1-1-1996

A numerical investigation of performance and optimization trends of direct contact heat exchangers

Richard Arlen Brickman
University of Nevada, Las Vegas

Repository Citation

Brickman, Richard Arlen, "A numerical investigation of performance and optimization trends of direct contact heat exchangers" (1996). *UNLV Retrospective Theses & Dissertations*. 3011.
http://dx.doi.org/10.25669/4zw2-onhk

INFORMATION TO USERS

This manuscript has been reproduced from the microfilm master. UMI films the text directly from the original or copy submitted. Thus, some thesis and dissertation copies are in typewriter face, while others may be from any type of computer printer.

The quality of this reproduction is dependent upon the quality of the copy submitted. Broken or indistinct print, colored or poor quality illustrations and photographs, print bleedthrough, substandard margins, and improper alignment can adversely affect reproduction.

In the unlikely event that the author did not send UMI a complete manuscript and there are missing pages, these will be noted. Also, if unauthorized copyright material had to be removed, a note will indicate the deletion.

Oversize materials (e.g., maps, drawings, charts) are reproduced by sectioning the original, beginning at the upper left-hand corner and continuing from left to right in equal sections with small overlaps. Each original is also photographed in one exposure and is included in reduced form at the back of the book.

Photographs included in the original manuscript have been reproduced xerographically in this copy. Higher quality 6" x 9" black and white photographic prints are available for any photographs or illustrations appearing in this copy for an additional charge. Contact UMI directly to order.

UMI

A Bell & Howell Information Company
300 North Zeeb Road, Ann Arbor MI 48106-1346 USA
313/761-4700 800/521-0600

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

A NUMERICAL INVESTIGATION OF PERFORMANCE AND OPTIMIZATION TRENDS OF DIRECT CONTACT HEAT EXCHANGERS

by

Richard A. Brickman

A dissertation submitted in partial fulfillment
of the requirements for the degree of

Doctor of Philosophy

in

Engineering

Mechanical Engineering Department
University of Nevada, Las Vegas
December 1996

UMI Number: 9718041

UMI
300 North Zeeb Road
Ann Arbor, MI 48103

The Dissertation of Richard A. Brickman for the degree of Doctor of Philosophy in Engineering is approved.

Chairperson, Robert F. Boehm, Ph.D.

Examining Committee Member, Samir F. Moujaes, Ph.D.

Darrell W. Pepper

Examining Committee Member, Darrell W. Pepper, Ph.D.

Examining Committee Member, William P. Graebel, Ph.D.

Graduate Faculty Representative, William C. Culbreth, Ph.D.

Dean of the Graduate College, Ronald W. Smith, Ph.D.

University of Nevada, Las Vegas
December 1996

ABSTRACT

A numerical model has been developed that can be used for industrial design of a wide range of direct contact heat exchangers (DCHXs). This model is built from some earlier work where a one-dimensional model was used to describe some very specific and limited DCHX spray-column configurations. Changes to the initial model are proposed to reflect more accurately the physics of the flows found in both two- and three-phase DCHXs. In addition, generalizations are incorporated into the model to be able to analyze a wide variety of fluid combinations and roles (either as the dispersed flow or as the continuous fluid). A major effort is directed to the development of an optimization that can handle the complete variety of fluids and configurations for spray columns. Parametric evaluations are then performed to demonstrate that the model does indeed perform appropriately, and comparisons are made to existing experimental data. Finally, an extension to a more complicated configuration is made where the two-dimensional effects of a sieve-tray-tower exchanger are incorporated. This latter aspect is the first step in developing multidimensional models.

TABLE OF CONTENTS

LIST OF FIGURES

NOMENCLATURE

A	Column cross sectional area, m^2.
A_B	Surface area of droplet/bubble, m^2.
a_b	Acceleration of droplet/bubble, m/s^2.
C_D	Drag coefficient.
c_p	Specific heat, J/kgK.
D	Diffusion coefficient, .
$\mathbf{D}$	An averaged droplet diameter, $(4R^2+4R_0^2)/8RR_0$, m.
E	Eccentricity of vapor cap of droplet undergoing phase change.
g	Acceleration of gravity, m/s^2.
Gr	Grashoff number.
H	Ratio of convective to conduction coefficients, h/k, m^{-1}.
h	Convective heat transfer coefficient, W/m^2K, or step size.
I	Number of nodes in x direction in two dimensional analysis.
i	Enthalpy, J/kg.
J	Number of nodes in y direction in two dimensional analysis.
K	Indexed parameter used in Runge-Kutta solution
k	Conduction coefficient, W/mK.
L	Column length, m.

$LMTD$	Log mean temperature difference, °K.
M	Ratio of liquid density to vapor density.
$\tilde{M}$	Molecular weight of fluid or gas.
m	Mass, kg.
$\dot{m}$	Fluid mass flow rate, kg/sec.
n_B	Number of bubbles in control volume.
Nu	Nusselt number.
P	Pressure, Pa.
Pe	Peclect number.
Pr	Prandtl number.
$\dot{Q}$	Heat transfer rate or duty, W.
R, r	Droplet/bubble radius, m.
Re	Reynolds number.
S	Reference cross sectional area of droplet/bubble, m^2.
Sc	Schmidt number.
Sh	Sherwood number
T	Fluid temperature, °C.
t	Time variable.
U	Fluid velocity, m/s.
V	Volume (control volume or droplet/bubble), m^3.
w_o	Optimized weighting
X, x	Fluid quality (0=liquid to 1=vapor) or location in two dimensional analysis.

y	Non-specific variable or location in two dimensional analysis.
z	Column location (axial), m.
z_D	Normal distance of droplet rise, $z/2R$.

Greek

α	Thermal diffusivity, $k/\rho c_p$, m^2/s.
β	Droplet half opening angle, radians, or grid aspect ratio.
γ	Constant coefficient of time describing rate of change of temperature of fluid surrounding sphere.
Δ, δ	Difference.
η	Heat loss coefficient, $\dot{Q}_c/\dot{Q}_d$
κ	Parameter used in equation 16 defined in equation 17.
μ	Viscosity, Pa-s.
ν	Kinematic viscosity, m^2/s.
ξ	Parameter used in equation 52 and defined in equation 53.
π	3.141592654
ρ	Density, kg/m^3.
σ	Molecular property used in Chapman-Enskog theory (Cussler, 1984).
ϕ	Holdup ratio (also known as void fraction).
ψ	Stream function.
Ω	Molecular property used in Chapman-Enskog theory (Cussler, 1984).
ω	Vorticity, rad/s.

Subscripts

0	Initial condition.
1	From point of droplet injection or indexing.
2	From point at which bubble completes phase change or indexing.
3	Indexing
4	Indexing
B, b	Droplet/bubble.
C, c	Continuous fluid.
cd	From continuous fluid to dispersed fluid.
D, d	Dispersed fluid.
fg	Fluid to gas state change
FT	According to work by Fan and Tsuchiya (1990).
hc	High circulation regime.
i	Spatial index.
inj	Droplet injection regime.
j	Spatial index.
JS	According to work by Sideman (1966) cited by Jacobs(1988).
k	Step index
max	Maximum.
n	Time step index.
o	Outside (exterior) of droplet/bubble. . . . .
ph	Phase change regime.

S	According to work by Sideman (1964).
sr	Stable rise regime.
STr	According to work by Steinberger and Treybal (1963).
surf	At droplet/bubble surface.
T	According to work by Tochtani (1977b).
v	Vapor
vbr	Vapor bubble rise regime.
x	With respect to the x-axis or direction.
y	With respect to the y-axis or direction.
z	With respect to the z-axis or direction

ACKNOWLEDGEMENTS

Much gratitude goes to Dr. Robert Boehm for providing the opportunity to pursue this research (and this degree) by way of an invitation four years ago to work with him on a grant awarded for the the study of the topics presented in this dissertation. Much gratitude also goes to Cray Research, Inc. for funding the grants and to the National Supercomputing Center for Energy and the Environment who administer the Cray supercomputer at the University of Nevada, Las Vegas. In particular, I wish to recognize Joe Lombardo of the NSCEE staff for the countless times he was called upon to solve technical difficulties getting software encoded and running. Finally, and perhaps most importantly, the sincerest of all thanks go to my wife, Bette, for her unfaultering enthusiasm and encouragement.

CHAPTER 1

INTRODUCTION

Direct contact heat exchange is a process that offers a highly efficient means of transferring heat between fluids. It is a process that sees use in chemical, geothermal, and manufacturing engineering, though surprisingly, its use is not wide spread in these industries despite its many commonalities with widely used direct contact mass transfer processes. Current research is equally sparse compared to other conventional forms of heat exchange processes. The reasons for the paucity of current research and use are the general lack of understanding of the direct contact processes and the availability (or absence) of simulation software or codes that can aid in producing useful designs.

As the name implies, a direct contact heat exchanger (DCHX) is a heat exchanger that functions by bringing two *working* fluids - one *continuous* and one *dispersed* - in direct contact with each other. By using this kind of process, convection heat transfer takes place sans the impedance of channel walls or tubing common in other types of heat exchangers. Direct contact processes offer benefits over other conventional types of heat exchangers that include: 1) reduced capital cost due to generally simpler design, 2) higher operating efficiencies due to the elimination of the physical interfacing barriers, 3) the ability to achieve close temperature approaches, and 4) reduced operational maintenance due to the elimination of the fouling problem normally associated with tubes and channels.

Direct contact processes are not without limitations and though relatively few, they include 1) the necessity that the working fluids be at the same pressure, 2) the need for sufficiently differing densities to facilitate proper flow (countercurrent in most cases) and reseparation of the fluids, and 3) the requirement that the fluids be immiscible.

As stated, DCHX processes see selected use in industries and research. A notable example in the area of geothermal power production is the East Mesa heat extraction study (Orlander et al., 1983) conducted on the Salton Sea in southern California. In the East Mesa project, a 10-meter vertical device and supporting facilities were constructed at a geothermally active area of the Salton Sea. Utilizing isobutane and geothermal well brine, a DCHX was operated and monitored to study the efficiency of extracting heat from geothermal wells. Another notable example of direct contact use is a hot water generation application in the U.K. (Baker, 1987) where a DCHX was built and operated at a tannery. The tannery used the system to generate a high volume of hot water needed for the hide tanning process. Exhaust from a combuster resembling a jet engine is forced through a vessel of water, heating the water to a desired temperatures.

