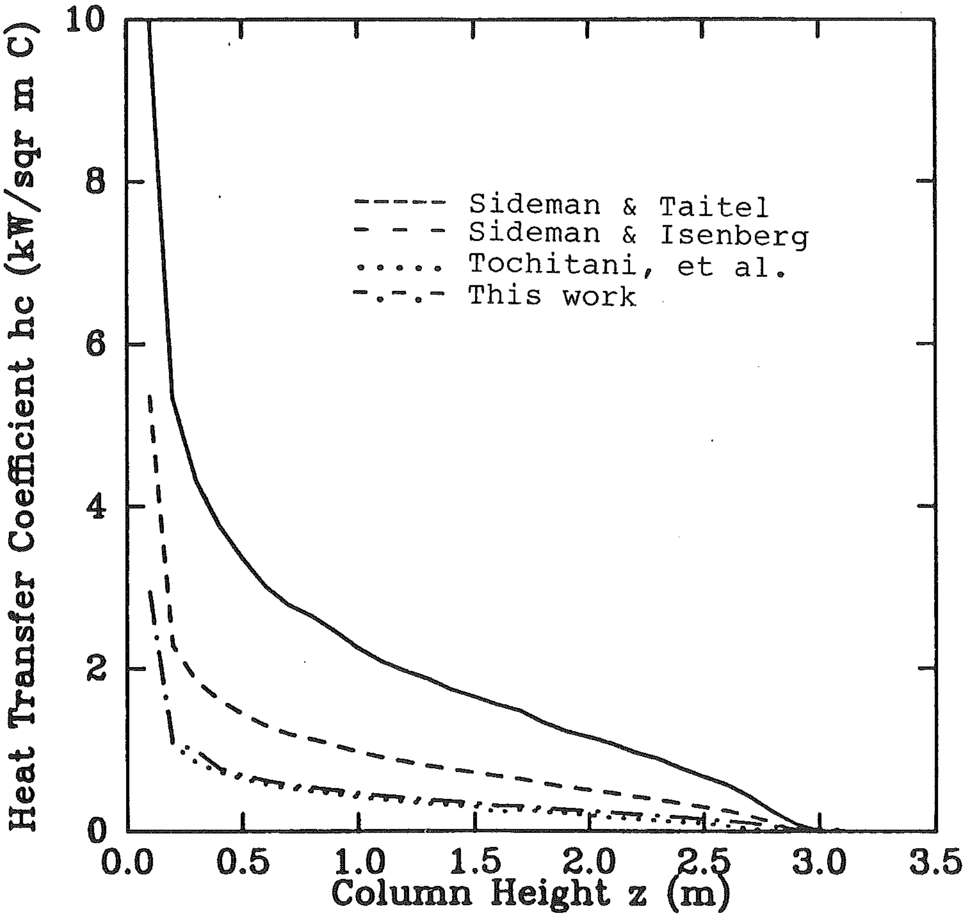

Figure 33. Comparison of various heat transfer predictions.

In order to determine the validity of spherical droplet conception, the Eotvos number (Eo) is checked. It is found that the Eo number is in the range of 0.05 to 1.2. According to Grace [5], this range is basically a spherical regime. At the end of the boiling process an entry into the ellipsoidal regime is possible, but the cases analyzed here never deviate from the spherical case very much.

Relative error of the calculations is checked for 50 and 200 iteration steps. The following error percentages are obtained;

Total heat transfer : 0.0315 %

Holdup ratio : 0. %

Pressure : 0.0257 %

Temperature (continuous) : 0.2246 %

Temperature (dispersed) : 0.0908 %

Velocity (continuous) : 0. %

CHAPTER IV

CONCLUSIONS AND RECOMMENDATIONS

A one dimensional steady state model of a three-phase spray column counterflow direct-contact heat exchanger has been developed by considering the total flow inside the column along with the microphenomena influencing the motion and heat transfer to the dispersed phase droplets. A numerical model was succcessfully applied to solve for the local variation of these properties. A mixed temperature quantity has been developed to compare to the temperatures read with thermocouples mounted in a column. The model is checked for the accuracy of predicting total heat transfer for several experimental tests, and it is found to yield almost the same values as those obtained in the experiment. Effects of several variables on the heat transfer are investigated. It is found that the mass flow rate of the dispersed phase is the prime variable to set the total heat transfer. Other important variables are found to be the initial drop diameter, column pressure, and the initial drop temperature. It is shown that insufficient column height can be an important factor causing incomplete boiling. But after the completion of boiling, additional column height has only a small effect on the total heat transfer.

Various models of heat transfer are incorporated into the solution routine and the validity of each model is examined. Furthermore, the validity of the spherical droplet assumption is checked and found to be a good representation of the physical situation.

Based on this study which investigates total flow conditions of the three-phase direct-contact heat exchanger, the following conclusions are made:

1. The assumption of spherical droplets used here is justified for the small droplets that are predicted to exist in most cases. In a few cases the droplet may grow slightly into the ellipsoidal region for the final part of the evaporation process.

2. Three phase modeling provides an accurate description of temperature fields and therefore this three phase modeling can be used as a design tool to predict physical dimensions of the next generation boilers. Existing experimental data and those predicted by the model are found to be in good agreement.

3. Variations of (i) flow rates of either stream, (ii) pressure, (iii) incoming temperatures, (iv) height of the column and (v) droplet radius can adversely effect the operational conditions of the column. An increase of flow rates of the streams and column height has a positive effect on the total heat transfer. An increase of incoming dispersed

phase temperatures and bubble droplet radius has a decreasing effect on the total heat transfer.

4. Even though the effect of heat transfer to the vapor phase is very small compared to the heat transfer to the liquid portion of the evaporating drop, the heat transfer to the vapor is very important for the completion of the evaporation and heat transfer in the superheated region of the column.

5. Although approximate models such as those used here are possible to develop, it is almost impossible to make a model to define the complete physical description of the direct contact behavior due to the enormous complexity of the heat transfer and fluid mechanics in a spray column.

As far as future research is concerned, there is much to be done in the area of three phase heat transfer. However as an extension of the problem studied herein, the following recommendations are made:

1. The effect on the heat transfer of baffles, packings and other kind of heat transfer enchancement devices should be examined.

2. For a more complete description of the heat transfer and fluid motion, coalescence and break-up of the drops should be accounted for in the model. Since there are not many experimental and theoretical studies on these phenomena, more research should be done.

3. The physical behavior of different shaped, or

generally shaped, drops should be examined and the heat transfer mechanisms to such drops should be investigated.

4. The flow characteristics inside of the droplets should be examined and more detailed work should be done for the heat transfer coefficient inside of the bubble when a phase change is taking place.

5. An effect of stripping of the continuous fluid due to evaporation into the dispersed fluid bubble could be incorporated into the model.

APPENDIX A

HEAT TRANSFER AND RELATED EQUATIONS

Sideman and Taitel [28] Equations

This equation assumes that the droplet is a perfect sphere and heat transfer occurs only to the liquid phase of the bubble. Furthermore it assumed that outside heat transfer dominates the total heat transfer. The bubble and half opening angle β are shown in the Figure A-1. The Nusselt number for the outside of the droplet is given as equation A-1-1 and the definition of half opening angle as a function of dispersed fluid quality is given in equation A-1-2.

A-1-1 $$Nu_c = \frac{[\,3\cos\beta - \cos^3\beta + 2\,]^{0.5}}{\pi^{0.5}} (Pr_c\, Re_c)^{0.5}$$

A-1-2 $$3\cos\beta - \cos^3\beta + 2 = \frac{4\,(1 - x)}{1 + x\,(M - 1)}$$

Sideman and Isenberg [24] Equation

This equation is the same as the Sideman and Taitel equation with the exception of a correction factor *k*. The relationship for the correction factor and new form of Nusselt number are given by equations A-2-1 and A-2-2.

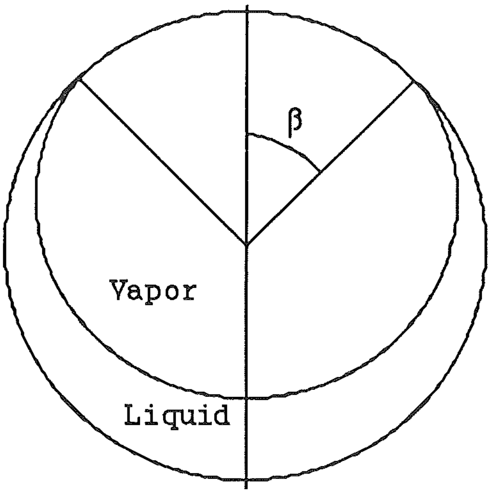

Figure A-1 Definition of half opening angle of vapor phase according to Sideman and Taitel [28]

A-2-1 $$k = 0.25\ Pr_c^{-1/3}$$

A-2-2 $$Nu_c = \frac{[\ 3\cos\beta - \cos^3\beta + 2\ ]^{0.5}}{\pi^{0.5}}\ (0.25\ Pr_c Re_c)^{0.5}$$

Tochiani et al. [35] Equations

This is a theoretically developed relation. It assumes that the surface of the two phase bubble is rigid, and that the flow around the bubble obeys Stokes Law. The terms in the first parenthesis in the equation A-3-1 represents heat transfer coefficient for rigid spheres and the terms in the second parenthesis represents the effect of surface area change of the bubble.

A-3-1 $$Nu_c = (Pr_c\ Re_c)^{1/3}\{0.466\ (\ \pi - \beta + 0.5\ \sin 2\beta)^{2/3}\}$$

Raina and Grover Equation [16]

This theoretical model is developed by considering the effect of viscous shear on the spreading of the dispersed liquid over the bubble surface. The configurations of the angle α and β are shown in Figure A-2. The heat transfer coefficient is defined as follows

A-4-1 $$h_c = 0.314\ \frac{k_c}{R}\left[\frac{R\ U}{\alpha_c}\right]^{1/3}\left[\beta - \alpha - \frac{\sin 2\beta - \sin 2\alpha}{2}\right]^{2/3}$$

A-4-1 $$h_c = 0.314\ \frac{k_c}{R}\left[\frac{R\ U}{\alpha_c}\right]^{1/3}\left[\beta - \alpha - \frac{\sin 2\beta - \sin 2\alpha}{2}\right]^{2/3}$$

The definition of β is the same as Sideman and Taitel. α_c represents the thermal diffusivity. The angle α is defined as

A-4-2 $$1.5\ \frac{\mu_c\ U}{S_d} = \frac{1}{\cos\alpha - \cos\beta}$$

In this equation S_d is the total surface tension effect on the bubble and is given by

A-4-3 $$S_d = \sigma_c + \sigma_d - \sigma_{cd}$$

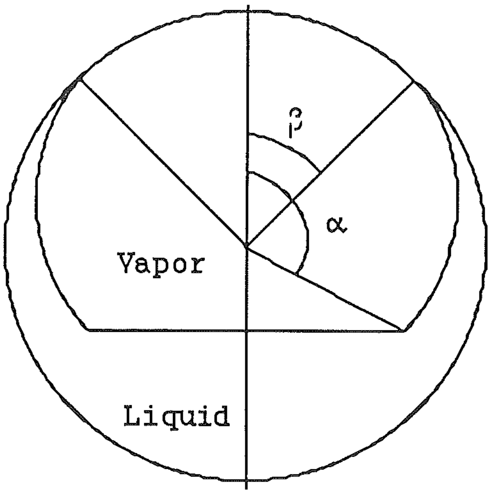

Figure A-2 Definition of α and β according to Raina and Grover [9]

where σ_c, and σ_d are the surface tensions of the continuous and the dispersed phases, respectively, interacting with their own vapors. σ_{cd} represents the surface tension during interaction of the two fluids.

A. E. S. Adams and K. L. Pinder Equation [1]

This equation is a result of an experimental study in which a dilatometric method was combined with a cine-photographic technique to obtain the average heat transfer coefficents during the evaporation of droplets. These data are correlated by curve fitting to obtain equation A-5-1.

A-5-1 $$Nu_d = 7550.\ Pr_c^{-0.75} \left[\frac{\mu_c}{\mu_c + \mu_d}\right]^{4.3} Bo^{0.33}$$

A-5-2 $$Bo = (\rho_c - \rho_d)\ D^2_i\ g\ /\ \sigma$$

Where Bo is the liquid Bond number.

C. B. Parakash and K. L. Pinder Equation [15]

These equations are also experimentally derived. The authors studied furan, isopentane and cyclopentane to determine the heat transfer coefficient via photographic methods. Two correlation equations were given by this experimental study.

A-6-1 $$Nu = 0.0505\ (Pe')^{0.417} \left[\frac{\rho_c}{\rho_d}\right]^{1.25}$$

A-6-2 $$Nu = 0.5\ (Pe')^{0.445} \left[\frac{\rho_c - \rho_d}{\rho_c}\right]^{1.81}$$

A-6-3 $$Pe' = \frac{Re\ Pr}{1 + \frac{\mu_d}{\mu_c}}$$

The above mentioned authors claimed that the best fit to experimental data was obtained by equation A-6-1. Pe' in this equations is the modified Peclet number and this is defined by equation A-6-3.

Simpson, Beggs and Nazir equation [30]

This is an experimental equation for the heat transfer coefficients of butane droplets. The authors reported that their values for the heat transfer coefficients were in better agreement with Sideman butane-water data than was the Sideman theory.

$$\text{A-7-1} \qquad h = \left(2.57 + \left(\frac{D}{D_0}\right)^{0.25}\right) / \left(1 + 0.206 \left(\frac{D}{D_0}\right)^{5/12}\right)$$

In this equation D_0 and D represent the initial bubble diameter and the bubble diameter for a given point, respectively.

Raina, Wanchoo and Grover velocity equations [19]

This paper deals with the motion of a vaporizing two-phase bubble. First, the basic equation for the terminal velocity is derived from the force balance equation [A-8-1]. The result of this equation is correlated to account for the imbalance between the initial drop diameters and the initial diameter and instantaneous diameter [A-8-2]. Finally, the experimental values of the heat transfer coefficient for a given initial diameter of the dispersed liquid drop were found to decrease with increases in temperature driving force and a correction factor were added. The equation [A-8-2] then took the form of [A-8-3].

A-8-1 $$U = \left[\frac{4}{3}\left[1 - \frac{\rho_d}{\rho_c}\left(\frac{D_0}{D}\right)^3\right]\left(\frac{D\,g}{C_D}\right)\right]^{1/2}$$

A-8-2 $$U = \frac{1.1547\left[\left\{1 - \frac{\rho_d}{\rho_c}\left(\frac{D_0}{D}\right)^3\right\}\left(\frac{D\,g}{C_D}\right)\right]^{1/2} D^{\left(\frac{5}{6} - \frac{1}{T_c}\mathbf{D}\right)}}{\left[\frac{T_c^2 + T_d^2}{2\,T_c T_d}\right]^{\mathbf{D}}}$$

A-8-3 $$U = \frac{1.1547\left[\left\{1 - \frac{\rho_d}{\rho_c}\left(\frac{D_0}{D}\right)^3\right\}\left(\frac{D\,g}{C_D}\right)\right]^{1/2} D^{\left(\frac{5}{6} - \frac{1}{T_c}\mathbf{D}\right)}}{\left[\frac{T_c^2 + T_d^2}{2\,T_c T_d}\right]^{\mathbf{D}}\left[\frac{Cp_c \mu_c}{k_c}\right]^{\frac{D_0}{1.6\mathbf{D}}}}$$

A-8-4 $$\mathbf{D} = \frac{D^2 + D_0^2}{2\,D\,D_0}$$

A-8-5 $$U = U_d - U_c$$

In equations [A-8-2] and [A-8-3] $\mathbf{D}$ represents an average diameter which is defined by equation [A-8-3]. All of the velocities given in these equations are obtained by assuming that there are no continuous flow velocities. Therefore, these velocities are converted to the velocities of our model as the differences of continuous and dispersed fluid velocities. Due to counter flow directions of the two flows, these two velocities are actually added up to each other.

G.K. Raina and R. K. Wanchoo equation [17]

These authors derived an analytical expression for the instantaneous velocity of a two phase bubble evaporating while passing through an immiscible liquid. They claimed that their equation predicts very well the experimental data for n-pentane and furan drops evaporating through high viscosity aqueous glycerol. This equation has the following form

A-9-1 $$U = \frac{D^2 \rho_c g}{36 \mu_c}\left[1 - \frac{\rho_d}{\rho_c}\left(\frac{D_0}{D}\right)^3\right]\left[\frac{1}{1+\left(\frac{1}{1+x(M-1)}\right)^{1/3}}\right]$$

Where M represents the density ratio of the dispersed fluid to the continuous fluid. U has the same definition as in the Raina, Graver and Wanchoo equation.

Sideman and Shabtai Equation [27]

This equation provides an expression for the inside heat transfer coefficient of a one phase bubble. The equation has the following form

A-10-1 $$Nu_d = 0.00375\ (Re_d\ Pr_d)\ /\ [1 + \mu_c/\mu_d]$$

APPENDIX B

EXPERIMENTAL DESIGN AND PROCEDURE

Experimental Design

As part of this work, experiments were performed by this author and Paul Goodwin, a Master of Science student. The experimental apparatus consisted of a direct contact tower and two flow loops, one each for the dispersed and the continuous phases. Water and commercial grade n-pentane were chosen as the continuous and dispersed fluids. A schematic diagram of the apparatus is shown Figure B-1. Due to the flammability aspects of the pentane all of the electrical wiring and pumps were designed to be explosion proof. A small shed was located about 15 meters from the experiment where all the control equipment was installed. The total system could be considered as three basic sections; the direct contact tower, the water side loop and the pentane side loop. A general drawing of the vessel is shown in Figure B-2.

The tower was designed from standard 24 inch (0.61 m) pipe. It was constructed in two pieces. The pieces were connected together by rectangular plates 0.0525 cm thick which were welded together. The same kind of plates were used to cover the top and the bottom of the column. The total height of the column was 6.096 meters. Typical working height

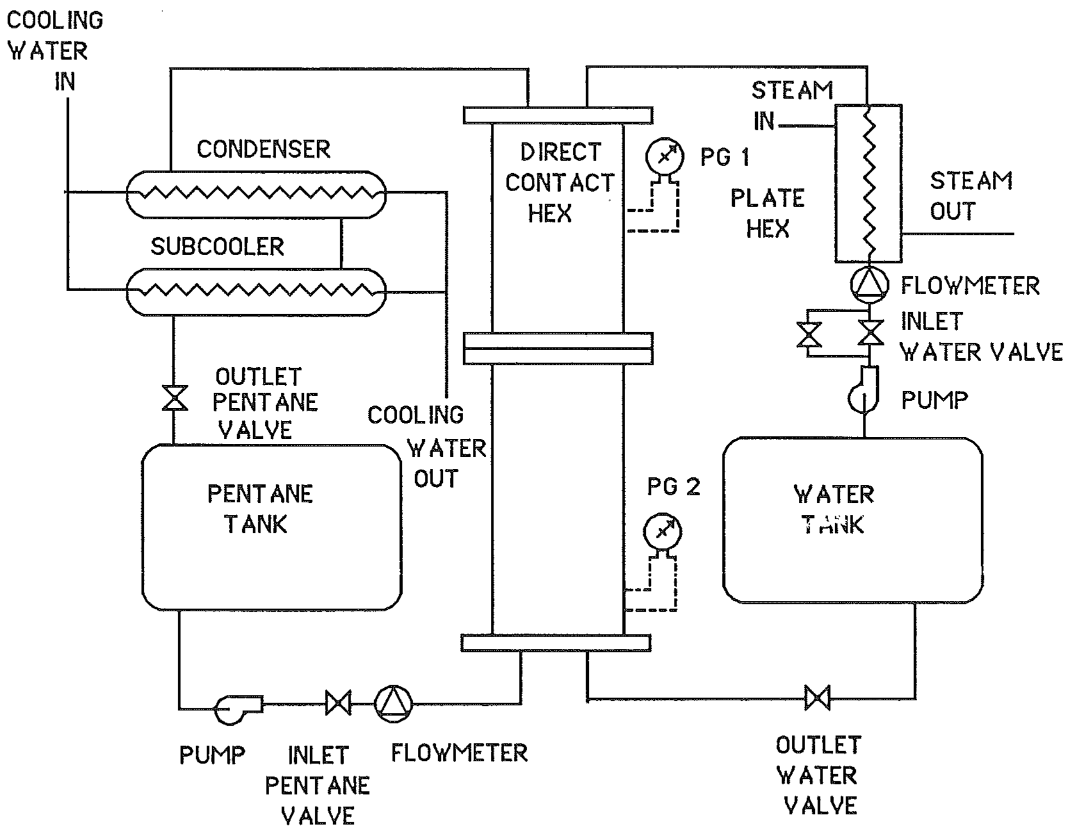

Figure B-1 A schematic diagram of the apparatus

was approximately 3 meters. The benefit of using a longer column than is required is to allow other configurations to be evaluated later that may require greater working heights. The column had three plexiglass viewing windows in the column midsection so that visual observations of the interior and active column level checks could be made easily.

The vessel was equipped with two pressure gauges, one at the bottom and another at the top of the vessel. The temperature measurements were performed with the use of chromel-alumel four were used for input and output

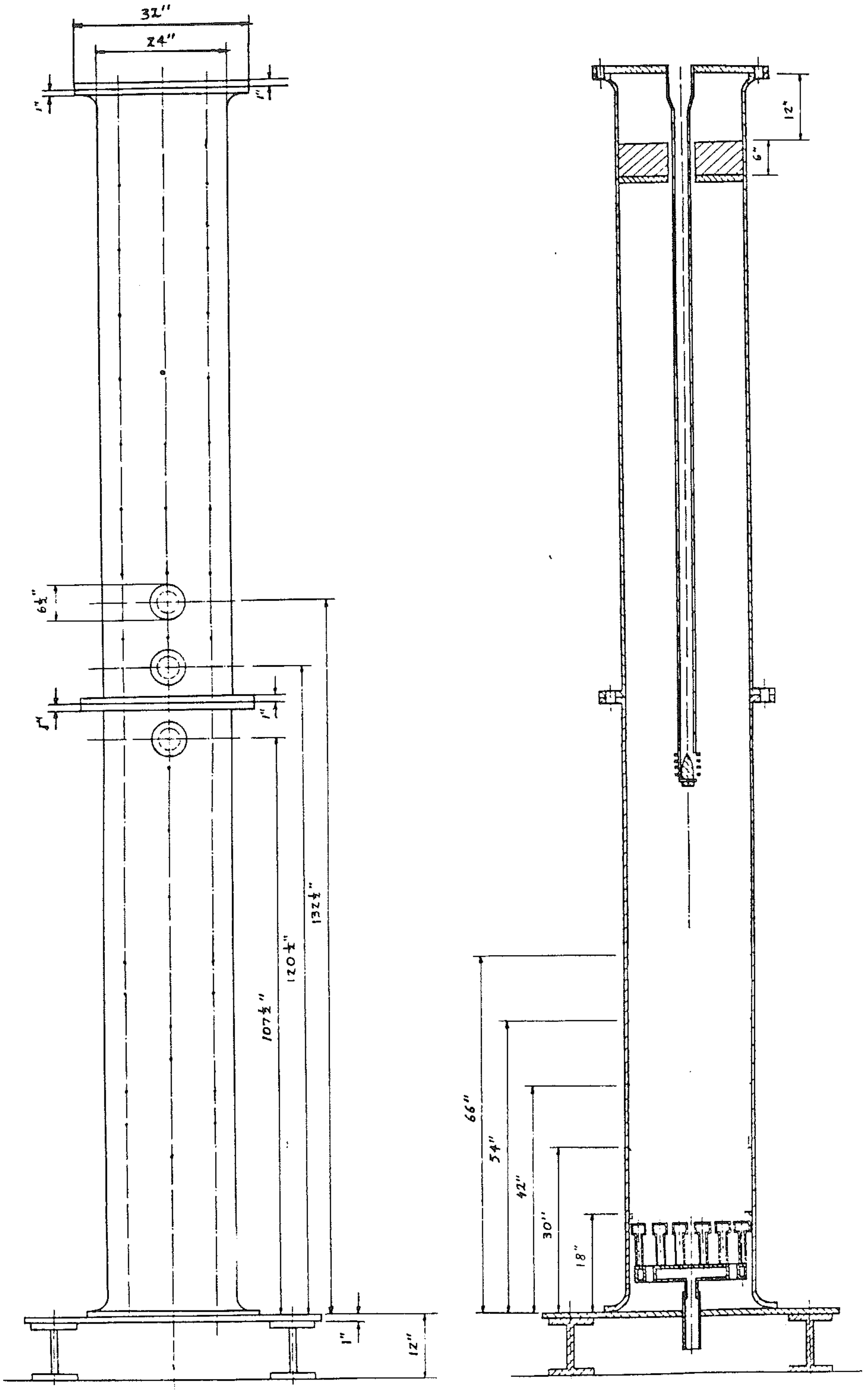

Figure B-2 Elevation and cutaway of the direct contact heat exchangers.

conditions for each phase. By using a couple of thermocouples at the same height but at the different radial dimensions within the vessel an average of mixed temperatures were obtained.The distribution of the thermocouples is shown in Figure B-3.

Water entered the tower through an inlet nozzle at about the middle of the column. The nozzle was designed so that the water would be efficiently distributed. Pentane entered the direct contact heat exchanger from the bottom and was distributed to the inlet nozzles by a cylindrical manifold. This part contains not only connections for the nozzles but also holes which facilitate easier flow of the continuous phase. The spray nozzles were constructed of hard plastic casing with a thin metal cap. A total of 14 nozzles were used with 25 holes in each nozzle. The diameter of the holes was 0.8 mm.

The water loop of the system consists of a storage tank, two parallel centrifugal pumps, a main valve accompanied by a smaller bypass valve and a plate heat exchanger. The storage tank was a 500 gallon galvanized steel container. Tap water was fed to the top of the tank by a 1 1/2 inch pipeline. The pipe was equipped with a standard gate valve and a check valve to prevent backflow. A pressure gauge was also located at the top of the tank. In addition a pressure relief valve was placed on the water container tank. It was set to open at approximately 7.5 bars. The relief

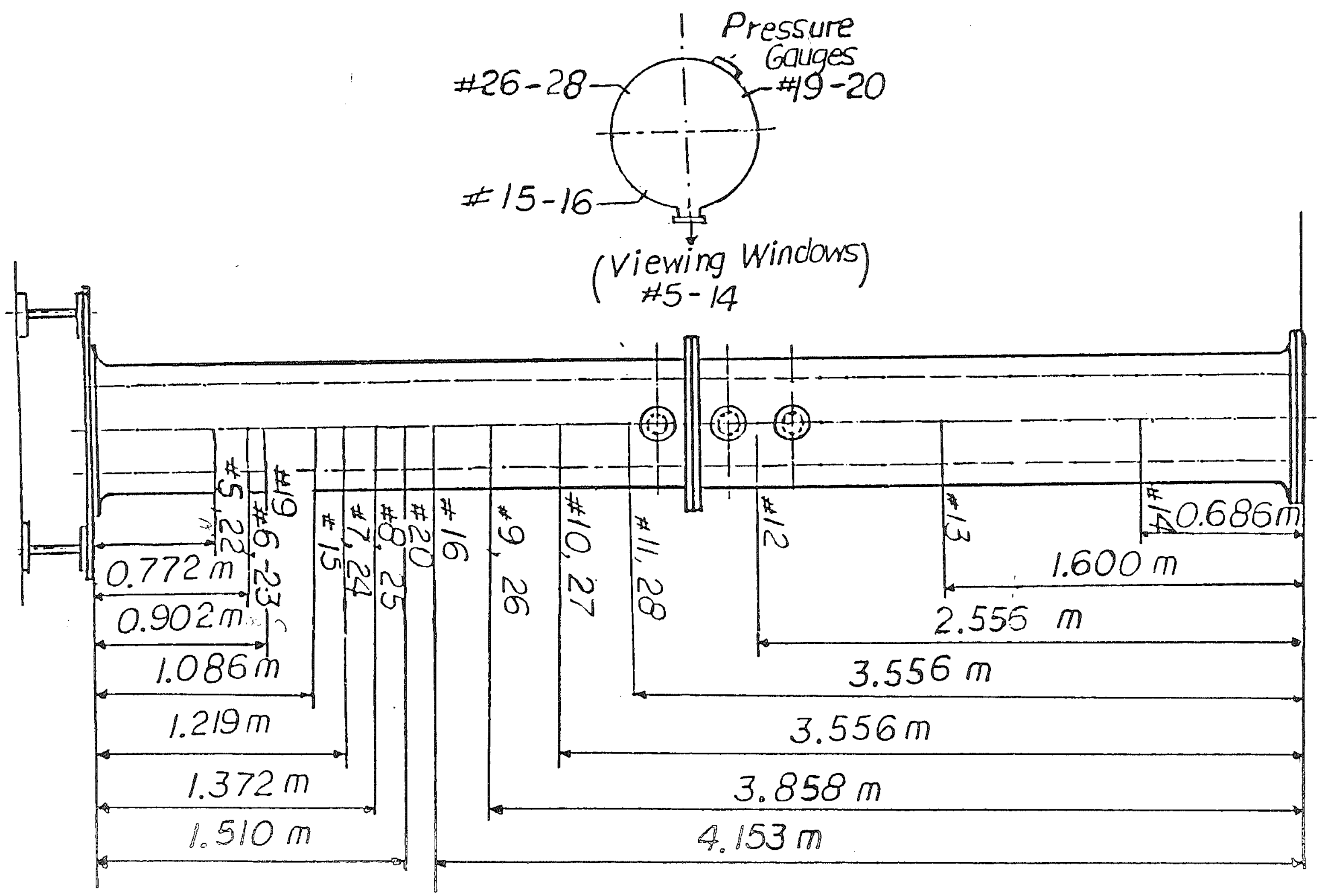

Figure B-3 Distribution of the thermocouples on the Direct contact heat exchanger vessel.

valve never opened during the operations except during a special pressure test run. The last outfittings of the water tank were two drain lines; the first one was fixed at the bottom of the column to drain the water while the second one was at the top of the tank to discharge pentane vapor collection in the tank.

Pumping was accomplished by a pair of centrifugal explosion proof pumps. These pumps were originally designed for N_2O_4 with an output of 60 gpm and 73 feet of head. The pumps were connected in parallel with each other and merged to a single 3 inch line. The next piece of equipment on this line was a main water valve. A small bypass line parallel to this was also equipped with a small air-controlled valve. This valve was controlled by a float control arrangement in order to keep free water level constant. During the operational program this component did not work properly, hence most of the level adjustments were done by observing the water level visually following manual adjustments of the main control valve. Measurements of water mass flow rate were accomplished by using a 1.89+/-0.003 centimeter orifice plate. The orifice plate was connected to a mercury filled U-tube manometer and a Taylor pressure transmitter which sent voltage signals to the data logger. Next, voltage signals were converted to the pressure drop information by using a set of equations which were also calibrated by actual manometer readings. In actual runs both readings were recorded continuously. The water which passed through the orifice plates then entered a plate heat exchanger. The heat exchanger was constructed with 33 separate plates. It was designed to operate at 6.9 bars pressure and has a total surface area of 8.676 square meters. Steam from the Utah Power and Light Company Gatsby Power Unit was used to heat the incoming water. The amount of steam entering through the heat exchanger was controlled by a gate valve. The steam use variation in the power unit affected the total steam pressure slightly, consequently small changes in mass flow rate of steam occurred. Because of this, continuous checking and subsequent slight adjustments of steam mass flow rate were performed manually and evaluated by (1) using water inlet temperature of the direct contact heat exchanger and (2) condensed steam mass flow rates. Water inlet temperature of the column was kept constant at 85+/- 1 °C. A few data points were also taken for the water inlet temperature of the column specifically at 76+/- 1 °C and 94+/- 1 °C. The water exited from the bottom of the tower and flowed back to the water tank.

The pentane loop included a pentane storage tank, a centrifugal pump, main control bypass valves and condensers. The storage tank was the same kind of 500 gallon container used for the water. This had a capacity of about 1500 liters pentane. The empty portion of the column was filled with water. The water level at the bottom of the tank was about 30 to 45 centimeters high. A water drainage system was located at the bottom of the tank which was also used for small pressure adjustments. Two glass tubes were used to check the water level in the storage tank. Even though a big percentage of vapor coming out of the direct contact boiler was pentane vapor, a small percentage of it was the steam which entered to the tank in condensed form. In order to keep the system pressure constant a small amount of water drained from the tank periodically. Inlet of the pentane pipe was designed to be 50 cm high from the bottom to insure that water was not mixed with the pentane.

The pumping was carried out by a centrifugal explosion-proof pump. It was originally designed with a capacity of 120 gpm at a head of 85 feet. An air actuated valve was located just after the pump. In actual system runs, it was observed that pump capacity was too large. As a solution to this, a secondary parallel by-pass system with a feedback line to the tank was developed. Double sealing ball valves were used in this line for both feedback and main flow. The pentane flow rate measurements were done by a similar system as that of the water loop. Two flat shaped orifice plates were used with diameters of 1.072+/- 0.003 and 1.778+/- 0.003 centimeters. Pressure measurements were accomplished using a mercury filled incline manometer and a Taylor transmitter. The reason for using two orifice plates was, first, to keep the incline manometer head within the bounds of the manometer and, second, to ensure that the voltage signals were within the reading limits of the analog voltmeter. Transmitter voltage was also recorded on a digital datalogger. In order to save mercury in the case of an accidential overpressurization, a mercury trapping system was attached to mercury manometer. No temperature control was planned for the pentane stream going into the tower. Actual runs caused overheating of the pentane and drove temperatures out of the desired limits. Therefore a water stream was channeled along a section of the inlet pipe to partially control the inlet pentane stream.

The pentane vapor coming out of the direct contact tower entered into the first of two serial shell and tube heat exchangers. The first heat exchanger operated as a condenser. Condensed pentane was cooled furher by the second heat exchanger. A thermocouple was placed at the exit of the subcooler to check the exit temperature of pentane to adjust the cooling water flow rate. Tap water was used as cooling water for both heat exchangers. Cooling water flow rates for each heat exchanger could be independently controlled by valves. The liquid pentane returned to the pentane tank and completed the loop.

Experimental Procedure

A setup procedure was developed to restart the system after a long period of down time. The pentane loop was insulated from the rest of the system by closing all the valves connecting the pentane loop to the tower. The water side valves were opened and tap water was added to the system through the feedwater line. The water side pumps were started and water was heated in the plate heat exchanger by steam. Thermocouples at the top of the column and at the exit of the pentane side heat exchanger were removed and the air in the system was let out. The process was continued until all the air came out and the column was completely filled with water. Hot water was ejected through thermocouple holes for about another half an hour to be sure that no air was left in the system. Then the thermocouples were installed, and the pumps were stopped. Part of the additional water was ejected from the column by letting the water out from the water side by a drainage valve. When one third of the column was drained, vacuum and pressure forces balanced each other and the flow of water stopped. The pentane valve was opened for a brief time and a small amount of pentane was added to the column. Pentane was evaporated in the column by hot water and repressurized the system so that more water could be ejected to bring the water level in the vessel to the desired operational level.

A different daily startup procedure was applied as follows: The pentane side was closed the same way as described above. The water side valves were opened and pumps were started. The mass flow rate of the water was adjusted manually according to a mercury manometer reading. The steam valve was opened and the water heated up during the circulation. Condenser and subcooler cooling water valves were opened to prepare them for operation. When the water temperature had reached about 60 °C, the first pentane inlet stream was introduced into the column. During the heating of the system, the pentane flow rate was kept less than the desired operational mass flow rates. When the water inlet temperature approached the desired operational temperature, the actual pentane flow rate was controlled by adjusting the pentane inlet valve and checking the incline mercury manometer across the pentane orifice. When the operational water inlet temperature was reached, the steam flow rate was adjusted to keep the inlet temperature constant. This adjustment was made by checking the thermocouples and the condensed steam flow rate coming out of the plate heat exchanger. After the system was thought to be stabilized an additional 10 to 20 minutes were spent in waiting to insure steady state operation. The data-logger was started to take readings at 2 minute intervals and actual experimental data were taken for a period of half an hour. During this period, pressure differences of manometers, pressures and active level of the column were recorded manually for 5 minute periods. After each measurement, the stability of the system was checked for 10 to 20 minutes before taking another set of measurements. Column pressure and mass flow rates were changed for different data points. In order to obtain higher pressures, the cooling water flow for both the condenser and subcooler was decreased.

Heat loss data were taken during the shutdown of the system. Two different methods were used to obtain data for the heat loss. In the first method, column temperatures were kept constant after shutting down the pentane flow. This was accomplished by a small amount of heat addition to the system through the plate heat exchanger. Since of all energy in the heat exchanger goes to counterbalance the heat loss, the heat loss is calculated easily from the amount of heat added to the system. In the second method, both sides of the flow were turned off and the temperature of the system was recorded for 1 minute periods. The heat loss was calculated from the time dependent cooling of the system. After the heat loss data were taken, all of the electrical equipment and switches were turned off.

Data Reduction

Mass Flow Rates Calculations

The mass flow rate calculations for the square-edged orifice plates was carried out by using the following equations:

$$\text{B-3-1} \qquad m = 0.52502 \left(\frac{C\, \nu\, d^2\, F_a}{1 - \beta^4}\right) \sqrt{\rho_1 (P_1 - P_2)}$$

B-3-2 $$C = K\sqrt{1 - \beta^4}$$

B-3-3 $$K = K_0 \left(1 + \frac{A}{R_d}\right)$$

B-3-4 $$K_0 = K_e \left(\frac{10^6 d}{10^6 d + 15 A}\right)$$

B-3-5 $$K_e = 0.5993 + \frac{0.007}{D} + \left(0.364 + \frac{0.076}{\sqrt{D}}\right)\beta^4 + 0.4\left(1.6 - \frac{1}{D}\right)^5 \left[\left(0.07 + \frac{0.5}{D}\right) - \beta\right]^{5/2} + E$$

where E is an additional term that changes with β and D as shown below:

If $\left(0.007 + \frac{0.5}{D} - \beta\right)$ is greater than zero, and if $0.5 - \beta$ is greater than zero, then

B-3-6 $$E = -\left(0.009 + \frac{0.0314}{D}\right)(0.5 - \beta)^{3/2}$$

and if $\beta - 0.7$ is greater than zero, then

B-3-7 $$E = \left(\frac{65}{D^2} + 3\right)(\beta - 0.7)^{5/2}$$

If $\left(0.007 + \frac{0.5}{D} - \beta\right)$ is smaller than zero, then E is zero. The term A is calculated as,

$$\text{B-3-8} \qquad A = d(830 - 5000\,\beta + 9000\,\beta^2 - 4200\,\beta^3 + \frac{530}{\sqrt{D}})$$

where A, K, K_e and K_0 are the flow coefficients, C is the discharge coefficient, and d and D are the diameters of the orifice plate and inside of the pipe respectively. β is the ratio of the orifice to the pipe diameter. ν is the dynamic viscosity (lbm/ft-sec). For operating temperatures the thermal expansion coefficient F_a can be taken as 1.002 from the tables. A subroutine was developed which computed the mass flow rates. This used inputs of pressure difference, radius of the pipe and orifice and densities of manometer fluid and the flowing fluid.

Energy Balance

The energy balance is the application of the First Law of Thermodynamics to the experimental apperatus. For these calculations, it was assumed that the mass flow rates of pentane entering and leaving the vessel are constant. Furthermore the inlet mass flow rate was assumed to be equal to the sum of the liquid and vapor outlet mass flow rates. It was further assumed that the partial pressure of the water vapor was equal to the saturation pressure of the water at the pentane exit temperature.

These assumptions led to the following equation

$$\text{B-3-9} \qquad m_{wi}^{l} h_{wi}^{l} + m_{pi}^{l} h_{pi}^{l} = m_{po}^{v} h_{po}^{v} + m_{wo}^{l} h_{wo}^{l} + (m_{wi}^{l} - m_{wo}^{l}) \, h_{wo}^{v} + Q_{loss}$$

subscripts i and o represent inlet and outlet conditions and w and p stand for pentane and water, respectively. Superscripts l and v represent the liquid and vapor cases. Q_{loss} is the vessel heat loss. In order to determine the water mass flow rate, it is assumed that partial pressures of water and pentane vapor are proportional to mole flow rates of the phases.

$$\text{B-3-10} \qquad \frac{P_{wt}(T_{po})}{P_{pt}} = \frac{(m_{wi}^{l} - m_{wo}^{l}) \, / \, M_w}{m_{po}^{v} \, / \, M_p}$$

where M_w and M_p are the molecular weights of water and pentane respectively. Thus, the outlet mass flow rate of water can be found as,

$$\text{B-3-11} \qquad m_{wo}^{l} = m_{wi}^{l} - \frac{P_{wt} \, M_w}{P_{pt} \, M_p} \, m_{po}^{v}$$

Heat Loss

The heat loss was calculated by two methods. In the first method the steady state heat loss data were used. The equation for the heat loss in this case is

B-3-12 $$Q_{loss} = h^{1}_{wi} \; (h^{1}_{wi} - h^{1}_{wo})$$

The second method was based on the change of heat transfer as a function of time. The temperature data as a function of time were obtained when the system was shut down. Hence, all temperature changes were considered to be due to heat loss. The column was computed discretely as the sum of partial volumes . Each volume was assumed to be in a constant temperature. The total heat loss in the column is,

B-3-13 $$Q_{loss} = \sum_{i=1}^{13} \rho_w (T_i)\, V_i (\Delta 1) \;\; C_{pw} (T_i) \; \frac{\partial T_i}{\partial t} + \sum_{i=1}^{15} \rho_{wall}(T_i) \;\; V_{wall} (\Delta 1) \;\; C_{p\ wall} \; \frac{\partial T_i}{\partial t}$$

or by replacing the differential terms by difference equation becomes,

B-3-14 $$Q_{loss} = \sum_{i=1}^{13} \rho_w(T_i) V_i (\Delta 1) \;\; C_p (T_i) \; \frac{T_i (t_2) \; - \; T_i (t_1)}{t_2 \; - \; t_1} + \sum_{i=1}^{15} \rho_{wall} C_{p\ wall} V_{wall} (\Delta 1) \; \frac{T_i (t_2) \; - \; T_i (t_1)}{t_2 \; - \; t_1}$$

Volumetric Heat Transfer Coefficient

The volumetric heat transfer coefficient is defined as the total heat transfer divided by logarithmic mean temperature difference. The total heat transfer can be assumed to be the heat transfer to the pentane or the heat transfer from the water phase, minus the heat loss. For the logarithmic mean temperature difference the standard definition for heat exchangers was taken. The definition of the logarithmic mean temperature difference has the following form,

B-3-15 $$\mathrm{LMTD} = \frac{(T_{wi} - T_{po}) - (T_{wo} - T_{pi})}{\ln \frac{(T_{wi} - T_{po})}{(T_{wo} - T_{pi})}}$$

APPENDIX C

SUBROUTINE FOR THERMODYNAMIC PROPERTIES

Subroutines had been developed at University of Utah [20],[21],[22] for calculating the thermodynamic properties of water, light hydrocarbons and Freons in British units. Additional subroutines are added to this programs in order to use them in SI units. Furthermore an additional subroutine was written to get access to all of the available fluids without using different subprograms.

In order to use the thermodynamic routines, the following statement should be added to the program written in Fortran 77 language.

```
CALL SIFLUID(CYCLE,LFTYPE,FGIVEN,SGIVEN,QL,ENTR,ENTH,VOL,U,
            TEMP,PRESS,SL,HL,VL,UL,SV,HV,VV,UV)
```

SI units used in this subroutine and the following units should be used for the particular thermodynamic properties

Temperature : Degree C

Pressure : kPa ($N/m^2 *10^{-3}$)

Internal energy : kJ/kg

Enthalpy : kJ/kg

Entropy : kJ/kg deg C

Specific volume : m^3/kg

The input variables used in the subroutine are;

LFTYPE :type of the substance thermodynamic properties to be evaluated. This is a 6 letter character type variable. Specific names should be used for each fluid as follows

'H2O   ' : water and steam

Hydrocarbons :

'ISOB ' : Isobutane

'BUT   ' : Butane

'ISOP ' : Isopentane

'PENT ' : Pentane

'METH ' : Methane

'PROP ' : Propane

'HEX   ' : Hexane

'HEPT ' : Heptane

'OCT   ' : Octane

'ETHYL ' : Ethylene

'PROPYL' : Propylene

Freons :

'F113 ' : Freon-113

'F114 ' : Freon-114

'F11   ' : Freon-11

'F12   ' : Freon-12

'F13   ' : Freon-13

'F14   ' : Freon-14

'F21   ' : Freon-21

'F22   ' : Freon-22

'F23   ' : Freon-23

CYCLE : Identifier of known thermodynamic variables. This is a 2 letter character type variable. Specific names shoud be used to identify for each thermodynamic known variable pairs are as follows;

'TP' : Temperature - Pressure
'TX' : Temperature - Quality
'PS' : Pressure - Entropy
'PH' : Pressure - Enthalpy
'PX' : Pressure - Quality

FGIVEN : First given thermodynamic variable
SGIVEN : Second given thermodynamic variable

The output variables are ;

QL : Quality of the fluid. QL is equal to -2 for liquid and 2 for superheated gases.
ENTR : Entropy of the given state.
ENTH : Enthalpy of the given state.
VOL : Specific volume of the given state.
U : Internal energy of the given state.
TEMP : Temperature of the given state.
PRESS : Pressure of the given state.
SL : Entropy of the saturation liquid.
HL : Enthalpy of the saturation liquid.
UL : Internal energy of the saturation liquid.
SV : Entropy of the saturation vapor.
HV : Enthalpy of the saturation vapor.
UV : Internal energy of the saturation vapor.

REFERENCES

1. A. E. Adams and K. L. Pinder, "Average Heat Transfer Coefficient During the Direct Evaporation of a Liquid Drop," The Canadian Journal of Chemical Engineering, Vol 45, pp 215-219, August 1967.

2. P. Batya, V. R. Raghavan and K. N. Seetheramu, " A Theoretical Analysis of Direct Contact Evaporation in Spray Columns," Int. Comm. Heat Mass Transfer, Vol 10, pp. 553-543, 1983.

3. H. Carlslaw and J. Jeager, Conduction of Heat in Solids, Oxford, 1959.

4. S. W. Churchill, and M. Bernstein, "A Correlating Equation for Forced Convection from Gases and Liquids to a Circular Cylinder in Crossflow," J. Heat Transfer, Trans. ASME, Vol. 59, p 583, 1937.

5. J. R. Grace, "Hydrodynamics of Liquid Drops in Immiscible Liquids," Chapter 38 in Handbook of Fluids in Motion (n. Cheremisinoff and R. Gupta, Eds.), Ann Arbor Science, p 1008, 1983.

6. P. Goodwin, M. T. Coban and R. Boehm, "Evaluation of the Flooding Limits and Heat Transfer of a Direct Contact Three Phase Spray Column Direct Contact Heat Exchanger," ASME Paper 85-HT-49, 1985.

7. H. Jacobs and R. Boehm, "Direct Contact Binary Cycles," in Sourcebook on the Production of Electricity from Geothermal Energy (J. Kestin et al. Eds.), DOE/RAW/4051-1, 1980.

8. H. Jacobs and M. Golafshani, "A Hueristic Evaluation of the Governing Mode of Heat Transfer in a Liquid-Liquid Spray Column," presented at the 1985 National Heat Transfer Conference, Denver, Colorado, 1985.

9. J. H. Keenan, F. G. Keys, P. G. Hill, and J. G. Moore, Steam Tables, Wiley-Inter Science Publication, 1978.

10. J. Klee and R. E. Treybal, "Rate of Rise or Fall of Liquid Drops," A.I.Ch.E. Journal, pp 444-447, December 1956.

11. M. Mokharzadeh and A. El-Shirbini, "A Theoretical Analysis of An Evaporating Droplet in an Immiscible Liquid," Int. J. Heat Mass Transfer, 22, pp. 27-38, 1979.

12. M. R. Mokhtarzadeh-Deghan and A. A. El-Shirbini, "Dynamics of Two-Phase Bubble-Droplets in Immiscible Liquids," Wårme- undStoffübertragung 19, pp. 53-59, 1985.

13. Y. Mori and K. Komotori, "Boiling Modes of Volatile Liquid Drops in an Immiscible Liquid Depending on Degree of Superheat," presented at 1976 National Heat Transfer Conference, 1976.

14. R. Orlander, S. Oshmyansky, K. Nichols, and D. Werner, "Final Phase Testing and Evaluation of the 500 kW Direct Contact Pilot Plant at East Mesa," US DOE Report DOE/SF/11700-T1 (DE840044781), December 1983.

15. C. B. Parakash and K.L. Pinder, "Direct Contact Heat Transfer Between Two Immiscible Liquids During Vaporization, Part I Measurements of Heat Transfer Coefficient," The Canadian Journal Of Chemical Engineering,50, pp. 707-713, 1972.

16. G. K. Raina and P. D. Grover, "Direct Contact Heat Transfer with Change of Phase: Theoretical Model," AIChE Journal,28, pp. 515-517, 1982.

17. G. K. Raina and R. Wanchoo, "Direct Contact Heat Transfer with Phase Change: Theoretical Expression for Instantaneous Velocity of a Two-phase Bubble," Int. Comm. Heat Mass Transfer,11, pp. 227,237, 1984.

18. G. K. Raina, R. Wanchoo and P. Grover, "Direct Contact Heat Transfer with Phase Change: Motion of Evaporating Droplets," AIChE Journal,30, pp. 835-837, 1984.

19. G. K. Raina, R. Wanchoo and P. Grover, "Direct Contact Heat Transfer with Change of Phase: Theoretical Model Incorporating Sloshing Effects," AIChE Journal,31, pp. 507-510, 1985.

20. P. M. Rapier, "Design Features and Equilibrium Flash Modeling of Direct Binary Fluid Heat Exchangers for Use with Geothermal Brines," Paper presented at the 20th ASME- AIChE national Heat Transfer Conference, Augus 2-5, 1981.

21. D. Reimer, H. Jacobs and R. Boehm, " A Computer Program for Determining the Thermodynamic Properties of Water," University of Utah Report UTEC ME-76-171, 1976.

22. D. Reimer, H. Jacobs and R. Boehm, " A Computer Program for Determining the Thermodynamic Properties of Freon Refrigerants," University of Utah Report UTEC ME-76-212, 1976.

23. D. Reimer, H. Jacobs, R. Boehm and D. S. Cook, " A Computer Program for Determining the Thermodynamic Properties of Light Hydrocarbons," University of Utah Report UTEC ME-76-211, 1976.

24. S. Sideman and J. Isenberg, "Direct Contact Heat Transfer with Change of Phase: Bubble Growth in Three-Phase Systems," Desalination,2, pp. 207,214, 1967.

25. S. Sideman and G. Hirsch, "Direct Contact Heat Transfer with Change of Phase III: Analysis in the Transfer Mechanism of Drops Evaporating in Immiscible Liquid Media," Israel Journal of Technology Vol 2, No 2, pp 234-241,1965.

26. S. Sideman, G. Hirsch and Y. Gat, "Direct Contact Heat

Transfer with Change of Phase: Effect of the Initial Drop Size in Three Phase Heat Exchanger," A.I.Ch.E. Journal, pp 1081,1086, 1965.

27. S. Sideman and S. Shabtai, "Direct Contact Heat Transfer Between a Single Drop and an Immiscible Medium," Canadian Journal of Chemical Engineering, pp. 107-117, June 1964.

28. S. Sideman and Y. Taitel, "Direct Contact Heat Transfer with Change of Phase: Evaporation of Drops in Immiscible Liquid Medium," Int. J. Heat Mass Transfer,7, pp. 1273- 1289, 1964.

29. A. V. Simms, "Final Report Geothermal Direct Contact Heat Exchange," Prepared for the United States Research and Development Administration Contract No. E(04-3)-1116, 1979.

30. H. C. Simpson , G. Beggs and M. Nazir, "Evaporation of Butane Drops in Brine," Desalination,15,pp. 11-23, 1974.

31. R. C. Smith, W. M. Rohsenow, M. S. Kazimi, "Volumetric Heat Transfer Coefficients for Direct-Contact Evaporation," Transactions of ASME, vol 104, pp. 264-270, May 1982.

32. B. Sudhoff, M. Plische and P. M. Weinspach, "Direct Contact Heat Transfer with Change of Phase-Condensation or Evaporation of a Drobble," Ger. Chem. Eng. 5, pp. 24-43, 1982.

33. W. B. Suratt et al., "Study and Testing of Direct Contact Heat Exchangers for Geothermal Brines," United States Department of Commerce National Technical Information Service Report ORO-4891-1, January 1977.

34. L. Tadris, P. Seguin, R. Santini, J. Pantoloni, "Experimental and Numerical Study of Direct Contact Heat Exchangers," Int. J. Heat Mass Transfer, Vol 28, no 6, pp 1215 - 1227,1985.

35. Y. Tochitani, T. M. Mori and K. Komotori, "Vaporization of a Liquid Injected into an Immiscible Liquid through a Single Nozzle," Warme- und Stoffubertragung, 8, pp. 249-259, 1975.

36. F. White, Viscous Fluid Flow, McGraw-Hill, 1974.

37. N. B. Vargaftick, Table on Thermophysical Properties of liquids and Gases, Second Edition, Hemisphere Publishing, 1975.

APPENDIX D

PROGRAM LISTING

```
C  ************************************************
C  *                                              *
C  *           DIRECT CONTACT BOILER              *
C  *                                              *
C  *        This program is written by :          *
C  *          MUSTAFA TURHAN COBAN                *
C  *                                              *
C  ************************************************
C
C  INPUT VARIABLES :
C
C  FILANA  : File name which input data is written in
C  ICHO    : Dispersed phase liquid identification code
C  XL      : Column Height                          (m)
C  DINS    : Column Diameter                        (m)
C  AMASSC  : Mass flow rate of the continuous fluid(kg/sec)
C  AMASSD  : Mass flow rate of the dispersed fluid (kg/sec)
C  X0      : Starting point of the column           (m)
C  N       : Number of division of the column
C  TDINIT  : Initial Temperature of the dis. fluid (deg C)
C  TCINIT  : Initial Temperature of the con. fluid (deg C)
C  PDINIT  : Initial pressure of the column        (Pa)
C  RINIT   : Initial radius of the bubble diameter (m)
C  HUR     : Initial guess for the holdup ratio
C
C  ----------------------------------------------------------
C
          IMPLICIT DOUBLE PRECISION (A-H,O-Z)
   PARAMETER ND=200,ID=8
   DIMENSION X(0:ND),Y(ID,ND),Y0(ID),NBCF(ID),YF(ID),XI(20)
    #      ,YI(20),XII(ID),YII(ID),DELTAX(ID),E(20),CVISPL(20),
    &      CVISPV(20),CKPL(2),CKPV(2),ZZ(15,100)
   CHARACTER*20 FILENA,FILE
   CHARACTER*6 LFTYPE,FTYPE
   CHARACTER*2 CYCLE
   CHARACTER*4 FNA
   COMMON/NPRT/NPRT
   COMMON/NCOEF/NCOEF
   COMMON/TEMC/TEMC
   COMMON/FTYPE/FTYPE
   COMMON/CKLD/CKLD
   COMMON/CKVD/CKVD
   COMMON/CKC/CKC
```

```
    COMMON/CKAIR/CKAIR
    COMMON/VISLD/VISLD
    COMMON/VISVD/VISVD
    COMMON/VISC/VISC
    COMMON/VISAIR/VISAIR
    COMMON/SIGW/SIGW
    COMMON/SIGD/SIGD
    COMMON/ROAIR/ROAIR
C   WIND VELOCITY,AIR TEMPERATURE
    COMMON/VELAIR/VELAIR
    COMMON/TAIR/TAIR
C   HEIGHT OF THE BAFFLES
    COMMON/XL/XL
    COMMON/DINS/DINS
C   INITIAL RADIUS,HOLDUP
    COMMON/RINIT/RINIT
    COMMON/RODINT/RODINT
    COMMON/VELINT/VELINT
    COMMON/SINIT/SINIT
C   NORMALIZE HC,HD,ROC,P
    COMMON/HINITC/HINITC
    COMMON/HINITD/HINITD
    COMMON/ROINIC/ROINIC
    COMMON/IQUALD/IQUALD
    COMMON/QUALD/QUALD
    COMMON/PINIT/PINIT
C   MASS FLOW RATES
    COMMON/AMASSC/AMASSC
    COMMON/AMASSD/AMASSD
C
    COMMON/FLAG/FLAG
    COMMON/XQUAL/XQUAL
    COMMON/HFG/HFG
    COMMON/ITOT/ITOT
    COMMON/TOTHOL/TOTHOL
    COMMON/THOLD/THOLD
    COMMON/TD/TD
        COMMON/ERFLAG/ERFLAG
    COMMON/IFLAG/IFLAG
    COMMON/FILE/FILE
C   END OF COMMONS
C   NORMALIZATION DATA
    LFTYPE='H2O   '
    CYCLE='TP'
    TT=4.
    PP=101.325
    PINIT=PP*1.E03
    CALL SIFLUID(CYCLE,LFTYPE,TT,PP,QL,ENTR,HINITC,VOL,U,TEMP,
     1      PRESS,SL,HL,VL,UL,SV,HV,VV,UV)
    ROINIC=1./VOL
    TT=4.
```

```
      PP=101.325
      PRINT*,'Dispersed phase liquid'
      PRINT*,'1 : pentane'
      PRINT*,'2 : Butane'
      PRINT*,'3 : Isobutane'
      PRINT*,'4 : Isopentane'
      PRINT*,'Choose Your dispersed phase liquid'
      READ*, ICHO
      IF(ICHO.EQ.1) THEN
      FTYPE='PENT  '
      ELSEIF(ICHO.EQ.2) THEN
      FTYPE='BUT   '
      ELSEIF(ICHO.EQ.3) THEN
      FTYPE='ISOB  '
      ELSEIF(ICHO.EQ.4) THEN
      FTYPE='ISOP  '
      ENDIF
      WRITE(6,*) '*',FTYPE,'*'
      LFTYPE=FTYPE
      CALL SIFLUID(CYCLE,LFTYPE,TT,PP,QL,ENTR,HINITD,VOL,U,TEMP,
     1      PRESS,SL,HL,VL,UL,SV,HV,VV,UV)
      ROINID=1./VOL
      PRINT*, 'INPUT FILE NAME ='
      READ(5,66) FILENA
C     PRINT*,'Output file name ='
C     READ(*,'(A20)') FILE
      FNA=FILENA(2:5)
      WRITE(FILE,'(A2,A4)') 'OL',FNA
   66 FORMAT(A10)
   67 FORMAT(A6)
      OPEN(UNIT=10,FILE=FILENA,DISPOSE='SAVE'
     #  ,ERR=999,STATUS='UNKNOWN')
      GOTO 99
  999        PRINT*, 'ERROR IN FILE READING'
   99 CONTINUE
C     COLUMN DATA
C     Length of the column,Inside Diameter
      READ(10,*) XL,DINS
c     Mass flow rates
c     M. F. R. of Continious phase,Dispersed Phase
      READ(10,*) AMASSC,AMASSD
      AMASSC=-AMASSC
      IE=7
C     INITIAL X VALUE =
      READ(10,*) X0
C     FINAL X VALUE   =
      XLIM=XL
C     NUMBER OF POINTS =
      READ(10,*) N
C     TEMPERATURE CONT,DISPERSED PRESSURE
      READ(10,*) TDINIT,TCINIT,PDINIT
```

```
      PRC=PDINIT
      CALL PROPTY(TDINIT,TCINIT,25.,PRC)
C     'INITIAL BUBBLE RADIUS'
      READ(10,*) RINIT
      Y0(7)=RINIT
C     'INITIAL HOLDUP RATIO ='
      READ(10,*) HUR
      Y0(6)=HUR
C     PCINIT=PDINIT-2*SIGD*1.E-03/RINIT
C     TDINIT,TCINIT Deg C
C     PINIT KPa
      PIN=PDINIT
      CYCLE='TP'
      LN=10
      NS=0
      NP=0
      FGIVEN=TCINIT
      SGIVEN=PIN*1.E-03
      LFTYPE='H2O    '
      CALL SIFLUID(CYCLE,LFTYPE,FGIVEN,SGIVEN,QUINIT,SINIT,
     1  HINIT,VINIT,UINIT,TEMP,PRESS,SL,HL,VL,UL,SV,HV,VV,UV)
      Y0(1)=PDINIT
      Y0(2)=HINIT*1.D03
C     'INITIAL Y VALUES ='
      PIN=PDINIT
      FGIVEN=TDINIT
      SGIVEN=PIN*1.D-03
      ROC=1./VINIT
      TD=TDINIT
      LFTYPE=FTYPE
      CYCLE='TP'
      CALL SIFLUID(CYCLE,LFTYPE,FGIVEN,SGIVEN,QUINIT,SINIT,
     1  HINIT,VINIT,UINIT,TEMP,PRESS,SL,HL,VL,UL,SV,HV,VV,UV)
      Y0(3)=HINIT*1.D03
      HINITD=Y0(3)
      ROD=1/VINIT
      RODINT=ROD
      PI=4.*ATAN(1.)
      VD=AMASSD/(PI/4.*DINS*DINS*ROD*HUR)
      SIG=ABS(SIGW-SIGD)
      Y0(4)=VD
      Y0(5)=AMASSC/(PI*DINS*DINS/4.*ROC*(1.-HUR))
      K=1
      WRITE(6,15) 'Y(',K,',0) = '
      DO 10 K=1,IE
C     IF(K.LE.4) GOTO 12
C     IF(K.EQ.7) GOTO 12
C     WRITE(6,15) 'Y(',K,',0) = '
C     READ(10,*) Y0(K)
   12         CONTINUE
   10         XII(K)=Y0(K)
```

```
    SINIT=HUR
    YØ(6)=SINIT
  15        FORMAT(1X,A2,I1,A6)
    WRITE(6,*) 'INITIAL OR FINAL=1 Y VALUE AS BOUNDARY CONDITION'
    DO 115 I=1,IE
 115        NBCF(I)=2.
C   'BOUNDARY CONDITION OF EQUATION',2
    PRINT*,'INITIAL CONDITION FOR WATER INLET  1 : END CONDITION'
    PRINT*,'                               OTHER : INITIAL COND.'
    READ(1Ø,*) NBCF(2)
 16 FORMAT(1X,A31,I2)
    IF(NBCF(2).EQ.1) THEN
C
    PRINT*,'END TEMPERATURE AT THE TOP OF THE COLUMN FOR THE CONT. PHASE'
    READ(1Ø,*) YF(2)
C   'STARTING INITIAL VALUE GUESS ;',YØ(I)
    PRINT*,'RANGE OF INITIAL CONT. PHASE TEMP'
    READ(1Ø,*) DELTAX(2)
    ENDIF
    CLOSE(UNIT=1Ø,STATUS='SAVE')
    PRINT*,'NCOEF ='
    READ*,NCOEF
    IF(NBCF(2).EQ.1)THEN
    DO I=1,IE
    XII(I)=YØ(I)
    ENDDO
    PRINT*,'YØ(2)=',YØ(2)
    TTC=TCINIT-DELTAX(2)/2.
    PRINT*,'TTC =',TTC,'TCINIT =',TCINIT
    JJ=3
    DO 14 J=1,JJ
    TEMPC=TTC+DELTAX(2)*DFLOAT(J-1)/DFLOAT(JJ-1)
    PRINT*,'TEMPC =',TEMPC,'TCINIT =',TCINIT
    FGIVEN=TEMPC
    SGIVEN=PIN*1.E-Ø3
    LFTYPE='H2O    '
    CYCLE='TP'
    CALL SIFLUID(CYCLE,LFTYPE,FGIVEN,SGIVEN,QUINIT,SINIT,
     1  HINIT,VINIT,UINIT,TEMP,PRESS,SL,HL,VL,UL,SV,HV,VV,UV)
    XII(2)=HINIT*1.DØ3
    PRINT*,'XII(2)',XII(2)
    NPRT=Ø
    CALL EULER(IE,X,Y,XLIM,N,XØ,XII)
C   PRINT*,'NEW FLAGS SHOUD BE SET UP IN THIS POINT'
    TD=TDINIT
    FLAG=Ø.
    ERFLAG=2.
    IFLAG=Ø.
    IQUALD=-1
    XQUAL=Ø.
    QUALD=-1.
```

```
      XI(J)=TEMC
      YI(J)=TEMPC
      PRINT*,'X =',XI(J),'Y =',YI(J)
   14 CONTINUE
      NN=JJ
      LD=JJ-1
      CALL LSTSQR(XI,YI,NN,LD,E)
          TCINIT=FUNC(E,LD,YF(2))
      FGIVEN=TCINIT
      SGIVEN=PIN*1.E-03
      LFTYPE='H2O   '
      CYCLE='TP'
      CALL SIFLUID(CYCLE,LFTYPE,FGIVEN,SGIVEN,QUINIT,SINIT,
       1  HINIT,VINIT,UINIT,TEMP,PRESS,SL,HL,VL,UL,SV,HV,VV,UV)
      XII(2)=HINIT*1.D03
      Y0(2)=XII(2)
      ENDIF
      NPRT=1
      CALL EULER(IE,X,Y,XLIM,N,X0,Y0)
      HDFIN=Y(3,IE)
      STOP
      END
      SUBROUTINE EULER(IE,X,Y,XLIM,N,X0,Y0)
C     ********************************************************
C     *                                                      *
C     *       EULER                SOLUTIONS                 *
C     *       FOR THE DIFFERENTIAL EQUATIONS                 *
C     *                                                      *
C     *  __________________________________________________  *
C     *           MUSTAFA TURHAN COBAN                       *
C     *  __________________________________________________  *
C     *                                                      *
C     ********************************************************
C
C     PROGRAM DEFINATION :
C
C     THIS SUBROUTINES CALCULATES SOLUTION FOR THE SYSTEM OF
C     DIFFERENTIAL EQUATIONS BY USING EULER
C     METHOD INITIAL VALUES OF SOLUTIONS ARE REQUIRED
C
C     ______________________________________________________
C                     VARIABLE IDENTIFICATION :
C     VARIABLES :
C     AK1...AK6 :     COEFFICIENTS FOR RUNGE KUTTE EQUATIONS
C     XX        :     DUMMY VARIABLES FOR X INCREMENTS
C     YY        :     DUMMY VARIABLES FOR RUNGE EQNS.
C     ARRAYS    :
C     X         :     INDEPENDENT VARIABLE
C     Y         :     SOLUTION OF DIFFERENTIAL EQNS. DEPENDENT
C                     VARIABLE
C     Y0        :     INITIAL CONDITIONS OF DEPENDENT VARIABLES
```

```
C     ----------------------------------------------------------
C
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      PARAMETER ND=200,ID=20
      DIMENSION YN(ID),X(0:ND),Y(ID,0:ND),Y0(ID),XX(ID)
      EXTERNAL F
      CHARACTER*6 LFTYPE,FTYPE
      CHARACTER*20 FILE
      COMMON/NPRT/NPRT
      COMMON/FTYPE/FTYPE
      COMMON/XX/XX
      COMMON/CKLD/CKLD
      COMMON/CKVD/CKVD
      COMMON/CKC/CKC
      COMMON/CKAIR/CKAIR
      COMMON/VISLD/VISLD
      COMMON/VISVD/VISVD
      COMMON/VISC/VISC
      COMMON/VISAIR/VISAIR
      COMMON/SIGD/SIGD
      COMMON/SIGW/SIGW
      COMMON/ROAIR/ROAIR
      COMMON/HOC/HOC
      COMMON/HOD/HOD
C     WIND VELOCITY
      COMMON/VELAIR/VELAIR
      COMMON/TAIR/TAIR
C     HEIGHT OF THE BAFFLES
      COMMON/XL/XL
      COMMON/DINS/DINS
C     INITIAL RADIUS
      COMMON/VELINT/VELINT
      COMMON/RINIT/RINIT
      COMMON/RODINT/RODINT
      COMMON/SINIT/   COMMON/HN/HN
C     MASS FLOW RATES
      COMMON/AMASSC/AMASSC
      COMMON/AMASSD/AMASSD
C     NORMALIZE HC,HD,ROC,P
      COMMON/HINITC/HINITC
      COMMON/HINITD/HINITD
      COMMON/ROINIC/ROINIC
      COMMON/PINIT/PINIT
      COMMON/FLAG/FLAG
C     OUTPUT  VARIABLES
      COMMON/TEMD/TEMD
      COMMON/TEMC/TEMC
      COMMON/ROOD/ROOD
      COMMON/ROOC/ROOC
      COMMON/QUAL/QUAL
      COMMON/HOLDUP/HOLDUP
```

```
      COMMON/RADIUS/RADIUS
      COMMON/VELD/VELD
      COMMON/PRES/PRES
      COMMON/VELC/VELC
      COMMON/XQUAL/XQUAL
      COMMON/IQUALD/IQUALD
      COMMON/QUALD/QUALD
      COMMON/HFG/HFG
      COMMON/NCOEF/NCOEF
      COMMON/DD0/DD0
      COMMON/HDROP/HDROP
      COMMON/QUALCH/QUALCH
      COMMON/IFLAG/IFLAG
      COMMON/ITOT/ITOT
      COMMON/TOTHOL/TOTHOL
      COMMON/TDP/TDP
      COMMON/TD/TD
      COMMON/THOLD/THOLD
      COMMON/HDROP0/HDROP0
          COMMON/ERFLAG/ERFLAG
      COMMON/FILE/FILE
C     END OF COMMONS
      IJ=3
      HN=(XLIM-X0)/DFLOAT(N)
C     'HN=',HN
      X(0)=X0
      DO 30 I=1,IE
      Y(I,0)=Y0(I)
   30 CONTINUE
      DO 20 J=0,N
      DO 11 I=1,IE
      YN(I)=Y(I,J)
   11 CONTINUE
      X(J)=DFLOAT(J)*HN+X(0)
      CALL F(X(J),YN)
      DO 13 I=1,IE
      Y(I,J+1)=YN(I)
   13 CONTINUE
   20 CONTINUE
      RETURN
      END
      FUNCTION FUNC(E,LD,XXX)
C     ************************************************
C     *                                              *
C     *   POLYNOMIAL OF DEGREE LD                    *
C     *   Y=E(1)+E(2)*XXX+...+E(LD+1)*XXX**LD        *
C     *                                              *
C     ************************************************
C
      IMPLICIT DOUBLE PRECISION(A-H,O-Z)
      DIMENSION E(20)
```

```
      LD=LD+1
      FUNC=E(1)
      DO 3 L=2,LD
      IM1=L-1
      FUNC=FUNC+E(L)*XXX**IM1
    3 CONTINUE
      RETURN
      END
      SUBROUTINE LSTSQR(XI,YI,NN,LD,E)
C     *********************************************************
C     *                                                       *
C     *              LEAST SQUARE CURVE FITTING               *
C     *                                                       *
C     *     ______________________________________________    *
C     *                  MUSTAFA TURHAN COBAN                 *
C     *     ______________________________________________    *
C     *                                                       *
C     *********************************************************
C
C      PROGRAM DEFINATION :
C
C     THIS PROGRAM CALCULATES LEAST SQUARE CURVE FITTING FOR THE
C     POLYNOMIAL OF THE DEGREE LD
C
C     FOR SOLVING OF THE EQUATIONS GAUSS ELIMINATION METHOD IS USED
C
C     ---------------------------------------------------------
C     VARIABLE IDENTIFICATION :
C
C     LSTSQR  : NAME OF THE SUBROUTINE
C     INPUT VARIABLES :
C     XI      : KNOWN VALUES OF THE INDEPENDENT VARIABLE
C     YI      : KNOWN VALUES OF THE DEPENDENT VARIABLE YI=F(XI)
C     LD      : DEGREE OF THE REQUIRED POLYNOMIAL (GIVEN DATA)
C     LD2     : 2*LD
C     CALCULATION VARIABLES :
C     G       : A DUMMY VARIABLE FOR CALCULATING LEAST SQUARE MATRIX
C     B       : RIGHT HAND SIDE OF THE LEAST SQUARE EQUATION
C     A       : MATRIX OF LD*LD THE LEFT HAND SIDE OF THE L. S. E.
C     GAUSS   : SUBROUTINE NAME FOR THE GAUSS ELIMINATION
C               TO SOLVE LEAST SQUARE SYSTEM OF EQUATIONS.
C     OUTPUT VARIABLE
C     E       : COEFFICIENTD OF THE LEAST SQUARE POLYNOMIAL OF THE
C               DEGREE LD
C
C
C     _________________________________________________________
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      PARAMETER M=20,MM=20
      DIMENSION XI(M),YI(M),A(MM,MM),E(MM),G(0:MM),B(MM)
      LD=LD+1
      LD2=LD*2
```

```
      G(0)=NN
      DO 1 K=1,LD2
              G(K)=0.
              DO 1 I=1,NN
                      G(K)=G(K)+XI(I)**K
C             END OF DO LOOP
 1    CONTINUE
C     CALCULATE THE LEFT HAND SIDE OF THE LEAST SQUARE EQUATION
      DO 2 K=1,LD
              B(K)=0.
              DO 2 I=1,NN
                      IF(K.EQ.1)THEN
                      B(K)=B(K)+YI(I)
                      ELSE
                      B(K)=B(K)+YI(I)*XI(I)**(K-1)
                      ENDIF
 2    CONTINUE
      DO 3 J=0,LD-1
C             SET THE TERMS OF THE EQUATION IN THE MATRIX FORM
              DO 3 I=0,LD-1
                      IPJ=I+J
                      A(I+1,J+1)=G(IPJ)
 3    CONTINUE
C     SOLVE THE SYSTEM OF THE EQUATION AND FIND THE COEFFICIENTS
C     OF THE LEAST SQUARE MATRIX
      CALL GAUSS(A,B,E,LD)
      RETURN
      END
      SUBROUTINE GAUSS(A,B,XX,N)
C     ***************************************************
C     *                                                 *
C     *       THIS SUBROUTINE IS WRITTEN BY             *
C     *       ----------------------------              *
C     *           MUSTAFA TURHAN COBAN                  *
C     *       ____________________________              *
C     ***************************************************
C     THIS SUBROUTINE CALCULATES THE SOLUTION OF AX=B SYSTEM OF
C     EQUATIONS BY USING GAUSS ELIMINATION METHOD. PIVOTING IS
C     ALSO USED IN THE CALCULATIONS.
C
C     ____________________________________________________________
C                     VARIABLE IDENTIFICATION :
C                     A(I,J)*XX(J)=B(J)
C     A        : N*N MATRIX OF THE COEFFICIENTS OF THE UNKNOWN XX
C     B        : N MATRIX OF THE CONSTANTS IN THE EQUATIONS
C     XX       : N MATRIX OF UNKNOWN
C     AMAX     : MAXIMUM ABSOLUTE ELEMENT IN PIVOTING COLUMN
C     C,D      : DUMMY INTERCHANGE VARIABLES.
C     CB,CD    : DUMMY INTERCHANGE VARIABLES.
C     KMAX     : ROW NUMBER OF MAXIMUM COEFFICIENT IN PIVOTING COLUMN
C     AM       : M MULTIPLICATION TEM OF GAUSIAN ALGORITHM
```