By no means a new concept - studies of various aspects of DCHX thermo and hydrodynamic mechanisms date to the 1940s and 1950s - the use of direct contact heat exchange is not widely employed, despite as stated, its close resemblance to direct-contact mass transfer. Examples like those cited are in fact relatively few. Much of the data and analyses being reported in the literature are derived from laboratory models or experimental prototypes.

One reason for the lack of practical use of this process has been the stated continued lack of understanding and adequate design tools for these systems. In some cases, classical means of calculating performance and efficiency fail to yield meaningful information when applied to direct-contact devices. Further, the thermo and hydrodynamic responses of fluids in direct contact are more complex. Performance evaluation requires detailed knowledge of the convection heat transfer processes associated with droplets and bubbles of one fluid passing through another. The complexity of two and three-phase convective transfer hinders finding simple rules of thumb. Systems that are in use today are generally the result of individual experience or iterative trial and error. There is a need for useable software code capable of simulating the direct-contact exchange processes.

A number of individuals have pursued studies in direct-contact heat exchange and related areas. Several exceptional works are particularly notable. Kreith and Boehm, Direct-Contact Heat Transfer (1988), is a compilation of articles addressing a broad range of issues related to the DCHX design process. A chapter of that text, authored by R. Letan, discusses rules of thumb for achieving DCHX designs. Jacobs (1988) presented a comprehensive treatise of the key mechanisms in DCHX processes.

A number of important works - Sideman (1964, 1966a, 1966b), Tochitani et al. (1975, 1977a, 1977b), Moresco and Marschall (1979), and Johnson and Marschall (1985) - have contributed to a better understanding of the various component processes such as the dynamics of and heat transfer to droplets and bubbles in various continuous media. A particularly in-depth treatise of the complex wake dynamics and heat transfer of bubbles is Fan and Tsuchiya, Bubble Wake Dynamics in Liquids and Liquid-Solid Suspensions (1990). The authors of this text provide many formulations and relationships, but suggest that much more work can be done in this complex field.

The purpose of the work herein is to create an integrated model using results and findings from the literature to simulate the performance of a selected range of DCHXs. In creating such a model, an economical means will have been developed that allows previewing DCHX designs and variations prior to actual construction and experimental testing. With energy use a constant issue in most industrial endeavors, the need for efficient production, extraction and use will always remain high. Direct-contact heat exchange is a highly efficient process that can potentially be used in more and more applications given that a clearer understanding of the processes can be achieved.

DCHX Systems and Processes

As stated in the opening paragraphs, the direct-contact processes function by bringing two working fluids in direct contact with each other. This is done in a device called the direct-contact heat exchanger. While variations of these devices are many, they generally consist of a vertical chamber or column accessible at both ends through which the fluids are introduced and extracted. The simplest of these devices is the open spray-column shown schematically in Figure 1. In this type of device, a continuous fluid (commonly water or brine) is pumped into the top of the column and allowed to pass unobstructed to the bottom of the column, where it exits from the device. As is often typical (e.g. geothermal applications), the continuous fluid enters at a high temperature and serves to heat a suitably selected dispersed fluid introduced at the bottom of the column. The dispersed fluid, generally of lesser density, rises buoyantly through the column, receiving heat from the continuous fluid until it reaches its exit at the top.

A system is *two-phased* if the continuous and dispersed fluid retain their state throughout the contact. If one fluid (often the dispersed fluid) undergoes phase change, the system becomes *three-phased.* Two-phase systems are further delineated as liquid-liquid, liquid-vapor systems, or (less frequently) liquid-solid systems.

Other types of DCHXs configuration have been developed with the intent of enhancing system performance and/or efficiency. Alternate design enhancements include packed beds, baffles, axial agitators, and sieve trays, to

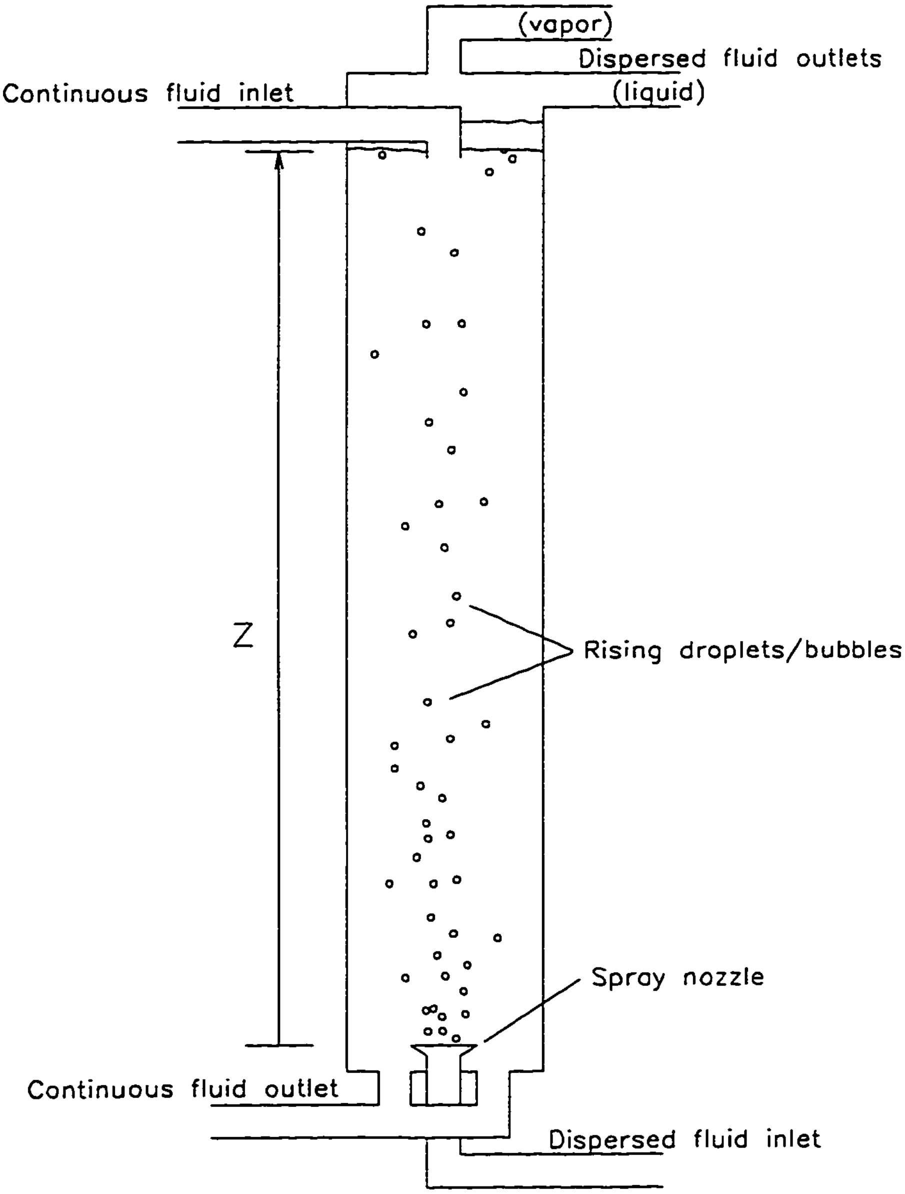

Figure 1 Schematic of Simple Spray-Column Direct-Contact Heat Exchanger (DCHX).

name a few. Each of these variations functions fundamentally the same as the spray-column in that counterflowing fluids are in direct contact. The enhancements are made to improve the heat transfer by creating better convection processes, each having merit to one degree or another.

In constructing a model that simulates the spray-column system, it is important to incorporate as many of the physical mechanisms as possible in as true a form as possible. By doing so, a resulting model has the most chance of working for a wide range of specifications and conditions. Realistically, this is not always possible, given the large numbers of factors influencing performance, but it is what should be strived for.

A number of works have been published on the modelling problem containing an assortment of approaches. Each has merits for the specific applications to which it applies. Some are limited in scope, describing a specific aspect of direct-contact exchange such as flooding (Golafshani and Jacobs, 1989) or droplet rise (Mansson and Marschall, 1994). Yet others provide for modelling of a specific configuration such as liquid-particulate solids (Tadrist et al., 1985) or two-layer melting (Hong and Saito, 1993). More generalized works on spray-column DCHXs have been published by Coban (1986) and Coban and Boehm (1986). Ay et al. (1994) closely followed the Coban and Boehm approach, work which to date is the most universally applicable. A one-dimensional model that models an enhanced version of a DCHX was developed by Summers and Crowe (1991). This work addresses a packed bed system and agrees reasonably with experimental data. The Coban and Coban and Boehm works comprise a fine attempt at modelling the hydro and thermodynamic responses of a three-phase counterflow direct-contact heat exchanger. The modelling sets down a solution algorithm for the governing one-dimensional differential equations, using a variety of supporting empirical expressions. The heat transfer processes are delineated for two and three-phase systems where liquid droplets or gas bubbles rise through a counterflow continuous fluid. Results of the model are compared (apparently favorably) with experimental data from a laboratory device.

Reconstruction of the Coban model - first by direct transcription of the published code and then by recreating a code from the analysis and algorithms - was undertaken as background for this work. Transcription of the code was unsuccessful. Recreating the code based on the algorithms given led to the discovery of suspected errors in some of the analyses. The apparent agreement between predicted and experimental data appeared to be due to offsetting errors, thus leaving open the need for a generalized model capable of correctly simulating DCHX processes.

The primary physics involved in spray-column DCHXs is the heat transfer to and from droplets and bubbles rising through a counterflowing medium. From injection to exit, a dispersed fluid can pass through six separate regimes. The six regimes are divided between the liquid and vapor phases of a typical three-phase system. These regimes - represented in Figure 2 - are *droplet injection, high circulation, and stable rise* for liquid droplets and *onset of phase change, two-phase* *rise, and vapor bubble rise* for vapor bubbles.

The *droplet injection* regime is characterized by rapid droplet growth and internal fluid circulation. It is found that convection heat transfer is high due to the internal circulation. Depending on the injection flow rate, orifice size, fluid properties, etc., a droplet being released from a nozzle takes on a myriad of elongated shapes until it separates. Time spent in this regime is very brief.

The *high circulation* regime begins at the point a droplet separates from a nozzle port. As the name implies, it is initially characterized by high internal droplet circulation. Droplets in this regime, however, also see a decay of the circulation as internal viscous forces dampen fluid motion. Time spent in this regime is also brief, lasting only ten to twenty droplet diameters. Droplet shapes continues to oscillate, elongating and flattening.

Culbreth (1983) and Johnson (1985) have studied and photographed the behavior of the first two regimes in simple water-oil experimental systems. Culbreth and Marschall observed that a significant amount of heat transfer occurs within the first ten droplet diameters after injection. This is an important point. Modelling works to date have not accounted for high circulation at or near injection, perhaps due to the brevity of these regimes. Droplets have, instead, been treated as stable, well-formed spheres during their traverse of the DCHX. Jacobs cited several works addressing heat transfer of circulating droplets.

The *stable rise* regime is that portion of the liquid droplets' travel in which the most time is spent. It is a condition that contrasts sharply with the injection regimes in that the droplet's oscillating shape dampens, taking on a constant spherical shape. Further, internal circulation has also decayed completely. A droplet remains in this regime until onset of vaporization in the case of a three-phase system or until it exits the exchanger in the case of a two-phase one. In a three-phase system, this regime, when combined with the injection and high circulation regimes, is considered the pre-heating portion of a DCHX.

Heat transfer to a droplet during stable rise differs dramatically from the highly circulating conditions preceding it. Heat transfer to a non-circulating droplet is dominated by the internal resistance to transfer, with the droplet taking on the thermal characteristics of a solid sphere. Construction of a heat transfer expression must account for both the exterior convection and internal resistance. In this case a series resistance is formulated. Sideman et al. (1964, 1966a, 1966b) studied heat transfer of stable droplets rising in a continuous medium. Tochitani et al. (1975, 1977b) studied the exterior convection of droplets of varying qualities. Expressions provided by both sources are used in this work.

Onset of phase-change begins at the end of the preheating of the dispersed fluid and like injection, is a brief regime. Here, very little literature material is available. It reasons that there is an increase in internal circulation due to the rapidly changing droplet/bubble shape. Typically the vaporized fluid collects in the upper portion of the droplet/bubble entity, forcing the liquid portion into a cup like shell. The change in shape is rapid, due to the extreme differences in densities. A droplet quality (mass) change of only a few percent can result in an order of magnitude change in volume. The heat transfer at this stage jumps, resembling that of the injection and circulation.

The *two-phase rise* regime is where a bulk of the overall heat transfer occurs in a three-phase system. This is due to high individual droplet heat transfer as a result of both circulation and the shifting of the droplet's shape from spherical to shell-like. The change in liquid to a shell eliminates much of the internal resistance, allowing the higher external convection to dominate the heat transfer process. Further, there is a high amount of absorption of the transferred heat by the vaporization process, which holds the dispersed fluid's temperature constant during this regime. Tochitani et al. (1977a, 1977b) studied the two-phase rise regime experimentally for a water/n-pentane system and offered expressions for the heat transfer and hydrodynamic motion for qualities up to approximately 0.5. At this point, the upper half of a droplet/bubble appears to be predominantly a vapor bubble having some degree of circulation. Recent work by Fan and Tsuchiya (1990) addresses vapor bubbles of varying shapes in diverse conditions, focusing on the mass transfer characteristics. An analogy is made between mass and heat transfer which provides a means of deriving expressions for use in modelling the heat transfer in this regime.

The final regime is the *vapor bubble rise*. Once again, little is found in the literature about the transition. Coban viewed the vapor bubble as being similar to stable liquid droplet with high internal resistance. Due to the generally low viscosities associated with gases, it is more likely that the bubble interior is

circulating. Better heat transfer expressions for vapor bubbles - making use of the heat transfer/mass transfer analogy - are found in the Fan and Tsuchiya work.

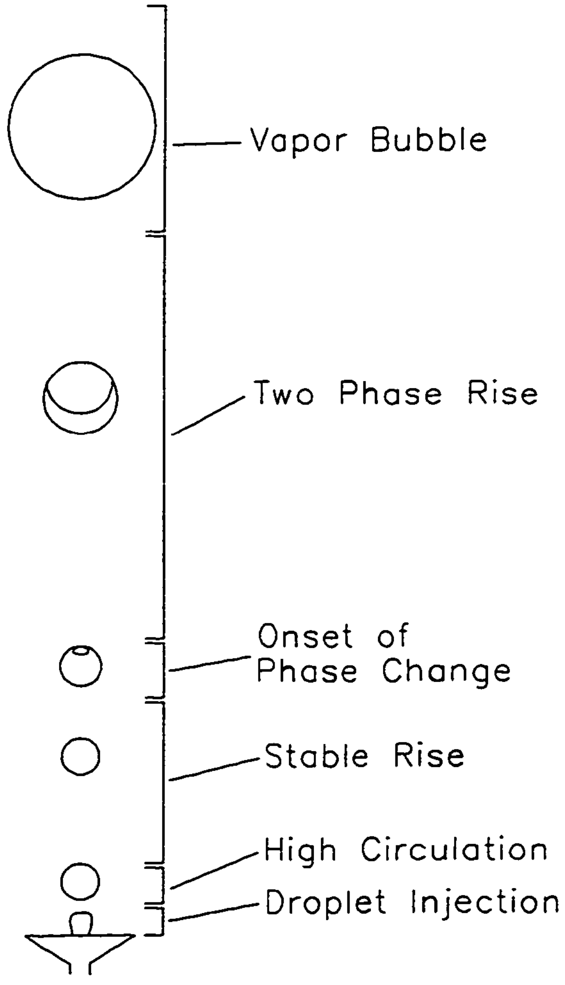

Figure 2 Droplet/Bubble Regimes Occurring in a Three-Phase Spray-Column DCHX.

The task of assembling a DCHX model is primarily one of assembling a matrix of expressions that describe the heat transfer for each of the regimes which can be used by an algorithm for solving the governing differential equations. Such a matrix must be constructed from available expressions reported in the literature. The selections and detailed discussion of the expressions used in this work are outlined in Chapter 2.

Dual Utility Optimization

Optimization of DCHX systems can be approached in several different ways depending on the type of outcome desired. One of the features of heat exchangers in general - direct-contact included - is the flexibility of their use. One of the uses is maximum heat extraction/addition to a working fluid. For this type of application, one might assume that an abundant amount of a source fluid is available and is used to extract or add as much heat as possible to a target fluid of limited supply. The resulting temperature profile might appear like the hypothetical case shown in Figure 3. An abundant low temperature continuous fluid (perhaps water) is supplied to the top of a DCHX to cool a high temperature dispersed fluid (perhaps oil from some application). The continuous fluid undergoes little change in temperature while the dispersed fluid experiences a wide temperature change. The criteria by which this type of system is considered optimized might be the point at which no appreciable improvement is made in the dispersed fluids temperature excursion, despite the continued increase in the continuous fluid's flow.

A second use of DCHXs is the specific control of a target fluid's temperature. In an application of this type, a DCHX is used to heat or cool a fluid to a specific temperature. This requires adding or extracting heat at a specific rate and presents more challenge since any one of a number of factors can affect DCHX transfer. A profile for this scenario might look like that shown in Figure 4, where the dispersed fluid might enter the system at a high temperature and must exit at a specific level. The continuous fluid's flow rate and inlet temperature are key factors affecting the outcome of the system as are other basic parameters such as device length, dispersed droplet size, operating pressure, etc. A parameter by parameter analysis would be required to determine the right combination of parameters needed to achieve the target exit temperature.

A third distinct use of a DCHX is the transfer of heat in such a manner as to utilize both fluids to the maximum extent possible. This might be desirable when both fluids are of limited supply or the cost of pumping the fluid to and/or through a system is significant. If achieved, a system can be said to have *dual utility optimization*. A characteristic of the fluids' temperature profiles for this type of optimization would be that they undergo equal excursions (are parallel) and that they have as close of an approach as possible. The primary factors affecting a successful balance of parameters are individual flow rates and specific heats.

Figure 5 depicts two cases of dual utilization profiles. In Case 1 (dashed lines), two fluids are input at rates that result in each having the same change in temperature from input to output. However, the approach of temperatures is not good and although both fluids are undergoing equal temperature excursions, the system is not optimized under the criteria stated for dual utility. Case 2 (solid lines) depicts a second set of conditions that does result in both equal excursions and a much improved, if not maximized, approach of temperatures. (It is noted that the ideal case of the temperature profiles lying on top of each other is not possible since there would be no temperature differential to create heat transfer.) Case 2 qualifies as being optimized for dual utility. It is the dual utility application that is the focus of the optimization analysis in this work.

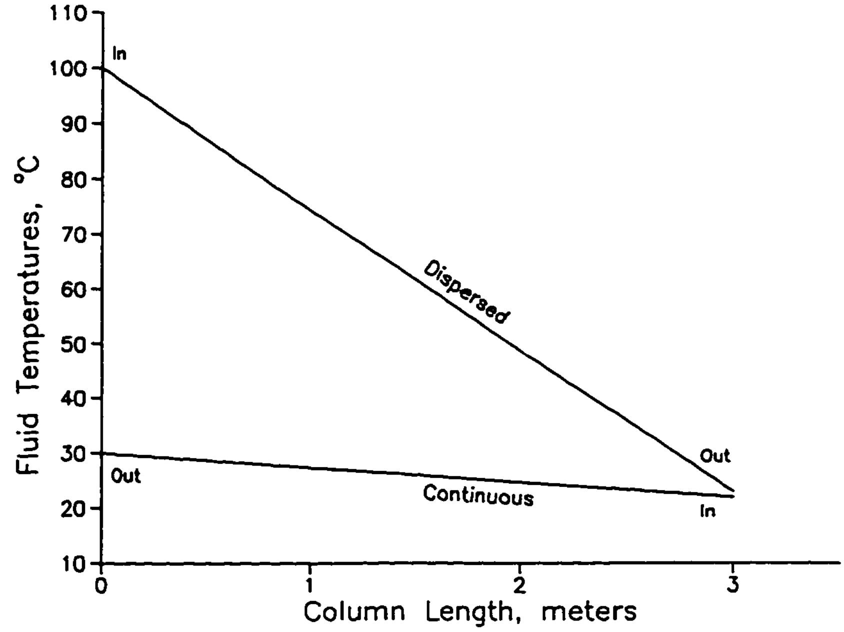

Figure 3 Example Temperature Profile of a DCHX Optimized for Maximum Heat Exchange.