TERM

```
C
C     -----------------------------------------------------------
C
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      PARAMETER MM=20
      DIMENSION A(MM,MM),B(MM),XX(MM)
      DO 1 K=1,N
              AMAX=0.
C             PIVOTING.....
              DO 2 I=K,N
C                     FIND THE BIGGEST ELEMENT IN THE COLUMN.
                      IF(ABS(A(I,K)).GT.AMAX ) THEN
                              AMAX=ABS(A(I,K))
                              KMAX=I
                      ENDIF
  2           CONTINUE
C             IF THE MATRIX IS SINGULAR STOP..
              IF(AMAX.EQ.0.) THEN
                      WRITE(5,*) 'MATRIX IS SINGULAR'
                      RETURN
              ELSE
C                     INTERCHANGE THE K"TH LINE WITH THE MAXIMUM LINE
C                     (APPLY PARTIAL PIVOTING)
                      DO 3 J=1,N
                              C=A(K,J)
                              D=A(KMAX,J)
                              A(K,J)=D
                              A(KMAX,J)=C
  3                   CONTINUE
                              CB=B(K)
                              CD=B(KMAX)
                              B(K)=CD
                              B(KMAX)=CB
C                     END OF PIVOTING.
              ENDIF
C             BEGIN TO GAUSS ELIMINATION.
              DO 4 I=K+1,N
                      AM=A(I,K)/A(K,K)
                      DO 5 J=K+1,N
                              A(I,J)=A(I,J)-AM*A(K,J)
  5                   CONTINUE
                      B(I)=B(I)-AM*B(K)
  4           CONTINUE
  1   CONTINUE
C     CALCULATE XX COEFFICIENTS FROM UPPER TRIANGULAR MATRIX
      DO 6 I=N,1,-1
              AX=0.
              IF(I.NE.N) THEN
                      DO 7 J=I+1,N
                              AX=AX+A(I,J)*XX(J)
  7                   CONTINUE
```

```
                ENDIF
                XX(I)=(B(I)-AX)/A(I,I)
  6     CONTINUE
 C      C AND HERE IS THE RESULTS IN THE ARRAY OF XX
        RETURN
        END
        SUBROUTINE DERIVW(LFTYPE,PG,HG,QUAL,ENTR,
         1      ENTH,VOL,U,TEMP,PRESS,SL,HL,VL,UL,SV,HV,VV,UV)
        IMPLICIT DOUBLE PRECISION (A-H,O-Z)
        CHARACTER*6 LFTYPE
        CHARACTER*2 CYCLE
        COMMON/CPC/CPC
        CYCLE='PH'
        IF(LFTYPE.EQ.'H2O   ') THEN
        X=HG
        X1=PG
        FACT=HG*5.D-03
        XX=X+FACT/2.
        HPLUS=XX
        CALL SIFLUID(CYCLE,LFTYPE,X1,XX,QL,ENTR,ENTH,VOL,U,
         1      TEMP,PRESS,SL,HL,VL,UL,SV,HV,VV,UV)
        TPLUS=TEMP
        A1=1./VOL
        XX=X-FACT/2
        HMINUS=XX
        X1=PG
        CALL SIFLUID(CYCLE,LFTYPE,X1,XX,QL,ENTR,ENTH,VOL,U,
         1      TEMP,PRESS,SL,HL,VL,UL,SV,HV,VV,UV)
        TMINUS=TEMP
        B2=1./VOL
        X1=PG
        X2=HG
        CALL SIFLUID(CYCLE,LFTYPE,X1,X2,QL,ENTR,ENTH,VOL,U,
         1      TEMP,PRESS,SL,HL,VL,UL,SV,HV,VV,UV)
        CPC=(HPLUS-HMINUS)/(TPLUS-TMINUS)*1.D+03
        ELSE
                WRITE(6,*) 'WRONG SUBROUTINE IS IN USE'
        ENDIF
        RETURN
        END
        SUBROUTINE DERIVD(LFTYPE,PG,HG,XG,QL,ENTR,
         1      ENTH,VOL,U,TEMP,PRESS,SL,HL,VL,UL,SV,HV,VV,
         2  UV,CPD,HFG,HLIQ,RATIO)
        IMPLICIT DOUBLE PRECISION (A-H,O-Z)
        CHARACTER*6 LFTYPE
        CHARACTER*2 CYCLE
        COMMON/FLAG/FLAG
        COMMON/IQUALD/IQUALD
        COMMON/CPDL/CPDL
        COMMON/CPDV/CPDV
        COMMON/XQUAL/XQUAL
```

```
    COMMON/TD/TD
        COMMON/ERFLAG/ERFLAG
    IF(LFTYPE.NE.'H2O   ') THEN
    IF(FLAG.EQ.2.) GOTO 30
    IF(IQUALD.GE.1) GOTO 30
    IF(((XG.GT.0.).AND.(XG.LE.1.)).OR.(FLAG.EQ.1.)) THEN
 10       X1=PG
      X2=XG
C   WRITE(6,*) 'INSIDE BOILING REGION '
      CYCLE='PX'
      CALL SIFLUID(CYCLE,LFTYPE,X1,X2,QL,ENTR,ENTH,VOL,U,
     1         TEMP,PRESS,SL,HL,VL,UL,SV,HV,VV,UV)
      FACT=2.
      TBOIL=TEMP
      HLIQ=HL
      HVAP=HV
      CYCLE='TP'
      XX=TBOIL-2.
      X1=PG
      CYCLE='TP'
      CALL SIFLUID(CYCLE,LFTYPE,XX,X1,QL2,ENTR2,ENTH2,VOL2,U2,
     1         TEMP2,PRESS2,SL2,HL2,VL2,UL2,SV2,HV2,VV2,UV2)
      TMIN1=XX
      HMIN1=ENTH2
      CYCLE='TP'
      XX=TBOIL-1.5
      X1=PG
      CYCLE='TP'
      CALL SIFLUID(CYCLE,LFTYPE,XX,X1,QL2,ENTR2,ENTH2,VOL2,U2,
     1         TEMP2,PRESS2,SL2,HL2,VL2,UL2,SV2,HV2,VV2,UV2)
      TMIN2=XX
      HMIN2=ENTH2
      X1=PG
      XX=TBOIL+2.
      X1=PG
      CALL SIFLUID(CYCLE,LFTYPE,XX,X1,QL2,ENTR2,ENTH2,VOL2,U2,
     1         TEMP2,PRESS2,SL2,HL2,VL2,UL2,SV2,HV2,VV2,UV2)
      TPLUS1=XX
      HPLUS1=ENTH2
      X1=PG
      XX=TBOIL+1.5
      X1=PG
      CALL SIFLUID(CYCLE,LFTYPE,XX,X1,QL2,ENTR2,ENTH2,VOL2,U2,
     1         TEMP2,PRESS2,SL2,HL2,VL2,UL2,SV2,HV2,VV2,UV2)
      TPLUS2=XX
      HPLUS2=ENTH2
      CPDL=(HMIN2-HMIN1)/(TMIN1-TMIN2)
      CPDV=(HPLUS1-HPLUS2)/(TPLUS1-TPLUS2)
      CPDL=ABS(CPDL)
      CPDV=ABS(CPDV)
      CPD=CPDL+XG*(CPDV-CPDL)
```

```
        CPD=CPD*1.D+03
        CPDL=CPDL*1.D+03
        CPDV=CPDV*1.D+03
        HFG=(HVAP-HLIQ)
        HFG=ABS(HFG)*1.D+03
        ROL=1./VL
        ROV=1./VV
        RATIO=ROL/ROV
        ROD=ROL*(1.-QL)+ROV*QL
        VOL=1./ROD
        HLIQ=HL*1.D+03
      ELSE
   20    CONTINUE
C     WRITE(6,*) 'INSIDE OF PREHEATING REGION'
        CYCLE='TX'
        XFIRST=TD
        XSEC=0.
        CALL SIFLUID(CYCLE,LFTYPE,XFIRST,XSEC,QL3,ENTR3,ENTH3,VOL3,U3,
       1        TEMP3,PRESS3,SL3,HL3,VL3,UL3,SV3,HV3,VV3,UV3)
          IPG=INT(PG)
          IPRES3=INT(PRESS3)
        IF(IPG.LE.IPRES3) THEN
          XG=0.
          FLAG=1.
          GOTO10
        ENDIF
        CYCLE='TP'
        X=HG
        X1=TD
        X2=PG
        TPLUS=X1
        CALL SIFLUID(CYCLE,LFTYPE,X1,X2,QL,ENTR,ENTH,VOL,U,
       1        TEMP,PRESS,SL,HL,VL,UL,SV,HV,VV,UV)
        IF(ERFLAG.EQ.1.) THEN
          XG=0.
          FLAG=1.
          GOTO10
        ENDIF
        HPLUS=ENTH
        CYCLE='TP'
        XX=TPLUS-1.
        TMINUS=XX
        X1=PG
        CALL SIFLUID(CYCLE,LFTYPE,XX,X1,QL2,ENTR2,ENTH2,VOL2,U2,
       1        TEMP2,PRESS2,SL2,HL2,VL2,UL2,SV2,HV2,VV2,UV2)
        HMINUS=ENTH2
        IF(TPLUS.NE.TMINUS) THEN
        CPD=(HPLUS-HMINUS)/(TPLUS-TMINUS)
        CPD=CPD*1.D+03
        CPDL=CPD
        ENDIF
```

```
      ENDIF
      ELSE
              WRITE(6,*) 'WRONG SUBROUTINE IS IN USE'
      ENDIF
      RETURN
 30     CYCLE='TP'
 C           WRITE(6,*) 'INSIDE OF SUPERHEATING REGION'
        X1=TD
        X2=PG
        TMINUS=X1
        CALL SIFLUID(CYCLE,LFTYPE,X1,X2,QL,ENTR,ENTH,VOL,U,
       1         TEMP,PRESS,SL,HL,VL,UL,SV,HV,VV,UV)
        HMINUS=ENTH
        TPLUS=TMINUS+.4
        CYCLE='TP'
        X=TPLUS
        X1=PG
        CALL SIFLUID(CYCLE,LFTYPE,X,X1,QL2,ENTR2,ENTH2,VOL2,U2,
       1         TEMP2,PRESS2,SL2,HL2,VL2,UL2,SV2,HV2,VV2,UV2)
        HPLUS=ENTH2
        B2=1./VOL
        IF(TPLUS.NE.TMINUS) THEN
        CPD=(HPLUS-HMINUS)/(TPLUS-TMINUS)
        CPD=CPD*1.D+03
        CPDV=CPD
        ENDIF
        QL=2.
      RETURN
      END
      SUBROUTINE CONDUC(CONDK,TIME,TC,TD,RBUBLE,HCONDV,RODVAP)
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      COMMON/CKVD/CKVD
      PI=4.*DATAN(1.D0)
      TOT=0.
      DO 1 I=1,10
      AN=DFLOAT(I)
      TOT=TOT+1./AN**4*EXP(-CKVD*AN**2*PI**2*TIME/RBUBLE**2)
  1   CONTINUE
      TAVG=CONDK*(TIME-RBUBLE**2/(15.*CKVD))+6.*CONDK*RBUBLE**3
       #      /CKVD/PI**4*TOT
      Q=RODVAP*TAVG/3.
      HCONDV=Q/(TC-TD)
      RETURN
      END
      SUBROUTINE NEWTON(XX,XQUAL,RATIO)
 C    NEWTON ITARATION FOR CUBIC ROOTS
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      ERR=1.E-5
      PI=4.*DATAN(1.D0)
      IF(XQUAL.LE.0.) THEN
      A=4.
```

```
      ELSEIF(XQUAL.GE.1) THEN
      A=0.
      ELSE
      A=4.*(1.-XQUAL)/(1.+XQUAL*(RATIO-1.))
      ENDIF
      XX=0.
    1 FX=3.*XX-XX*XX*XX+2.-A
      FXP=3.-3.*XX*XX
      XN=XX-FX/FXP
      ER=ABS(XX-XN)
      XX=XN
      IF(ER.LE.ERR) THEN
              YY=ACOS(XX)
              ZZ=YY*180./PI
              RETURN
      ELSE
              GOTO 1
      ENDIF
      RETURN
      END
      SUBROUTINE F(XN,YN)
C     ************************************************
C     *                                              *
C     *       DIFFERENTIAL EQUATIONS                 *
C     *                                              *
C     *______________________________________________*
C     *                                              *
C     ************************************************
C
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      PARAMETER MM=20
      DIMENSION YN(20),AN(MM,MM),BN(MM),XX(MM),DYN(20),ZZ(15,300)
      CHARACTER*6 LFTYPE,FTYPE
      CHARACTER*2 CYCLE
      CHARACTER*20 FILE
C
C     COMMON STATEMENTS
C
      COMMON/NPRT/NPRT
      COMMON/FTYPE/FTYPE
      COMMON/XX/XX
      COMMON/CKLD/CKLD
      COMMON/CKVD/CKVD
      COMMON/CKC/CKC
      COMMON/CKAIR/CKAIR
      COMMON/VISLD/VISLD
      COMMON/VISVD/VISVD
      COMMON/VISC/VISC
      COMMON/VISAIR/VISAIR
      COMMON/SIGD/SIGD
      COMMON/SIGW/SIGW
```

```
      COMMON/ROAIR/ROAIR
C
C     WIND VELOCITY,AIR TEMPERATURE
      COMMON/VELAIR/VELAIR
      COMMON/TAIR/TAIR
C     HEIGHT OF THE BAFFLES
      COMMON/XL/XL
      COMMON/DINS/DINS
C     INITIAL RADIUS
      COMMON/RINIT/RINIT
      COMMON/SINIT/SINIT
      COMMON/RODINT/RODINT
      COMMON/VELINT/VELINT
      COMMON/HN/HN
      COMMON/ERFLAG/ERFLAG
C     MASS FLOW RATES
      COMMON/AMASSC/AMASSC
      COMMON/AMASSD/AMASSD
C     NORMALIZE HC,HD,ROC,P
      COMMON/HINITC/HINITC
      COMMON/HINITD/HINITD
      COMMON/ROINIC/ROINIC
      COMMON/PINIT/PINIT
      COMMON/FLAG/FLAG
C     OUTPUT  VARIABLES
      COMMON/TEMD/TEMD
      COMMON/TEMC/TEMC
      COMMON/ROOD/ROOD
      COMMON/ROOC/ROOC
      COMMON/QUAL/QUAL
      COMMON/HOLDUP/HOLDUP
      COMMON/RADIUS/RADIUS
      COMMON/VELD/VELD
      COMMON/VELC/VELC
      COMMON/PRES/PRES
      COMMON/XQUAL/XQUAL
      COMMON/IQUALD/IQUALD
      COMMON/CPC/CPC
      COMMON/HFG/HFG
      COMMON/NCOEF/NCOEF
      COMMON/DD0/DD0
      COMMON/HDROP/HDROP
      COMMON/QUALCH/QUALCH
      COMMON/IFLAG/IFLAG
      COMMON/HDROP0/HDROP0
      COMMON/CPDL/CPDL
      COMMON/CPDV/CPDV
      COMMON/ITOT/ITOT
      COMMON/TOTHOL/TOTHOL
      COMMON/THOLD/THOLD
      COMMON/TDP/TDP
```

```
    COMMON/TD/TD
    COMMON/FILE/FILE
C   END OF COMMONS
    PI=4.*ATAN(1.)
C
 10 CONTINUE
C   ________EQUATIONS OF STATES AND DERIVATIVES ______
    PW=YN(1)*1.D-03
    HW=YN(2)*1.D-03
C   PW KPa        YN(3) KPa
C   HW KJ/Kg      YN(2) KJ/Kg
    LFTYPE='H2O   '
    CALL  DERIVW(LFTYPE,PW,HW,QUALC,ENTRC,ENTHC,
     1        VOLC,UC,TC,PRC,SL,HL,VL,UL,SV,HV,VV,UV)
    ROC=1/VOLC
    PC=YN(1)
    HC=YN(2)
    PD=YN(1)*1.D-03
    HD=YN(3)*1.D-03
C   PD KPa        YN(3) Pa
C   HD KJ/Kg      YN(5) J/Kg
    LFTYPE=FTYPE
 110        CALL DERIVD(LFTYPE,PD,HD,XQUAL,QUALD,ENTRD,ENTHD,
     #        VOLD,UD,TD,PRD,SL,HL,VL,UL,SV,HV,VV,
     2  UV,CPD,HFG,HLIQ,RATM)
    IF(IQUALD.GE.1) THEN
            YN(3)=ENTHD*1.D+03
    ENDIF
    IQUALD=INT(QUALD+.001)
    IF(FLAG.EQ.2.) THEN
            GOTO 117
    ENDIF
    IF((IQUALD.EQ.0.).AND.(XQUAL.EQ.0.).AND.(FLAG.EQ.0.)) THEN
     FLAG=1.
             GOTO 110
        ENDIF
 117        CONTINUE
    RODL=1./VL
    RODV=1./VV
    ROD=1/VOLD
    RRRTIO=RODL/RODV
    DELRO=ABS(ROC-ROD)
    SIG=ABS(SIG)
    PD=YN(1)
    HD=YN(3)
    CALL NEWTON(COSBET,XQUAL,RRRTIO)
    AREARL=(1+COS(BETA))/2.
    AREARV=(1.-COS(BETA))/2.
    BETA=ACOS(COSBET)
    BETAR=BETA*180./PI
    PRC=YN(1)
```

```
C
      CALL PROPTY(TD,TC,TAIR,PRC)
C     _______THERMOPHYSICAL PROPERTIES__________
      IF(XQUAL.LT.1) THEN
              CKD=(1.-XQUAL)*CKLD+XQUAL*CKVD
      ELSE
              CKD=CKVD
      ENDIF
C     CKD W./m deg C
C     VOLUME RATIO OF THE BUBBLE
      IF(XQUAL.LE.0.) THEN
          VOLRAT=0.
      ELSEIF(XQUAL.GE.1) THEN
          VOLRAT=1.
      ELSE
          VOLRAT=(XQUAL/RODV)/(XQUAL/RODV+(1.-XQUAL)/RODL)
      ENDIF
      VISD=(1.-VOLRAT)*VISLD+VOLRAT*VISVD
C     VISD N.s/m**2
C     Avarage viscosity of inside of the column N.s/m**2
      TEMPEF=(1-YN(6))*TC+YN(6)*TD
c     TEMPEF Deg c
c     Avarage temperature of inside of the column
      VEF=ABS(YN(4)-YN(6))
c     Avarage velocity of inside of the column m/sec
      ROEF=(1-YN(6))*ROC+YN(6)*ROD
c     Avarage Density of inside of the column Kg/m**3
      DE=2.*YN(7)
c     DE  m
      DELRO=ROC-ROD
      DELRO=ABS(DELRO)
      EOTZ=DELRO*9.8106*DE*DE/SIGW
      AMGRP=9.8106*VISC**4*DELRO/(ROC*ROC)/SIGW**3
      AKAPPA=VISC/VISD
      AKOFF=1./AKAPPA
c     Reynolds number (inside of the column)
      RE=ROC*DE*YN(4)/VISC
      RE=ABS(RE)
      CDCOR=(1.+2./AKAPPA/3.)/(1.+VISC/VISD)
      CD=(24./RE+6./(1.+DSQRT(RE))+.4)*(1.+2./3.*AKAPPA)/
     #        (1.+AKAPPA)
      CD2=24./RE+3.73/RE**.5-4.83D-03*RE**.5/(1.+3.D-06*RE**1.5)
     #       +.49
      CD2=CD2*CDCOR
      FD1=ROC*(YN(4)-YN(5))**2*PI*YN(7)*YN(7)*CD/2.
      FD2=ROC*(YN(4)-YN(5))**2*PI*YN(7)*YN(7)*CD2/2.
      FB=(ROC-ROD)*9.8106*4./3.*PI*YN(7)**3
      DELF1=FB-FD1
      DELF2=FB-FD2
      CDW=CD
      CDDD=4./3.*EOTZ**1.5/RE**2/AMGRP**.5
```

```
C   CD=1./(AKAPPA+1.)*(AKAPPA*(24./RECD+4./RECD**(1./3.))+
C    #      14.9/RECD**.78)
c   CD Drag Coefficient
    ABSK=3./8./YN(7)*YN(6)*CD*ROC*(YN(4)-YN(5))
    COLARE=PI*DINS*DINS/4.
        ROAVG=(ROD+ROC)/2.
        XIDMOL=.1998181
        XICMOL=.8001816
        IF(XQUAL.LE.0) THEN
            SIG=(ROAVG*(XIDMOL*SIGD**.25/ROD+XICMOL*SIGW**.25/ROC))**4
            SIG=ABS(SIG)
        ELSEIF((XQUAL.GT.0.).AND.(XQUAL.LT.1)) THEN
            SIG=(ROAVG*(XIDMOL*SIGD**.25/ROD+XICMOL*SIGW**.25/ROC))**4
            SIG=AREARL*SIG+AREARV*SIGW
        ELSE
            SIG=SIGW
        ENDIF
        DELRO=ABS(ROC-ROD)
        EOTVOS=DELRO*9.8106*(2.*YN(7))**2/SIG
        SD=SIGW-(SIGD+SIG)
            CDDD=2./3.*(ROC*9.8106*YN(7)*YN(7)/1.82/SIG)
        IF((XQUAL.NE.0.).AND.(QUAL.EQ.0.)) THEN
            DD0=YN(7)
        ENDIF
            IF(XN.LE.HN) THEN
            TIME=0.
        ENDIF
        IF(XQUAL.GT.0) THEN
        TIME=TIME+HN/YN(4)
        ENDIF
        IF((XQUAL.LE.0.)) THEN
        D0=YN(7)
        ELSEIF((DDO.NE.0.).AND.(XQUAL.GT.0.)) THEN
        D0=DD0
        ELSEIF(DD0.EQ.0.) THEN
        D0=YN(7)
        ENDIF
        DO HJGJG=1,10
C       IF(XQUAL.LE.0.) THEN
C               YN(4)=SQRT(4./3.*((ROC-ROD)/ROD)*YN(7)*9.8106/CD)
C       ELSE
                YYY=1.-1./TC*(.5*(D0/YN(7)+YN(7)/D0))
                DDD=(YN(7)*YN(7)+D0*D0)/2./YN(7)/D0
                TL=(TC+TD)/2.
                YP1=((TC*TC+TL*TL)/2./TC/TL)**DDD
                YP2=(CPC*VISC/CKC)**(D0/1.6/YN(7))
                YN(4)=SQRT(4./3.*(1.-ROD/ROC*(D0/YN(7))**3))
     #                      *SQRT(YN(7)*9.8106/CD)
                YYY=1.-1./TC*DDD
                YN(4)=YN(4)*DDD**YYY/YP1
C       YN4=YN(4)
```

```
C           YN(4)=(2.*YN(7))**2*ROC*9.8106/36./VISC*
C     #             ((1.-ROD/ROC*(DD0/YN(7))**3)*(1./(1.-DD0/YN(7))))
C       ENDIF
C       YN(4)=SQRT(4./3.*(1.-ROD/ROC))
C     #         *SQRT(YN(7)*9.8106/CD)
C       VER1=1.+XQUAL*(RATM-1.)
C       VER2=1.-(1./VER1)**(1./3.)
C       VER3=1./VER2
C       VER4=1.-(ROD/ROC)*(DD0/YN(7))**3
C       YN(4)=YN(7)*YN(7)*ROC*9.8106/36./VISC*VER4*VER3
        IF(XN.LE.HN) THEN
          VELINT=YN(4)
        ENDIF
C.          YN(6)=AMASSD/(COLARE*YN(4))/ROD
        IF(XQUAL.LE.0.0) THEN
                    YN(6)=AMASSD/(COLARE*YN(4))/ROD
            HURL=YN(6)
            HURV=0.
            HUR=YN(6)
        ELSEIF(XQUAL.GE.1.0) THEN
                    YN(6)=AMASSD/(COLARE*YN(4))/ROD
            HURL=0.
            HURV=YN(6)
            HUR=YN(6)
        ELSE
            HURL=AMASSD/(COLARE*YN(4))*(1.-XQUAL)/RODL
            HURV=AMASSD/(COLARE*YN(4))*XQUAL/RODV
            YN(6)=AMASSD/(COLARE*YN(4))*((1.-XQUAL)/RODL+XQUAL/RODV)
            HUR=YN(6)
        ENDIF
        IF(XN.LE.HN) THEN
            SINIT=YN(6)
        ENDIF
        YN(5)=AMASSC/COLARE/(ROC*(1.-YN(6)))
C       IF NB=CONST
        IF(XN.LE.HN) THEN
          RINIT=YN(7)
        ENDIF
        LCOAL=0.
C       IF(YN(6).LT.0.3) THEN
C           LCOAL=0.
C       ELSEIF((HOLDUP.LT.0.3).AND.(YN(6).GE.0.3)) THEN
C           LCOAL=XN
C       ENDIF
        IF(LCOAL.EQ.0) THEN
           YN(7)=RINIT*(YN(6)/SINIT)**(1./3.)
        ELSE
           YN(7)=RINIT*(YN(6)/SINIT*EXP(XN/(2.*LCOAL))
     #         )**(1./3.)
        ENDIF
    ENDDO
```

```
      TOTAL=0.
      NSIV=20
      DO 113 K=-NSIV,NSIV
  113          TOTAL=TOTAL+(COS(K*PI/2./NSIV))**2/(2.*NSIV+1.)
      XN3=HN
      IF(XN.LT.XN3) THEN
          DELVD=0.
          DELVC=0.
      ELSE
          DELVD=(YN(4)-VELD)/HN
          DELVC=(YN(5)-VELC)/HN
      ENDIF
          TEMD=TD
          TEMC=TC
          ROOD=ROD
          ROOC=ROC
          QUAL=XQUAL
          HOLDUP=YN(6)
          RADIUS=YN(7)
          ENTHC=YN(2)
          ENTHD=YN(3)
          VELD=YN(4)
          VELC=YN(5)
          PRES=YN(1)
C     ________HEAT TRANSFER COEFFICIENTS_____________
C
C     HEAT TRANSFER COEFFICIENT FROM THE BUBBLE
C     SOURCE : DIRECT CONTACT HEAT TRANSFER WITH CHANGE OF PHASE :
C     BUBBLE GROWTH IN THREE PHASE SYSTEMS
C      S. SIDEMAN and J. ISENBERG
C     Desalination, 2 (1967) 207-214
      RATIO=YN(7)/RINIT
c     RATIO density ratio of liquid/gas for dispersed phase
C
C
c     CP Specific Heat at constant pressure D:Dispersed C:continious

      ALPHAD=CKD/(ROD*CPD)
      PRD=VISD/ALPHAD/ROD
      IF(XQUAL.LE.0.) THEN
          ALPHDL=ALPHAD
          PRDL=PRD
      ELSEIF(XQUAL.GE.1.) THEN
          ALPHDV=ALPHAD
          PRDV=PRD
      ELSE
          ALPHDL=CKLD/(RODL*CPDL)
          ALPHDV=CKVD/(RODV*CPDV)
          PRDL=VISLD/ALPHDL/RODL
          PRDV=VISVD/ALPHDV/RODV
      ENDIF
```

```
      ALPHAC=CKC/(ROC*CPC)
C     ALPHA Thermal Diffusivity
      DE=2.*YN(7)
      VE=YN(4)-YN(5)
      RE=ROC*DE*VE/VISC
      REDL=RODL*DE*VE/VISLD
      REDV=RODV*DE*VE/VISVD
      RED=ROD*DE*VE/VISD
      RE=ABS(RE)
      PEDL=PRDL*REDL
      PEDLM=PEDL/(1.+VISLD/VISC)
      PRC=VISC/ALPHAC/ROC
      PED=RE*PRD
c     PR Prandtl number
      PREF=(1.-YN(6))*PRC+YN(6)*PRD
C     PREF Prandtl number inside of the column (mix fluid)
      UINF=ABS(YN(4)-YN(5))
      AREARL=0.46619*(PI-BETA+SIN(2.*BETA)/2.)**(2./3.)
      AREARV=1.-AREARL
        IF(XQUAL.GT.1) THEN
        AREARL=0.
        ENDIF
      HDRIV=5.*CKVD/YN(7)
      HDROP2=HDRIV
      HDROP3=5.*CKLD/YN(7)
      IF(NCOEF.EQ.1) THEN
        HDRCL=CKC/(2.*YN(7))*(ABS(REDL*PRDL))**(1./3.)
        HDROP1=HDRCL
C       HDROP2=1./(1./HDROP2+1./HDROP1)
      ELSEIF(NCOEF.EQ.4) THEN
        IF(XQUAL.GT.1) THEN
        AREARL=0.
        ENDIF
        HDRCL=SQRT((3.*COSBET-COSBET**3+2.)/PI)*(.25/PRC**(1./3.)*
       #        (RE*PRC))**.5*CKC/(2.*YN(7))
        HDROP1=HDRCL
        HDROP=(1.-VOLRAT)*HDROP1+VOLRAT*HDROP2
C       HDROP2=1./(1./HDROP2+1./HDROP1)
      ELSEIF(NCOEF.EQ.5) THEN
        BO=(ROC-ROD)*(2.*YN(7))**2*9.8106/SIGD
        HDRCL=(238.75196/PRC**.75*(VISC/(VISC+VISD))**4.3*
       #        BO**.33)*(CKD/(2.*YN(7)))
        HDROP1=HDRCL
        HDROP=(1.-VOLRAT)*HDROP1+VOLRAT*HDROP2
C       HDROP2=1./(1./HDROP2+1./HDROP1)
      ELSEIF(NCOEF.EQ.6) THEN
        PECLET=PED/(1.+VISD/VISC)
        HDROP1=(CKD/(2.*YN(7)))*0.0505*PECLET**.417*(ROC/ROD)**1.25
        HDROP=(1.-VOLRAT)*HDROP1+VOLRAT*HDROP2
C       HDROP2=1./(1./HDROP2+1./HDROP1)
      ELSEIF(NCOEF.EQ.7) THEN
```

```
      PECLET=PED/(1.+VISD/VISC)
      HDROP1=(CKD/(2.*YN(7)))*0.5*PECLET**.445*((ROC-ROD)/ROC)**1.81
      HDROP=(1.-VOLRAT)*HDROP1+VOLRAT*HDROP2
C     HDROP2=1./(1./HDROP2+1./HDROP1)
    ELSEIF(NCOEF.EQ.8) THEN
            BCOF=HDROP0*(TC-TD)/VELINT/(RINIT*HFG)
     #              *(RODL-RODV)/(RODL*RODV)
            COFM=2.-AREARL
            HVOL=2.*COFM*YN(6)*HDROP0/(2.*RINIT)*
     #              ((1.+COFM*BCOF*XN)**(3./COFM)-1.)/
     &          (COFM*BCOF*XN)
                QOVV=HVOL*(TC-TD)
            HDROP1=QOVV/(TC-TD)*YN(7)/3./YN(6)
            HDROP=(1.-VOLRAT)*HDROP1+VOLRAT*HDROP2
C     HDROP2=1./(1./HDROP2+1./HDROP1)
    ELSEIF(NCOEF.EQ.11) THEN
      HDRCL=SQRT((3.*COSBET-COSBET**3+2.)/PI)*(
     #        (RE*PRC))**.5*CKC/(2.*YN(7))
      HDROP1=HDRCL
      HDROP=(1.-VOLRAT)*HDROP1+VOLRAT*HDROP2
C     HDROP2=1./(1./HDROP2+1./HDROP1)
    ELSEIF(NCOEF.EQ.12) THEN
      HDROP1=(2.+0.6*RED**.5*PRD**(1./3.))*CKD/(2.*YN(7))
      HDROP=(1.-VOLRAT)*HDROP1+VOLRAT*HDROP2
C     HDROP2=1./(1./HDROP2+1./HDROP1)
    ELSEIF(NCOEF.EQ.13) THEN
      BO=(ROC-ROD)*(2.*YN(7))**2*9.8106/SIGD
      HDROP1=(238.75196/PRC**.75*(VISC/(VISC+VISD))**4.3*
     #        BO**.33)*(CKD/(2.*YN(7)))
      HDROP=(1.-VOLRAT)*HDROP1+VOLRAT*HDROP2
C     HDROP2=1./(1./HDROP2+1./HDROP1)
    ELSEIF(NCOEF.EQ.14) THEN
      IF(XQUAL.LE.0.) THEN
            CCC=CKLD/(2.*YN(7))
            HDRCL=CCC*(RE*PRC)**(1./3.)
            HDRDL=0.00375*PEDLM
            HDRCL=1./(1./HDRCL+1./HDRDL)
      ELSE
            HDRCL=CKC/(2.*YN(7))*(RE*PRC)**(1./3.)
      ENDIF
      HDROP1=HDRCL
      IF(XQUAL.LE.0.) THEN
        HDROP=HDROP1
      ELSEIF(XQUAL.GE.1.) THEN
        HDROP=HDROP2
      ELSE
        HDROP=AREARL*HDROP1+AREARV*HDROP2
      ENDIF
    ELSEIF(NCOEF.EQ.15) THEN
      HDRCL=CKC/(2.*YN(7))*(RE*PRC)**(1./3.)
      HDROP1=HDRCL
```

```
      HDROP=(1.-VOLRAT)*HDROP1+VOLRAT*HDROP2
C     HDROP2=1./(1./HDROP2+1./HDROP1)
      HDROP1=1./(1./HDROP1+1./HDROP3)
    ELSEIF(NCOEF.EQ.16) THEN
      HDRCL=ABS(CKC/(2.*YN(7))*(ABS(RE*PRC))**(1./3.))
      HDROP1=HDRCL
      HDROP=(1.-VOLRAT)*HDROP1+VOLRAT*HDROP2
C     HDROP2=1./(1./HDROP2+1./HDROP1)
    ELSEIF(NCOEF.EQ.17) THEN
      HDRCL=CKC/(2.*YN(7))*(.97+.68*RE**.5)*PRC**.3
      HDRIL=5.*CKLD/YN(7)
      HDROP1=HDRCL/6.
      HDROP=(1.-VOLRAT)*HDROP1+VOLRAT*HDROP2
    ELSEIF(NCOEF.EQ.18) THEN
            CCC=CKLD/(2.*YN(7))
            HDRCL=CCC*(RE*PRC)**(1./3.)
            HDRDL=0.00375*PEDLM
            HDROP1=1./(1./HDRCL+1./HDRDL)*0.38
            HDROP=HDROP1
    ENDIF
      HS1=SQRT((3.*COSBET-COSBET**3+2.)/PI)*(
     #        (RE*PRC))**.5*CKC/(2.*YN(7))
      HS2=SQRT((3.*COSBET-COSBET**3+2.)/PI)*(.25/PRC**(1./3.)*
     #        (RE*PRC))**.5*CKC/(2.*YN(7))
      HTOCH=CKC/(2.*YN(7))*(RE*PRC)**(1./3.)*AREARL
      BO=(ROC-ROD)*(2.*YN(7))**2*9.8106/SIGD
      HCAN=(238.75196/PRC**.75*(VISC/(VISC+VISD))**4.3*
     #        BO**.33)*(CKD/(2.*YN(7)))
C   HDROP=CKC/YN(7)*(.272*(RE*PRC)**.5)
C   QCONV=3.*YN(6)/YN(7)*HDROP*ABS(TC-TD)
C   HDROP Convective Heat Transfer Coefficient for the drop surface
    IF(NCOEF.NE.3) THEN
C     IF(XN.EQ.0.) THEN
C        QOVV=200000.
C     ENDIF
    QOVV=3.*YN(6)/YN(7)*HDROP*(TC-TD)
C        IF(XQUAL.LE.0.) THEN
C           QOVV=(3.*AREARL*YN(6)/YN(7)*HDROP1)*(TC-TD)
C        ELSEIF(XQUAL.GE.1.) THEN
C           QOVV=(3.*AREARV*YN(6)/YN(7)*HDROP2)*ABS(TC-TD)
C        ELSE
C           QOVV=3.*AREARL/(1.-VOLRAT)*HURL/YN(7)*HDROP1*(TC-TD)
C    #          +3.*AREARV/VOLRAT*HURV/YN(7)*HDROP2*(TC-TD)
C        ENDIF
    ENDIF
    HVOL=QOVV/(TC-TD)
C   FLODDING SEARCH.........
    VELRAT=ABS(YN(4)/YN(5))
    ABVC=ABS(YN(5))
    FLODD=1.+1.8*((ROD/ROC)*SQRT(VELRAT))
     #      -.4674693*(2.*YN(4))**0.25/(ABVC**2/9.8106*
```

```
      & (ROC-ROD)/ROC*VISC**0.25)**(.25)
     HVOL=QOVV/(TC-TD)
C
C    ------------DIFFERRENTIAL EQUATIONS SET----------
C
     S1=AMASSD/COLARE*(1./ROD/YN(4))
     S2=AMASSC/COLARE*(1./ROC/YN(5))
C
     G1=1./(ROD*YN(4)*YN(4))
     GS1=S1*G1
     G3=1./(ROC*YN(5)*YN(5))
     GS3=S2*G3
     G5=AMASSD/COLARE/(YN(4)*ROD*ROD)
     G6=AMASSC/COLARE/(YN(5)*ROC*ROC)
     G7=1./(YN(4)*ROD*ROD)
     G8=1./(YN(5)*ROC*ROC)
C
     XX(1)=-9.8106*(S1*ROD+S2*ROC)
C     #  -COLARE/AMASSC*DELVC-COLARE/AMASSD*DELVD
     YN1=YN(1)
     YN(1)=YN(1)+XX(1)*HN