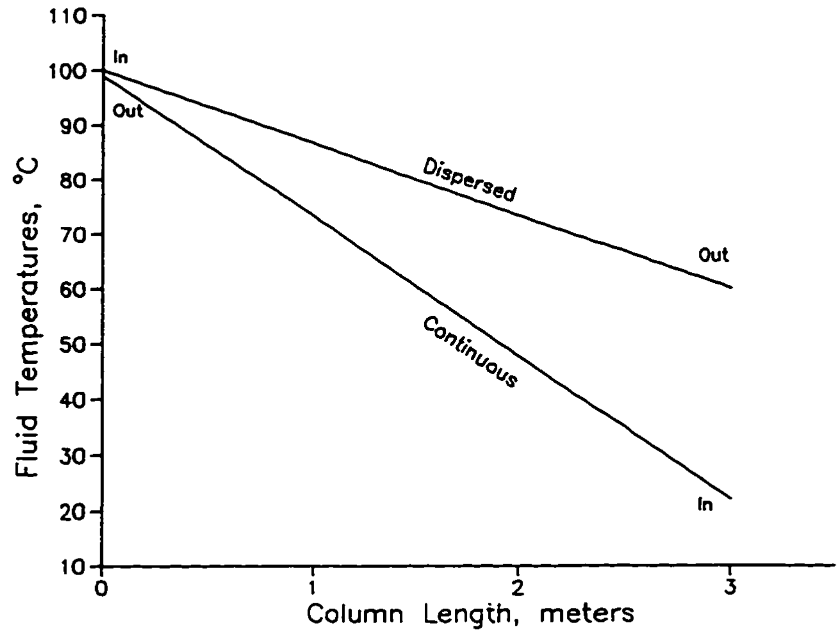

Figure 4 Example Temperature Profile of a DCHX Optimized for Specific Target Temperatures.

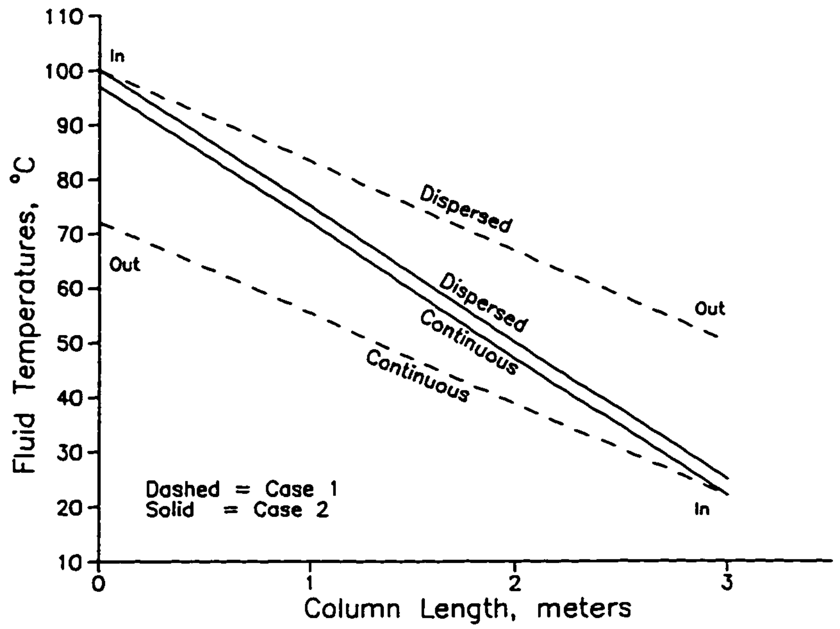

Figure 5 Example Temperature Profiles of a DCHX Optimized for Dual Utility. (Dashed Lines = Partially Optimized, Solid Lines = Optimized.)

A need then arises for an indicator that measures the conditions - parallel and close approach of the temperature profiles - and determines dual utility optimization. Classical expressions such as effectiveness and log mean temperature difference (LMTD) are not useful in providing a measure for this type of outcome. LMTD is of some value in indicating closeness of approach. The lower the value, the smaller (closer) the *average* temperature difference is. However, it does not provide any information as to the parallelism exhibited by the profiles. Effectiveness, as classically defined, offers insight into the system's efficiency and might be of use in determining maximum performance as a trade off to some parameter. A new indicator is therefore needed and developed in this work.

The indicator, referred to as the Dual Utility Optimization (DUO) characteristic, combines a measure of profile parallelism and closeness of approach into a single expression that ranges between 0 and 1. Values approaching 0 indicate poor utilization, and values approaching 1 indicate high utilization noting that the extreme cases - values *equal* to 1 - are not possible as this situation would indicate coincidental profiles, a condition precluded by the physics of heat transfer. The development of the characteristic is discussed in Chapter 2 and results presented in Chapter 3. The indicator is especially beneficial in that it is a well-behaved and stand alone indicator for comparing a wide variety of parameter combinations.

Sieve Tray Enhancement of DCHXs: Two-Dimensional Analysis

It has been found that DCHX output can be measurably enhanced by the adaptation of a spray-column into one of several alternate types of DCHX devices. These adaptations include the addition of packing materials, sieve trays, or column agitators, to name a few. In all cases, fluid flow is altered in such a way that heat transfer is increased. The mechanism responsible for the increase varies from adaptation to adaptation, and it is not completely known what specifically about the altered flows causes the enhanced output, though it is shown that increased convection is generally a greater factor than residence time. In a packed column, hundreds of irregularly shaped objects are randomly packed within the counterflow column, inhibiting the free rise of droplets and bubbles. Droplets continuously collide with the objects and must pass to either side as they restart their rise. It is believed that this slows the rise, thus increasing the droplet/bubble residence time. A second belief is that the continual colliding with the packing objects creates high circulation and strips off unwanted wake, which leads to axial mixing (Kehat, 1971, Letan, 1965).

A second type of enhancement is the addition of a series of partially spanning alternating sieve trays throughout the column. The sieve trays force the continuous fluid to snake back and forth along the length of the column. As a result of this motion, cross flows are set up. Droplets of dispersed fluid are swept left and right as they travel the length of the column. The heat transfer enhancement mechanism is believed to be several fold. First, there is a higher

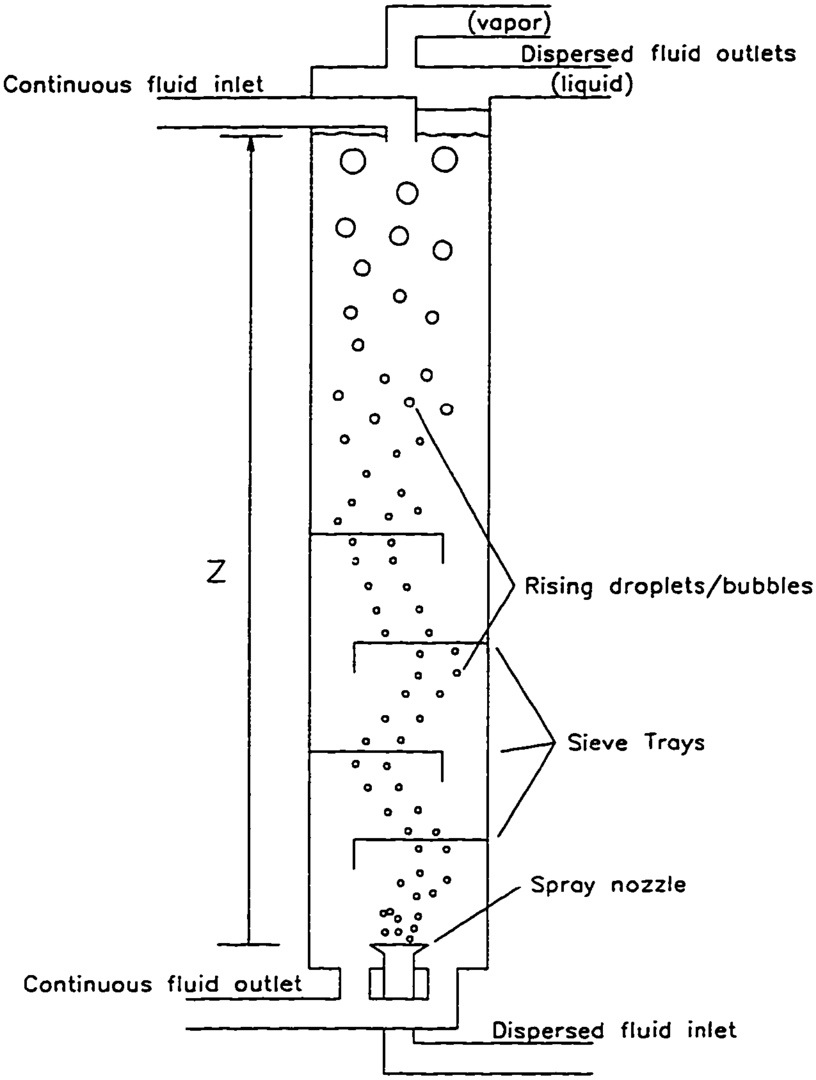

Figure 6 Schematic of Sieve-Tray Enhanced Direct-Contact Heat Exchanger.

relative velocity since the droplets now have a two component velocity vector. Second, the droplets' size, circulation and wake effects are reset each time the dispersed fluid passes through the perforations of successive trays. Experimental tests by Mines (1983) and Jacobs (1985) have shown gains in efficiency and output by the use of trays.

It would be desirable to investigate which mechanisms are responsible for enhancing performance. To accomplish this, multi-dimensional modelling is needed. The sieve-tray enhancement is a likely candidate for study in that it can be viewed from a two-dimensional perspective. Since sieve-tray design is basically modular (stacked trays), a single module is readily adapted to two-dimensional modeling. Ultimately, the goal is to create a two-dimensional mapping of the flow for analysis. With this mapping, droplet dynamics and thermal response are studied and compared to unenhanced DCHX cases. It is shown that there is an increase in efficiency in heating (preheating) dispersed liquid droplets.

Modelling can be done by using a commercial fluid dynamics code or a written routine. Any identified influences can then be incorporated back into the one-dimensional model and compared to experimental results. Two-dimensional modelling of DCHXs is not known to have been accomplished by anyone to date. Since results can lead to valuable insights, it is a logical step in the investigation into the sieve-tray effect.

CHAPTER 2

MODELLING DEVELOPMENT AND BACKGROUND

A simulation of the one-dimensional (axial) heat transfer occurring between a dispersed and continuous fluid in a DCHX device requires a numerical modelling of governing and supporting equations describing the hydrodynamic and thermodynamic mechanisms. In this work, both a one-dimensional and a simplified two-dimensional model are developed and discussed. In addition, a new indicator for evaluating one particular case of optimization is presented, and its merit is shown.

One-Dimensional Model

The one-dimensional (axial) model serves as the principle tool for the work studied. It is based on a governing system of ordinary differential equations (ODEs) describing the physics of one-dimensional momentum and energy transfer. A set of supporting expressions is included to determine the various components comprising the ODEs. The model simulates a spray-column DCHX which consists of a vertical tube through which two working fluids pass in counterflow: the heavier introduced at the top and the lighter at the bottom. The dispersed fluid - generally the fluid with the smaller flow rate - is introduced into the continuous fluid through a spray nozzle which causes the dispersed fluid to form droplets or bubbles. The fundamental conditions for this type of system to operate is that there is a sufficient density differential to support a counterflow and that the two fluids are immiscible and at different temperatures. Either fluid may be the heavier or lighter. Similarly, either fluid may be the hot or cold component. A robust model should be flexible enough to account for a variety of combinations, making appropriate selections of transfer expressions to accommodate any one of the four possible configurations.

Governing Equations

In modelling the one-dimensional response of the above described device, the axial dimension of the column is divided into control volumes. A set of governing ODEs for the control volumes is derived (Coban, 1986) from the Navier-Stokes equations by applying fundamental assumptions germane to one-dimensional steady state two phase flow. The result is a set of three - one momentum and two energy - equations describing the hydrodynamic and thermodynamic activity. The equation set is solved numerically for each control volume given a set of conditions existing at the boundaries. The governing set of ODEs are as follows:

$$\frac{dP}{dz} = -\frac{\dot{m}_d}{A}\frac{dU_d}{dz} - \frac{\dot{m}_c}{A}\frac{dU_c}{dz} - [\rho_c(1-\phi) + \rho_d\phi]\,g \qquad 1$$

$$\frac{di_d}{dz} = \frac{A}{\dot{m}_d}\frac{\dot{Q}_d}{V} \qquad 2$$

$$\frac{di_c}{dz} = -\frac{A}{\dot{m}_c}\frac{\eta\dot{Q}_d}{V} \qquad 3$$

with friction between the continuous fluid and device walls assumed negligible. This system can be solved numerically by the application of a Runge-Kutta scheme of an appropriate order. In this work a 4th order scheme is chosen.

Since the ODEs are comprised of a number of components which must be determined at each control volume location, it is necessary to have appropriate expressions for these components. A review of the literature - particularly of works by the individuals mentioned in Chapter 1 - provides candidate expressions for each of the various components. Generally, the expressions are empirically based. Where needed, theoretical derivations and modifications are developed. An underlying criterion for selecting expressions for use in the model is that they must be as universally applicable as possible. This presents some difficulty in that the set of expressions is large and some must be extended to cover ranges of conditions not specifically stated but not specifically excluded. Theoretical variations and modifications are discussed where appropriate in the following expressions.

Dispersed Fluid Velocity

Two separate expressions are used for determining dispersed droplet or bubble velocities. An expression from Raina et al. (1984) provides a correlation for rising ellipsoidal fluid particles.

Rising droplets

$$U_d = \frac{1.1547\left[\left\{1-\frac{\rho_d}{\rho_c}\left(\frac{R_0}{R}\right)^3\right\}\left(\frac{2Rg}{C_D}\right)\right]^{1/2}\mathbf{D}^{\left(\frac{5}{6}-\frac{\mathbf{D}}{\mathbf{T}_c}\right)}}{\left[\frac{T_c^2+T_d^2}{2T_cT_d}\right]^{\mathbf{D}}\left[\frac{C_{p_c}\mu_c}{k_c}\right]^{\frac{R_o}{1.6R}}} \qquad 4$$

Falling Droplets

A simpler expression cited by Raina which accounts for the effect of gravity on the density differential is used for falling droplets.

$$U_d = \left[\frac{4}{3}\left(\frac{\rho_d - \rho_c}{\rho_d}\right)\frac{2Rg}{C_D}\right]^{\frac{1}{2}} \qquad 5$$

Holdup

The holdup ratio is the ratio of dispersed fluid volume to total volume of a control volume segment. For any given location, the ratio is found by

$$\phi = \frac{\dot{m}_d}{\rho_d A U_d} \qquad 6$$

Continuous Fluid Velocity

The continuous fluid velocity is found from simple continuity.

$$U_c = \frac{\dot{m}_c}{A(1-\phi)\rho_c} \qquad 7$$

Drag Coefficient

Equations 4 and 5 require a droplet drag coefficient. A number of expressions are given in various references. A suitable expression is taken from White (1974, 1991):

$$C_D = \left[\frac{24}{Re} + \frac{6}{1+Re^{1/2}} + 0.4\right] \cdot \left[\frac{1+\frac{2\mu_c}{3\mu_d}}{1+\frac{\mu_c}{\mu_d}}\right] \qquad 8$$

Equations 1-8 comprise the hydrodynamic modelling of a spray-column DCHX and can be shown to simulate with reasonable accuracy, observed reactions in a selection of applications. It must be noted that a single universally applicable expression for every combination of fluids, flow rates, and DCHX geometries is not found. The above set of expressions represent a generally good simulation approximation of DCHX dynamics. Quantitative use of such a simulation for micro analysis of DCHX operation must be kept in perspective.

Dispersed Droplet/Bubble Radius

The dispersed fluid droplet/bubble size is a key element of the velocity expressions and Reynolds number calculations. Droplet/bubble radius is readily determined if one notes that eliminating droplet/bubble breakup and/or coalescence results in the conditions that droplet/bubble integrity is maintained. Thus the droplet/bubble radius is determined by a simple conservation of mass.

$$R = R_0\left(\frac{\rho_{d_0}}{\rho_d}\right)^{1/3} \qquad 9$$

Droplet/Bubble Population

Droplet/bubble population is a key element in the heat transfer calculation. It is calculated from control volume size, droplet/bubble size and knowledge of the holdup. In some works it has been assumed that control volume droplet/bubble population per control volume remains constant along the length of a column. A more detailed examination of continuity requirements suggests otherwise. Asuming droplet/bubble integrity, if the control volume population were constant then an increase in droplet/bubble size would affect the holdup. However, holdup is determined analytically from equation 6 and not linked to droplet/bubble size or population; therefore, a constant population is not possible. By noting that the mass flow rate and density in equation 6 reduces to a volumetric flow rate and that the time variable cancels with the time variable of the velocity term, the holdup ratio, ϕ (Equation 6), can be rewritten as

$$\phi = \frac{\frac{4}{3}\pi R^3 n_B}{A dz} \qquad 10$$

where n_B, the control volume droplet/bubble population, is introduced. This expresses the holdup more fundamentally to its definition and gives a means for determining the population for any given control volume.

$$n_B = \frac{\phi A dz}{\frac{4}{3}\pi R^3} \qquad 11$$

The total number of bubbles passing through a control volume *during a specified length of time* will be constant in keeping with continuity. However, the instantaneous population increases or decreases with a decrease or increase in droplet/bubble velocity, respectively.

Heat Transfer

The heat transfer mechanisms are of the most concern in modelling. It is described in the ODEs by the $\dot{Q}$ term in equations 2 and 3 and is the aggregate sum of the convective transfer to or from all droplets/bubble present in the control volume. It is simply and classically expressed as

$$\dot{Q} = hA_B(T_c - T_d)\,n_B \qquad 12$$

The key element of equation 12 is the convection transfer coefficient, h. The coefficient must be determined for the two-phase (liquid or gas) or three-phase condition present at any given location in a DCHX. As pointed out in Chapter 1, the dispersed fluid can pass through a number of regimes (e.g. circulating, stable, phase change, etc.) as it passes through a device. This potentially requires a matrix of expressions for modelling since any or all regimes can be present in a given DCHX configuration.

Expressions for droplet/bubble convection heat transfer are available in the literature. Most are empirically based. Many are application specific. In selecting suitable candidates for use in modelling, like the hydrodynamics, only those that are most universally applicable are retained.

A difficulty exists in that not all regimes can be precisely modelled. Heat transfer at droplet injection is an example. This regime is characterized by high droplet circulation and can account for a substantial portion of the total heat transfer in two-phase liquid system (Culbreth, 1986). The findings upon which this statement is made are specific to a water/oil system where single oil droplets were injected into the continuous stream from a port consisting of a small tube. The tube allowed the continuous medium to flow nearly completely around the forming droplet and continue unimpeded. This injection process contrasts significantly with what might actually be found in a commercial system where a spray nozzle large enough to provide numerous streams of droplets would be required and around which the continuous fluid would have to pass. The continuous flow would, at injection, be radically different from the interaction described by Culbreth. This suggests that the injection heat transfer may be attenuated, especially since the effect was found to occur in only approximately the first ten droplet diameters of rise.

In this modelling, the first two regimes are combined and not uniquely modelled. Rather, they are treated as a weighted influence. The influence is added to expressions used for describing the transfer coefficient for droplets in the next regime - stable rise. Reasonable modelling effort precludes any greater detail.