     IF(XQUAL.LT.1.0) THEN
     CO=1.6
     ELSEIF(XQUAL.GE.1.0) THEN
     CO=1.
     ELSE
     CO=1.
     ENDIF
     XX(2)=-COLARE/AMASSC*QOVV*CO
     YN(2)=YN(2)+XX(2)*HN
     XX(3)=COLARE/AMASSD*(QOVV)
     DXX3=XX(3)*HN
     XX3=XX(3)
     IF(IQUALD.EQ.0) THEN
      XX3=XX(3)/HFG
      XQUAL=XQUAL+XX3*HN
          X1=PRES*1.D-03
      X2=XQUAL
      CYCLE='PX'
      LFTYPE=FTYPE
      CALL SIFLUID(CYCLE,LFTYPE,X1,X2,QL,ENTR,ENTH,VOL,U,
      1       TEMP,PRESS,SL,HL,VL,UL,SV,HV,VV,UV)
      YN3=ENTH*1.D+03
      XQUAL=QL
      IQUALD=INT(XQUAL+0.001)
      IF(IQUALD.GT.0) THEN
        TTD=TEMP
      ENDIF
        IF(IQUALD.EQ.0) THEN
         YN(3)=YN3
         FLAG=1.
```

```
            X1=YN(1)*1.D-03
        X2=YN(3)*1.D-03
        CYCLE='PH'
        LFTYPE=FTYPE
        CALL SIFLUID(CYCLE,LFTYPE,X1,X2,QL,ENTR,ENTH,VOL,U,
     1          TEMP,PRESS,SL,HL,VL,UL,SV,HV,VV,UV)
        YN(3)=ENTH*1.D+03
        XQUAL=QL
       ELSE
C       PRINT*,'**** SUPERHEATING STARTED ****'
            X1=YN1*1.D-03
        X2=1.
        CYCLE='PX'
        LFTYPE=FTYPE
        CALL SIFLUID(CYCLE,LFTYPE,X1,X2,QL,ENTR,ENTH,VOL,U,
     1          TEMP,PRESS,SL,HL,VL,UL,SV,HV,VV,UV)
        ENT1=ENTH*1.E+03
            TD=TEMP+.2
        X1=TD
        X2=YN(1)*1.D-03
        CYCLE='TP'
        LFTYPE=FTYPE
        CALL SIFLUID(CYCLE,LFTYPE,X1,X2,QL,ENTR,ENTH,VOL,U,
     1          TEMP,PRESS,SL,HL,VL,UL,SV,HV,VV,UV)
        ENT2=ENTH*1.E+03
        DELTHD=ENT2-ENT1
        IF(DELTHD.GT.0) THEN
        YN(3)=ENT2
        ELSE
        YN(3)=ENT1
        ENDIF
        IQUALD=1
        XQUAL=1.
        FLAG=2.
        YN(3)=YN3
           ENDIF
    ELSEIF(IQUALD.LT.0) THEN
      TD=TD+XX(3)*HN/CPD
          YN(3)=YN(3)+XX(3)*HN
      FLAG=0.
C     PRINT*, '****** PREHEATER ******'
    ELSE
C           PRINT*, '****** SUPERHEATER ******'
      FLAG=2
      TD=TD+XX(3)*HN/CPDV*.5
    ENDIF
 873        QUALCH=XQUAL
C   PRINT*,'DISTANCE =',XN
C   PRINT*,'XQUAL =',XQUAL
C   PRINT*,'TD =',TD
C   PRINT*,'TC=',TC
```

```
      TAVGV=(TD*YN(6)*YN(4)-TC*(1.-YN(6))*YN(5))/(YN(6)*YN(4)-
     #      (1.-YN(6))*YN(5))
      IF((XQUAL.LE.0.).OR.(XQUAL.GT.1.)) THEN
      TAVGM=(TD*YN(6)*ROD/72.15*YN(4)-TC*(1.-YN(6))*ROC/18.016*YN(5))/
     #  (YN(6)*ROD/72.15*YN(4)-(1-YN(6))*ROC/18.016*YN(5))
      ELSE
      TAVGM=(TD*(HURL*RODL+HURV*RODV)*YN(4)-TC*(1.-YN(6))*ROC*YN(5))/
     #  ((HURL*RODL+HURV*RODV)*YN(4)-(1-YN(6))*ROC*YN(5))
      ENDIF
      TAVGR=(TD*YN(6)*ROD*YN(4)-TC*(1.-YN(6))*ROC*YN(5))/
     #  (YN(6)*ROD*YN(4)-(1-YN(6))*ROC*YN(5))
      DTC=0.001
      DLC=.2
      DELTT=ABS(TAVGR-TD)
      IF(XQUAL.LE.0.) THEN
      REDTV=0.
      REDTL=RODL*YN(4)*DTC/VISLD
      PRDTL=VISLD*CPDL/CKLD
      PRDTV=0.
      REDT=REDTL
      PRDT=PRDTV
      ELSEIF(XQUAL.GE.1.) THEN
      REDTL=0.
      PRDTL=0.
      REDTV=RODV*YN(4)*DTC/VISVD
      PRDTV=VISVD*CPDV/CKVD
      REDT=REDTV
      PRDT=PRDTV
      ELSE
      REDTL=ROD*YN(4)*DTC/VISLD
      REDTV=ROD*YN(4)*DTC/VISVD
      PRDTL=VISLD*CPD/CKLD
      PRDTV=VISVD*CPD/CKVD
      REDT=REDTL*(1.-XQUAL)+REDTV*XQUAL
      PRDT=PRDTL*(1.-XQUAL)+PRDTV*XQUAL
      ENDIF
      RECT =ROC*ABS(YN(5))*DTC/VISC
      PRCT=VISC*CPC/CKC
c     A. Zhukauskas equation "Heat transfer from tubes in cross flow"
c          in J. P. Hartnett and T. F. Irvine Jr. Edts., Advances in Heat
c     Transfer., Vol 8 Academic Press, New York, 1972
      IF(REDTL.LE.40.) THEN
              CDTL=.75
              ANDTL=.4
      ELSEIF((REDTL.GT.40.).AND.(REDTL.LE.1000.)) THEN
              CDTL=.51
              ANDTL=0.5
      ELSEIF((REDTL.GT.1000.).AND.(REDTL.LE.2.E05)) THEN
              CDTL=0.26
              ANDTL=0.6
      ELSEIF(REDTL.GT.2E5) THEN
```

```
          CDTL=0.076
          ANDTL=0.7
    ENDIF
    IF(RECT.LE.40.) THEN
          CCT=.75
          ANCT=.4
    ELSEIF((RECT.GT.40.).AND.(RECT.LE.1000.)) THEN
          CCT=.51
          ANCT=0.5
    ELSEIF((RECT.GT.1000.).AND.(RECT.LE.2.E05)) THEN
          CCT=0.26
          ANCT=0.6
    ELSEIF(RECT.GT.2E5) THEN
          CCT=0.076
          ANCT=0.7
    ENDIF
    IF(REDTV.LE.40.) THEN
          CDTV=.75
          ANDTV=.4
    ELSEIF((REDTV.GT.40.).AND.(REDTV.LE.1000.)) THEN
          CDTV=.51
          ANDTV=0.5
    ELSEIF((REDTV.GT.1000.).AND.(REDTV.LE.2.E05)) THEN
          CDTV=0.26
          ANDTV=0.6
    ELSEIF(REDTV.GT.2E5) THEN
          CDTV=0.076
          ANDTV=0.7
    ENDIF
    IF(PRCT.LE.10) THEN
          CTM=.37
    ELSE
          CTM=.36
    ENDIF
    IF(PRDTL.LE.10) THEN
          DLTM=.37
    ELSE
          DLTM=.36
    ENDIF
    IF(PRDTV.LE.10) THEN
          DVTM=.37
    ELSE
          DVTM=.36
    ENDIF

    HCT=CKC/DTC*CCT*RECT**ANCT*PRCT**CTM
    HDTL=CKLD/DTC*CDTL*REDTL**ANDTL*PRDTL**DLTM
    HDTV=CKVD/DTC*CDTV*REDTV**ANDTV*PRDTV**DVTM
    HDT=(1.-VOLRAT)*HDTL+VOLRAT*HDTV
C   PRINT*,'HDT=',HDT
    TAVG1=(TD*YN(4)*(YN(6)*HDT)+TC*ABS(YN(5))*
```

```
     #  (1.-YN(6))*HCT)/(YN(4)*(YN(6)*HDT)+ABS(YN(5))*
     $  (1.-YN(6))*HCT)
      HDT=(1.-VOLRAT)*HDTL+VOLRAT*HDTV
      TAVG2=(TD*YN(4)*(YN(6)*HDT)+TC*ABS(YN(5))*
     #  (1.-YN(6))*HCT)/(YN(4)*(YN(6)*HDT)+ABS(YN(5))*
     $  (1.-YN(6))*HCT)
      HDT=(1.-VOLRAT)*HDTL+VOLRAT*HDTV
      TAVG4=(TD*YN(4)*(YN(6)*HDT)+TC*ABS(YN(5))*
     #  (1.-YN(6))*HCT)/(YN(4)*(YN(6)*HDT)+ABS(YN(5))*
     $  (1.-YN(6))*HCT)
C     Churchill and Bernstein equation, J of Heat transfer 99,300,1977
      HDTL=CKLD/DTC*(.3+(.62*REDTL**.5*PRDTL**(1./3.)/(1.+(.4*PRDTL)
     #   **(2./3.))**.25)*(1.+(REDTL/28200.)**(5./8.))**0.8)
      HDTV=CKVD/DTC*(.3+(.62*REDTV**.5*PRDTV**(1./3.)/(1.+(.4*PRDTV)
     #   **(2./3.))**.25)*(1.+(REDTV/28200.)**(5./8.))**0.8)
      HDT=(1.-VOLRAT)*HDTL+VOLRAT*HDTV
      HCT=CKC/DTC*(.3+(.62*RECT**.5*PRCT**(1./3.)/(1.+(.4*PRCT)**(2./3.)
     #  )**.25)*(1.+(RECT/28200.)**(5./8.))**0.8)
      TAVG3=(TD*YN(4)*(YN(6)*HDT)+TC*ABS(YN(5))*
     #  (1.-YN(6))*HCT)/(YN(4)*(YN(6)*HDT)+ABS(YN(5))*
     $  (1.-YN(6))*HCT)
C     S. Whitaker Equation
c     AIChE J., 22, 264, 1976
c
      HDTL=CKVD/DTC*(.4*REDTV**.5+.06*REDTV**(2./3.))*PRDTV**.4
      HDTL=CKC/DTC*(.4*RECT**.5+.06*RECT**(2./3.))*PRCT**.4
      HDT=(1.-VOLRAT)*HDTL+VOLRAT*HDTV*1.4
      TAVG4=(TD*YN(4)*(HUR*HDT)+TC*ABS(YN(5))*
     #  (1.-YN(6))*HCT)/(YN(4)*(HUR*HDT)+ABS(YN(5))*
     $  (1.-YN(6))*HCT)
              HC=CKC/DTC*((0.4*RECT**.5+.06*RECT**.67))*PRCT**.4
      IF(XQUAL.LE.0.) THEN
              HD=CKLD/DTC*((0.4*REDTL**.5+.06*REDTL**.67))*PRDTL**.4
      ELSEIF(XQUAL.GE.1.) THEN
              HD=CKVD/DTC*((0.4*REDTV**.5+.06*REDTV**.67))*PRDTV**.4
      ELSE
      DELTT=ABS(TD-TAVGR)
      ALAMDP=HFG*(1.+(.4*CPDV*DELTT/HFG))
              HD=2.7*SQRT((YN(4)-ABS(YN(5)))*CKVD*RODV*ALAMDP/(DTC*DELTT))
      ENDIF
      TAVG=(TD*YN(4)*YN(6)*HD+TC*ABS(YN(5))*(1.-YN(6))*HC)/
     #  (YN(4)*YN(6)*HD+ABS(YN(5))*(1.-YN(6))*HC)
      HDT=(1.-VOLRAT)*HDTL+VOLRAT*HDTV
      IF(XQUAL.LE.0.) THEN
      XAVG=0.
      ELSEIF(XQUAL.GE.1.) THEN
      XAVG=1.
      ELSE
      XAVG=XQUAL
      ENDIF
      IF(XQUAL.LE.1.) THEN
```

```
      TOTHOL=0.
      ITOT=0
      ELSE
      ITOT=ITOT+1
      TOTHOL=TOTHOL+YN(6)
      THOLD=TOTHOL/FLOAT(ITOT)
      ENDIF
      IF(NPRT.NE.0) THEN
      OPEN(UNIT=20,FILE=FILE,DISPOSE='SAVE'
     #   ,STATUS='UNKNOWN')
      N=1
543         READ(20,*,END=542) (ZZ(J,N),J=1,14)
      N=N+1
      GOTO 543
542         CONTINUE
      CLOSE(UNIT=20,STATUS='DELETE')
      OPEN(UNIT=20,FILE=FILE,DISPOSE='SAVE'
     #   ,STATUS='NEW')
      PRINT*,'ENTHD =',ENTHD,'HINITD =',HINITD,'YN(3)= ',YN(3)
      Q=AMASSD*(ENTHD-HINITD)
      ZZ(1,N)=XN
      ZZ(2,N)=AMASSC
      ZZ(3,N)=AMASSD
      ZZ(4,N)=TEMC
      ZZ(5,N)=TEMD
      ZZ(6,N)=TAVG1
      ZZ(7,N)=PRES
      ZZ(8,N)=Q
      ZZ(9,N)=QUAL
      ZZ(10,N)=HOLDUP
      ZZ(11,N)=RADIUS
      ZZ(12,N)=TAVG2
      ZZ(13,N)=TAVG3
      ZZ(14,N)=TAVG4
      DO I=1,N
      WRITE(20,'(1X,6(F6.2,1X),F10.1,1X,E11.5,1X,F5.3,1X,F7.4,1X,F7.4,
     #  1X,3(F5.1,1X))') (ZZ(J,I),J=1,14)
      WRITE(6,'(1X,6(F6.2,1X),F10.1,1X,E11.5,1X,F5.3,1X,F7.4,1X,F7.4,
     #  1X,3(F5.1,1X))') (ZZ(J,I),J=1,14)
      ENDDO
      CLOSE(UNIT=20,STATUS='KEEP')
      ENDIF
C     WRITE(18,*) XN,XAVG,YN(6),TEMC,TEMD,TAVG1,TAVG2,TAVG3,TAVGM,VELC,VELD
C     WRITE(17,*) '*****************************************************'
C     WRITE(17,*) '             X            =',XN
C     WRITE(17,*) '____________________________________________________'
C     WRITE(17,*) ' AVARAGE TEMPERATURE (MASS    )           =',TAVGM
C     WRITE(17,*) ' TEMPERATURE Deg C (DISPERSED PHASE )     =',TEMD
C     WRITE(17,*) ' TEMPERATURE         (CONTINIOUS PHASE)   =',TEMC
C     WRITE(17,*) ' ENTHALPY  J/KG C  (DISPERSED PHASE )     =',ENTHD
C     WRITE(17,*) ' ENTHALPY  J/KG C  (CONTINIOUS PHASE )    =',ENTHC
```

```
C   WRITE(17,*) '  PRESSURE    Pa                                =',PRES
C   WRITE(17,*) '  DENSITY   Kg/m**3 (DISPERSED PHASE )   =',ROOD
C   WRITE(17,*) '  DENSITY   Kg/m**3 (CONTINIOUS PHASE)   =',ROOC
C   WRITE(17,*) '  QUALTY OF DISPERSED PHASE              =',QUAL
C   WRITE(17,*) '  BETA OPENING ANGLE                     =',BETAR
C   WRITE(17,*) '  HOLDUP RATIO                           =',HOLDUP
C   WRITE(17,*) '  RADIUS OF DROP    (METERS)             =',RADIUS
C   WRITE(17,*) '  VELOCITY  m/s     (DISPERSED PHASE )   =',VELD
C   WRITE(17,*) '  VELOCITY  m/s     (CONTINIOUS PHASE)   =',VELC
C   WRITE(17,*) ' '
    RETURN
    END
    SUBROUTINE PROPTY(TD,TC,TAIR,PRC)
C   *********************************************************
C   *                                                       *
C   *          THERMOPHYSICAL PROPERTIES OF                 *
C   *         - WATER,,AIR,N-PENTANE,BUTANE,ISOBUTANE       *
C   *          PROPERTIES :                                 *
C   *          HYDROCARBON: LIQUID, VAPOR KINEMATIC VIS.    *
C   *                    LIQUID, VAPOR THERMAL CONDUCTIVITY *
C   *                    SURFACE TENSION                    *
C   *          WATER    : LIQUID KINEMATIC VISCOSITY        *
C   *                    LIQUID THERMAL CONDUCTIVITY        *
C   *                    SURFACE TENSION                    *
C   *          AIR      : KINEMATIC VISCOSITY               *
C   *                    THERMAL CONDUCTIVITY               *
C   *                                                       *
C   *       ________________________________________        *
C   *       REFERENCE : Tables on the Thermophysical        *
C   *       Properties of Liquids and Gases N. B. Vargaftic *
C   *       John Willey & Sons Inc.                         *
C   *                                                       *
C   *********************************************************
C   VARIABLES IDENTIFICATION
C   CKLD    :  CONDUCTIVE HEAT TRANSFER COEFFICIENT W/m deg C
C              (PENTANE LIQUID)
C   CKVD    :  CONDUCTIVE HEAT TRANSFER COEFFICIENT W/m deg C
C              (PENTANE VAPOR )
C   CK
    IMPLICIT DOUBLE PRECISION (A-H,O-Z)
    CHARACTER*6 FTYPE
    COMMON/FTYPE/FTYPE
    COMMON/CKLD/CKLD
    COMMON/CKVD/CKVD
    COMMON/CKC/CKC
    COMMON/CKAIR/CKAIR
    COMMON/VISLD/VISLD
    COMMON/VISVD/VISVD
    COMMON/VISC/VISC
    COMMON/VISAIR/VISAIR
    COMMON/SIGD/SIGD
```

```
      COMMON/SIGW/SIGW
      COMMON/ROAIR/ROAIR
      IF(PRC.LT.2.E06) THEN
      IF(TC.LT.100) THEN
C       P = 1 BAR
        A0 =    0.5689999177E+00
        A1 =    0.2237388258E-02
        A2 =   -0.4194219792E-04
        A3 =    0.8461035643E-06
        A4 =   -0.1064290936E-08
        A5 =   -0.2195261464E-09
        A6 =    0.2867647080E-11
        A7 =   -0.1108776851E-13
                CKC1=A0+A1*TC+A2*TC*TC+A3*TC**3+A4*TC**4+A5*TC**5
        1       +A6*TC**6+A7*TC**7
C           P=20 BARS
        A0 =    0.5000276018E+00
        A1 =    0.1842722906E-01
        A2 =   -0.1449807789E-02
        A3 =    0.6254021227E-04
        A4 =   -0.1490806953E-05
        A5 =    0.1984199371E-07
        A6 =   -0.1381372569E-09
        A7 =    0.3915733022E-12
        CKC2=A0+A1*TC+A2*TC*TC+A3*TC**3+A4*TC**4+A5*TC**5
        1       +A6*TC**6+A7*TC**7
        CKC=CKC1+(CKC2-CKC1)/(1.9E+06)*(PRC-.1E+06)
      ELSE
C          P=20 BARS
        A0 =    0.5034354585E+00
        A1 =    0.1080602021E-01
        A2 =   -0.4031109874E-03
        A3 =    0.8187132370E-05
        A4 =   -0.8981143005E-07
        A5 =    0.5378607955E-09
        A6 =   -0.1659310095E-11
        A7 =    0.2062792380E-14
        CKC2=A0+A1*TC+A2*TC*TC+A3*TC**3+A4*TC**4+A5*TC**5
        1       +A6*TC**6+A7*TC**7
        CKC=CKC2
      ENDIF
      ELSE
C           P=20 BARS
        A0 =    0.5034354585E+00
        A1 =    0.1080602021E-01
        A2 =   -0.4031109874E-03
        A3 =    0.8187132370E-05
        A4 =   -0.8981143005E-07
        A5 =    0.5378607955E-09
        A6 =   -0.1659310095E-11
        A7 =    0.2062792380E-14
```

```
      CKC1=A0+A1*TC+A2*TC*TC+A3*TC**3+A4*TC**4+A5*TC**5
      1       +A6*TC**6+A7*TC**7
C     P=40 BARS
      A0 =    0.5719129275E+00
      A1 =    0.1915405391E-02
      A2 =   -0.8020192384E-05
      A3 =   -0.2277327713E-07
      A4 =    0.4561862413E-09
      A5 =   -0.3196354991E-11
      A6 =    0.1089066733E-13
      A7 =   -0.1478755907E-16
      CKC2=A0+A1*TC+A2*TC*TC+A3*TC**3+A4*TC**4+A5*TC**5
      1       +A6*TC**6+A7*TC**7
      CKC=CKC1+(CKC2-CKC1)/(2.E+06)*(PRC-2.E+06)
     ENDIF
     CKAIR=.0236999993+7.00033131E-05*TAIR-2.8057219E-10*TAIR*TAIR
      #       +7.88027674E-12*TAIR**3-8.96694155E-14*TAIR**4+
      &       3.56213081E-16*TAIR**5
     IF(PRC.LT.2.E06) THEN
     IF(TC.LT.100) THEN
C     P = 1 BAR
      A0 =    0.1752498865E+05
      A1 =   -0.5623342898E+03
      A2 =    0.1293770770E+02
      A3 =   -0.2316151658E+00
      A4 =    0.3092855426E-02
      A5 =   -0.2800988617E-04
      A6 =    0.1494852986E-06
      A7 =   -0.3495565033E-09
      VISC1=A0+A1*TC+A2*TC*TC+A3*TC**3+A4*TC**4+A5*TC**5
      1        +A6*TC**6+A7*TC**7
C         P=20 BARS
      A0 =    0.1751398227E+05
      A1 =   -0.5621437186E+03
      A2 =    0.1301016255E+02
      A3 =   -0.2372883464E+00
      A4 =    0.3278885154E-02
      A5 =   -0.3108219012E-04
      A6 =    0.1745751680E-06
      A7 =   -0.4300887168E-09
      VISC2=A0+A1*TC+A2*TC*TC+A3*TC**3+A4*TC**4+A5*TC**5
      1       +A6*TC**6+A7*TC**7
      VISC=VISC1+(VISC2-VISC1)/(1.9E+06)*(PRC-.1E+06)
      VISC=VISC*1.E-07
     ELSE
C        P=20 BARS
      A0 =    0.4230490200E+04
      A1 =    0.1884581715E+03
      A2 =   -0.6676100270E+01
      A3 =    0.9206814826E-01
      A4 =   -0.6904918772E-03
```

```
       A5 =    0.2967543200E-05
       A6 =   -0.6887934270E-08
       A7 =    0.6712245273E-11
       VISC2=A0+A1*TC+A2*TC*TC+A3*TC**3+A4*TC**4+A5*TC**5
     1        +A6*TC**6+A7*TC**7
       VISC=VISC2*1.E-07
      ENDIF
      ELSE
      IF(TC.LT.100) THEN
C          P=20 BARS
       A0 =    0.1751398227E+05
       A1 =   -0.5621437186E+03
       A2 =    0.1301016255E+02
       A3 =   -0.2372883464E+00
       A4 =    0.3278885154E-02
       A5 =   -0.3108219012E-04
       A6 =    0.1745751680E-06
       A7 =   -0.4300887168E-09
       VISC1=A0+A1*TC+A2*TC*TC+A3*TC**3+A4*TC**4+A5*TC**5
     1        +A6*TC**6+A7*TC**7
C      P=40 BARS
       A0 =    0.1750199292E+05
       A1 =   -0.5612118942E+03
       A2 =    0.1295640067E+02
       A3 =   -0.2346663975E+00
       A4 =    0.3202648351E-02
       A5 =   -0.2986536004E-04
       A6 =    0.1646732071E-06
       A7 =   -0.3979925446E-09
       VISC2=A0+A1*TC+A2*TC*TC+A3*TC**3+A4*TC**4+A5*TC**5
     1        +A6*TC**6+A7*TC**7
       VISC=VISC1+(VISC2-VISC1)/(2.E+06)*(PRC-2.E+06)
       VISC=VISC*1.E-07
      ELSE
C      P=20 BARS
       A0 =    0.4230490200E+04
       A1 =    0.1884581715E+03
       A2 =   -0.6676100270E+01
       A3 =    0.9206814826E-01
       A4 =   -0.6904918772E-03
       A5 =    0.2967543200E-05
       A6 =   -0.6887934270E-08
       A7 =    0.6712245273E-11
       VISC1=A0+A1*TC+A2*TC*TC+A3*TC**3+A4*TC**4+A5*TC**5
     1        +A6*TC**6+A7*TC**7
C      P=40 BARS
       A0 =    0.9542057174E+04
       A1 =   -0.7263813086E+02
       A2 =   -0.1240393524E+01
       A3 =    0.2999460681E-01
       A4 =   -0.2705631795E-03
```

```
       A5 =    0.1284340755E-05
       A6 =   -0.3185213540E-08
       A7 =    0.3261966188E-11
       VISC2=A0+A1*TC+A2*TC*TC+A3*TC**3+A4*TC**4+A5*TC**5
     1        +A6*TC**6+A7*TC**7
       VISC=VISC1+(VISC2-VISC1)/(2.E+06)*(PRC-2.E+06)
       VISC=VISC*1.E-07
      ENDIF
      ENDIF
      VISAIR=17.4560012+.0215431014*TAIR+1.54701747E-03*TAIR*TAIR
     #      -4.17692993E-05*TAIR**3+4.6410164E-07*TAIR**4-1.80483506E-09*TAIR**5
      VISAIR=VISAIR*1.E-06
      ROAIR=1.25200012-5.44373253E-03*TAIR+7.91997385E-05*TAIR**2
     #  -1.67800022E-06*TAIR**3+1.77186738E-08*TAIR**4-6.7749145E
     $   -11*TAIR**5
      SIGW=75.5041118881198-8.468347319355099 3E-02*TC
     #  -3.187966200459344 3E-03*TC*TC+ 5.076282051261255 2E-05*TC**3
     $  -3.326923076897280 5E-07*TC**4+6.025641025528711 9E-10*TC**5
      SIGW=SIGW*1.D-03
C
      IF(FTYPE.EQ.'PENT  ') THEN
      CKVD=11.7000943+.0974954646*TD+1.2506797E-04*TD*TD-
     #       3.20993285E-10*TD**3
      CKVD=CKVD*1.E-03
      CKLD=.120500001-2.19166518E-04*TD-1.4583602E-06*TD*TD
     #       +4.1667695E-08*TD**3-4.16677821E-10*TD**4
      VISLD= 0.2830648416302071-4.703787707411947 5E-03*TD
     1       +3.945715737879038 1E-04*TD*TD-2.204082880904841 2E-05*TD*TD*TD
     2       +5.804206632965768 9E-07*TD**4-7.813455978274381 4E-09*TD**5
     3       +5.235702691586973 0E-11*TD**6-1.386554646319268 1E-13*TD**7
      VISLD=VISLD*1.E-03
      VISVD=49.3702925+.938695079*TD-.0151413279*TD*TD+
     #       1.38067486E-04*TD**3-4.01657705E-07*TD**4
      VISVD=VISVD*1.E-07
      SIGD=18.200088-.104142117*TD-1.05600806E-03*TD*TD
     #       +5.42164101E-05*TD**3-7.87026653E-07*TD**4
      SIGD=SIGD*1.D-03
      ELSEIF(FTYPE.EQ.'BUT   ') THEN
      CKVD=13.09999999999999+ 9.508333333341403 2E-02*TD+
     1       1.322916666659804 0E-03*TD*TD-3.697916666647379 1E-05*TD**3
     2       +4.427083333311390 6E-07*TD**4-1.822916666657954 0E-09*TD**5
      CKVD=CKVD*1.E-03
      CKLD= 0.1181211990095500 -4.972357499505042 2E-04*TD+
     1       1.355191265165391 6E-06*TD*TD-1.842615955340561 0E-08*TD**3
     2  +4.619392963326019 2E-10*TD**4+3.092514125794470 0E-12*TD**5
     3  -4.535467829200628 9E-14*TD**6-3.280959766719931 3E-16*TD**7
      VISLD=0.2077242494148349-1.777955047600862 0E-03*TD+
     1       1.717933227325748 8E-05*TD*TD-1.558329493543794 5E-07*TD**3
     2  -1.188377428451111 5E-10*TD**4+2.052088850663739 2E-11*TD**5
     3  +1.827032574239229 0E-15*TD**6-2.455097963686549 7E-15*TD**7
     4  +1.218472694674582 7E-17*TD**8
```

```
VISLD=VISLD*1.E-03
VISVD=68.19999999999985+0.2594409882007312*TD
 1      +5.671968009962751lE-04*TD*TD+2.038580285381017 2E-07*TD**3
 2  -6.098120365949611 0E-07*TD**4+ 1.104700212070080 0E-08*TD**5
 3  -5.480874240608572 6E-11
VISVD=VISVD*1.E-07
SIGD= 14.69887586454411-0.1213412813637348*TD
 1  +1.2537287655336214E-04*TD*TD+4.7908659759907817E-07*TD**3
 2  -1.7322815304702313E-08*TD**4+1.3739211170486623E-11*TD**5
 3  +1.1427589337992400E-12*TD**6-3.5658670411439808E-15*TD**7
SIGD=SIGD*1.E-03
ELSEIF(FTYPE.EQ.'ISOP  ') THEN
  VISLD= 0.2123060252441059-2.2961653171389990E-03*TD
 1     +1.7430732184911213E-05*TD*TD-2.0511381678177523E-07*TD**3
 2     -1.0966731496914415E-09*TD**4+6.5600936730970486E-12*TD**5
 3     +3.2980736154107037E-13*TD**6
  VISLD=VISLD*1.E-03
  VISVD=68.90192088367745 +0.2882120928681750*TD
 1     -6.6015871489698662E-04*TD*TD+5.1584268075363516E-06*TD**3
 2      -1.7540446681309466E-08*TD**4+2.1335967723032119E-11*TD**5
       VISVD=VISVD*1.E-07
  CKLD=0.1034320425645717-5.3617612354974583E-04*TD
 1     -3.4818547890720563E-06*TD*TD+4.0047672524736692E-07*TD**3
 2     +7.9117669178277704E-09*TD**4-4.5916087532261178E-10*TD**5
 3     +4.0224793055692234E-12*TD**6
  CKVD=14.02360414148952+0.1208272189021553*TD
 1     +1.0860013903942589E-04*TD*TD-1.9940388279179968E-05*TD**3
 2     +4.9048950395836884E-07*TD**4-4.9294337623315290E-09*TD**5
 3     +2.2305489177839495E-11*TD**6-3.7421511943571895E-14*TD**7
  CKVD=CKVD*1.E-03
  SIGD=13.10249170232413-0.1195441595693908*TD
 1     -4.6563796290468780E-04*TD*TD-8.6947909578871229E-06*TD**3
 2     +5.5756978432797757E-08*TD**4+1.9965171835269199E-09*TD**5
 3     +1.0499489901682764E-11*TD**6
  SIGD=SIGD*1.E-03
ELSEIF(FTYPE.EQ.'ISOB  ') THEN
IF(TD.LT.120.) THEN
 A0 =     0.2688582269E+05
 A1 =    -0.5578866926E+03
 A2 =     0.6186873157E+01
 A3 =    -0.2669591904E-01
 VISLD=A0+A1*TD+A2*TD*TD+A3*TD*TD*TD
ELSE
 A0 =     0.2123381538E+05
 A1 =    -0.2574826211E+03
 A2 =     0.1159664588E+01
 VISLD=A0+A1*TD+A2*TD*TD
ENDIF
 VISLD=VISLD*1.E-08
IF(TD.LT.100) THEN
 A0 =     0.3151068247E+03
```

```
       A1 =    0.1393842979E+02
       A2 =   -0.3417497132E-01
       VISVD=A0+A1*TD+A2*TD*TD
      ELSE
       VISVD=1370.
      ENDIF
       VISVD=VISVD*1.E-08
       A0 =    0.1300000000E+02
       A1 =   -0.1130000000E+00
       A2 =    0.9999999998E-04
       A3 =   -0.1396763564E-15
       SIGD=A0+A1*TD+A2*TD*TD+A3*TD*TD*TD
       SIGD=SIGD*1.E-03
C      PROBLEM :CKISV.DAT
       A0 =    0.1381380000E+02
       A1 =    0.1088360000E+00
       A2 =    0.3069733333E-03
       A3 =   -0.3348800000E-05
       A4 =    0.1116266667E-07
       CKVD=A0+A1*TD+A2*TD*TD+A3*TD*TD*TD
       CKVD=CKVD*1.E-03
C      PROBLEM :CKISL
       A0 =    0.1054573000E+02
       A1 =   -0.5897775000E-01
       A2 =    0.2765750000E-03
       CKLD=A0+A1*TD+A2*TD*TD
       CKLD=CKLD*1.E-02
      ENDIF
      RETURN
      END
        SUBROUTINE SIFLUID(CYCLE,LFTYPE,FGIVEN,SGIVEN
       1 ,QL,ENTR,ENTH,VOL,U,TEMP,PRESS,SL,HL,VL,UL,SV,HV,VV,UV)
              IMPLICIT DOUBLE PRECISION (A-H,O-Z)
        CHARACTER*6 LFTYPE
        CHARACTER*6 IFREON(9),ICARBN(12)
        CHARACTER*2 CYCLE
        COMMON/ERFLAG/ERFLAG
        DATA(IFREON(I),I=1,9)/'F11   ','F12   ','F13   ','F14   ',
       1    'F21   ','F22   ','F23   ','F113  ','F114  '/
        DATA(ICARBN(I),I=1,12)/'METH  ','ETH   ','PROP  ','BUT   ',
       1    'PENT  ','HEX   ','HEPT  ','OCT   ','ETHYL ','PROPYL',
       2    'ISOB  ','ISOP  '/
        PARAMETER M2=6
        DO 100 I=1,9
        IF(LFTYPE.EQ.IFREON(I)) GO TO 400
    100 CONTINUE
        DO 200 I=1,12
        IF(LFTYPE.EQ.ICARBN(I)) GO TO 500
    200 CONTINUE
        IF(LFTYPE.EQ.'H2O   ') GO TO 300
        WRITE(M2,9000) LFTYPE
```

```
      RETURN
  300 LN=10
      NS=0
      NP=0
      CALL SIWATER(CYCLE,FGIVEN,SGIVEN,LN,NS,NP,QL,ENTR,ENTH,
     1    VOL,U,TEMP,PRESS,SL,HL,VL,UL,SV,HV,VV,UV,NERROR)
      RETURN
  400 CALL SIFREON(CYCLE,FGIVEN,SGIVEN,LFTYPE,ENTH,
     1  ENTR,VOL,TEMP,PRESS,HV,SV,VV,HL,SL,VL,QL,NPRINT)
      U=0.0
      UL=0.0
      UV=0.0
      RETURN
 500  CONTINUE
      CALL SICARBON(CYCLE,FGIVEN,SGIVEN,LFTYPE,ENTH,ENTR,VOL,TEMP,
     1  PRESS,NP,SL,HL,VL,SV,HV,VV,QL)
      U=0.0
      UL=0.0
      UV=0.0
      RETURN
 9000 FORMAT(1H,4X,'***ERROR IN FLUID*** THE FLUID',A6,'CANNOT BE EVALUA
     1TED')
! Univac Statement      DEBUG SUBCHK
      END
      SUBROUTINE WATER(CYCLE,FGIVEN,SGIVEN,CALLIN,NSTAT1,NPRINA,QUAL1,
     1    ENTR1,ENTH1,VOL1,UINT1,TEMPF1,PRESS1,LSATS1,LSATH1,LSATV1,
     2    LSATU1,VSATS1,VSATH1,VSATV1,VSATU1,NERROR)
      IMPLICIT REAL(L)
      INTEGER CALLIN
      CHARACTER*2 CYCLE
      PARAMETER M2=6,M1=5
      COMMON/WATER1/VSATS,VSATH,VSATU,VSATV,LSATS,LSATH,LSATU,LSATV,
     1    TEMPK, TCRIT, PCRIT, TAUCRT, TGUESS, GAS, GAS1, CONVR,
     2    DPSDT, DGUESS, DENSIT, DPDRHO, TEMPF, PRESS, UINT, ENTH, VOL,
     3    ENTR, DHDT, DHDRHO, DPDT, SCAL, HCAL,DSDRHO, SJOULE, SKNOWN,