Stable Rise

Stable rise is characterized by a droplet rising with negligible internal circulation and stable geometric shape. Thus the transfer mechanism is that of exterior convection coupled with internal conduction, with the internal conduction resistance being the dominant factor. An expression for the external convection comes from Tochitani et al. (1977b), and is given as

$$h_T = \frac{k_c}{2R}\left(\frac{2|U_d|R}{\alpha_c}\right)^{1/3} \qquad 13$$

The internal resistance was studied by Sideman (1964), and is given by

$$h_S = .00375\frac{2|U_d|R}{\alpha_c}\left[\frac{1}{1+\frac{\mu_c}{\mu_d}}\right]\frac{k_c}{2R} \qquad 14$$

The above two expressions are coupled in a series resistance combination which gives the convection transfer coefficient for the stable rise regime.

$$h_{sr} = \frac{h_S h_T}{h_S+h_T} \qquad 15$$

Injection/High Circulation Influence

A correlation for the internal transfer for rapidly circulating drops is cited by Jacobs (1988) as being recommended by Sideman (1966) for best determining the transfer for this type of droplet. The correlation is given as

$$h_{JS} = [2 + 1.13(Pe\kappa)^{0.5}]\frac{k_c}{2R} \qquad 16$$

where

$$\kappa = 1 - 1.45\frac{1+3\frac{\mu_d}{\mu_c}}{1+\left(\frac{\rho_d\mu_d}{\rho_c\mu_c}\right)^{\frac{1}{2}}}Re_c^{-\frac{1}{2}} \qquad 17$$

Like the stable rise expression, the above expression is used in a series combination with the Tochitani expression for the exterior transfer. A decay function is used to facilitate the transition between this and the stable rise regime. The weighted decay is effective to approximately thirty droplet diameters. The final expression becomes

$$h_{inj/hc} = \frac{h_T h_{JS}}{h_T + h_{JS}}\left[1 - \left(\frac{z_{D_1}}{30}\right)^2\right] \qquad 18$$

Phase Change

The convection process for a droplet undergoing phase change was specifically studied with some limitations (Tochitani et al., 1977). One notes that a phase changing droplet adopts a liquid shell-like structure that cups around the vapor region. Tochitani studied the heat transfer process around a phase changing droplet to include a change of fluid quality up to approximately 0.5. This corresponds with a bubble that appears to be quite far along in the phase change since the quality is based on the ratio of mass changed. The physical size of the vapor portion would be considerable compared to the remaining liquid at a quality of 0.5. His observations led to the expression given below.

$$Nu = 0.463 Pe_c^{1/3}(\pi - \beta + 0.5\sin 2\beta)^{2/3} \qquad 19$$

wherein a transfer coefficient is obtained

$$h_T = .463 Pe_c^{1/3}(\pi - \beta + 0.5\sin 2\beta)^{2/3}\frac{k_c}{2R} \qquad 20$$

β, the droplet half opening angle used in the above two expressions, is described as shown in Figure 6 and defined analytically as

$$3\cos\beta - \cos^3\beta + 2 = \frac{4(1 - X_d)}{1 + X_d(M-1)} \qquad 21$$

where X_d is the droplet's quality and M is the ratio of liquid density to vapor density.

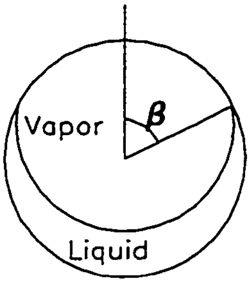

Figure 6 Geometry of Droplet Undergoing Phase Change with Droplet Half Opening Angle, β, Defined.

At a point at which the quality reaches approximately 0.5, an expression is required for the predominantly vapor bubble structure. This structure exhibits some circulation and rises at a higher velocity than while the fluid was liquid. Some works, including Coban's, viewed the vapor portion as non circulating and thus attributed a high resistance to this region, resulting in little or no heat transfer. It can be argued that low internal bubble viscosity allows for circulation. Of the several heat transfer expressions explored, the most reasonable is one for a spherical cap bubble derived from a mass transfer analogy of the heat transfer taken from Fan and Tsuchiya (1990), given by

$$Sh = 1.79\frac{(3E^2+4)^{2/3}}{E^2+4}Pe^{1/2} \tag{22}$$

Noting that heat and mass transfer are approximately related by

$$\frac{Sh}{Nu} = \left(\frac{Sc}{Pr}\right)^{1/3} \tag{23}$$

then

$$h_{FT} = \left[1.79\frac{(3E^2+4)^{2/3}}{E^2+4}Pe^{1/2}\right]\left(\frac{Pr}{Sc}\right)^{\frac{1}{3}}\frac{k_c}{2R} \qquad 24$$

As was the case in the earlier regimes, a weighted combination of this and the Tochitani correlation is constructed. The weighting gradually filters in the spherical cap correlation. The Tochitani correlation is primary for droplet/bubble qualities less than 0.5, whereas the Fan and Tsuchiya expression is primary for qualities greater than 0.5. The weighted expression is

$$h_{ph} = (1-X_d^2)h_T + X_d^2 h_{FT} \qquad 25$$

Vapor Bubble Rise

Coban and Boehm (1986), developed a method for determining the heat transfer to or from a vaporized bubble based on the assumption that the bubble's vapor interior is stable (noncirculating). They noted that for the experimental cases they studied, the vapor bubble traveled through a portion of the continuous medium in which the temperature profile was more or less linear. As a result, they used an expression from Carslaw and Jaeger (1959) for the temperature distribution inside a sphere subject to a linearly increasing surface temperature. Neglecting the oscillatory terms, they used the expression to find the difference between the average temperature inside the sphere and its surface, which then would allow them to set up a conduction expression for the sphere at the surface. As mentioned, an assumption was made that the temperature of the surface would be that of the continuous fluid, and for most cases this would be approximately so, with only small error.

Following a similar analysis but beginning with an alternative expression from Carslaw and Jaeger that gives the interior temperature profile of a sphere due to the heat transfer from a continuous medium with a linearly varying bulk temperature, an expression is derived for the vapor bubble rise. This expression differs from the one used by Coban in that it does not presume the bubble surface equals the continuous medium temperature but rather relies on an exterior convective transfer coefficient. Like the Coban and Boehm analysis, only the steady state portion of this expression is used. The average interior and surface temperatures are found, and a conduction relation is set up. The starting expression is given by

$$T(r,t) = \gamma\left[t+\frac{r^2RH-R^2(2+RH)}{6\alpha RH}\right] \qquad 26$$

Appendix I contains a full derivation of the expression, describing the step by step reduction and use in a heat transfer expression given by

$$\dot{Q} = k_v A_B \frac{\gamma R}{3\alpha} = h_B A_B \gamma\left[\frac{R}{3\alpha\frac{h_o}{k_v}}+\frac{R^2}{15\alpha}\right] \qquad 27$$

An expression for the overall transfer coefficient which accounts for the outside convection and interior conduction into a vapor bubble in the vapor bubble rise regime, h_{vbr} (h_B in above), can be defined as

$$h_{vbr} = \frac{5h_o k_v}{5k_v + Rh_o} \qquad 28$$

This expression is similar to the expression used by Coban and Boehm in that it represents the series combination of the exterior convection resistance and the internal conduction resistance (See Appendix I). By noting that a separation of terms yields

$$h_{vbr} = \left[\frac{1}{h_o} + \frac{R}{5k_v}\right]^{-1} \qquad 29$$

the individual resistances are revealed. The last term is the same "*internal convection*" term derived by Coban and Boehm.

The h_o term is the convection coefficient acting between the continuous medium and the exterior surface of the bubble. A suitable expression for this term comes from work by Steinberger and Treybal (1963), cited by Sideman (1966) and is given as

$$h_o = h_{STr} = [2 + .569(GrPr)^{.25} + .347Re_c^{.62}Pr^{.31}]\left(\frac{k_c}{2R}\right) \qquad 30$$

It is of note to observe that if $5k_v/R << h_o$, then the heat transfer will be dominated by the $5k_v/R$ term

$$h_{vbr} = \frac{5k_v}{R} \; ; \quad \left(\frac{5k_v}{R} << h_o\right) \qquad 31$$

which is the Coban and Boehm result. In selecting the alternative starting expression, the same conclusion is essentially reached. The improvement in this version is that the external convection resistance around a vapor bubble is retained vis a vis assuming it to be negligible. Though a low exterior resistance (compared to higher internal resistance) is usually the case, an exception might occur in cases where bubbles of a gas with highly conductive properties rise through a liquid of poorly conductive properties. For modeling accuracy and versatility in simulating a wide range of candidate fluids, the new expression is incorporated.

The difference between the development of equation 28 and the development for the phase change regime is the assumption of a stable vapor bubble compared to a circulating vapor cap on the two-phase bubble. To facilitate a smooth transition between the regions, a weighting function is once again used to gradually eliminate the influence of the previous regime during the first ten bubble diameters of the stable vapor rise. The expression becomes

$$h_{vbr_{transition}} = .5h_{vbr} + .5h_{PT}\left[1-\left(\frac{z_{D_2}}{10}\right)^2\right] \qquad 32$$

Equations 13-32 comprise the matrix of expressions required for determining the heat transfer to/from droplets/bubbles.

Mass Transfer

Mass transfer effects are not modelled, though with some modification to the governing equations and the addition of several more supporting expressions, they could be. An alternative form of equation 3 would appear as

$$\frac{di_c}{dz}=-\frac{A}{\dot{m}_c}\frac{(\eta\dot{Q}_d+\dot{m}_{cd}i_{fg_c})}{V} \tag{33}$$

Note the inclusion of the term which accounts for the small amount of mass transfer occurring between the continuous fluid and a dispersed bubble. It is mentioned here for completeness though its significance is minor since mass transfer effects only apply to vapor-liquid interface conditions. If one eliminates the two-phase liquid-liquid and the three-phase preheating and phase change cases, and considering that the small amount of heat transferred due to vaporizing mass is negligible compared to the normal convective transfer in large systems, the total influence of mass transfer in modelling is small.

Thermodynamic, Thermophysical, and Diffusion Variables

Key variables, such as pressure, temperature, quality, density (or specific volume), and enthalpy, as well as additional supporting properties such as viscosity and conduction coefficients are determined from a specialized ensemble of subroutines. In the case of the thermodynamic (state) variables, a set of subroutines developed by Reimer et al. (1976a, 1976b, 1976c) is used. These routines take the input of any two of the five state variables and return the remaining three. Separate ensembles for water, hydrocarbons, and freons are included in the simulation and are integral to the modelling architecture. For those fluids not included in the Reimer routines, a special subroutine was constructed to provide thermodynamic values based on a curve fit of published tabulated data. Appendix IV lists the curve fit expressions used.