     4    ITERAT, PSAT, DENTOL, TTOLER, BTOLER, PCAL, QUAL, PKNOWN, DSDT
      COMMON/PRINTA/NPRINT,NSTATE
      DATA  TTOLER,BTOLER,CONVR/1.005,0.995,0.4299229/
      DATA TCRIT,PCRIT,TAUCRT,GAS1,GAS/374.136,3203.6,1.544912,66.9366,
     1    0.46151/
      DATA ITERAT/300/
      NSTATE=NSTAT1
      NPRINT=NPRINA
      TEMPF=TEMPF1
      ENTR=ENTR1
      ENTH=ENTH1
      PRESS=PRESS1
      LSATH=0.0
      LSATU=0.0
      LSATV=0.0
      LSATS=0.0
```

```
      VSATH=0.0
      VSATU=0.0
      VSATV=0.0
      VSATS=0.0
      NERROR=0.0
      IF(NPRINT.EQ.1) WRITE(M2,9000)
      IF (CYCLE.EQ.'TP') GO TO 1930
      IF (CYCLE.EQ.'TX'.OR.CYCLE.EQ.'PX') GO TO 1720
      IF (CYCLE.EQ.'PS') GO TO 1100
      IF (CYCLE.EQ.'PH') GO TO 1330
      WRITE (M2,9010) CYCLE, CALLIN
      NERROR=1
      RETURN
C------------------PRESSURE-ENTROPY--------------------------------
 1100 SKNOWN=SGIVEN*1.8
      PKNOWN=FGIVEN
      PRESS   =FGIVEN
      ENTR=SGIVEN
      IF(NPRINT.EQ.1) WRITE(M2,9050)
      IF(PRESS.LT.PCRIT) GO TO 1200
      TEMPK=500.0
      IF(ENTR.GT.1.05800) DENSIT=1.0E-05
      IF(ENTR.GT.1.05800) GO TO 1550
      DENSIT=0.96
      GO TO 1630
 1200 CALL TSATUR
      CALL VAPDEN(NSTAT1)
      IF(ENTR/SCAL.GT.TTOLER) GO TO 1550
      CALL LIQDEN(NSTAT1)
      IF(ENTR/SCAL.LT.BTOLER) GO TO 1630
      QUAL=(ENTR-LSATS)/(VSATS-LSATS)
      ENTH=(1.0-QUAL)*LSATH + QUAL*VSATH
      UINT=(1.0-QUAL)*LSATU + QUAL*VSATU
      VOL =(1.0-QUAL)*LSATV + QUAL*VSATV
      CALL PRINT
      GO TO 2200
C------------------PRESSURE-ENTHALPY-------------------------------
 1330 PKNOWN=FGIVEN
      HKNOWN=SGIVEN
      PRESS  =FGIVEN
      ENTH=SGIVEN
      IF(NPRINT.EQ.1) WRITE(M2,9060)
      IF(PRESS.LT.PCRIT) GO TO 1400
      TEMPK=500.0
      IF(ENTH.GT.902.5) DENSIT=1.0E-05
      IF(ENTH.GT.902.5) GO TO 1550
      DENSIT=0.96
      GO TO 1630
 1400 CALL TSATUR
      CALL VAPDEN(NSTAT1)
      IF(HKNOWN/HCAL.GT.TTOLER) GO TO 1550
```

```
      CALL LIQDEN(NSTAT1)
      IF(HKNOWN/HCAL.LT.BTOLER) GO TO 1630
      QUAL=(HKNOWN-LSATH)/(VSATH-LSATH)
      ENTR=(1.0-QUAL)*LSATS + QUAL*VSATS
      UINT=(1.0-QUAL)*LSATU + QUAL*VSATU
      VOL =(1.0-QUAL)*LSATV + QUAL*VSATV
      CALL PRINT
      GO TO 2200
 1550 DENTOL=1.0E-10
      IF (CYCLE.EQ.'PH') CALL PHFIND
      IF (CYCLE.EQ.'PS') CALL PSFIND
      ENTH=HCAL
      ENTR=SCAL
      IF(PRESS.GT.PCRIT) QUAL=-3.0
      IF(PRESS.LE.PCRIT) QUAL=+2.0
      CALL PRINT
      GO TO 2200
 1630 DENTOL=1.0E-05
      IF (CYCLE.EQ.'PH') GO TO 1680
      TEMPK=180.0*(5.4+9.630628*ENTR-2.640989*SQRT(ENTR)-0.10592*ENTR**2
     1)
      DENSIT=10.0/(62.4283*(0.12    -.5276609*ENTR+.2878426*SQRT(ENTR)
     1    +.4208526*ENTR**2))
      CALL PSFIND
      GO TO 1700
 1680 TEMPK=1.8*(512.0+1.370531*ENTH+4.737845*SQRT(ENTH)+4.540232E-04*
     1    ENTH**2)
      DENSIT=1.0E+04/(62.4283*(149.2-.185626*ENTH+2.671239*SQRT(ENTH)
     1    +.3508786E-03*ENTH**2))
      CALL PHFIND
 1700 ENTH=HCAL
      ENTR=SCAL
      IF(PRESS.GT.PCRIT) QUAL=-3.0
      IF(PRESS.LE.PCRIT) QUAL=-2.0
      CALL PRINT
      GO TO 2200
 1720 QUAL=SGIVEN
      IF (QUAL.GE.0.0) GO TO 1780
      WRITE (M2,9020)
      QUAL=0.0
 1780 IF (QUAL.LE.1.0) GO TO 1810
      WRITE (M2,9030)
      QUAL=1.0
C-----------------PRESSURE-QUALITY---------------------------------
 1810 IF (CYCLE.EQ.'TX') GO TO 1870
      PRESS  =FGIVEN
      PKNOWN=FGIVEN
      IF(NPRINT.EQ.1) WRITE(M2,9070)
      IF(PRESS.LE.PCRIT*0.99) GO TO 1890
      NERROR=1
      WRITE(M2,9100)
```

```
      RETURN
 1890 CALL TSATUR
      GO TO 2120
C------------------TEMPERATURE-QUALITY   ----------------------------
 1870 TEMPF=FGIVEN
      IF(NPRINT.EQ.1) WRITE(M2,9080)
      IF(TEMPF.LE.705.0) GO TO 1900
      NERROR=1
      WRITE(M2,9110)
      RETURN
 1900 TEMPK=(TEMPF+459.67)/1.8
      CALL PSATUR
      PRESS  =PSAT
      GO TO 2120
C------------------PRESSURE-TEMPERATURE------------------------------
 1930 TEMPF=FGIVEN
      PRESS  =SGIVEN
      IF(NPRINT.EQ.1) WRITE(M2,9090)
      TEMPK=(TEMPF+459.67)/1.8
      IF(TEMPF.LT.705.0) GO TO 1960
      IF(PRESS.GT.PCRIT) GO TO 2080
      GO TO 2035
 1960 CALL PSATUR
      IF(PRESS  /PSAT.GT.TTOLER.OR.PRESS  /PSAT.LT.BTOLER) GO TO 2030
      WRITE(M2,9040) CALLIN,TEMPF,PRESS
      NERROR=1
      RETURN
 2030 IF(PRESS  /PSAT.GT.TTOLER) GO TO 2080
 2035 QUAL=2.0
      DGUESS=1.0E-05
      DENTOL=1.0E-10
      GO TO 2090
 2080 DGUESS=0.99
      DENTOL=1.0E-07
      IF(TEMPF.LT.705.) QUAL=-2.0
      IF(TEMPF.GE.705.) QUAL=-3.0
 2090 CALL DFIND(NSTAT1)
      ENTH=HCAL
      ENTR=SCAL
      CALL PRINT
      GO TO 2200
 2120 IF(ABS(QUAL).LT.0.000001) GO TO 2150
      CALL VAPDEN(NSTAT1)
      IF(ABS(QUAL-1.0).LT.0.000001) GO TO 2160
 2150 CALL LIQDEN(NSTAT1)
 2160 ENTR=(1.0-QUAL)*LSATS + QUAL*VSATS
      UINT=(1.0-QUAL)*LSATU + QUAL*VSATU
      ENTH=(1.0-QUAL)*LSATH + QUAL*VSATH
      VOL =(1.0-QUAL)*LSATV + QUAL*VSATV
      CALL PRINT
 2200 TEMPF1=TEMPF
```

```
      VOL1=VOL
      ENTR1=ENTR
      ENTH1=ENTH
      UINT1=UINT
      PRESS1=PRESS
      QUAL1=QUAL
      LSATS1=LSATS
      LSATH1=LSATH
      LSATU1=LSATU
      LSATV1=LSATV
      VSATS1=VSATS
      VSATH1=VSATH
      VSATU1=VSATU
      VSATV1=VSATV
 9000 FORMAT(1H1,//,13X,'********    THERMODYNAMIC PROPERTIES OF WATER
     1  ********',////)
 9010 FORMAT(1H ,4X,'***ERROR*** CYCLE = ',A3,' CYCLE MUST EQUAL TP, TX,
     1 PX, PS, PH, LAST CALL ON WATER CAME FROM LINE',I6,//,10X,'WATER T
     2ERMINATED')
 9020 FORMAT(1H ,4X,'***ERROR*** A QUALITY OF < 0 IS NOT DEFINED, ASSUME
     1D TO BE ZERO')
 9030 FORMAT(1H ,4X,'***ERROR*** A QUALITY OF > 1 IS NOT DEFINED, ASSUME
     1D TO BE ONE')
 9040 FORMAT(1H ,4X,'***ERROR*** TEMPERATURE AND PRESSURE ARE NOT INDEPE
     1NDENT UNDER THE SATURATION DOME',//,4X,'WATER TERMINATED, LAST CAL
     2L CAME FROM LINE ',I6,/,10X,'TEMPERATURE = ',F15.5,10X,'PRESSURE =
     3 ',F15.5)
 9050 FORMAT(1H ,16X,'PRESSURE AND ENTROPY')
 9060 FORMAT(1H ,16X,'PRESSURE AND ENTHALPY')
 9070 FORMAT(1H ,16X,'PRESSURE AND QUALITY')
 9080 FORMAT(1H ,16X,'TEMPERATURE AND QUALITY')
 9090 FORMAT(1H ,16X,'TEMPERATURE AND PRESSURE')
 9100 FORMAT(1H ,4X,'***ERROR*** A SATURATION STATE DOES NOT EXIST FOR P
     1RESSURES ABOVE THE CRITICAL PRESSURE(3203.6PSIA), WATER TERMINATED
     2   ')
 9110 FORMAT(1H ,4X,'***ERROR*** A SATURATION STATE DOES NOT EXIST FOR T
     1EMPERATURES ABOVE THE CRITICAL TEMPERATURE(705 DEG.F), WATER TERMI
     2NATED')
      RETURN
! Univac Statement      DEBUG SUBCHK
      END
C     $$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
C     $                                                                 $
C     $                        T S A T U R                              $
C     $                                                                 $
C     $    USES NEWTON-RAPHSON ITERATION TO LOCATE SATURATION TEMPERATURE$
C     $GIVEN SATURATION PRESSURE.                                       $
C     $                                                                 $
C     $$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
      SUBROUTINE TSATUR
      IMPLICIT REAL(L)
```

```
      PARAMETER M2=6
      COMMON/WATER1/VSATS,VSATH,VSATU,VSATV,LSATS,LSATH,LSATU,LSATV,
     1    TEMPK, TCRIT, PCRIT, TAUCRT, TGUESS, GAS, GAS1, CONVR,
     2    DPSDT, DGUESS, DENSIT, DPDRHO, TEMPF, PRESS, UINT, ENTH, VOL,
     3    ENTR, DHDT, DHDRHO, DPDT, SCAL, HCAL,DSDRHO, SJOULE, SKNOWN,
     4    ITERAT, PSAT, DENTOL, TTOLER, BTOLER, PCAL, QUAL, PKNOWN, DSDT
      TSAT=340.0
      TEMPK=TSAT
      CALL PSATUR
      DO 2320 I=1, ITERAT
 2280 TEMPK=TSAT-(PSAT-PKNOWN)/DPSDT
      IF(TEMPK-273.15.GT.TCRIT) TEMPK=TCRIT+273.15
      CALL PSATUR
      IF(PSAT/PKNOWN.GT.BTOLER.AND.PSAT/PKNOWN.LT.TTOLER) GO TO 2330
      TSAT=TEMPK
 2320 CONTINUE
      WRITE(M2,9000) ITERAT
 9000 FORMAT(1H ,'*** SATURATED TEMPERATURE SEARCH FAILED TO CONVERGE IN
     1',I5,' ITERATIONS, RESULTS MAY NOT BE VALID***')
 2330 TSAT=TEMPK
      RETURN
! Univac Statement     DEBUG SUBCHK
      END
C     $$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
C     $                                                                $
C     $                     V A P D E N                                $
C     $                                                                $
C     $    SETS UP SATURATED VAPOR DENSITY SEARCH                      $
C     $                                                                $
C     $$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
      SUBROUTINE VAPDEN(NSTAT1)
      IMPLICIT REAL(L)
      COMMON/WATER1/VSATS,VSATH,VSATU,VSATV,LSATS,LSATH,LSATU,LSATV,
     1    TEMPK, TCRIT, PCRIT, TAUCRT, TGUESS, GAS, GAS1, CONVR,
     2    DPSDT, DGUESS, DENSIT, DPDRHO, TEMPF, PRESS, UINT, ENTH, VOL,
     3    ENTR, DHDT, DHDRHO, DPDT, SCAL, HCAL,DSDRHO, SJOULE, SKNOWN,
     4    ITERAT, PSAT, DENTOL, TTOLER, BTOLER, PCAL, QUAL, PKNOWN, DSDT
      DGUESS=1.0E-06
      DENTOL=1.0E-12
      CALL DFIND(NSTAT1)
      VSATV=VOL
      VSATU=UINT
      VSATS=SCAL
      VSATH=HCAL
      RETURN
! Univac Statement     DEBUG SUBCHK
      END
C     $$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
C     $                                                                $
C     $                     L I Q D E N                                $
C     $                                                                $
```

```
C     $    SET UP SEARCH FOR SATURATED LIQUID DENSITY                        $
C     $                                                                      $
C     $$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
       SUBROUTINE LIQDEN(NSTAT1)
       IMPLICIT REAL (L)
       COMMON/WATER1/VSATS,VSATH,VSATU,VSATV,LSATS,LSATH,LSATU,LSATV,
      1    TEMPK, TCRIT, PCRIT, TAUCRT, TGUESS, GAS, GAS1, CONVR,
      2    DPSDT, DGUESS, DENSIT, DPDRHO, TEMPF, PRESS, UINT, ENTH, VOL,
      3    ENTR, DHDT, DHDRHO, DPDT, SCAL, HCAL,DSDRHO, SJOULE, SKNOWN,
      4    ITERAT, PSAT, DENTOL, TTOLER, BTOLER, PCAL, QUAL, PKNOWN, DSDT
       DGUESS=0.96
       DENTOL=1.0E-07
       CALL DFIND(NSTAT1)
       LSATV=VOL
       LSATU=UINT
       LSATS=SCAL
       LSATH=HCAL
       RETURN
! Univac Statement      DEBUG SUBCHK
       END
C     $$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
C     $                                                                      $
C     $                           P R I N T                                  $
C     $                                                                      $
C     $$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
       SUBROUTINE PRINT
       IMPLICIT REAL (L)
       COMMON/WATER1/VSATS,VSATH,VSATU,VSATV,LSATS,LSATH,LSATU,LSATV,
      1    TEMPK, TCRIT, PCRIT, TAUCRT, TGUESS, GAS, GAS1, CONVR,
      2    DPSDT, DGUESS, DENSIT, DPDRHO, TEMPF, PRESS, UINT, ENTH, VOL,
      3    ENTR, DHDT, DHDRHO, DPDT, SCAL, HCAL,DSDRHO, SJOULE, SKNOWN,
      4    ITERAT, PSAT, DENTOL, TTOLER, BTOLER, PCAL, QUAL, PKNOWN, DSDT
       COMMON/PRINTA/NPRINT,NSTATE
       PARAMETER M2=6
       TEMPF=TEMPK*1.8-459.67
       IF(NPRINT.EQ.0) RETURN
       WRITE(M2,9040) NSTATE
       WRITE(M2,9050) TEMPF,PRESS
       IF (QUAL.LE.1.0.AND.QUAL.GE.0.0) GO TO 3280
       IF (QUAL.GT.1.0) WRITE(M2,9060)
       IF(QUAL.LT.0.0.AND.QUAL.GT.-2.5) WRITE(M2,9070)
       IF(QUAL.LT.-2.5) WRITE(M2,9030)
       GO TO 3370
  3280 WRITE(M2,9080) QUAL
       WRITE(M2,9130)
       WRITE(M2,9090) VSATU,LSATU
       WRITE(M2,9100) VSATH,LSATH
       WRITE(M2,9110) VSATS,LSATS
       WRITE(M2,9120) VSATV,LSATV
  3370 WRITE(M2,9140)
       WRITE(M2,9090) UINT
```

```
      WRITE(M2,9100) ENTH
      WRITE(M2,9110) ENTR
      WRITE(M2,9120) VOL
 9030 FORMAT(1H ,16X,'CRITICAL SOLUTION',//)
 9040 FORMAT(1H ,16X,'PROPERTIES OF WATER AT STATE ',I4,/)
 9050 FORMAT(1H ,16X,'TEMPERATURE = ',F7.2,' DEG.F',//,
     1    17X,'PRESSURE    = ',F10.5,' PSIA')
 9060 FORMAT(1H ,16X,'SUPERHEATED VAPOR')
 9070 FORMAT(1H ,16X,'COMPRESSED LIQUID')
 9080 FORMAT(1H ,16X,'SATURATED LIQUID-VAPOR EQUILIBRUIM',/,17X,
     1    'QUALITY = ',F6.4)
 9090 FORMAT(1H ,16X,'INTERNAL ENERGY (BTU/LBM)',5X,F9.3,8X,F9.3)
 9100 FORMAT(1H ,16X,'ENTHALPY',8X,'(BTU/LBM)',5X,F9.3,8X,F9.3)
 9120 FORMAT(1H ,16X,'SPECIFIC VOLUME (CU.FT/LBM )',2X,F9.4,8X,F9.4)
 9110 FORMAT(1H ,16X,'ENTROPY',9X,'(BTU/LBM-R)',3X,F9.6,8X,F9.6)
 9130 FORMAT(1H ,/,47X,'SATURATED',8X,'SATURATED',/,47X,'VAPOR',12X,
     1    'LIQUID',/,17X,60('#'))
 9140 FORMAT(1H ,/)
      RETURN
! Univac Statement      DEBUG SUBCHK
      END
C    $$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
C    $                                                                   $
C    $                        P H F I N D                                $
C    $                                                                   $
C    $    USES NEWTON-RAPHSON ITERATION TO FIND DENSITY AND TEMPERATURE$
C    $GIVEN PRESSURE AND ENTHALPY FOR SUPERHEATED VAPOR AND COMPRESSED $
C    $LIQUID.                                                            $
C    $                                                                   $
C    $$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
      SUBROUTINE PHFIND
      IMPLICIT REAL(L)
      PARAMETER M2=6
      COMMON/WATER1/VSATS,VSATH,VSATU,VSATV,LSATS,LSATH,LSATU,LSATV,
     1    TEMPK, TCRIT, PCRIT, TAUCRT, TGUESS, GAS, GAS1, CONVR,
     2    DPSDT, DGUESS, DENSIT, DPDRHO, TEMPF, PRESS, UINT, ENTH, VOL,
     3    ENTR, DHDT, DHDRHO, DPDT, SCAL, HCAL,DSDRHO, SJOULE, SKNOWN,
     4    ITERAT, PSAT, DENTOL, TTOLER, BTOLER, PCAL, QUAL, PKNOWN, DSDT
      REAL JACOB
      TT=1.0005
      BT=.9995
      DENT=1.0E-11
      DLAST=DENSIT
      TLAST=TEMPK
      CALL PROPER
      DO 3600 I=1, 400
 3470 JACOB=DPDRHO*DHDT-DHDRHO*DPDT
      DENSIT=DLAST-((PCAL-PRESS   )*DHDT-(HCAL-ENTH)*DPDT)/JACOB
      IF(DENSIT.LE.1.0E-10) DENSIT=DLAST/10.0
      TEMPK=TLAST-((HCAL-ENTH)*DPDRHO-(PCAL-PRESS   )*DHDRHO)/JACOB
      IF(TEMPK.LE.0.0) TEMPK=300.0
```

```
      CALL PROPER
      IF(PCAL/PRESS   .LT.TT.AND.PCAL/PRESS   .GT.BT.AND.
     1    HCAL/ENTH.LT.TT.AND.HCAL/ENTH.GT.BT) RETURN
      IF(ABS(DLAST-DENSIT).LT.DENT.AND.ABS(TLAST-TEMPK).LT.1.0E-05)
     1                                                               RETURN
      DLAST=DENSIT
      TLAST=TEMPK
 3600 CONTINUE
      WRITE(M2,9000) ITERAT
9000  FORMAT(' ***TEMPERATURE-DENSITY SEARCH, GIVEN PRESSURE & ENTHALPY,
     1FAILED TO CONVERGE IN ',I5,' ITERATIONS, RESULTS MAY NOT BE VALID'
     2  )
      RETURN
! Univac Statement      DEBUG SUBCHK
      END
C    $$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
C    $                                                                 $
C    $                       P S F I N D                               $
C    $                                                                 $
C    $    USES NEWTON-RAPHSON ITERATION TO FIND DENSITY AND TEMPERATURE$
C    $GIVEN PRESSURE AND ENTROPY FOR SUPERHEATED VAPOR AND COMPRESSED  $
C    $LIQUID                                                           $
C    $                                                                 $
C    $$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
      SUBROUTINE PSFIND
      IMPLICIT REAL(L)
      PARAMETER M2=6
      COMMON/WATER1/VSATS,VSATH,VSATU,VSATV,LSATS,LSATH,LSATU,LSATV,
     1    TEMPK, TCRIT, PCRIT, TAUCRT, TGUESS, GAS, GAS1, CONVR,
     2    DPSDT, DGUESS, DENSIT, DPDRHO, TEMPF, PRESS, UINT, ENTH, VOL,
     3    ENTR, DHDT, DHDRHO, DPDT, SCAL, HCAL,DSDRHO, SJOULE, SKNOWN,
     4    ITERAT, PSAT, DENTOL, TTOLER, BTOLER, PCAL, QUAL, PKNOWN, DSDT
      REAL JACOB
      DLAST=DENSIT
      TLAST=TEMPK
      CALL PROPER
      DO 4000 I=1, ITERAT
 3790 JACOB=DPDRHO*DSDT-DSDRHO*DPDT
      DENSIT=DLAST-((PCAL-PRESS   )*DSDT-(SJOULE-SKNOWN)*DPDT)/JACOB
      IF(DENSIT.LE.1.0E-10) DENSIT=DLAST/10.0
      TEMPK=TLAST-((SJOULE-SKNOWN)*DPDRHO-(PCAL-PRESS   )*DSDRHO)/JACOB
      IF(TEMPK.LE.0.0) TEMPK=300.0
      CALL PROPER
      IF(PCAL/PRESS.LT.TTOLER.AND.PCAL/PRESS.GT.BTOLER.AND.SCAL/ENTR.LT.
     1    TTOLER.AND.SCAL/ENTR.GT.BTOLER) RETURN
      IF(ABS(DLAST-DENSIT).LT.DENTOL.AND.ABS(TLAST-TEMPK).LT.1.0E-04)
     1                                                                 RETURN
      DLAST=DENSIT
      TLAST=TEMPK
 4000 CONTINUE
      WRITE(M2,9000) ITERAT
```

```
 9000 FORMAT(' ***TEMPERATURE-DENSITY SEARCH GIVEN PRESSURE AND ENTROPY
     1FAILED TO CONVERGE IN ',I5,' ITERATIONS, RESULTS MAY NOT BE VALID'
     2   )
      RETURN
! Univac Statement      DEBUG SUBCHK
      END
C     ***********************************************************
C     *                                                         *
C     *                     P S A T U R                         *
C     *                                                         *
C     *   COMPUTES SATURATION GIVEN SATURATION TEMPERATURE      *
C     *                                                         *
C     ***********************************************************
      SUBROUTINE PSATUR
      IMPLICIT REAL(L)
      COMMON/WATER1/VSATS,VSATH,VSATU,VSATV,LSATS,LSATH,LSATU,LSATV,
     1    TEMPK, TCRIT, PCRIT, TAUCRT, TGUESS, GAS, GAS1, CONVR,
     2    DPSDT, DGUESS, DENSIT, DPDRHO, TEMPF, PRESS, UNIT, ENTH, VOL,
     3    ENTR, DHDT, DHDRHO, DPDT, SCAL, HCAL, DSDRHO, SJOULE, SKNOWN,
     4    ITERAT, PSAT, DENTOL, TTOLER, BTOLER, PCAL, QUAL, PKNOWN, DSDT
      DIMENSION F(8)
      DATA (F(I), I=1,8) /-741.9242, -29.721, -11.55286, -0.8685635,
     1    0.1094098, 0.439993, 0.2520658, 0.05218684/
      TAU=1000.0/TEMPK
      TEMPC=TEMPK-273.15
      SUM1=0.0
      SUM2=0.0
      DO 2420 I=1,8
      SUM1=SUM1+F(I)*(0.65-0.01*TEMPC)**(I-1)
      SUM2=SUM2+F(I)*(I-1)*0.01*(0.65-0.01*TEMPC)**(I-2)
 2420 CONTINUE
      PSAT=EXP(TAU*1.0E-05*(TCRIT-TEMPC)*SUM1)*PCRIT
      DPSDT=PSAT/100.0*(((TEMPC-TCRIT)/TEMPK**2-1.0/TEMPK)*SUM1
     1    -(TCRIT/TEMPK-TEMPC/TEMPK)*SUM2)
      RETURN
! Univac Statement      DEBUG SUBCHK
      END
C    $$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
C    $                                                                  $
C    $                        D F I N D                                 $
C    $                                                                  $
C    $   USES NEWTON-RAPHSON ITERATION TO LOCATE DENSITY GIVEN PRESSURE$
C    $AND TEMPERATURE.                                                  $
C    $                                                                  $
C    $$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
      SUBROUTINE DFIND(NSTAT1)
      IMPLICIT REAL(L)
      PARAMETER M2=6
      COMMON/WATER1/VSATS,VSATH,VSATU,VSATV,LSATS,LSATH,LSATU,LSATV,
     1    TEMPK, TCRIT, PCRIT, TAUCRT, TGUESS, GAS, GAS1, CONVR,
     2    DPSDT, DGUESS, DENSIT, DPDRHO, TEMPF, PRESS, UINT, ENTH, VOL,
```

```
     3    ENTR, DHDT, DHDRHO, DPDT, SCAL, HCAL,DSDRHO, SJOULE, SKNOWN,
     4    ITERAT, PSAT, DENTOL, TTOLER, BTOLER, PCAL, QUAL, PKNOWN, DSDT
      DENSIT=DGUESS
      LASTD2=DGUESS
      CALL PROPER
      DO 2560 I=1, ITERAT
 2510 DENSIT=LASTD2-(PCAL-PRESS  )/DPDRHO
      IF(DENSIT.LE.1.0E-10) DENSIT=LASTD2/10.0
      CALL PROPER
      IF(PCAL/PRESS  .GT.BTOLER.AND.PCAL/PRESS  .LT.TTOLER.OR.
     1    ABS(LASTD2-DENSIT).LT.DENTOL) RETURN
      LASTD2=DENSIT
 2560 CONTINUE
      WRITE(M2,9000) ITERAT,PRESS,PCAL
 9000 FORMAT(' *** DENSITY SEARCH FAILED TO CONVERGE IN',I5,'ITERATIONS
     1 , RESULTS MAY NOT BE VALID***',/,' DESIRED PRESSURE=',F10.3,
     2  ' CALCULATED PRESSURE=',F10.3)
      WRITE(M2,9001) NSTAT1
 9001 FORMAT(1H ,'STATE NUMBER',I3)
      RETURN
! Univac Statement      DEBUG SUBCHK
           END
C    $$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
C    $                                                                  $
C    $                       P R O P E R                                $
C    $                                                                  $
C    $    COMPUTES PROPERTIES AND THEIR DERIVATIVES.                    $
C    $                                                                  $
C    $$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$

      SUBROUTINE PROPER
      IMPLICIT REAL(L)
      COMMON/WATER1/VSATS,VSATH,VSATU,VSATV,LSATS,LSATH,LSATU,LSATV,
     1    TEMPK, TCRIT, PCRIT, TAUCRT, TGUESS, GAS, GAS1, CONVR,
     2    DPSDT, DGUESS, DENSIT, DPDRHO, TEMPF, PRESS, UINT, ENTH, VOL,
     3    ENTR, DHDT, DHDRHO, DPDT, SCAL, HCAL,DSDRHO, SJOULE, SKNOWN,
     4    ITERAT, PSAT, DENTOL, TTOLER, BTOLER, PCAL, QUAL, PKNOWN, DSDT
      DIMENSION A(10,7), C(8)
      DATA ((A(I,J), J=1,7), I=1,10)/ 29.492937, -5.198586, 6.8335354,
     1    -0.1564104, -6.3972405, -3.9661401, -0.69048554, -132.13917,
     2    7.7779182, -26.149751, -0.72546108, 26.409282, 15.453061,
     3    2.7407416, 274.64632, -33.301902, 65.326396, -9.2734289,
     4    -47.740374, -29.142470, -5.1028070, -360.93828, -16.254622,
     5    -26.181978, 4.312584, 56.32313, 29.568796, 3.9636085, 342.1843
     6    1, -177.31074, 0.0, 0.0, 0.0, 0.0, 0.0, -244.50042, 127.48742,
     7    0.0, 0.0, 0.0, 0.0, 0.0, 155.18535, 137.46153, 0.0, 0.0, 0.0,
     8    0.0, 0.0, 5.9728487, 155.97836, 0.0, 0.0, 0.0, 0.0, 0.0,
     9    -410.30848, 337.31180, -137.46618, 6.7874983, 136.87317,
     A    79.84797, 13.041253, -416.0586, -209.88866, -733.96848,
     B    10.401717, 645.8188, 399.1757, 71.531353 /
      DATA (C(I), I=1,8) /1857.065, 3229.12, -419.465, 36.6649, -20.5516
```

```
     1    , 4.85233, 46.0, -1011.249/
      VOL=1.0/(62.4283*DENSIT)
      TAU=1000.0/TEMPK
      Z3SUM=0.0
      Z4SUM=0.0
      Z5SUM=0.0
      D2QDTA=0.0
      W2SUM=0.0
      Z9SUM=0.0
      DO 2890 J=1,7
      IF(J.EQ.1) GO TO 2680
      DCON=1.0
      TAUCON=2.5
      GO TO 2700
 2680 DCON=0.634
      TAUCON=TAUCRT
 2700 CONTINUE
      ZSUM=0.0
      Z2SUM=0.0
      WSUM=0.0
      DO 2740 I=1,8
      ZSUM=ZSUM+A(I,J)*(DENSIT-DCON)**(I-1)
      Z2SUM=Z2SUM+A(I,J)*(I-1)*(DENSIT-DCON)**(I-2)
      WSUM=WSUM+A(I,J)*(I-1)*(I-2)*(DENSIT-DCON)**(I-3)
 2740 CONTINUE
      Z1SUM=0.0
      DO 2780 I=9,10
      Z1SUM=Z1SUM+A(I,J)*DENSIT**(I-9)
 2780 CONTINUE
      Z1SUM=Z1SUM*EXP(-4.8*DENSIT)
      W1SUM=Z1SUM*23.04
      Z3SUM=Z3SUM+(TAU-TAUCON)**(J-2)*(ZSUM+Z1SUM)
      Z4SUM=Z4SUM+(TAU-TAUCON)**(J-2)*(Z2SUM-4.8*Z1SUM+EXP(-4.8*DENSIT)
     1    *A(10,J))
      Z5SUM=Z5SUM+((TAU-TAUCON)**(J-2)+(TAU-TAUCRT)* (J-2)*(TAU-TAUCON)
     1    **(J-3))*(ZSUM+Z1SUM)
      Z9SUM=Z9SUM+(J-2)*(TAU-TAUCON)**(J-3)*(Z2SUM-4.8*Z1SUM+
     1   EXP(-4.8*DENSIT)*A(10,J))*(TAU-TAUCRT)
      W2SUM=W2SUM+(TAU-TAUCON)**(J-2)*(WSUM+W1SUM-9.6*EXP(-4.8*DENSIT)
     1    *A(10,J))
      D2QDTA=D2QDTA+((TAU-TAUCRT)*(J-2)*(J-3)*(TAU-TAUCON)**(J-4)+
     1   2.0*(J-2)*(TAU-TAUCON)**(J-3))*(ZSUM+Z1SUM)
 2890 CONTINUE
      Q=(TAU-TAUCRT)*Z3SUM
      DQDRHO=(TAU-TAUCRT)*Z4SUM
      DQDTAU=Z5SUM
      Z6SUM=0.0
      Z7SUM=0.0
      Z8SUM=0.0
      X7SUM=0.0
      DO 2990  I=1,6
```

```
      X7SUM=X7SUM+(2-I)*(1-I)*C(I)/TAU**I
      Z6SUM=Z6SUM+(2-I)*C(I)/TAU**(I-1)
      Z7SUM=C(I)*(I-1)*TAU**(2-I)/1000.0 +Z7SUM
      Z8SUM=Z8SUM+C(I)*(I-1)*(I-2)*TAU**(3-I)
 2990 CONTINUE
      D2CDT2=Z8SUM*1.0E-06-C(7)/TEMPK**2+C(8)/(1000.0*TEMPK)
      D2QDD2=(TAU-TAUCRT)*W2SUM
      D2CTDT=X7SUM-C(7)/TAU+C(8)/TAU**2
      DCTDT=Z6SUM+C(7)*(LOG(TEMPK)-1.0)-C(8)/TAU
      DCDT=Z7SUM+C(7)/TEMPK+C(8)/1000.0*(LOG(TEMPK)+1.0)
      PCAL=DENSIT*GAS1*TEMPK*(1.0+Q*DENSIT+DENSIT**2*DQDRHO)
      UINT=CONVR*(1000.0*GAS*DENSIT*DQDTAU+DCTDT)
      SCAL=CONVR*(-GAS*(LOG(DENSIT)+DENSIT*Q-DENSIT*TAU*DQDTAU)
     1    -DCDT)/1.8
      SJOULE=SCAL*1.8
      HCAL=CONVR*(GAS*TEMPK*(DENSIT*TAU*DQDTAU+1.0+DENSIT*Q
     1    +DENSIT**2*DQDRHO)+DCTDT)
      DQDDDT=(Z9SUM+Z4SUM)*(-TAU/TEMPK)
      DPDRHO=GAS1*TEMPK*(1.0+2.0*Q*DENSIT+4.0*DENSIT**2*DQDRHO
     1    +DENSIT**3*D2QDD2)
      DHDT=CONVR*(GAS*(1.0+DENSIT*Q+DENSIT**2*DQDRHO-TAU*DENSIT*DQDTAU
     1   -DENSIT*TAU**2*D2QDTA+TEMPK*DENSIT**2*DQDDDT)-TAU/TEMPK*D2CTDT)
      DHDRHO=GAS*CONVR*TEMPK*(TAU*DQDTAU-TEMPK*DENSIT*DQDDDT+Q
     1    +3.0*DENSIT*DQDRHO+DENSIT**2*D2QDD2)
      DPDT=DENSIT*GAS1*(1.0+DENSIT*Q+DENSIT**2*DQDRHO+TEMPK*DENSIT*
     1    (-TAU/TEMPK*DQDTAU+DENSIT*DQDDDT))
      DSDRHO=-CONVR*GAS*(1.0/DENSIT+Q+DENSIT*DQDRHO-TAU*DQDTAU
     1    +DENSIT*TEMPK*DQDDDT)
      DSDT=-CONVR*(GAS*TAU**2/TEMPK*DENSIT*D2QDTA+D2CDT2)/1.8
      RETURN
! Univac Statement      DEBUG SUBCHK
      END

    SUBROUTINE SIWATER(CYCLE,FGIVEN,SGIVEN,CALLIN,NSTAT1,NPRINA,QUAL1,
     #  ENTR1,ENTH1,VOL1,UINT1,TEMPF1,PRESS1,SATLS1,SATLH1,SATLV1,SATLU1,
     $      ,VSATS1,VSATH1,VSATV1,VSATU1,NERROR)
    INTEGER CALLIN
    CHARACTER*2 CYCLE
    IF(CYCLE.EQ.'TP') THEN
    FGIVEN=FGIVEN*1.8+32.
    SGIVEN=.145037743897283*SGIVEN
    ELSEIF(CYCLE.EQ.'TX') THEN
    FGIVEN=FGIVEN*1.8+32.
    ELSEIF(CYCLE.EQ.'PS') THEN
    FGIVEN=.145037743897283*FGIVEN
    SGIVEN=SGIVEN/4.1868
    ELSEIF(CYCLE.EQ.'PH') THEN
    FGIVEN=.145037743897283*FGIVEN
    SGIVEN=SGIVEN/2.326
    ELSEIF(CYCLE.EQ.'PX') THEN
    FGIVEN=FGIVEN*.145037743897283
```

```
      ELSE
      PRINT*,'WRONG CYCLE VALUE IS GIVEN'
      RETURN
      ENDIF
      CALLIN=10
      NSTAT1=0
      NPRINA=0
    10        CALL WATER(CYCLE,FGIVEN,SGIVEN,CALLIN,NSTAT1,NPRINA,QUAL1,
       #  ENTR1,ENTH1,VOL1,UINT1,TEMPF1,PRESS1,SATLS1,SATLH1,SATLV1,SATLU1,
       $       ,VSATS1,VSATH1,VSATV1,VSATU1,NERROR)
      ENTR1=ENTR1*4.1868
      ENTH1=ENTH1*2.326
      VOL1=VOL1*0.0624219
      UINT1=UINT1*2.326
      TEMPF1=(TEMPF1-32.)/1.8
      PRESS1=PRESS1/.145037743897283
      SATLS1=SATLS1*4.1868
      SATLH1=SATLH1*2.326
      SATLV1=SATLV1*0.0624219
      SATLU1=SATLU1*2.326
      VSATS1=VSATS1*4.1868
      VSATH1=VSATH1*2.326
      VSATV1=VSATV1*0.0624219
      VSATU1=VSATU1*2.326
      RETURN
      END
      SUBROUTINE SICARBON(CYCLE,FGIVEN,SGIVEN,LFTYPE,H,S,V,TR,P,NPRIN,
       #       SL,HL,VL,SV,HV,VV,QUAL)
      IMPLICIT REAL(L)
      INTEGER CALLIN
      CHARACTER CYCLE*2,LFTYPE*6
          COMMON/ERFLAG/ERFLAG
      IF(CYCLE.EQ.'TP') THEN
      FGIVEN=FGIVEN*1.8+32.
      SGIVEN=.145037743897283*SGIVEN
      ELSEIF(CYCLE.EQ.'TX') THEN
      FGIVEN=FGIVEN*1.8+32.
      ELSEIF(CYCLE.EQ.'PS') THEN
      FGIVEN=.145037743897283*FGIVEN
      SGIVEN=SGIVEN/4.1868
      ELSEIF(CYCLE.EQ.'PH') THEN
      FGIVEN=.145037743897283*FGIVEN
      SGIVEN=SGIVEN/2.326
      ELSEIF(CYCLE.EQ.'PX') THEN
      FGIVEN=FGIVEN*.145037743897283
      ELSEIF(CYCLE.EQ.'TV') THEN
      FGIVEN=FGIVEN*1.8+32.
      SGVEN=SGIVEN*16.01846
      ELSE
      PRINT*,'WRONG CYCLE VALUE IS GIVEN'
      RETURN
```

```
    ENDIF
    CALL CARBON(CYCLE,FGIVEN,SGIVEN,LFTYPE,H,S,V,TR,P,NPRIN,
     #      SL,HL,VL,SV,HV,VV,QUAL)
    S=S*4.1868
    H=H*2.326
    V=V*0.0624219
    TR=(TR-32.)/1.8
    P=P/.145037743897283
    SL=SL*4.1868
    HL=HL*2.326
    VL=VL/16.01846
    SV=SV*4.1868
    HV=HV*2.326
    VV=VV/16.01846
    RETURN
    END
C     ******************************************************************
C     *                                                                *
C     *                        C A R B O N                             *
C     *                                                                *
C     *    THIS SUBROUTINE CALCULATES THE THERMODYNAMIC PROPERTIES FOR *
C     *LIGHT HYDROCARBONS BY USING K.E. STARLINGS MODIFICATION TO THE  *
C     *BENEDICT-WEBB RUBIN EQUATION OF STATE. GIVEN THE FOLLOWING      *
C     *PAIRS OF KNOWNS THE PROGRAM WILL CALCULATE THE REMAINING STATE  *
C     *PROPERTIES:                                                     *
C     *            1) TEMPERATURE-PRESSURE                             *
C     *            2) TEMPERATURE-QUALITY                              *
C     *            3) TEMPERATURE-SPECIFIC VOLUME                      *
C     *            4) PRESSURE-QUALITY                                 *
C     *            5) PRESSURE-ENTROPY                                 *
C     *            6) PRESSURE-ENTHALPY                                *
C     *                                                                *
C     *                                                                *
C     ******************************************************************
      SUBROUTINE CARBON(CYCLE,FGIVEN,SGIVEN,LFTYPE,H,S,V,TR,P,NPRIN,
     1  SL,HL,VL,SV,HV,VV,QUAL)
      CHARACTER LFTYPE*6,CYCLE*2
      COMMON/CAR1/C(11),CS(6),CH(6),WM,R,CONVR,LIMIT,ABST,TTOLER,BTOLER
      COMMON/ERFLAG/ERFLAG
      PARAMETER M2=6
      DIMENSION XISOB(27),XISOP(27),XMETH(27),ETH(27),PROP(27),
     1PENT(27),HEX(27),BUT(27),HEPT(27),OCT(27),ETHYL(27),PROPYL(27)
      DATA ABST,TTOLER,BTOLER,LIMIT,R/459.67,1.005,0.995,100,10.7335/
      DATA CONVR/0.185057/
      DATA (XISOB(I),I=1,27)/479907.0E-01,37264.0,8.58663,1.87890,
     1406763.0E05,101413.0E05,216863.0E02,853176.0E06,840860.0E08,
     24.23987,7.11486,0.86401,9.079E-04,-2.073E-07,0.885E-10,-0.241E-13,
     30.0,-776.11,6.872E-02,3.071E-04,0.0,0.0,0.0,529.06,274.96,0.2373,
     458.12/
      DATA(PENT(I),I=1,27)/162185.0,51108.2,16.6070,2.44417,135286.0E06,
     1223931.0E05,388521.0E02,101769.0E07,390860.0E08,7.06702,11.8593,
```

```
20.9179,1.511E-04,14.048E-07,-13.785E-10,4.560E-13,0.0,-688.09,
322.267E-02,-0.89E-04,5.018E-07,-3.048E-10,0.641E-13,489.51,385.42,
40.2007,72.146/
 DATA(PROP(I),I=1,27)/40066.4,18634.7,5.46248,0.964762,274461.0E05,
1796178.0E04,150520.0E02,453708.0E06,256053.0E08,2.01402,4.56182,
20.9353,26.158E-04,-56.478E-07,85.177E-10,-63.595E-13,18.451E-16,
3-804.84,25.477E-02,-1.602E-04,5.160E-07,-2.110E-10,0.0,617.400,
4206.13,0.3121,44.094/
 DATA(XMETH(I),I=1,27)/2574.89,7520.29,0.925404,0.723251,
1437222.0E03,271092.0E03,47489.1,107737.0E05,301122.0E05,0.468828,
21.48640,1.7491,68.131E-04,-167.165E-07,244.251E-10,-180.176E-13,
352.320E-16,-1798.16,53.978E-02,-1.345E-04,0.676E-07,1.966E-10,
4-1.037E-13,673.1,-116.43,0.6274,16.042/
 DATA(ETH(I),I=1,27)/22404.5,13439.3,3.11206,0.826059,681826.0E04,
1295195.0E04,702189.0,257477.0E06,146819.0E08,0.90681,2.99656,
21.3021,23.761E-04,-32.627E-07,28.695E-10,-9.485E-13,0.0,-993.32,
330.668E-02,-1.427E-04,4.203E-07,-1.567E-10,0.0,709.8,90.03,
40.4218,30.068/
 DATA(PROPYL(I),I=1,27)/81880.4,6051.36,7.64114,0.114457,
1294141.0E05,974762.0E04,54193.5E02,705921.0E06,341250.0E08,
21.36532,4.07919,1.2501,7.946E-04,-1.155E-07,-0.025E-10,0.054E-13,
30.0,376.86,9.868E-02,2.347E-04,0.295E-07,-0.196E-10,0.0,670.27,
4197.4,0.3449,42.08/
 DATA(HEX(I),I=1,27)/434517.0,45333.1,29.4983,2.66233,318412.0E06,
1526067.0E05,327460.0E02,552158.0E07,626433.0E09,9.70230,14.8720,
21.3746,-17.514E-04,35.508E-07,-21.907E-10,4.769E-13,0.0,
3-618.38,.240E-02,4.148E-04,-.614E-07
3,0.0,0.0,439.4,453.45,0.1696,86.172/
 DATA(XISOP(I),I=1,27)/204344.0,35742.0,19.8384,1.27752,129083.0E06
1,228430.0E05,349220.0E02,142115.0E07,241326.0E08,6.16154,11.7384,
2.7737,9.412E-04,-3.298E-07,2.306E-10,-.953E-13,.129E-16,
3-709.28,7.482E-02,2.545E-04,.857E-07,
3-0.464E-10,0.0,483.5,369.0,0.2027,72.146/
 DATA(BUT(I),I=1,27)/71181.8,32544.7,9.14066,1.56588,700044.0E05
1,137436.0E05,364238.0E02,333159.0E06,230902.0E07,4.00985,7.54122,
20.9008,10.06E-04,-3.006E-07,0.846E-10,0.0,0.0,-739.77,22.389E-02,
3-0.601E-04,3.93E-07,-1.594E-10,0.0,551.1,305.67,0.2448,58.12/
 DATA(HEPT(I),I=1,27)/359087.0,77826.9,27.4415,3.60493,374876.0E06,
1615662.0E05,8351150.0,777123.0E07,636251.0E07,21.8782,24.7604,
21.3170,-17.863E-04,36.047E-07,-22.278E-10,4.856E-13,0.0,-596.40,
30.549E-02,3.989E-04,-0.587E-07,0.0,0.0,396.8,512.85,0.1465,100.198
4/
 DATA(OCT(I),I=1 ,27)/131646.0,81690.6,10.5907,4.86965,642053.0E06,
1996546.0E05,185906.0E03,790575.0E07,346419.0E08,34.5124,21.9888,
1.2641,-17.734E-04,35.827E-07,-22.131E-10,4.823E-13,0.0,-577.59,
30.612E-02,4.094E-04,-0.605E-07,0.0,0.0,364.4,563.79,0.1284,114.224
4/
 DATA(ETHYL(I),I=1,27)/15978.1,12133.9,2.62914,0.747945,409725.0E04
1,163203.0E04,903550.0,517563.0E05,161706.0E05,0.589158,2.27971,
21.3982,23.551E-04,-34.268E-07,30.347E-10,-10.071E-13,0.0,928.25,
332.949E-02,-2.477E-04,4.768E-07,-1.711E-10,0.0,742.0,49.82,0.5035,
```

```
   428.05/
     ERFLAG=2.
     IF(LFTYPE.EQ.'ISOB  ') GO TO 10
     IF(LFTYPE.EQ.'PENT  ') GO TO 20
     IF(LFTYPE.EQ.'ISOP  ') GO TO 30
     IF(LFTYPE.EQ.'METH  ') GO TO 40
     IF(LFTYPE.EQ.'ETH   ') GO TO 50
     IF(LFTYPE.EQ.'PROP  ') GO TO 60
     IF(LFTYPE.EQ.'HEX   ') GO TO 70
     IF(LFTYPE.EQ.'BUT   ') GO TO 80
     IF(LFTYPE.EQ.'HEPT  ') GO TO 90
     IF(LFTYPE.EQ.'OCT   ') GO TO 100
     IF(LFTYPE.EQ.'ETHYL ') GO TO 110
     IF(LFTYPE.EQ.'PROPYL') GO TO 120
     WRITE(M2,9050)
     RETURN
  10 CALL SETUP(XISOB,PCRIT,TCRIT,DCRIT)
     GO TO 180
  20 CALL SETUP(PENT,PCRIT,TCRIT,DCRIT)
     GO TO 180
  30 CALL SETUP(XISOP,PCRIT,TCRIT,DCRIT)
     GO TO 180
  40 CALL SETUP(XMETH,PCRIT,TCRIT,DCRIT)
     GO TO 180
  50 CALL SETUP(ETH,PCRIT,TCRIT,DCRIT)
     GO TO 180
  60 CALL SETUP(PROP,PCRIT,TCRIT,DCRIT)
     GO TO 180
  70 CALL SETUP(HEX,PCRIT,TCRIT,DCRIT)
     GO TO 180
  80 CALL SETUP(BUT,PCRIT,TCRIT,DCRIT)
     GO TO 180
  90 CALL SETUP(HEPT,PCRIT,TCRIT,DCRIT)
     GO TO 180
 100 CALL SETUP(OCT,PCRIT,TCRIT,DCRIT)
     GO TO 180
 110 CALL SETUP(ETHYL,PCRIT,TCRIT,DCRIT)
     GO TO 180
 120 CALL SETUP(PROPYL,PCRIT,TCRIT,DCRIT)
 180 IF(CYCLE.EQ.'TP') GO TO 200
     IF(CYCLE.EQ.'TX') GO TO 400
     IF(CYCLE.EQ.'PX') GO TO 600
     IF(CYCLE.EQ.'PS') GO TO 800
     IF(CYCLE.EQ.'PH') GO TO 1000
     IF(CYCLE.EQ.'TV') GO TO 1200
     WRITE(M2,9000)CYCLE
     RETURN
 200 T=FGIVEN
     P=SGIVEN
     TR=T+ABST
     IF(T.GE.TCRIT) GO TO 220
```

```
      TCRITR=TCRIT+ABST
      CALL PRESAT(TR,PSAT,DL,DV,DCRIT,TCRITR,LFTYPE)
      IF(P/PSAT.GT.TTOLER.OR.P/PSAT.LT.BTOLER) GO TO 210
      WRITE(M2,9010)T,P
      ERFLAG=1.
      RETURN
  210 IF(P/PSAT.GT.TTOLER) GO TO240
  220 QUAL=2.0
      D=1.0E-06
      GO TO 260
  240 D=0.66569
      QUAL=-2.0
  260 CALL VFIND(TR,P,D)
      CALL ENTH(TR,D,H,DHDT,DHDD)
      CALL ENTR(TR,D,S,DSDT,DSDD)
      V=1.0/(D*WM)
      CALL PRINTC(TR,P,H,S,V,QUAL,0.0,0.0,0.0,0.0,0.0,0.0,LFTYPE,NPRIN)
      RETURN
  400 T=FGIVEN
      QUAL=SGIVEN
      TR=T+ABST
      IF(QUAL.LE.1.0.AND.QUAL.GE.0.0) GO TO 410
      WRITE(M2,9020)QUAL
      RETURN
  410 IF(T.LE.TCRIT) GO TO 420
      WRITE(M2,9030)
      RETURN
  420 TCRITR=TCRIT+ABST
      CALL PRESAT(TR,P,DL,DV,DCRIT,TCRITR,LFTYPE)
      CALL SATC(TR,DL,DV,HL,SL,HV,SV)
      S=SV*QUAL+SL*(1.0-QUAL)
      H=HV*QUAL+HL*(1.0-QUAL)
      VV=1.0/(DV*WM)
      VL=1.0/(DL*WM)
      V=VV*QUAL+VL*(1.0-QUAL)
      CALL PRINTC(TR,P,H,S,V,QUAL,HL,SL,VL,HV,SV,VV,LFTYPE,NPRIN)
      RETURN
  600 P=FGIVEN
      QUAL=SGIVEN
      IF(QUAL.LE.1.0.AND.QUAL.GE.0.0) GO TO 610
      WRITE(M2,9020)QUAL
      RETURN
  610 IF (P.LE.PCRIT) GO TO 620
      WRITE(M2,9040)
      RETURN
  620 CALL TEMSAT(P,TR,DL,DV,TCRIT,DCRIT)
      CALL SATC(TR,DL,DV,HL,SL,HV,SV)
      S=SV*QUAL+SL*(1.0-QUAL)
      H=HV*QUAL+HL*(1.0-QUAL)
      VL=1.0/(DL*WM)
      VV=1.0/(DV*WM)
```

```
      V=VV*QUAL+VL*(1.0-QUAL)
      CALL PRINTC(TR,P,H,S,V,QUAL,HL,SL,VL,HV,SV,VV,LFTYPE,NPRIN)
      RETURN
  800 P=FGIVEN
      S=SGIVEN
      IF(P.LT.PCRIT) GO TO 820
      TR=TCRIT*1.1
      DV=1.01*DCRIT
      GO TO 840
  820 CALL TEMSAT(P,TR,DL,DV,TCRIT,DCRIT)
      CALL SATC(TR,DL,DV,HL,SL,HV,SV)
      IF(S.LT.0.0)GO TO 830
      IF(S/SV.GT.TTOLER) GO TO 840
      IF(S/SL.LT.BTOLER) GO TO 860
      GO TO 835
  830 IF(S/SV.LT.BTOLER)GO TO 840
      IF(S/SL.GT.TTOLER)GO TO 860
  835 QUAL=(S-SL)/(SV-SL)
      H=HV*QUAL+HL*(1.0-QUAL)
      VL=1.0/(DL*WM)
      VV=1.0/(DV*WM)
      V=VV*QUAL+VL*(1.0-QUAL)
      CALL PRINTC(TR,P,H,S,V,QUAL,HL,SL,VL,HV,SV,VV,LFTYPE,NPRIN)
      RETURN
  840 QUAL=2.0
      DV=0.01
      CALL PSKNOW(P,S,TR,DV)
      D=DV
      GO TO 880
  860 QUAL=-2.0
      DL=0.55
      CALL PSKNOW(P,S,TR,DL)
      D=DL
  880 CALL ENTH(TR,D,H,DHDT,DHDD)
      V=1.0/(D*WM)
      CALL PRINTC(TR,P,H,S,V,QUAL,0.0,0.0,0.0,0.0,0.0,0.0,LFTYPE,NPRIN)
      RETURN
 1000 P=FGIVEN
      H=SGIVEN
      IF(P.LT.PCRIT) GO TO 1020
      TR=TCRIT*1.1
      DV=1.01*DCRIT
      GO TO 1040
 1020 CALL TEMSAT(P,TR,DL,DV,TCRIT,DCRIT)
      CALL SATC(TR,DL,DV,HL,SL,HV,SV)
      IF(H.LE.0.0)GO TO 1030
      IF(H/HV.GT.TTOLER) GO TO 1040
      IF(H/HL.LT.BTOLER) GO TO 1060
      GO TO 1035
 1030 IF(H/HV.LT.BTOLER)GO TO 1040
      IF(H/HL.GT.TTOLER)GO TO 1060
```

```
1035 QUAL=(H-HL)/(HV-HL)
     S=SV*QUAL+SL*(1.0-QUAL)
     VL=1.0/(DL*WM)
     VV=1.0/(DV*WM)
     V=VV*QUAL+VL*(1.0-QUAL)
     CALL PRINTC(TR,P,H,S,V,QUAL,HL,SL,VL,HV,SV,VV,LFTYPE,NPRIN)
     RETURN
1040 QUAL=2.0
     DV=0.01
     CALL PHKNOW(P,H,TR,DV)
     D=DV
     GO TO 1080
1060 QUAL=-2.0
     DL=0.55
     CALL PHKNOW(P,H,TR,DL)
     D=DL
1080 CALL ENTR(TR,D,S,DSDT,DSDD)
     V=1.0/(D*WM)
     CALL PRINTC(TR,P,H,S,V,QUAL,0.0,0.0,0.0,0.0,0.0,0.0,LFTYPE,NPRIN)
     RETURN
1200 T=FGIVEN
     TR=FGIVEN+ABST
     V=SGIVEN
     D=1.0/(V*WM)
     IF(T.GE.TCRIT) GO TO 1240
     CALL PRESAT(TR,P,DL,DV,DCRIT,TCRITR,LFTYPE)
     IF(D/DL.LT.BTOLER) GO TO 1260
     IF(D/DV.GT.TTOLER) GO TO 1240
     CALL SATC(TR,DL,DV,HL,SL,HV,SV)
     QUAL=(D-DL)/(DV-DL)
     VL=1.0/(DL*WM)
     VV=1.0/(DV*WM)
     H=HV*QUAL+HL*(1.0-QUAL)
     S=SV*QUAL+SL*(1.0-QUAL)
     CALL PRINTC(TR,P,H,S,V,QUAL,HL,SL,VL,HV,SV,VV,LFTYPE,NPRIN)
     RETURN
1240 QUAL=2.0
     GO TO 1280
1260 QUAL=-2.0
1280 CALL PRES(TR,D,P,DPDT,DPDD)
     CALL ENTH(TR,D,H,DHDT,DHDD)
     CALL ENTR(TR,D,S,DSDT,DSDD)
     CALL PRINTC(TR,P,H,S,V,QUAL,0.0,0.0,0.0,0.0,0.0,0.0,LFTYPE,NPRIN)
     RETURN
9050 FORMAT(1H ,4X,'***ERROR IN CARBON, THE FLUID',A6,'CANNOT BE
    1EVALUATED, CARBON TERMINATED')
9010 FORMAT(1H ,4X,'***ERROR IN CARBON, THE TEMPERATURE',F10.2,'AND THE
    1PRESSURE',F10.4,'DEFINE A SATURATION STATE, CARBON TERMINATED')
9020 FORMAT(1H ,4X,'***ERROR IN CARBON, A QUALITY OF,'F8.2,'IS
    1MEANINGLESS')
9030 FORMAT(1H ,4X,'*** ERROR IN CARBON, A SATURATION STATE DOES NOT
```

```
     1EXIST FOR TEMPERATURES GREATER THAN THE CRITICAL TEMPERATURE')
 9040 FORMAT(1H ,4X,'***ERROR IN CARBON, A SATURATION STATE DOES NOT
     1EXIST FOR PRESSURES GREATER THAN THE CRITICAL PRESSURE')
 9000 FORMAT(1H ,4X,'*** ERROR IN CARBON, THE CYCLE',A6,'CANNOT BE
     1EVALUATED')
! Univac Statement      DEBUG SUBCHK
      END
C     ***************************************************************
C     *                                                             *
C     *                          S E T U P                          *
C     *                                                             *
C     *   THIS SUBROUTINE SETS UP THE REQUIRED COEFFICIENTS FOR THE *
C     *PARTICULAR FLUID TO BE ANALYZED.                             *
C     *                                                             *
C     ***************************************************************
      SUBROUTINE SETUP(GIVEN,PCRIT,TCRIT,DCRIT)
      COMMON/CAR1/C(11),CS(6),CH(6),WM,R,CONVR,LIMIT,ABST,TTOLER,BTOLER
      DIMENSION GIVEN(27)
      DO 100 I=1,11
      C(I)=GIVEN(I)
  100 CONTINUE
      DO 200 I=1,6
      CS(I)=GIVEN(I+11)
      CH(I)=GIVEN(I+17)
  200 CONTINUE
      PCRIT=GIVEN(24)
      TCRIT=GIVEN(25)
      DCRIT=GIVEN(26)
      WM=GIVEN(27)
      RETURN
! Univac Statement      DEBUG SUBCHK
      END
C     ***************************************************************
C     *                                                             *
C     *                          V F I N D                          *
C     *                                                             *
C     *   THIS SUBROUTINE FINDS THE DENSITY GIVEN TEMPERATURE AND   *
C     *PRESSURE FOR COMPRESSED LIQUID AND SUPERHEATED VAPOR STATES  *
C     *                                                             *
C     ***************************************************************
      SUBROUTINE VFIND(T,P,D)
      COMMON/CAR1/C(11),CS(6),CH(6),WM,R,CONVR,LIMIT,ABST,TTOLER,BTOLER
      PARAMETER M2=6
      DLAST=D
      CALL PRES(T,D,PCAL,DPDT,DPDD)
      DO 100 I=1,LIMIT
      D=DLAST-(PCAL-P)/DPDD
      IF(D.LE.0.0)D=DLAST/10.0
      CALL PRES(T,D,PCAL,DPDT,DPDD)
      IF (PCAL/P.GT.BTOLER.AND.PCAL/P.LT.TTOLER) RETURN
      DLAST=D
```

```
  100 CONTINUE
      WRITE(M2,9000) LIMIT
 9000 FORMAT(1H ,4X,'***VFIND FAILED TO CONVERGE IN',I6,'ITERATIONS')
      RETURN
! Univac Statement      DEBUG SUBCHK
      END
C     ****************************************************************
C     *                                                              *
C     *                           E N T H                            *
C     *                                                              *
C     *   THIS SUBROUTINE CALCULATES THE ENTHALPY GIVEN DENSITY AND   *
C     *TEMPERATURE                                                   *
C     *                                                              *
C     ****************************************************************
      SUBROUTINE ENTH(T,D,H,DHDT,DHDD)
      COMMON/CAR1/C(11),CS(6),CH(6),WM,R,CONVR,LIMIT,ABST,TTOLER,BTOLER
      TERM1=(C(4)*R*T-2.0*C(2)-4.0*C(6)/T**2+5.0*C(8)/T**3-6.0*C(9)
     1/T**4)
      TERM2=(2.0*C(3)*R*T-3.0*C(1)-4.0*C(7)/T)
      TERM3=C(10)*(6.0*C(1)+7.0*C(7)/T)
      TERM4=C(5)*(3.0-(3.0+0.5*C(11)*D**2-C(11)**2*D**4)*EXP(-C(11)*
     1D**2))/C(11)
      H=TERM1*D+TERM2*D**2/2.0+TERM3*D**5/5.0+TERM4/T**2
      DHDD=TERM1+TERM2*D+TERM3*D**4+C(5)/(C(11)*T**2)*(5.0*C(11)*D+5.0*
     1C(11)**2*D**3-2.0*C(11)**3*D**5)*EXP(-C(11)*D**2)
      TERM1=(C(4)*R+8.0*C(6)/T**3-15.0*C(8)/T**4+24.0*C(9)/T**5)*D
      TERM2=0.5*(2.0*C(3)*R+4.0*C(7)/T**2)*D**2
      TERM3=-C(10)*D**5*7.0*C(7)/(T**2*5.0)
      TERM4=-2.0*TERM4/T**3
      H=H*CONVR/WM+CH(1)+CH(2)*T+CH(3)*T**2+CH(4)*T**3+CH(5)*T**4+CH(6)
     1*T**5
      DHDD=DHDD*CONVR/WM
      DHDT=(TERM1+TERM2+TERM3+TERM4)*CONVR/WM+CH(2)+2.0*CH(3)*T+3.0
     1*CH(4)*T**2+CH(5)*T**3*4.0+5.0*CH(6)*T**4
      RETURN
! Univac Statement      DEBUG SUBCHK
      END
C     ****************************************************************
C     *                                                              *
C     *                           E N T R                            *
C     *                                                              *
C     *   THIS SUBROUTINE CALCULATES THE ENTROPY GIVEN DENSITY AND    *
C     *TEMPERATURE                                                   *
C     *                                                              *
C     ****************************************************************
      SUBROUTINE ENTR(T,D,S,DSDT,DSDD)
      COMMON/CAR1/C(11),CS(6),CH(6),WM,R,CONVR,LIMIT,ABST,TTOLER,BTOLER
      TERM1=C(4)*R+2.0*C(6)/T**3-3.0*C(8)/T**4+4.0*C(9)/T**5
      TERM2=C(3)*R+C(7)/T**2
      TERM3=1.0-(1.0+0.5*C(11)*D**2)*EXP(-C(11)*D**2)
      S=CONVR*(-R*LOG(D*R*T)-TERM1*D-0.5*TERM2*D**2
```

```
     1+C(10)*C(7)*D**5/(5.0*T**2)+2.0*C(5)*TERM3/(C(11)*T**3))/WM+CS(1)
     2+CS(2)*T+CS(3)*T**2+CS(4)*T**3+CS(5)*T**4+CS(6)*T**5
      DSDT=CONVR*(-R/T-(-6.0*C(6)/T**4+12.0*C(8)/T**5-20.0*C(9)/T**6)*D
     1+C(7)*D**2/T**3-0.4*C(10)*C(7)*D**5/T**3-6.0*C(5)*TERM3/
     2(C(11)*T**4))/WM+CS(2)+2.0*CS(3)*T+3.0*CS(4)*T**2+4.0*CS(5)*T**3
     3+5.0*CS(6)*T**4
      TERM1=-(C(4)*R+2.0*C(6)/T**3-3.0*C(8)/T**4+4.0*C(9)/T**5)
      TERM2=-(C(3)*R+C(7)/T**2)*D
      DSDD=(-R/D+TERM1+TERM2+C(10)*C(7)*D**4/T**2+2.0*C(5)*
     1(D+C(11)*D**3)*EXP(-C(11)*D**2)/T**3)*CONVR/WM
      RETURN
! Univac Statement     DEBUG SUBCHK
      END
C     ****************************************************************
C     *                                                              *
C     *                         P R I N T C                          *
C     *                                                              *
C     *  THIS SUBROUTINE PRINTS OUT THE RESULTS OF SUBROUTINE CARBON *
C     *                                                              *
C     ****************************************************************
      SUBROUTINE PRINTC(T,P,H,S,V,QUAL,HL,SL,VL,HV,SV,VV,LFTYPE,NPRIN)
      CHARACTER*6 LFTYPE
      PARAMETER M2=6
      COMMON/CAR1/C(11),CS(6),CH(6),WM,R,CONVR,LIMIT,ABST,TTOLER,BTOLER
      T=T-ABST
      IF(NPRIN.EQ.0) RETURN
      WRITE(M2,9000)LFTYPE
      WRITE(M2,9010)T,P
      IF(QUAL.LE.1.0.AND.QUAL.GE.0.0) GO TO 3280
      IF(QUAL.GT.1.0) WRITE(M2,9020)
      IF(QUAL.LT.0.0) WRITE(M2,9030)
      GO TO 3370
 3280 WRITE(M2,9040) QUAL
      WRITE(M2,9050)
      WRITE(M2,9070) HV,HL
      WRITE(M2,9080) SV,SL
      WRITE(M2,9060) VV,VL
 3370 WRITE(M2,9090)
      WRITE(M2,9070) H
      WRITE(M2,9080) S
      WRITE(M2,9060) V
 9000 FORMAT(1H1,4X,'***THERMODYNAMIC PROPERTIES OF ',A6,' USING
     1STARLING-BWR EQUATION OF STATE***')
 9010 FORMAT(1H ,4X,'TEMPERATURE = ',F7.1,'DEG.F',/,5X,'PRESSURE    = ',
     1    F10.5,'PSIA')
 9020 FORMAT(1H ,4X,'SUPERHEATED VAPOR',//////)
 9030 FORMAT(1H ,4X,'COMPRESSED LIQUID',//////)
 9040 FORMAT(1H ,4X,'SATURATED LIQUID-VAPOR EQUILIBRUIM',/,5X,'QUALITY =
     1 ',F6.4)
 9050 FORMAT(1H ,//,35X,'SATURATED',8X,'SATURATED',/,35X,'VAPOR',12X,
     1    'LIQUID',/,5X,60('#'))
```

```
 9060 FORMAT(1H ,4X,'SPECIFIC VOLUME (CU.FT/LBM)',5X,F9.4,8X,F9.4,/////
     1    /)
 9070 FORMAT(1H ,4X,'ENTHALPY',8X,'(BTU/LBM)',5X,F9.3,8X,F9.3)
 9080 FORMAT(1H ,4X,'ENTROPY',9X,'(BTU/LBM-R)',5X,F9.6,8X,F9.6)
 9090 FORMAT(1H ,//////)
      RETURN
! Univac Statement      DEBUG SUBCHK
      END
C     **************************************************************
C     *                                                            *
C     *                         P R E S A T                        *
C     *                                                            *
C     *   THIS SUBROUTINE FINDS THE SATURATION PRESSURE AND SATURATED *
C     *LIQUID AND VAPOR DENSITIES GIVEN TEMPERATURE                 *
C     *                                                            *
C     **************************************************************
      SUBROUTINE PRESAT(T,P,DL,DV,DCRIT,TCRITR,FTYPE)
      CHARACTER*6 FTYPE
      COMMON/CAR1/C(11),CS(6),CH(6),WM,R,CONVR,LIMIT,ABST,TTOLER,BTOLER
      PARAMETER M2=6
      TRED=T/TCRITR
      DL=DCRIT*(2.0+2.5*(1.0-TRED))
      IF(FTYPE.NE.'PROP  ') GO TO 10
      DL=DCRIT*(1.8+3.4*(1.0-TRED)-2.7*(1.0-TRED)**2.0)
      DV=DCRIT/(275.0*(1.0-TRED))
      GO TO 200
   10 IF(FTYPE.NE.'ISOP  ') GO TO 20
      DL=DCRIT*(1.8+3.4*(1.0-TRED)-2.7*(1.0-TRED)**2.0)
      DV=DCRIT/(225.0*(1.0-TRED))
      GO TO 200
   20 IF(FTYPE.NE.'PENT  ') GO TO 30
      DL=DCRIT*(1.8+3.4*(1.0-TRED)-2.7*(1.0-TRED)**2.0)
      DV=DCRIT/(300.0*(1.0-TRED)-6.0)
      GO TO 200
   30 DV=DCRIT/(150.0*(1.0-TRED))
  200 CALL FUGAC(T,DL,FL,DFLDT,DFLDDL)
      CALL FUGAC(T,DV,FV,DFVDT,DFVDDV)
      CALL PRES(T,DL,PL,DPLDT,DPLDDL)
      CALL PRES(T,DV,PV,DPVDT,DPVDDV)
      DVLAST=DV
      DLLAST=DL
      DO 1000 I=1,LIMIT
      FP=PL-PV
      FF=FL-FV
      DFPDDV=-DPVDDV
      DFPDDL=DPLDDL
      DFFDDV=-DFVDDV
      DFFDDL=DFLDDL
      XJACOB=DFPDDV*DFFDDL-DFPDDL*DFFDDV
      DV=DVLAST-(FP*DFFDDL-FF*DFPDDL)/XJACOB
      DL=DLLAST-(FF*DFPDDV-FP*DFFDDV)/XJACOB
```

```
      IF(DV.LE.0.0)DV=DVLAST/10.0
      IF(DL.LE.0.0)DL=DLLAST/10.0
      IF(DV.GE.DCRIT) DV=DCRIT*0.9
      IF(DL.LE.DCRIT) DL=DCRIT*1.1
      IF(DL.GT.1.0) DL=DLLAST*1.01
      CALL FUGAC(T,DL,FL,DFLDT,DFLDDL)
      CALL FUGAC(T,DV,FV,DFVDT,DFVDDV)
      CALL PRES(T,DL,PL,DPLDT,DPLDDL)
      CALL PRES(T,DV,PV,DPVDT,DPVDDV)