Thermophysical properties are determined from a subroutine written specifically for this work. Values are determined from a curve fit of tabulated data spanning reasonably wide ranges of temperatures. Data is derived from numerous sources such as Vargaftik (1975) and handbooks, (ASHRAE, 1989).

Diffusion

As noted above, a portion of the heat transfer analysis is based on the mass transfer analogy. Cussler (1984) details a method for determining the diffusion coefficient from the Chapman-Enskog theory. The calculations are easily coded into a subroutine and require only a few key characteristics of the gasses involved. The coefficient is found by

$$D = \frac{1.86\cdot 10^{-3} T^{3/2}\left(1/\tilde{M}_1 + 1/\tilde{M}_2\right)^{1/2}}{P\sigma_{12}^2\Omega} \qquad 34$$

where $\tilde{M}_1$ and $\tilde{M}_1$ are the molecular weights of the fluids/gasses and σ_{12} and Ω are molecular property characteristics detailed in the theory presented by Cussler. Expressions for these quantities, while not extensive, require some study of the theory. The quantity Ω, for example, is determined from tabulated data. As a result, the expressions are not repeated here.

ODE Solution Scheme

A solution to the governing equations (Eqs 1-3) is accomplished with a fourth order Runge-Kutta numerical scheme which is discussed in any of a number of texts on numerical analysis. Simply expressed, the scheme takes the following form for a function *y(x)* utilizing a step size, *h*.

$$y_{i+1} = y_i + \frac{1}{6}(K_1 + 2K_2 + 2K_3 + K_4) \qquad 35$$

where

$$K_1 = hF(x_i, y_i) \qquad 36$$

$$K_2 = hF(x_i+\frac{h}{2}, y_i+\frac{1}{2}K_1) \qquad 37$$

$$K_3 = hF(x_i+\frac{h}{2}, y_i+\frac{1}{2}K_2) \qquad 38$$

$$K_4 = hF(x_i+h, y_i+K_3) \qquad 39$$

and noting that the function, *F*, is the first derivative - defined by one of the ODEs - of the variable of interest.

The solution requires boundary conditions for the variables in the ODEs. Pressure, mass flow rates and both fluid velocities, as well as densities can be determined at the starting boundary (bottom of the column). However, only one temperature is known at this location. The other is known at the opposing end of the counterflow system. A solution, then, requires a guess of the temperature at the unknown location to fill out the compliment of boundary conditions. The Runge-Kutta is then allowed to *march* along the solution until the top of the column is reached, where a check is made of the temperature which was guessed at with the known specification. If the check reveals a temperature that is too high (or too low) , a downward (or upward) revision of the guess is made by bisecting the interval between the previous guess and upper or lower inlet temperature limit upon which the Runge-Kutta scheme is run again. This process - well known as a *shooting method* - is repeated until the solution marches to a correct ending temperature for the guessed variable.

Variable Mapping

The solution of the differential equations is accomplished by first calculating all the necessary supporting variables from known conditions from boundary conditions or previous control volumes and then using these variables in the differential equation set, always using updated values as they become available in the calculation order. An algorithm sets the order of calculation for the variables. It is desirable, for obvious reasons, that dependent variables are calculated with updated values of the variables upon which they depend as the solution *marches* along the length of the DCHX. Values of the fluids' densities, enthalpies, pressure, and temperature are readily available and independent of many of the physical quantities such as velocity and droplet size. The physical quantities are often dependent on other physical quantities and therefore occasionally mutually dependent. For these cases, values of certain variables from previous step in the solution scheme are used. Fortunately, the necessity for these approximations are quite few if the following mapping order is followed.

First, known values of the system pressure, P, and the fluids' temperatures are assigned and tracked. From this information, enthalpy, density, and quality are found from subroutines. The temperature is then used to ascertain the thermophysical properties - also from subroutines. Velocities are determined from input mass flow rates and entry conditions which then yield beginning holdup ratios. Velocity gradients are assumed to be zero at z=0 and then calculated by

$$\left(\frac{dU}{dz}\right)_i = \frac{U_i - U_{i-1}}{dz} \qquad 40$$

The simplest of the remaining dependent variables to calculate is the droplet radius, which depends only on the changing density, one of the first items known in the variable calculation algorithm. The complete order of variable determination is then

$\dot{m}_d \; \dot{m}_c \; R_0 \; A \; L \; dz \; x_c$ - input values

$P \; i_c \; i_d$ - from boundary conditions or previous step

$T_c \; T_d \; \rho_c \; \rho_d \; x_d$ - from fluid state subroutines

$\mu_c \; \mu_d \; k_c \; k_d \; c_p \; Pr$ - from thermophysical property routines - function of (T,P)

For the i_{th} calculation

$R \; (\rho_d)$

$\beta \; (\rho_d \; x_d)$

$Re \; (\rho \; U_{i-1} \; R \; \mu)$

$C_D \; (Re \; \mu)$

$U_d \; (\rho \; R \; g \; C_D \; T c_p \; k)$

$\phi \quad (\dot{m}_d \; A \; U_d \; \rho)$

$U_c \quad (\phi \; \dot{m}_c \; A \; \rho_c)$

$U_d' \; (U_d \; dz)$

$U_c' \; (U_c \; dz)$

$n_B \quad (A \; dz \; R \; \phi)$

$h \quad (Re \; Pr \; Gr \; R \; \beta$ *... and properties dependent on above)*

$\dot{Q}_d \quad (h \; R \; n_B \; T_c \; T_d \; k_c \; k_d)$

For the (i*th*+1) calculation (a result of a step in the Runge-Kutta solution of the ODEs)

$P \quad (\dot{m}_d \; \dot{m}_c \; U_d' \; U_c' \; \rho_d \; \rho_c \; g \; \phi \; A)$

$i_d \quad (A \; \dot{m}_d \; Q_d \; R)$

$i_c \quad (A \; \dot{m}_c \; Q_d \; R)$

The significance of this mapping is that only the Reynolds number requires using the dispersed velocity from the previous control volume so that it can be used in the drag coefficient and subsequently to find the new dispersed velocity value. With the variables calculated, the fourth order Runge-Kutta scheme can be implemented to find the pressure and enthalpy for the next (i*th*+1) location from which the temperatures, densities, and quality are found and carried forward for the next calculations.

Optimizing a Configuration for Dual Fluid Utility

As discussed in Chapter 1, optimization of a given DCHX depends on a number of parameters, not the least of which is the type of optimization - maximum extraction, target temperature, dual fluid utility, etc. - desired. The conditions of dual utilization are that both fluids undergo the same temperature excursion and that the differential at any given location, particularly at the ends, be a minimum. An indicator that accurately portrays whether or not this condition is met must account for both the parallelism in the profiles and the closeness of approach. Such an indicator is developed for evaluating the case of dual utility optimization. With this indicator, the ideal case - coincidental curves are achieved - is reflected by a value of 1.0. Any condition away from the ideal shows up as a degraded value (less than 1.0) with less and less desirable cases tending asymptotically to 0.

Heretofore, system evaluation has relied on traditional log mean temperature difference and various effectiveness measures to assess system performance. As stated, for the dual utility criterion, these measures fail, requiring an alternative means of measure. Appendix II outlines the development of a new indicator - hereafter referred to as the Dual Utilization Optimization (DUO) indicator - that identifies when dual utility is achieved. The indicator is defined by the following expression

$$DUO = \left(1 - \frac{LMTD}{\Delta T_{max}}\right)\left(1 - \left|\frac{\Delta T_{inlet} - \Delta T_{outlet}}{\Delta T_{max}}\right|\right) \qquad 41$$

The utility of this indicator is that it yields a value ranging between 0 and 1 with 1 being, theoretically, the ideal case. (It is noted that one cannot achieve the so called *ideal* case - coincidental profiles - since temperature profiles falling on top of each other would mean zero differential, resulting in no heat exchange.) The indicator is calculated and recorded as one or more parameters are varied over a target range. The optimum value of the parameter is found when a peak occurs in the indicator value, signifying that optimum dual utility has been reached. The peak will be some value less than 1 but generally will range above .75 in systems achieving dual utility. The indicator is of most use in two-phase systems due to more well behaved nature of the temperature profiles. The advantages of using this indicator are shown in the results presented in Chapter 3.

Two-Dimensional Modelling for Sieve Tray Analysis

It is desirable to expand one-dimensional modelling to two-dimensions for analyzing certain effects of DCHX enhancements such as the introduction of sieve-trays into the axial flow. To accomplish this, it is desirable to limit the analysis to examining a single module (two consecutive trays) of a sieve-tray stack and generating a schematically correct two-dimensional representation. Modelling of the continuous fluid flow is accomplished using commonly used computational fluid dynamics (CFD) methods. With the continuous fluid flow field known, the path of a rising droplet can be determined and plotted. Droplet path traces allow for calculating the heat transfer increase due to the apparent increase in relative velocity seen by a droplet. A comparison is made to a spray-column case. The first step of this process is the modelling of the two-dimensional flow of a sieve-tray module.

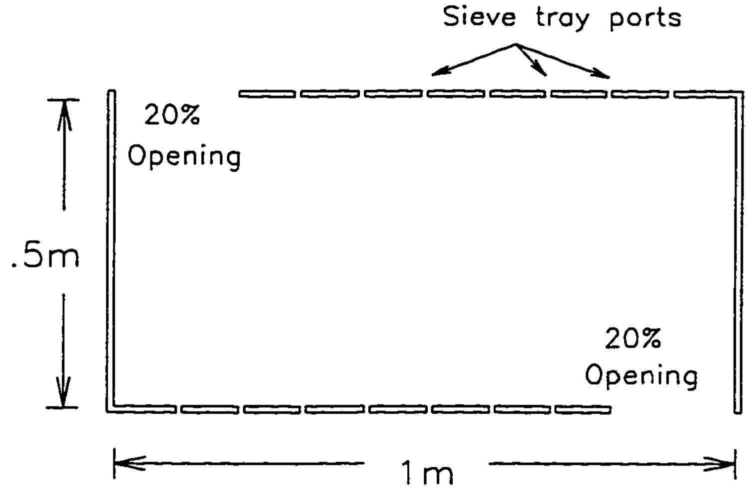

Figure 7 Schematic of Sieve-Tray Module (Two-Dimensional Model).