      P=(PV+PL)/2.0
      IF(FL/FV.LT.TTOLER.AND.FL/FV.GT.BTOLER.AND.PL/PV.LT.TTOLER.
     1AND.PL/PV.GT.BTOLER) RETURN
      DLLAST=DL
      DVLAST=DV
 1000 CONTINUE
C     WRITE(M2,9000)LIMIT
C9000 FORMAT(1H ,4X,'***PRESAT FAILED TO CONVERGE IN',I6,'ITERATIONS')
      RETURN
! Univac Statement      DEBUG SUBCHK
      END
C     *****************************************************************
C     *                                                               *
C     *                           S A T C                             *
C     *                                                               *
C     *   THIS SUBROUTINE DETERMINES SATURATED LIQUID AND VAPOR        *
C     *PROPERTIES                                                     *
C     *                                                               *
C     *****************************************************************
      SUBROUTINE SATC(T,DL,DV,HL,SL,HV,SV)
      CALL ENTH(T,DL,HL,DHLDT,DHLDD)
      CALL ENTR(T,DL,SL,DSLDT,DSLDD)
      CALL ENTH(T,DV,HV,DHVDT,DHVDD)
      CALL ENTR(T,DV,SV,DSVDT,DSVDD)
      RETURN
! Univac Statement      DEBUG SUBCHK
      END
C     *****************************************************************
C     *                                                               *
C     *                         T E M S A T                           *
C     *                                                               *
C     *   THIS SUBROUTINE FINDS THE SATURATION TEMPERATURE AND        *
C     *SATURATED LIQUID AND VAPOR DENSITIES GIVEN PRESSURE            *
C     *                                                               *
C     *****************************************************************
      SUBROUTINE TEMSAT(P,T,DL,DV,TCRIT,DCRIT)
      COMMON/CAR1/C(11),CS(6),CH(6),WM,R,CONVR,LIMIT,ABST,TTOLER,BTOLER
      DIMENSION XJACOB(3,3),F(3),DX(3)
      PARAMETER M2=6
      TCRITR=TCRIT+ABST
      T=640.0
      DV=DCRIT/20.0
```

```
      DL=DCRIT*2.5
      TLAST=T
      DVLAST=DV
      DLLAST=DL
      CALL PRES(T,DL,PL,DPLDT,DPLDDL)
      CALL PRES(T,DV,PV,DPVDT,DPVDDV)
      CALL FUGAC(T,DL,FL,DFLDT,DFLDDL)
      CALL FUGAC(T,DV,FV,DFVDT,DFVDDV)
      DO 1000 I=1,LIMIT
      F(1)=PL-P
      F(2)=PV-P
      F(3)=FV-FL
      XJACOB(1,1)=DPLDDL
      XJACOB(1,2)=0.0
      XJACOB(1,3)=DPLDT
      XJACOB(2,1)=0.0
      XJACOB(2,2)=DPVDDV
      XJACOB(2,3)=DPVDT
      XJACOB(3,1)=-DFLDDL
      XJACOB(3,2)=DFVDDV
      XJACOB(3,3)=DFVDT-DFLDT
      CALL INVERT(XJACOB,F,DX)
      DL=DLLAST-DX(1)
      DV=DVLAST-DX(2)
      T=TLAST-DX(3)
  750 IF(T.LE.0.0) T=TLAST/10.0
      IF(DL.LE.0.0) DL=DLLAST/10.0
      IF(DV.LE.0.0) DV=DVLAST/10.0
      IF(DV.GE.0.55) DV=DVLAST*2.0
      IF(DL.GE.1.0) DL=DLLAST*2.0
      IF(T.GT.TCRITR) T=TCRITR
      CALL PRES(T,DL,PL,DPLDT,DPLDDL)
      CALL PRES(T,DV,PV,DPVDT,DPVDDV)
      CALL FUGAC(T,DL,FL,DFLDT,DFLDDL)
      CALL FUGAC(T,DV,FV,DFVDT,DFVDDV)
      IF(PL/P.LT.TTOLER.AND.PL/P.GT.BTOLER.AND.PV/P.LT.TTOLER.
     1AND.PV/P.GT.BTOLER.AND.FL/FV.LT.TTOLER.AND.FL/FV.GT.BTOLER) RETURN
      DLLAST=DL
      DVLAST=DV
      TLAST=T
 1000 CONTINUE
C     WRITE(M2,9000)LIMIT
C9000 FORMAT(1H ,4X,'***TEMPSAT FAILED TO CONVERGE IN',I6,'ITERATIONS')
      RETURN
! Univac Statement      DEBUG SUBCHK
      END
C     *****************************************************************
C     *                                                               *
C     *                        P S K N O W                            *
C     *                                                               *
C     *   THIS SUBROUTINE FINDS THE DENSITY AND TEMPERATURE GIVEN     *
```

```
C     *PRESSURE AND ENTROPY FOR COMPRESSED LIQUID AND SUPERHEATED VAPOR*
C     *STATES                                                           *
C     *                                                                 *
C     *******************************************************************
      SUBROUTINE  PSKNOW(P,S,T,D)
      COMMON/CAR1/C(11),CS(6),CH(6),WM,R,CONVR,LIMIT,ABST,TTOLER,BTOLER
      PARAMETER M2=6
      DLAST=D
      TLAST=T
      CALL PRES(T,D,PCAL,DPDT,DPDD)
      CALL ENTR(T,D,SCAL,DSDT,DSDD)
      DO 1000 I=1,LIMIT
      XJACOB=DPDD*DSDT-DPDT*DSDD
      D=DLAST-((PCAL-P)*DSDT-(SCAL-S)*DPDT)/XJACOB
      T=TLAST-((SCAL-S)*DPDD-(PCAL-P)*DSDD)/XJACOB
      IF(T.LE.0.0)T=TLAST/10.0
      IF(D.LE.0.0) D=DLAST/10.0
      CALL PRES(T,D,PCAL,DPDT,DPDD)
      CALL ENTR(T,D,SCAL,DSDT,DSDD)
      IF(PCAL/P.LT.TTOLER.AND.PCAL/P.GT.BTOLER.AND.SCAL/S.LT.TTOLER.
     1AND.SCAL/S.GT.BTOLER)RETURN
      DLAST=D
      TLAST=T
 1000 CONTINUE
      WRITE(M2,9000)LIMIT
 9000 FORMAT(1H ,4X,'***PSKNOW FAILED TO CONVERGE IN',I6,'ITERATIONS')
      RETURN
! Univac Statement      DEBUG SUBCHK
      END
C     *******************************************************************
C     *                                                                 *
C     *                          P H K N O W                            *
C     *                                                                 *
C     *   THIS SUBROUTINE FINDS THE DENSITY AND TEMPERATURE GIVEN       *
C     *PRESSURE ANS ENTHALPY FOR COMPRESSED LIQUID AND SUPERHEATED VAPO*
C     *STATES                                                           *
C     *                                                                 *
C     *******************************************************************
      SUBROUTINE  PHKNOW(P,H,T,D)
      COMMON/CAR1/C(11),CS(6),CH(6),WM,R,CONVR,LIMIT,ABST,TTOLER,BTOLER
      PARAMETER M2=6
      TT=1.0005
      BT=.9995
      DLAST=D
      TLAST=T
      CALL PRES(T,D,PCAL,DPDT,DPDD)
      CALL ENTH(T,D,HCAL,DHDT,DHDD)
      DO 1000 I=1,100
      XJACOB=DPDD*DHDT-DPDT*DHDD
      D=DLAST-((PCAL-P)*DHDT-(HCAL-H)*DPDT)/XJACOB
      T=TLAST-((HCAL-H)*DPDD-(PCAL-P)*DHDD)/XJACOB
```

```
      IF(D.LE.0.0) D=DLAST/10.0
      IF(T.LE.0.0)T=TLAST/10.0
      CALL PRES(T,D,PCAL,DPDT,DPDD)
      CALL ENTH(T,D,HCAL,DHDT,DHDD)
      IF(PCAL/P.LT.TT.AND.PCAL/P.GT.BT.AND.HCAL/H.LT.TT.
     1AND.HCAL/H.GT.BT)RETURN
      DLAST=D
      TLAST=T
 1000 CONTINUE
      WRITE(M2,9000)LIMIT
 9000 FORMAT(1H ,4X,'***PHKNOW FAILED TO CONVERGE IN',I6,'ITERATIONS')
      RETURN
! Univac Statement      DEBUG SUBCHK
      END
C     ****************************************************************
C     *                                                              *
C     *                          F U G A C                           *
C     *                                                              *
C     *   THIS SUBROUTINE CALCULATES THE FUGACITY GIVEN DENSITY AND  *
C     *TEMPERATURE                                                   *
C     *                                                              *
C     ****************************************************************
      SUBROUTINE FUGAC(T,D,F,DFDT,DFDD)
      COMMON/CAR1/C(11),CS(6),CH(6),WM,R,CONVR,LIMIT,ABST,TTOLER,BTOLER
      TERM1=C(4)*R*T-C(2)-C(6)/T**2+C(8)/T**3-C(9)/T**4
      TERM2=C(3)*R*T-C(1)-C(7)/T
      TERM3=C(1)+C(7)/T
      TERM4=1.0-(1.0-0.5*C(11)*D**2-C(11)**2*D**4)*EXP(-C(11)*D**2)
      F=LOG(D*R*T)+2.0*TERM1*D/(R*T)+3.0*D**2*TERM2/(2.0*R*T)+6.0*C(10)
     1*TERM3*D**5/(5.0*R*T)+C(5)*TERM4/(C(11)*R*T**3)
      F=EXP(F)
      DFDD=F*(1.0/D+2.0*TERM1/(R*T)+3.0*TERM2*D/(R*T)+6.0*C(10)*D**4
     1*TERM3/(R*T)+C(5)/(R*C(11)*T**3)*(3.0*C(11)*D+3.0*C(11)**2*D**3
     2-2.0*D**5*C(11)**3)*EXP(-C(11)*D**2))
      TERM1=C(2)+3.0*C(6)/T**2-4.0*C(8)/T**3+5.0*C(9)/T**4
      TERM2=C(1)+2.0*C(7)/T
      DFDT=F*(1.0/T+2.0*TERM1*D/(T**2*R)+3.0*TERM2*D**2/(2.0*R*T**2)
     1-6.0*C(10)*TERM2*D**5/(5.0*R*T**2)-3.0*C(5)*TERM4/(C(11)*R*T**4))
      RETURN
! Univac Statement      DEBUG SUBCHK
      END
C     ****************************************************************
C     *                                                              *
C     *                           P R E S                            *
C     *                                                              *
C     *   THIS SUBROUTINE CALCULATES THE PRESSURE GIVEN DENSITY AND  *
C     *TEMPERATURE                                                   *
C     *                                                              *
C     ****************************************************************
      SUBROUTINE PRES(T,D,P,DPDT,DPDD)
      COMMON/CAR1/C(11),CS(6),CH(6),WM,R,CONVR,LIMIT,ABST,TTOLER,BTOLER
```

```
      TERM1=C(4)*R*T-C(2)-C(6)/T**2+C(8)/T**3-C(9)/T**4
      TERM2=C(3)*R*T-C(1)-C(7)/T
      TERM3=C(1)+C(7)/T
      TERM4=1.0+C(11)*D**2
      P=D*R*T+TERM1*D**2+TERM2*D**3+C(10)*TERM3*D**6+C(5)*D**3*TERM4*
     1EXP(-C(11)*D**2)/T**2
      DPDD=R*T+2.0*TERM1*D+3.0*TERM2*D**2+6.0*C(10)*TERM3*D**5
     1+3.00*C(5)*D**2*TERM4*EXP(-C(11)*D**2)/T**2-2.0*C(11)**2*D**6
     2*C(5)*EXP(-C(11)*D**2)/T**2
      DPDT=D*R+D**2*(C(4)*R+2.0*C(6)/T**3-3.0*C(8)/T**4+4.0+C(9)/T**5)
     1+D**3*(C(3)*R+C(7)/T**2)-D**6*C(10)*C(7)/T**2-2.0*C(5)*D**3*TERM4*
     2EXP(-C(11)*D**2)/T**3
      RETURN
! Univac Statement      DEBUG SUBCHK
      END
C     ******************************************************************
C     *                                                                *
C     *                          I N V E R T                           *
C     *                                                                *
C     *   THIS SUBROUTINE SOLVES THREE SIMULTANEOUS EQUATIONS BY       *
C     *GUASSIAN ELIMINATION                                            *
C     *                                                                *
C     ******************************************************************
      SUBROUTINE INVERT(XJACOB,F,DX)
      DIMENSION XJACOB(3,3),DX(3),F(3),W(3,4)
      DO 5 I=1,3
      DO 5 J=1,4
      IF(J-4)2,3,5
    2 W(I,J)=XJACOB(I,J)
      GO TO 5
    3 W(I,J)=F(I)
    5 CONTINUE
      N=0
      DO 100 K=1,2
      N=N+1
      DO 95 L=2,3
      IF(K-L)10,90,90
   10 C=-W(L,N)/W(K,N)
      DO 80 M=1,4
      W(L,M)=W(L,M)+C*W(K,M)
   80 CONTINUE
   90 CONTINUE
   95 CONTINUE
  100 CONTINUE
      DX(3)=W(3,4)/W(3,3)
      DX(2)=(W(2,4)-DX(3)*W(2,3))/W(2,2)
      DX(1)=(W(1,4)-DX(3)*W(1,3)-DX(2)*W(1,2))/W(1,1)
      RETURN
! Univac Statement      DEBUG SUBCHK
      END
C     ******************************************************************
```

```
C     *                                                                *
C     *                          F R E O N                             *
C     *                                                                *
C     *   THIS SUBROUTINE CALCULATES THE THERMODYNAMIC PROPERTIES OF   *
C     *FREONS BY USING THE MARTIN-HOU EQUATION OF STATE. THE TYPES OF  *
C     *STATES WHICH CAN BE EVALUATED ARE, GIVEN THE FOLLOWING PAIRS OF *
C     *KNOWNS:                                                         *
C     *              1) TEMPERATURE-PRESSURE                           *
C     *              2) TEMPERATURE-QUALITY                            *
C     *              3) PRESSURE-ENTROPY                               *
C     *              4) PRESSURE-ENTHALPY                              *
C     *              5) PRESSURE-QUALITY                               *
C     *              6) TEMPERATURE-SPECIFIC VOLUME                    *
C     *   COMPRESSED LIQUID STATES ARE APPROXIMATED BY SATURATED       *
C     *LIQUID STATES.                                                  *
C     *THIS PROGRAM WAS ADAPTED FROM A PROGRAM BY DAVID H. RIEMER.     *
C     ******************************************************************
      SUBROUTINE FREON(CYCLE,FGIVEN,SGIVEN,FTYPE,H,S,V,T,PSIA,
     1   HV,SV,VV,HL,SL,VL,QUAL,NPRT)
      CHARACTER FTYPE*6,CYCLE*2
      PARAMETER M2=6
      DIMENSION F113(4,12),F114(4,12),F11(4,12),F12(4,12),F13(4,12),
     1F14(4,12),F21(4,12),F22(4,12),F23(4,12)
      COMMON/FREON1/Q(4,12),TTOLER,BTOLER,LIMIT,DERROR,NTYPE
      DATA LIMIT,DERROR,TTOLER,BTOLER/50,1.0E-10,1.005,0.995/
      DATA ((F113(I,J),I=1,4),J=1,12)/33.0655,-4330.98,-9.2635,
     1   0.0020539,0.0,0.0,0.0,0.0,0.0,0.05728,0.0,417.4,122.872,
     2   -4.035,0.002618,0.0,-0.0128,-0.0214,5.0E-05,0.0,-6.36E-05,0.0,
     3   0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.07963,1.159E-04,0.0,
     4   0.0,0.0,25.198,-0.40552,0.1699,459.6,109.49,3.0,0.02781,0.0,
     5   0.0,0.0,498.9/
      DATA((F114(I,J),I=1,4),J=1,12)/27.071306,-5113.7021,-6.3086761,
     1   6.913003E-04,0.78142111002,768.35,0.0,0.0,0.0,0.062780807,
     2   5.914907E-03,294.35,36.32,-2.3856704,1.0801207E-03,-6.5643648,
     3   0.021776,0.034055687,-5.3336494E-06,0.16366057,0.63649,
     4   -3.857481E-04,0.0,0.0,6.7186,1.6017659E-06,6.2632341E-10,
     5   -1.0165314E-05,1.97E-06,0.0,0.0,0.0,0.0175,3.49E-04,-1.67E-07,
     6   0.0,0.0,25.33966211,-0.1151371756,0.15842,459.6,95.4,2.0,
     7   0.027533,0.0,3.0,0.0,473.187/
      DATA((F11(I,J),I=1,4),J=1,12)/42.14702865,-4344.343807,
     1-12.84596753,0.004008372507,0.0313605356,862.07,0.0,0.0,0.0,
     20.078117,0.00190,388.47,34.57,-3.126759,1.318523E-03,-35.769990,
     357.63811,-0.025341,4.8751212E-05,1.220367,43.63220,1.687277E-03,
     4-1.805062E-06,0.0,-42.82356,-2.358930E-05,2.448303E-08,
     5-1.478379E-04,36.70663,1.057504E08,-9.472103E04,0.0,0.023815,
     62.798823E-04,-2.123734E-07,5.999018E-11,-336.807030,50.5418,
     7-0.0918395,0.17219,459.6,112.080,1.0,0.028927,0.0,4.50,580.00,
     8639.50/
      DATA((F12(I,J),I=1,4),J=1,12)/39.88381727,-3436.632228,
     1-12.47152228,0.004730442442,0.0,0.0,0.0,0.0,0.0,0.088734,
     20.0065093886,233.70,34.84,-3.409727134,0.00159434848,-56.7627671,
```

```
30.02696,0.06023944654,-1.879618431E-05,1.311399084,0.834921,
4-0.000548737007,0.0,0.0,6.02683,0.0,3.468834E-09,-2.54390678E-05,
5-0.655549E-05,0.0,0.0,0.0,0.0080945,0.000332662,-2.413896E-07,
66.72363E-11,0.0,39.55655122,-0.0165379361,0.1359,459.6,78.86,2.0,
70.02870,0.0,5.475,0.0,596.9/
 DATA((F13(I,J),I=1,4),J=1,12)/25.96797498,-2709.538217,
1-7.172343913,0.00254515398,0.2803010913,546.00,0.0,0.0,0.0,
20.102728,0.0048,84.00,36.06996128,-3.083417,2.341695E-03,
3-18.212643,0.01566,0.058854,-5.671268E-05,0.571958,1.110,
4-1.026061E-03,1.338679E-06,0.0,6.665,5.290649E-06,-7.395111E-9,
5-3.874233E-05,3.245E-05,7.378601E07,-7.435565E04,0.0,0.01602,
62.823E-04,-1.159E-07,0.0,0.0,20.911,-0.05676,0.08898,459.6,
745.271,2.0,0.0277239,0.0,4.00,625.00,561.30/
 DATA((F14(I,J),I=1,4),J=1,12)/20.71545389,-2467.505285,
1-4.69017025,0.00064798076,0.770707795,424.0,0.0,0.0,0.0,0.1219336,
20.0015,-50.1,39.06,-2.162959,2.135114E-03,-18.941131,69.56848907,
34.404057E-03,1.282818E-05,0.539776,4.58661139,1.921072E-04,
4-3.918263E-07,0.0,36.17166615,-4.481049E-06,9.062318E-09,
5-4.836678E-05,-8.05898583,5.838823E07,-9.263923E04,0.0,
63.00559282E-02,2.37043352E-04,-2.85660077E-08,-2.95338806E-11,0.0,
786.102162,0.36172528,0.5284,459.6,100.636,1.0,0.02560163,0.0,4.00,
8661.199997,543.16/
 DATA((F21(I,J),I=1,4),J=1,12)/42.7908,-4261.34,-13.0295,0.0039851,
10.0,0.0,0.0,0.0,0.0,0.10427,0.0,353.2,116.37962,-7.316,0.00464210,
20.0,-0.03106808,-0.20382376,0.0003593,0.0,-0.0000501,0.0,0.0,0.0,
30.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0427,0.000140,0.0,0.0,0.0,0.0,
40.0,0.1906,459.6,120.45,3.0,0.030675,0.0,0.0,0.0,750.0/
 DATA((F22(I,J),I=1,4),J=1,12)/29.35754453,-3845.193152,
1-7.86103122,0.002190939044,0.445746703,686.1,0.0,0.0,0.0,0.124098,
20.002,204.81,32.76,-4.353547,0.002407252,-44.066868,54.6344093,
3-0.017464,7.62789E-05,1.483763,36.74892,0.002310142,-3.605723E-06,
40.0,-22.2925657,-3.724044E-05,5.355465E-08,-1.845051E-04,
520.47328862,1.363387E08,-1.672612E05,0.0,0.02812836,2.255408E-04,
6-6.509607E-08,0.0,257.341,62.4009,-0.0453335,0.16016,459.69,
791.329,1.0,0.030525,0.0,4.2,548.2,721.906/
 DATA((F23(I,J),I=1,4),J=1,12)/328.90853,-7952.76913,-144.5142304,
10.2421150182,0.0,0.0,0.0,-2.128066524E-04,9.43495542E-08,0.153270,
20.00125,78.73,32.7758,-4.679499,3.472778E-03,-159.775232,63.37784,
3-0.012475,7.733388E-05,5.941212,-25.30533,2.068042E-03,
4-3.684238E-06,0.0,144.16182,-3.868546E-05,6.455643E-08,
5-7.394214E-04,-106.13280,7.502357E07,-1.114202E05,0.0,0.07628087,
6-7.561805E-06,3.9065696E-07,-2.454905E-10,0.0,0.0,0.0,0.1198,
7459.6,60.77,1.0,0.030510,0.0,5.50,520.00,701.42/
 RJ=0.185053
 H=0.0
 S=0.0
 V=0.0
 T=0.0
 PSIA=0.0
 HV=0.0
 SV=0.0
```

```
      VV=0.0
      HL=0.0
      SL=0.0
      VL=0.0
      QUAL=0.0
      IF(FTYPE.EQ.'F113  ') GO TO 10
      IF(FTYPE.EQ.'F114  ') GO TO 20
      IF(FTYPE.EQ.'F11   ')GO TO 30
      IF(FTYPE.EQ.'F12   ')GO TO 40
      IF(FTYPE.EQ.'F13   ')GO TO 50
      IF(FTYPE.EQ.'F14   ')GO TO 60
      IF(FTYPE.EQ.'F21   ')GO TO 70
      IF(FTYPE.EQ.'F22   ')GO TO 80
      IF(FTYPE.EQ.'F23   ')GO TO 90
      WRITE(M2,9060) FTYPE
      RETURN
   10 DO 15 I=1,4
      DO 15 J=1,12
      Q(I,J)=F113(I,J)
   15 CONTINUE
      GO TO 350
   20 DO 25 I=1,4
      DO 25 J=1,12
      Q(I,J)=F114(I,J)
   25 CONTINUE
      GO TO 350
   30 DO 35 I=1,4
      DO 35 J=1,12
      Q(I,J)=F11(I,J)
   35 CONTINUE
      GO TO 350
   40 DO 45 I=1,4
      DO 45 J=1,12
      Q(I,J)=F12(I,J)
   45 CONTINUE
      GO TO 350
   50 DO 55 I=1,4
      DO 55 J=1,12
      Q(I,J)=F13(I,J)
   55 CONTINUE
      GO TO 350
   60 DO 65 I=1,4
      DO 65 J=1,12
      Q(I,J)=F14(I,J)
   65 CONTINUE
      GO TO 350
   70 DO 75 I=1,4
      DO 75 J=1,12
      Q(I,J)=F21(I,J)
   75 CONTINUE
      GO TO 350
```

```
 80 DO 85 I=1,4
    DO 85 J=1,12
    Q(I,J)=F22(I,J)
 85 CONTINUE
    GO TO 350
 90 DO 95 I=1,4
    DO 95 J=1,12
    Q(I,J)=F23(I,J)
 95 CONTINUE
350 NTYPE=Q(3,11)
    HCRIT=Q(2,11)
    SCRIT=Q(4,10)
    PCRIT=Q(4,12)
    TCRIT=Q(4,3)
    IF(CYCLE.EQ.'TP') GO TO 400
    IF(CYCLE.EQ.'TX') GO TO 600
    IF(CYCLE.EQ.'PX') GO TO 700
    IF(CYCLE.EQ.'PS') GO TO 900
    IF(CYCLE.EQ.'PH') GO TO 1000
    IF(CYCLE.EQ.'TV') GO TO 1100
    WRITE(M2,9000)
    RETURN
400 TF=FGIVEN
    PSIA=SGIVEN
    T=TF+Q(1,11)
    IF(TF.LT.Q(4,3)) GO TO 430
    V=1.0/Q(1,4)
    GO TO 480
430 CALL VPEQM(PSAT,T,1)
    IF(PSIA/PSAT.GT.TTOLER.OR.PSIA/PSAT.LT.BTOLER) GO TO 470
    WRITE(M2,9010)
    RETURN
470 IF(PSIA/PSAT.GT.TTOLER) GO TO 500
480 QUAL=2.0
    CALL VOLUME(T,PSIA,V)
    CALL ENTHAL(PSIA,T,V,H,DHDT,DHDV)
    CALL ENTROP(T,V,S,DSDT,DSDV)
    CALL PRINTF(T,PSIA,QUAL,V,H,S,0.0,0.0,0.0,0.0,0.0,0.0,FTYPE,NPRT)
    RETURN
500 CALL VOLUME(T,PSAT,V)
    CALL DRVPT(DPSDT,PSAT,T,DPSDT2)
    CALL ENTHAL(PSAT,T,V,H,DHDT,DHDV)
    CALL ENTROP(T,V,S,DSDT,DSDV)
    CALL FLQDEN(VL,T,DVLDT)
    DELTH=RJ*DPSDT*(V-VL)*T
    H=H-DELTH
    S=S-DELTH/T
    QUAL=-2.0
    CALL PRINTF(T,PSIA,QUAL,VL,H,S,0.0,0.0,0.0,0.0,0.0,0.0,FTYPE,NPRT)
    V=VL
    RETURN
```

```
600 TF=FGIVEN
    QUAL=SGIVEN
    IF(QUAL.LE.1.0.AND.QUAL.GE.0.0) GO TO 610
    WRITE(M2,9070) QUAL
    RETURN
610 T=TF+Q(1,11)
    IF(TF.LE.Q(4,3)) GO TO 630
    WRITE(M2,9020)TCRIT
    RETURN
630 CALL VPEQM(PSIA,T,1)
    GO TO 800
700 PSIA=FGIVEN
    QUAL=SGIVEN
    IF(QUAL.LE.1.0.AND.QUAL.GE.0.0) GO TO 710
    WRITE(M2,9070) QUAL
    RETURN
710 IF(PSIA.LE.Q(4,12)) GO TO 730
    WRITE( M2,9030) PCRIT
    RETURN
730 CALL VPEQM(PSIA,T,2)
    TF=T-Q(1,11)
800 IF(TF-Q(4,3)) 830,810,810
810 V=1.0/Q(4,11)
    GO TO 840
830 CALL VOLUME(T,PSIA,VV)
840 CALL SATN(PSIA,T,VV,HV,SV,HL,SL,VL)
    H=(1.0-QUAL)*HL+QUAL*HV
    S=(1.0-QUAL)*SL+QUAL*SV
    V=(1.0-QUAL)*VL+QUAL*VV
    CALL PRINTF( T,PSIA,QUAL,V,H,S,VV,HV,SV,VL,HL,SL,FTYPE,NPRT)
    RETURN
900 PSIA=FGIVEN
    S=SGIVEN
    IF(PSIA.LT.Q(4,12)) GO TO 910
    IF(S.LE.SCRIT) GO TO 905
    T=Q(4,3)+Q(1,11)
    V=1.0/Q(4,11)
    GO TO 940
905 WRITE(M2,9040)
    RETURN
910 CALL VPEQM(PSIA,T,2)
    CALL VOLUME(T,PSIA,V)
    VV=V
    CALL SATN(PSIA,T,VV,HV,SV,HL,SL,VL)
    IF(S/SV.GT.TTOLER) GO TO 940
    IF(S/SL.LT.BTOLER) GO TO 960
    QUAL=(S-SL)/(SV-SL)
    H=(1.0-QUAL)*HL+QUAL*HV
    V=(1.0-QUAL)*VL+QUAL*VV
    CALL PRINTF( T,PSIA,QUAL,V,H,S,VV,HV,SV,VL,HL,SL,FTYPE,NPRT)
    RETURN
```

```
 940 CALL SFIND(T,PSIA,V,S)
     CALL ENTHAL(PSIA,T,V,H,DHDT,DHDV)
     QUAL=2.0
     CALL PRINTF(T,PSIA,QUAL,V,H,S,0.0,0.0,0.0,0.0,0.0,0.0,FTYPE,NPRT)
     RETURN
 960 P=PSIA
     CALL COMPS(P,T,V,S,VL)
     CALL ENTHAL(P,T,V,H,DHDT,DHDV)
     CALL DRVPT(DPSDT,P,T,DPSDT2)
     DELTH=RJ*DPSDT*(V-VL)*T
     H=H-DELTH
     QUAL=-2.0
     CALL PRINTF(T,PSIA,QUAL,VL,H,S,0.0,0.0,0.0,0.0,0.0,0.0,FTYPE,NPRT)
     V=VL
     RETURN
1000 PSIA=FGIVEN
     H=SGIVEN
     IF(PSIA.LT.Q(4,12)) GO TO 1020
     IF(H.GT.HCRIT) GO TO 1010
     WRITE(M2,9050)
     RETURN
1010 T=Q(1,11)+Q(4,3)+10.0
     V=1.0/Q(4,11)
     GO TO 1040
1020 CALL VPEQM(PSIA,T,2)
     CALL VOLUME(T,PSIA,V)
     VV=V
     CALL SATN(PSIA,T,VV,HV,SV,HL,SL,VL)
     IF(H/HV.GT.TTOLER) GO TO 1040
     IF(H/HL.LT.BTOLER) GO TO 1060
     QUAL=(H-HL)/(HV-HL)
     S=(1.0-QUAL)*SL+QUAL*SV
     V=(1.0-QUAL)*VL+QUAL*VV
     CALL PRINTF( T,PSIA,QUAL,V,H,S,VV,HV,SV,VL,HL,SL,FTYPE,NPRT)
     RETURN
1040 CALL HFIND(T,PSIA,V,H)
     CALL ENTROP(T,V,S,DSDT,DSDV)
     QUAL=2.0
     CALL PRINTF(T,PSIA,QUAL,V,H,S,0.0,0.0,0.0,0.0,0.0,0.0,FTYPE,NPRT)
     RETURN
1060 P=PSIA
     CALL COMPH(P,T,V,H,VL)
     CALL ENTROP(T,V,S,DSDT,DSDV)
     CALL DRVPT(DPSDT,P,T,DPSDT2)
     DELTH=RJ*DPSDT*(V-VL)
     S=S-DELTH
     QUAL=-2.0
     CALL PRINTF(T,PSIA,QUAL,VL,H,S,0.0,0.0,0.0,0.0,0.0,0.0,FTYPE,NPRT)
     V=VL
     RETURN
1100 TF=FGIVEN
```

```
      T=TF+Q(1,11)
      V=SGIVEN
      IF(T.GE.TCRIT) GO TO 1140
      CALL VPEQM(PSAT,T,1)
      CALL FLQDEN(VL,T,DVLDT)
      CALL VOLUME(T,PSAT,VV)
      IF(V/VL.LT.BTOLER) GO TO 1160
      IF(V/VV.GT.TTOLER) GO TO 1140
      CALL SATN(PSAT,T,VV,HV,SV,HL,SL,VL)
      QUAL=(V-VL)/(VV-VL)
      H=(1.0-QUAL)*HL+QUAL*HV
      S=(1.0-QUAL)*SL+QUAL*SV
      CALL PRINTF(T,PSAT,QUAL,V,H,S,VV,HV,SV,VL,HL,SL,FTYPE,NPRT)
      PSIA=PSAT
      RETURN
 1140 CALL EOS(PCAL,T,V,DPDT,DPDV,DPDVDT,D2PDT2)
      CALL ENTHAL(PCAL,T,V,H,DHDT,DHDV)
      CALL ENTROP(T,V,S,DSDT,DSDV)
      QUAL=2.0
      CALL PRINTF(T,PCAL,QUAL,V,H,S,0.0,0.0,0.0,0.0,0.0,0.0,FTYPE,NPRT)
      PSIA=PCAL
      RETURN
 1160 CALL VPEQM(PSIA,T,1)
      CALL VOLUME(T,PSIA,VV)
      CALL DRVPT(DPSDT,PSIA,T,DPSDT2)
      CALL ENTHAL(PSIA,T,VV,H,DHDT,DHDV)
      CALL ENTROP(T,VV,S,DSDT,DSDV)
      QUAL=-2.0
      DELTH=RJ*DPSDT*(VV-V)
      S=S-DELTH
      H=H-DELTH*T
      CALL PRINTF(T,PSIA,QUAL,V,H,S,0.0,0.0,0.0,0.0,0.0,0.0,FTYPE,NPRT)
      RETURN
 9000 FORMAT(1H ,4X,'***ERROR*** CYCLE = ',A3,' CYCLE MUST EQUAL TP, TX,
     1 PX, PS, PH',//,5X,'FREON TERMINATED')
 9010 FORMAT(1H ,4X,'***ERROR*** TEMPERATURE AND PRESSURE ARE NOT INDEPE
     1NDENT UNDER THE SATURATION DOME',//,5X,'FREON TERMINATED')
 9020 FORMAT(1H ,4X,'***ERROR*** A SATURATION STATE DOES NOT EXIST FOR T
     1EMPERATURES ABOVE THE CRITICAL TEMPERATURE',F10.3,'DEG.F',//,5X,
     2    'FREON TERMINATED')
 9030 FORMAT(1H ,4X,'***ERROR*** A SATURATION STATE DOES NOT EXIST FOR P
     1RESSURES ABOVE THE CRITICAL PRESSURE',F10.3,'PSIA',//,5X,'FREON TE
     2RMINATED')
 9040 FORMAT(1H ,4X,'THE MARTIN-HOU EQUATION OF STATE IS NOT VALID IN TH
     1IS REGION,2.E. P>PC AND S;SC',//,5X,'FREON TERMINATED')
 9050 FORMAT(1H ,4X,'THE MARTIN HOU EQUATION OF STATE IS NOT VALID IN TH
     1IS REGION,2.E. P>PC AND H;HC',//,5X,'FREON TERMINATED')
 9060 FORMAT(1H ,4X,'***ERROR*** THE FLUID ',A6,' CAN NOT BE EVALUATED U
     1SING THE SUBROUTINE FREON',//,5X,'FREON TERMINATED')
 9070 FORMAT(1H ,4X,'***ERROR*** A QUALITY OF ',F6.3,'IS NOT ALLOWED',
     1//, 5X,'FREON TERMINATED')
```

```
! Univac Statement      DEBUG SUBCHK
      END
C     ****************************************************************
C     *                                                              *
C     *                           V P E Q M                          *
C     *   THIS SUBROUTINE CALCULATES THE SATURATION PRESSURE OR      *
C     *TEMPERATURE GIVEN THE OTHER                                   *
C     *                                                              *
C     ****************************************************************
      SUBROUTINE VPEQM(P1,T,M)
      PARAMETER M2=6
      COMMON/FREON1/Q(4,12),TTOLER,BTOLER,LIMIT,DERROR,NTYPE
      REAL LOGP
      TC=Q(4,3)+Q(1,11)
      IF (M.EQ.1) GO TO 100
      T1 = ABS(Q(2,1)/(ALOG10(P1)-Q(1,1)))
      TMISS = (T1-T)/ T1
      IF (ABS(TMISS).GT..2D + 00) T = T1
  100 DO 105 ITR=1,LIMIT
      T2 = T*T
      T3 = T*T2
      FT = Q(2,2)-T
      IF (FT.LT..1E-20) FT = 1.0
      LOGP=Q(1,1)+Q(2,1)/T+Q(3,1)*ALOG10(T)+Q(4,1)*T+Q(1,2)*FT/T*ALOG10(
     1 FT)
      P2 = 10.0 **LOGP
      IF(M.EQ.1 ) P1 = P2
      DELP = P1-P2
      ERROR = DELP/P1
      IF(P2/P1.LT.TTOLER.AND.P2/P1.GT.BTOLER) RETURN
      CALL DRVPT(DPDT,P2,T,D2PDT2)
      DELT = DELP/DPDT
      T = T + DELT
      IF(T.GT.TC) T=TC
  105 CONTINUE
      WRITE (M2, 115) LIMIT,P1,P2
  110 RETURN
  115 FORMAT(1H ,4X,'***VPEQM FAILED TO CONVERGE IN',I6,'ITERATIONS',
     1 /,'DESIRED PRESSURE=',F10.3,'CALCULATED PRESSURE=',F10.3)
! Univac Statement      DEBUG SUBCHK
      END
C     ****************************************************************
C     *                                                              *
C     *                          V O L U M E                         *
C     *                                                              *
C     *   THIS SUBROUTINE FINDS THE VAPOR DENSITY BY USE OF THE NEWTON-*
C     *RAPHSON ITERATION METHOD, GIVEN PPRESSURE AND TEMPERATURE.    *
C     *                                                              *
C     ****************************************************************
      SUBROUTINE VOLUME(T,P,V)
      PARAMETER M2=6
```

```
      COMMON/FREON1/Q(4,12),TTOLER,BTOLER,LIMIT,DERROR,NTYPE
      V=Q(2,3)*T/P
      VLAST=V
      CALL EOS(PCAL,T,V,DPDT,DPDV,DPDVDT,DP2DT2)
      DO 1000 I=1,LIMIT
      V=VLAST-((PCAL-P)/DPDV)
      IF(V.LE.0.0) V=VLAST/10.0
      CALL EOS(PCAL,T,V,DPDT,DPDV,DPDVDT,DP2DT2)
      IF(PCAL/P.LT.TTOLER.AND.PCAL/P.GT.BTOLER.OR.ABS(V-VLAST).LT.
     1    1.0E-10) RETURN
      VLAST=V
 1000 CONTINUE
      WRITE(M2,9000) LIMIT
 9000 FORMAT( 11X,'***VOLUME FAILED TO CONVERGE IN ',I5,' ITERATIONS')
      RETURN
! Univac Statement      DEBUG SUBCHK
      END
C     ******************************************************************
C     *                                                                *
C     *                          E N T H A L                           *
C     *                                                                *
C     *   THIS SUBROUTINE COMPUTES THE ENTHALPY OF THE VAPOR.          *
C     *                                                                *
C     ******************************************************************
      SUBROUTINE ENTHAL(P,T,V,H,DHDT,DHDV)
      COMMON/FREON1/Q(4,12),TTOLER,BTOLER,LIMIT,DERROR,NTYPE
      C=0. 185053
      T2=T*T
      T3=T2*T
      T4=T3*T
      VMB=V-Q(3,3)
      VMB2=VMB*VMB
      VMB3=VMB2*VMB
      VMB4=VMB3*VMB
      VMB5=VMB4*VMB
      EKIT=-Q(2,12)*T/(Q(4,3)+Q(1,11))
      EKIT=EXP(EKIT)
      EKA=Q(3,12)*V
      IF (EKA.GT.30.0) GO TO 105
      IF (ABS(EKA). LE. DERROR) GO TO 105
      EKA = EXP(EKA)
      IF (ABS(Q(1,12)).LE.DERROR) GO TO 100
      PART = Q(1,12)*ALOG((Q(1,12)*EKA+1.0)/(Q(1,12)*EKA))
      GO TO 110
  100 PART = 0.0
      GO TO 110
  105 EKA = 0.0
      RX = 0.0
      RZ = 0.0
      PART = 0.0
      TERMA=0.0
```

```
      TERMB=0.0
      GO TO 115
  110 RX = Q(2,8)/Q(3,12)*(1.0/EKA-PART)
      RZ=Q(4,8)/(Q(3,12)*EKA)-(Q(4,8)*PART/Q(3,12))
      TERMA=Q(2,8)*(-1.0/EKA+Q(1,12)/(Q(1,12)*EKA+1.0))
      TERMB=Q(4,8)/EKA
  115 CALL EOS(P,T,V,DPDT,DPDV,DPDVDT,D2PDT2)
      H1=Q(1,9)*T+Q(2,9)*T2/2.0+Q(3,9)*T3/3.0+Q(4,9)*T4/4.0-
     1 Q(1,10)/T+C*P*V
      H2=C*(Q(2,4)/VMB+Q(2,5)/(2.0*VMB2)+Q(2,6)/(3.0*VMB3)+Q(2,7
     1)/(4.0*VMB4)+RX)
      H3=C*(Q(4,4)/VMB+Q(4,5)/(2.0*VMB2)+Q(4,7)/(4.0*VMB4)+Q(4,6
     1 )/(3.0*VMB3)+RZ)*(1.0+Q(2,12)*T/(Q(4,3)+Q(1,11)))*EKIT+Q(
     22,10)
      H = H1 + H2 + H3
      TC=Q(4,3)+Q(1,11)
      PART1=Q(1,9)+Q(2,9)*T+Q(3,9)*T2+Q(4,9)*T3+Q(1,10)/T2+C*DPDT*V
      PART2=Q(4,4)/VMB+Q(4,5)/(2.0*VMB2)+Q(4,6)/(3.0*VMB3)+Q(4,7)/(4.0*
     1    VMB4)+RZ
      PART2=-C*EKIT*Q(2,12)**2*T/TC**2*PART2
      DHDT=PART1+PART2
      PART1=C*DPDV*V+C*P+C*(-Q(2,4)/VMB2-Q(2,5)/VMB3-Q(2,6)/VMB4-Q(2,7)/
     1    VMB5+TERMA)
      PART2=-Q(4,4)/VMB2-Q(4,5)/VMB3-Q(4,6)/VMB4-Q(4,7)/VMB5-TERMB
     1    +Q(4,8)*Q(1,12)/(Q(1,12)*EKA+1.0)
      PART2=C*EKIT*(1.0+Q(1,12)*T/TC)*PART2
      DHDV=PART1+PART2
      RETURN
! Univac Statement     DEBUG SUBCHK
      END
C     ********************************************************************
C     *                                                                  *
C     *                           E N T R O P                            *
C     *                                                                  *
C     *   THIS SUBROUTINE COMPUTES THE ENTROPY OF THE VAPOR.             *
C     *                                                                  *
C     ********************************************************************
      SUBROUTINE ENTROP(T,V,S,DSDT,DSDV)
      COMMON/FREON1/Q(4,12),TTOLER,BTOLER,LIMIT,DERROR,NTYPE
      C=0.185053
      VMB=V-Q(3,3)
      VMB2=VMB*VMB
      VMB3=VMB2*VMB
      VMB4=VMB3*VMB
      VMB5=VMB4*VMB
      EKA=Q(3,12) *V
      IF(ABS(EKA).GT.30.0) GO TO 100
      IF(ABS(EKA).LT.DERROR) GO TO 100
      IF(ABS(Q(1,12)).LE.DERROR) GO TO 100
      EKA = EXP(EKA)
      PART = Q(1,12)*ALOG((Q(1,12)*EKA+1.0)/(Q(1,12)*EKA))
```

```
      GO TO 105
  100 RX = 0.0
      RZ = 0.0
      EKA = 0.0
      PART = 0.0
      TERMA=0.0
      TERMB=0.0
      GO TO 110
  105 RX = Q(3,8)/Q(3,12)*(1.0/EKA-PART)
      RZ = Q(4,8)/(Q(3,12)*EKA)-(Q(4,8)/Q(3,12)*PART)
      TERMA=Q(3,8)*(-1.0/EKA+Q(1,12)/(Q(1,12)*EKA+1.0))
      TERMB=Q(4,8)/EKA
  110 G=Q(4,4)/VMB+Q(4,5)/(2.0*VMB2)+Q(4,7)/(4.0*VMB4)+Q(4,6)/(
     1    3.0*VMB3)+RZ
      T2 = T*T
      T3 = T2*T
      EKIT=-Q(2,12)*T/(Q(4,3)+Q(1,11))
      IF (ABS(EKIT).GT.30.0) GO TO 115
      EKIT = EXP(EKIT)
      GO TO 120
  115 EKIT = 0.0
  120 ZZ = Q(2,12)**2.00/(Q(4,3)+Q(1,11))**2.0*EKIT
      S1=Q(1,9)*ALOG(T)+Q(2,9)*T+Q(3,9)*T2/2.0+Q(4,9)*T3/3.0-Q(1
     1 ,10)/(2.0*T2)
      S2=C*Q(2,3)*ALOG(VMB)
      S3=-C*(Q(3,4)/VMB+Q(3,5)/(2.0*VMB2)+Q(3,6)/(3.0*VMB3)+Q(3,
     1 7)/(4.0*VMB4)+RX)
      S4=C*(Q(2,12)/(Q(4,3)+Q(1,11))*EKIT)*(Q(4,4)/VMB+Q(4,5)/(2.0*V
     1 MB2)+Q(4,6)/(3.0*VMB3)+Q(4,7)/(4.0*VMB4)+RZ)+Q(3,10)
      S=S1+S2+S3+S4
      DSDT=Q(1,9)/T+Q(2,9)+Q(3,9)*T+Q(4,9)*T2+Q(1,10)/T3-C*G*ZZ
      TC=Q(4,3)+Q(1,11)
      PART1=C*Q(2,3)/VMB-C*(-Q(3,4)/VMB2-Q(3,5)/VMB3-Q(3,6)/VMB4-Q(3,7)/
     1    VMB5+TERMA)
      PART2=C*Q(2,12)*EKIT/TC*(-Q(4,4)/VMB2-Q(4,5)/VMB3-Q(4,6)/VMB4
     1    -Q(4,7)/VMB5-TERMB+Q(4,8)*Q(1,12)/(Q(1,12)*EKA+1.0))
      DSDV=PART1+PART2
  130 RETURN
! Univac Statement      DEBUG SUBCHK
      END
C     *****************************************************************
C     *                                                               *
C     *                         P R I N T F                           *
C     *                                                               *
C     *   THIS ROUTINE PRINTS THE RESULTS OF ROUTINE FREON.           *
C     *                                                               *
C     *****************************************************************
      SUBROUTINE PRINTF(T,P,QUAL,V,H,S,VV,HV,SV,VL,HL,SL,FTYPE,NPRIN)
      CHARACTER*6 FTYPE
      PARAMETER M2=6
      COMMON/FREON1/Q(4,12),TTOLER,BTOLER,LIMIT,DERROR,NTYPE
```

```
      T=T-Q(1,11)
      IF(NPRIN.EQ.0) RETURN
      WRITE(M2,9000) FTYPE
      WRITE(M2,9010) T,P
      IF(QUAL.LE.1.0.AND.QUAL.GE.0.0) GO TO 3280
      IF(QUAL.GT.1.0) WRITE(M2,9020)
      IF(QUAL.LT.0.0) WRITE(M2,9030)
      GO TO 3370
 3280 WRITE(M2,9040) QUAL
      WRITE(M2,9050)
      WRITE(M2,9070) HV,HL
      WRITE(M2,9080) SV,SL
      WRITE(M2,9060) VV,VL
 3370 WRITE(M2,9090)
      WRITE(M2,9070) H
      WRITE(M2,9080) S
      WRITE(M2,9060) V
 9000 FORMAT(1H1,4X,'***THERMODYNAMIC PROPERTIES OF ',A6,' USING MARTIN-
     1HOU EQUATION OF STATE***')
 9010 FORMAT(1H ,4X,'TEMPERATURE = ',F7.1,'DEG.F',/,5X,'PRESSURE    = ',
     1    F10.5,'PSIA')
 9020 FORMAT(1H ,4X,'SUPERHEATED VAPOR',//////)
 9030 FORMAT(1H ,4X,'COMPRESSED LIQUID-APPROXIMATED BY SATURATED LIQUID'
     1    ,//////)
 9040 FORMAT(1H ,4X,'SATURATED LIQUID-VAPOR EQUILIBRUIM',/,5X,'QUALITY =
     1 ',F6.4)
 9050 FORMAT(1H ,//,35X,'SATURATED',8X,'SATURATED',/,35X,'VAPOR',12X,
     1    'LIQUID',/,5X,60('#'))
 9060 FORMAT(1H ,4X,'SPECIFIC VOLUME (CU.FT/LBM)',5X,F9.4,8X,F9.4,/////
     1    /)
 9070 FORMAT(1H ,4X,'ENTHALPY',8X,'(BTU/LBM)',5X,F9.3,8X,F9.3)
 9080 FORMAT(1H ,4X,'ENTROPY',9X,'(BTU/LBM-R)',5X,F9.6,8X,F9.6)
 9090 FORMAT(1H ,//////)
      RETURN
! Univac Statement      DEBUG SUBCHK
      END
C     ******************************************************************
C     *                                                                *
C     *                            D R V P T                           *
C     *                                                                *
C     *   THIS SUBROUTINE CALCULATES THE DERIVATIVE OF THE SATURATION  *
C     *PRESSURE WITH RESPECT TO TEMPERATURE                            *
C     *                                                                *
C     ******************************************************************
      SUBROUTINE DRVPT(DPDT,P,T,D2PDT2)
      COMMON/FREON1/Q(4,12),TTOLER,BTOLER,LIMIT,DERROR,NTYPE
      T2 = T*T
      T3=T2*T
      FT=Q(2,2)-T
      IF (FT.LT.0.01) FT = 1.0
      XX = ALOG(10.0)
```

```
      DPDT = P*((-XX*(Q(2,1) + Q(1,2) *Q(2,2) *ALOG10(FT)))/T2+(Q(3,1)-Q
     1(1,2)) /T+Q(4,1)*XX)
      D2PDT2=P*(2.0*XX*(Q(2,1)+Q(1,2)*Q(2,2)*ALOG10(FT))/T3+
     1    XX/T2*(0.43429448/FT)+(Q(3,1)-Q(1,2))/T2)
      RETURN
! Univac Statement      DEBUG SUBCHK
      END
C     ****************************************************************
C     *                                                              *
C     *                         F L Q D E N                          *
C     *                                                              *
C     *   THIS SUBROUTINE COMPUTES THE SATURATED LIQUID SPECIFIC VOLUME*
C     *GIVEN TEMPERATURE USING ONE OF THREE EQUATIONS.                *
C     *                                                              *
C     ****************************************************************
      SUBROUTINE FLQDEN(VL,TR,DVLDT)
      COMMON/FREON1/Q(4,[illegible]),TTOLER,BTOLER,LIMIT,DERROR,NTYPE
      TC=Q(4,3)+Q(1,11)
  100 GO TO (200,300,400), NTYPE
  200 P1=1.0/3.0
      P2=2.0/3.0
      P3=4.0/3.0
      T=1.0-TR/TC
      DL=Q(1,4)+Q(1,5)*T**P1+Q(1,6)*T**P2+Q(1,7)*T**P3
      VL=1.0/DL
      T3=3.0*TC
      DVLDT=VL**2*(Q(1,5)/(T3*T**P2)+2.0*Q(1,6)/(T3*T**P1)+Q(1,7)/TC
     1    +4.0*Q(1,8)/(T3*T**P1))
      RETURN
  300 P1=1.0
      P2=1.0/2.0
      P3=1.0/3.0
      P4=2.0/3.0
      T=TC-TR
      DL=Q(1,4)+Q(1,5)*T**P1+Q(1,6)*T**P2+Q(1,7)*T**P3+Q(1,8)*T**2
      VL=1.0/DL
      DVLDT=VL**2*(Q(1,5)+Q(1,6)/(2.0*T**P2)+Q(1,7)/(3.0*T**P4)+2.0*
     1    Q(1,8)*T)
      RETURN
  400 DL=Q(1,4)+Q(1,5)*TR+Q(1,6)*TR**2
      VL=1.0/DL
      DVLDT=VL**2*(-Q(1,5)-Q(1,6)*2.0*TR)
      RETURN
! Univac Statement      DEBUG SUBCHK
      END
C     ****************************************************************
C     *                                                              *
C     *                            S A T N                           *
C     *                                                              *
C     *   THIS SUBROUTINE COMPUTES THE SATURATED VAPOR-LIQUID         *
C     *PROPERTIES GIVEN TEMPERATURE AND SPECIFIC VOLUME.              *
```

```
C     *                                                                *
C     ******************************************************************
      SUBROUTINE SATN(PSIA,T,V,HV,SV,HL,SL,VL)
      COMMON/FREON1/Q(4,12),TTOLER,BTOLER,LIMIT,DERROR,NTYPE
      CALL FLQDEN(VL,T,DVLDT)
      RJ=0.185053
      CALL ENTHAL(PSIA,T,V,H,DHDT,DHDV)
      CALL DRVPT(DPSDT,PSIA,T,DPSDT2)
      DELTH=RJ*DPSDT*(V-VL)*T
      HL=H-DELTH
      CALL ENTROP(T,V,S,DSDT,DSDV)
      SL=S-DELTH/T
      VV=V
      HV=H
      SV=S
      RETURN
! Univac Statement      DEBUG SUBCHK
      END
C     ******************************************************************
C     *                                                                *
C     *                          S F I N D                             *
C     *                                                                *
C     *                                                                *
C     *   THIS SUBROUTINE FINDS TEMPERATURE AND DENSITY GIVEN PRESSURE *
C     *AND ENTROPY FOR SUPERHEATED VAPOR STATES.                       *
C     *                                                                *
C     ******************************************************************
      SUBROUTINE SFIND(T,P,V,S)
      PARAMETER M2=6
      COMMON/FREON1/Q(4,12),TTOLER,BTOLER,LIMIT,DERROR,NTYPE
      VLAST=V
      TLAST=T
      CALL EOS(PCAL,T,V,DPDT,DPDV,DPDVDT,D2PDT2)
      CALL ENTROP(T,V,SCAL,DSDT,DSDV)
      DO 1000 I=1,LIMIT
      XJACOB=DPDV*DSDT-DPDT*DSDV
      V=VLAST-((PCAL-P)*DSDT-(SCAL-S)*DPDT)/XJACOB
      T=TLAST-((SCAL-S)*DPDV-(PCAL-P)*DSDV)/XJACOB
      IF(T.LE.0.0) T=TLAST/10.0
      IF(V.LE.0.0) V=VLAST/10.0
      CALL ENTROP(T,V,SCAL,DSDT,DSDV)
      CALL EOS(PCAL,T,V,DPDT,DPDV,DPDVDT,D2PDT2)
      IF(PCAL/P.LT.TTOLER.AND.PCAL/P.GT.BTOLER.AND.SCAL/S.LT.
     1    TTOLER.AND.SCAL/S.GT.BTOLER) GO TO 1050
      IF(ABS(T-TLAST).LT.1.0E-04.AND.ABS(V-VLAST).LT.1.0E-07) GO TO 1050
      VLAST=V
      TLAST=T
 1000 CONTINUE
      WRITE(M2,9000) LIMIT
 1050 RETURN
 9000 FORMAT(' ',10X,'***SFIND FAILED TO CONVERGE IN ',I5,' ITERATIONS')
! Univac Statement      DEBUG SUBCHK
```

```
      END
C     ****************************************************************
C     *                        C O M P S                             *
C     *                                                              *
C     *                                                              *
C     *   THIS SUBROUTINE FINDS THE TEMPERATURE AND DENSITY GIVEN    *
C     *PRESSURE AND ENTROPY FOR COMPRESSED LIQUID STATES.            *
C     *                                                              *
C     ****************************************************************
      SUBROUTINE COMPS(P,T,V,S,VL)
      PARAMETER M2=6
      COMMON/FREON1/Q(4,12),TTOLER,BTOLER,LIMIT,DERROR,NTYPE
      RJ=0.185053
      VLAST=V
      TLAST=T
      CALL EOS(PCAL,T,V,DPDT,DPDV,DPDVDT,D2PDT2)
      CALL VPEQM(PSAT,T,1)
      CALL ENTROP(T,V,SCAL,DSDT,DSDV)
      CALL DRVPT(DPSDT,PSAT,T,DPSDT2)
      CALL FLQDEN(VL,T,DVLDT)
      DO 1000 I=1,LIMIT
      FP=PCAL-PSAT
      DFPDT=DPDT-DPSDT
      FS=SCAL-RJ*DPSDT*(V-VL)-S
      DFSDT=DSDT-RJ*DPSDT2*(V-VL)+RJ*DPSDT*DVLDT
      DFSDV=DSDV-RJ*DPSDT
      XJACOB=DPDV*DFSDT-DFPDT*DFSDV
      V=VLAST-(FP*DFSDT-FS*DFPDT)/XJACOB
      T=TLAST-(FS*DPDV-FP*DFSDV)/XJACOB
      IF(T.LE.0.0) T=TLAST/10.0
      IF(V.LE.0.0) V=VLAST/10.0
      CALL EOS(PCAL,T,V,DPDT,DPDV,DPDVDT,D2PDT2)
      CALL VPEQM(PSAT,T,1)
      CALL ENTROP(T,V,SCAL,DSDT,DSDV)
      CALL DRVPT(DPSDT,PSAT,T,DPSDT2)
      CALL FLQDEN(VL,T,DVLDT)
      SL=SCAL-RJ*(V-VL)*DPSDT
      IF(PCAL/PSAT.LT.TTOLER.AND.PCAL/PSAT.GT.BTOLER.AND.SL/S.LT.
     1    TTOLER.AND.SL/S.GT.BTOLER) GO TO 1050
      IF(ABS(T-TLAST).LT.1.0E-04.AND.ABS(V-VLAST).LT.1.0E-07) GO TO 1050
      VLAST=V
      TLAST=T
 1000 CONTINUE
      WRITE(M2,9000) LIMIT
 1050 P=PCAL
      RETURN
 9000 FORMAT(' ',10X,'***COMPS FAILED TO CONVERGE IN ',I5,' ITERATIONS')
! Univac Statement        DEBUG SUBCHK
      END
C     ****************************************************************
C     *                                                              *
```

```
C     *                          H F I N D                              *
C     *                                                                 *
C     *   THIS SUBROUTINE FINDS TEMPERATURE AND DENSITY GIVEN PRESSURE *
C     *AND ENTHALPY FOR SUPERHEATED VAPOR STATES.                       *
C     *                                                                 *
C     *******************************************************************
      SUBROUTINE HFIND(T,P,V,H)
      PARAMETER M2=6
      COMMON/FREON1/Q(4,12),TTOLER,BTOLER,LIMIT,DERROR,NTYPE
      VLAST=V
      TLAST=T
      CALL EOS(PCAL,T,V,DPDT,DPDV,DPDVDT,DP2DT2)
      CALL ENTHAL(PCAL,T,V,HCAL,DHDT,DHDV)
      DO 1000 I=1,LIMIT
      XJACOB=DPDV*DHDT-DPDT*DHDV
      V=VLAST-((PCAL-P)*DHDT-(HCAL-H)*DPDT)/XJACOB
      T=TLAST-((HCAL-H)*DPDV-(PCAL-P)*DHDV)/XJACOB
      IF(T.LE.0.0) T=TLAST/10.0
      IF(V.LE.0.0) V=VLAST/10.0
      CALL EOS(PCAL,T,V,DPDT,DPDV,DPDVDT,DP2DT2)
      CALL ENTHAL(PCAL,T,V,HCAL,DHDT,DHDV)
      IF(PCAL/P.LT.TTOLER.AND.PCAL/P.GT.BTOLER.AND.HCAL/H.LT.
     1    TTOLER.AND.HCAL/H.GT.BTOLER) GO TO 1050
      IF(ABS(T-TLAST).LT.1.0E-04.AND.ABS(V-VLAST).LT.1.0E-07) GO TO 1050
      VLAST=V
      TLAST=T
 1000 CONTINUE
      WRITE(M2,9000) LIMIT
 1050 RETURN
 9000 FORMAT(' ',10X,'HFIND FAILED TO CONVERGE IN ',I5,' ITERATIONS')
! Univac Statement        DEBUG SUBCHK
      END
C     *******************************************************************
C     *                                                                 *
C     *                          C O M P H                              *
C     *                                                                 *
C     *   THIS SUBROUTINE FINDS THE TEMPERATURE AND DENSITY GIVEN       *
C     *PRESSURE AND ENTHALPY FOR COMPRESSED LIQUID STATES.              *
C     *                                                                 *
C     *******************************************************************
      SUBROUTINE COMPH(P,T,V,H,VL)
      PARAMETER M2=6
      COMMON/FREON1/Q(4,12),TTOLER,BTOLER,LIMIT,DERROR,NTYPE
      RJ=0.185053
      VLAST=V
      TLAST=T
      CALL EOS(PCAL,T,V,DPDT,DPDV,DPDVDT,DP2DT2)
      CALL VPEQM(PSAT,T,1)
      CALL ENTHAL(PCAL,T,V,HCAL,DHDT,DHDV)
      CALL DRVPT(DPSDT,PSAT,T,DPSDT2)
      CALL FLQDEN(VL,T,DVLDT)
```

```
      DO 1000 I=1,LIMIT
      FP=PCAL-PSAT
      DFPDT=DPDT-DPSDT
      FH=HCAL-RJ*DPSDT*(V-VL)*T-H
      DFHDT=DHDT-RJ*(DPSDT2*(V-VL)*T-DPSDT*DVLDT*T+DPSDT*(V-VL))
      DFHDV=DHDV-RJ*DPSDT*T
      XJACOB=DPDV*DFHDT-DFPDT*DFHDV
      V=VLAST-(FP*DFHDT-FH*DFPDT)/XJACOB
      T=TLAST-(FH*DPDV-FP*DFHDV)/XJACOB
      IF(T.LE.0.0) T=TLAST/10.0
      IF(V.LE.0.0) V=VLAST/10.0
      CALL EOS(PCAL,T,V,DPDT,DPDV,DPDVDT,DP2DT2)
      CALL VPEQM(PSAT,T,1)
      CALL ENTHAL(PCAL,T,V,HCAL,DHDT,DHDV)
      CALL DRVPT(DPSDT,PSAT,T,DPSDT2)
      CALL FLQDEN(VL,T,DVLDT)
      HL=HCAL-RJ*DPSDT*(V-VL)*T
      IF(PCAL/PSAT.LT.TTOLER.AND.PCAL/PSAT.GT.BTOLER.AND.HL/H.LT.
     1    TTOLER.AND.HL/H.GT.BTOLER) GO TO 1050
      IF(ABS(T-TLAST).LT.1.0E-04.AND.ABS(V-VLAST).LT.1.0E-07) GO TO 1050
      VLAST=V
      TLAST=T
 1000 CONTINUE
      WRITE(M2,9000) LIMIT
 1050 P=PCAL
      RETURN
 9000 FORMAT(' ',10X,'***COMPH FAILED TO CONVERGE IN ',I5,' ITERATIONS')
! Univac Statement      DEBUG SUBCHK
      END
C     ****************************************************************
C     *                                                              *
C     *                           E O S                              *
C     *                                                              *
C     *   THIS SUBROUTINE CALCULATES THE PRESSURE FOR VAPOR STATES AND *
C     *ITS DERIVATIVES GIVEN TEMPERATURE AND DENSITY                 *
C     *                                                              *
C     ****************************************************************
      SUBROUTINE EOS(P,T,V,DPDT,DPDV,DPDVDT,D2PDT2)
      COMMON/FREON1/Q(4,12),TTOLER,BTOLER,LIMIT,DERROR,NTYPE
      DIMENSION TERM (6), PART (7)
      DPDV = 0.0
      P1=0.0
      VMB=V-Q(3,3)
      EKIT=-Q(2,12)*T/(Q(4,3)+Q(1,11))
      EKIT=EXP(EKIT)
      EKA=Q(3,12)*V
      EKA2=2.00*Q(3,12)*V
      IF(ABS (EKA).LE.DERROR) GO TO 105
      IF (ABS(EKA).GT.30.0) GO TO 105
      EKA = EXP(EKA)
      EKA2 = EXP(EKA2)
```

```
      GO TO 115
  105 EKA = 0.0
      EKA2 = 0.0
  115 TERM(1) = Q(2,3)*T/VMB
      DO 125 I = 2,5
      TERM(I) = (Q(2,I+2)+Q(3,I+2)*T+Q(4,I+2)*EKIT)/VMB**I
  125 P1 = P1 +TERM(I)
      IF(ABS(EKA).LE.DERROR) TERM (6) = 0.0
      IF(ABS(EKA).LE.DERROR) GO TO 120
      TERM (6) = (Q(2,8) + Q(3,8)*T+Q(4,8)*EKIT)/(EKA*(1.0+Q(1,12)*E
     1KA))
  120 P1 = P1 + TERM(1) + TERM(6)
      P=P1
  140 PART (1) = Q(2,3)*T/VMB**2
      DO 150 I = 2,5
      PART(I)  = FLOAT (I)*(Q(2,I+2)+Q(3,I+2)*T+Q(4,I+2)*EKIT)/VMB**(I+1
     1)
  150 DPDV = DPDV + PART (I)
      IF (ABS (EKA).LE.DERROR) PART (6) = 0.0
      IF (ABS (EKA).LE.DERROR) GO TO 145
      IF (EKA2.GT.30.0) EKA2 = 0.0
      XMESS = (Q(3,12)*EKA+2.0*Q(3,12)*Q(1,12)*EKA2)/((EKA+Q(1,12)*E
     1KA2)**2)
      XMESS = -XMESS
      PART (6) = (Q(2,8)+Q(3,8)*T+Q(4,8)* EKIT)*XMESS
  145 DPDV =-(DPDV+PART(1) + PART (6))
      VMB2=VMB*VMB
      VMB3=VMB2*VMB
      VMB4=VMB2**2
      VMB5=VMB4*VMB
      VMB6=VMB3*VMB3
      TC=Q(4,3)+Q(1,11)
      CK2=Q(4,4)*Q(2,12)/TC
      CK3=Q(4,5)*Q(2,12)/TC
      CK4=Q(4,6)*Q(2,12)/TC
      CK5=Q(4,7)*Q(2,12)/TC
      CK6=Q(4,8)*Q(2,12)/TC
      EKA=Q(3,12)*V
      IF(ABS(EKA).LE.DERROR) GO TO 100
      IF(EKA.GT.30.0) GO TO 100
      EKA=EXP(EKA)
      TERMA=(Q(3,8)-CK6*EKIT)/(EKA*(1.0+Q(1,12)*EKA))
      TERMB=Q(4,8)/(EKA*(1.0+Q(1,12)*EKA))
      Z6=Q(3,12)*(1.0+2.0*Q(1,12)*EKA)/(EKA*(1.0+Q(1,12)*EKA))
      GO TO 180
  100 TERMA=0.0
      TERMB=0.0
      Z6=0.0
      EKA=0.0
  180 DPDT=Q(2,3)/VMB+(Q(3,4)-CK2*EKIT)/VMB2+(Q(3,5)-CK3*EKIT)/VMB3+
     1    (Q(3,6)-CK4*EKIT)/VMB4+(Q(3,7)-CK5*EKIT)/VMB5+TERMA
```

```
      D2PDT2=Q(2,12)**2*EKIT/TC**2*(Q(4,4)/VMB2+Q(4,5)/VMB3+Q(4,6)/VMB4+
     1    Q(4,7)/VMB5+TERMB)
      Z1=Q(3,4)-Q(2,12)*Q(4,4)*EKIT/TC
      Z2=Q(3,5)-Q(2,12)*Q(4,5)*EKIT/TC
      Z4=Q(3,7)-Q(2,12)*Q(4,7)*EKIT/TC
      Z3=Q(3,6)-Q(2,12)*Q(4,6)*EKIT/TC
      Z5=Q(3,8)-Q(2,12)*Q(4,8)*EKIT/TC
      DPDVDT=-Q(3,2)/VMB2-2.0*Z1/VMB3-3.0*Z2/VMB4-4.0*Z3/VMB5-5.0*Z4/
     1    VMB6-Z5*Z6
  175 RETURN
! Univac Statement      DEBUG SUBCHK
      END
    SUBROUTINE SIFREON(CYCLE,FGIVEN,SGIVEN,FTYPE,H,S,V,TR,P
     #       ,HV,SV,VV,HL,SL,VL,QUAL,NPRT)
    IMPLICIT REAL(L)
    INTEGER CALLIN
    CHARACTER CYCLE*2,FTYPE*6
    IF(CYCLE.EQ.'TP') THEN
    FGIVEN=FGIVEN*1.8+32.
    SGIVEN=.145037743897283*SGIVEN
    ELSEIF(CYCLE.EQ.'TX') THEN
    FGIVEN=FGIVEN*1.8+32.
    ELSEIF(CYCLE.EQ.'PS') THEN
    FGIVEN=.145037743897283*FGIVEN
    SGIVEN=SGIVEN/4.1868
    ELSEIF(CYCLE.EQ.'PH') THEN
    FGIVEN=.145037743897283*FGIVEN
    SGIVEN=SGIVEN/2.326
    ELSEIF(CYCLE.EQ.'PX') THEN
    FGIVEN=FGIVEN*.145037743897283
    ELSEIF(CYCLE.EQ.'TV') THEN
    FGIVEN=FGIVEN*1.8+32.
    SGIVEN=SGIVEN*.01602
    ELSE
    PRINT*,'WRONG CYCLE VALUE IS GIVEN'
    RETURN
    ENDIF
    CALL FREON(CYCLE,FGIVEN,SGIVEN,FTYPE,H,S,V,TR,P
     #       ,HV,SV,VV,HL,SL,VL,QUAL,NPRT)
    S=S*4.1868
    H=H*2.326
    V=V*0.0624219
    TR=(TR-32.)/1.8
    P=P/.145037743897283
    SL=SL*4.1868
    HL=HL*2.326
    VL=VL*0.0624219
    SV=SV*4.1868
    HV=HV*2.326
    VV=VV*0.0624219
```

```
RETURN
END
```