Flow Field Modelling

The velocity field of the sieve-tray module depicted in Figure 7 can be found by solving the two-dimensional vorticity transport equation given as

$$\frac{\partial\omega}{\partial t}+u\frac{\partial\omega}{\partial x}+v\frac{\partial\omega}{\partial y} = \nu\left(\frac{\partial^2\omega}{\partial x^2}+\frac{\partial^2\omega}{\partial y^2}\right) \qquad 42$$

The solution of the transport equation is then used in the stream function relationship

$$\nabla^2\psi = -\omega \qquad 43$$

In turn, the solution of the stream function equation yields the velocities from the commonly known relationships

$$u = \frac{\partial\psi}{\partial y} \qquad 44$$

and

$$v = -\frac{\partial\psi}{\partial x} \qquad 45$$

The above equations can be solved numerically through a variety of methods found in references such as Roache (1972). A method well suited for solving vorticity transport is the Alternating Direction Implicit (ADI) method, specifically, the Peaceman-Rachford ADI method. Formally, the two steps are denoted (using Roache's notation) by

$$\frac{\omega^{n+1/2}-\omega^{n}}{\Delta t/2} = -u\frac{\partial\omega^{n+1/2}}{\partial x}-v\frac{\partial\omega^{n}}{\partial y}+\nu\frac{\partial^2\omega^{n+1/2}}{\partial x^2}+\nu\frac{\partial^2\omega^{n}}{\partial y^2} \qquad 46$$

$$\frac{\omega^{n+1}-\omega^{n+1/2}}{\Delta t/2} = -u\frac{\delta\omega^{n+1/2}}{\delta x}-v\frac{\delta\omega^{n+1}}{\delta y}+\nu\frac{\delta^2\omega^{n+1/2}}{\delta x^2}+\nu\frac{\delta^2\omega^{n+1}}{\delta y^2} \qquad 47$$

The discretized forms of these equations are

$$\frac{\omega_{i,j}^{n+1/2}-\omega_{i,j}^{n}}{\Delta t/2} + u_{i,j}\left(\frac{\omega_{i+1,j}^{n+1/2}-\omega_{i-1,j}^{n+1/2}}{2\Delta x}\right) + v_{i,j}\left(\frac{\omega_{i,j+1}^{n}-\omega_{i,j-1}^{n}}{2\Delta y}\right)$$

$$= \nu\left(\frac{\omega_{i+1,j}^{n+1/2}-2\omega_{i,j}^{n+1/2}+\omega_{i-1,j}^{n+1/2}}{\Delta x^2}\right) + \nu\left(\frac{\omega_{i,j+1}^{n}-2\omega_{i,j}^{n}+\omega_{i,j-1}^{n}}{\Delta y^2}\right) \qquad 48$$

$$\frac{\omega_{i,j}^{n+1}-\omega_{i,j}^{n+1/2}}{\Delta t/2} + u_{i,j}\left(\frac{\omega_{i+1,j}^{n+1/2}-\omega_{i-1,j}^{n+1/2}}{2\Delta x}\right) + v_{i,j}\left(\frac{\omega_{i,j+1}^{n+1}-\omega_{i,j-1}^{n+1}}{2\Delta y}\right)$$

$$= \nu\left(\frac{\omega_{i+1,j}^{n+1/2}-2\omega_{i,j}^{n+1/2}+\omega_{i-1,j}^{n+1/2}}{\Delta x^2}\right) + \nu\left(\frac{\omega_{i,j+1}^{n+1}-2\omega_{i,j}^{n+1}+\omega_{i,j-1}^{n+1}}{\Delta y^2}\right) \qquad 49$$

In breaking up the process into two steps, the method creates two tri-diagonal matrices which are solved using commonly found subroutines.

Boundary conditions for set up described by Figure 7 are as follows. 1) Velocity (u,v) is zero along all walls. 2) Constant and equal vertical inlet/outlet velocity is imposed at the upper and lower 20% openings. 3) Vorticity is assumed zero along all boundaries; however, with each iteration on the solution, values at physical boundaries are updated using a forward or backward projection from locally adjacent nodes in the field. 4) The stream function is assumed zero along on continuous section of wall (i.e. left and lower segments). Incremental stream function changes across the field are summed until they reach the other continuous wall (upper and right). A check is made that to insure stream function is constant along the second surface.

A solution to Poisson's equation (Eqn 43) is found through the use of a Successive Over-Relaxation (SOR) method. Roache presents the following expression for implementing the SOR to the Poisson equation.

$$\psi^{k+1}_{i,j} = \psi^{k}_{i,j} + \frac{w_o}{2(1+\beta^2)}\left[\psi^{k}_{i+1,j} + \psi^{k+1}_{i-1,j} + \beta^2\psi^{k}_{i,j+1} + \beta^2\psi^{k+1}_{i,j-1} - \Delta x^2\omega_{i,j} - 2(1+\beta^2)\psi^{k}_{i,j}\right] \tag{50}$$

where

$$\beta = \frac{\Delta x}{\Delta y} \tag{51}$$

$$w_o = 2\left(\frac{1-\sqrt{1-\xi}}{\xi}\right) \tag{52}$$

$$\xi = \left[\frac{\cos\left(\frac{\pi}{I-1}\right) + \beta^2\cos\left(\frac{\pi}{J-1}\right)}{1+\beta^2}\right]^2 \tag{53}$$

ADI and SOR methods are discussed in detail in the cited reference. It is noted here that w_o is an optimized weighting factor based on the grid's aspect ratio.

Grid spacing is based on two factors - field location that is of most interest and computational requirements limitation. The aspect of most interest in this study is the path of droplets rising between the trays and their interaction with the continuous fluid in the central region of the module. Motion near the walls (in the boundary layer) is generally not important for this analysis. As a result, a relatively coarse grid suffices. The 1.0m x 0.5m module of Figure 7 is sectioned into a 22 x 22 grid. This sizing is selected so that when the boundary nodes are eliminated, a 20 x 20 grid of unknowns remains - a computationally economical size. A solution for the vorticity, stream function and velocities converges quickly, yielding a field that retains the characteristics of simple potential flow. A check of boundary layer buildup reveals that for the flows that are run, the buildup is less that the spacing of one grid. A coarse grid is satisfactory since a boundary layer study is not relevant to the overall scope of interest. Applying the above methods yields a mapping of the flow field velocity.

The paths of individual bubbles can be determined by numerically placing droplets at varying locations along the bottom tray and tracking them as they rise - two-dimensionally - through the flow field. Vertical velocity is determined using the expression stated for the one-dimensional simulation. Horizontal velocity is found through the simple application of physics. The horizontal component of flow will produce a horizontal drag force on an individual bubble that can be used in a simple mass-acceleration relationship.

$$F_{b_x} = m_b a_{b_x} = \frac{1}{2} C_{d_x} \rho \left(U_{c_x} - U_{b_x}\right)^2 S_b \qquad 54$$

wherein the horizontal acceleration is found by

$$a_{b_x} = \frac{\frac{1}{2}C_{d_x}\rho\,(U_{c_x}-U_{b_x})^2 S_b}{m_b} \qquad 55$$

With the acceleration known, both the droplet's location and total relative and absolute velocity can be determine from the classical physical relations

$$U_{b_x} = U_{b_{x_0}} + a_{b_x}\Delta t \qquad 56$$

$$x_b = x_0 + U_{b_x}\Delta t \qquad 57$$

$$y_b = y_0 + U_{b_y}\Delta t \qquad 58$$

$$|\vec{U}_b| = \sqrt{(U_{c_x}-U_{b_x})^2 + U_{b_y}^2} \qquad 59$$

noting that the droplet's relative vertical velocity is determined from the dispersed fluid velocity equation cited earlier in this chapter.

The heat transfer is determined from the appropriate transfer coefficient correlation - in this case, liquid drops rising through a liquid continuous fluid flow. It will be shown that the effect of the two-dimensional flow results in droplets that experience higher heat transfer due to the added horizontal component to the relative velocity.

Relatively few simplifying assumptions are made for this additional analysis over the one-dimensional case, the only significant one being that the continuous fluid remains at a constant temperature while within the confines of the sieve-tray module. This is reasonable since for cases such as water/n-pentane, the total temperature excursion of the continuous fluid (water) is only approximately 10° - 12°C over a column length of three meters, a bulk of which occurs during phase change. A one-meter module located in the preheating liquid-liquid portion experiences only a small temperature move. Another reason for this is that only single droplets are tracked, and thus any artificial temperature change would itself have to be an approximation.

Coding, Compiling, and Computing Requirements

Encoding of the algorithms and expressions described in this chapter is done in FORTRAN 77. The model was loaded and run on the Cray YMP2/216 supercomputer operated at the National Supercomputing Center for Energy and the Environment (NSCEE) located at the University of Nevada, Las Vegas. All modelling is written to be stand-alone, requiring no external library routines. A FORTRAN compiler resident on the NSCEE supercomputer optimally compiled the code (no special features invoked) for execution on the Cray YMP2/216.

Generally, runs made with the one-dimensional model required 30 to 50 seconds of CPU time on the Cray YMP2/216. Typically 20 to 35 iterations were required for a shooting method to find a solution. Solutions of the two-dimensional flow field converged quickly due to the relatively coarse (20x20) grid. Typically, solutions were reached in less than 6 iterations.

CHAPTER 3

RESULTS AND ANALYSES

One-Dimensional Simulation

A comparison is made of the results predicted by the one-dimensional simulation model to experimental results. The comparison is made to validate the selection of expressions and algorithms. Generally, the model predicts reasonably well.

Experimental data comes from work by Coban, performed at the University of Utah, where a 3-meter spray-column of .61 meter diameter was constructed for operating a three-phase spray-column DCHX. The system dispersed n-pentane through a flow of hot water at pressures ranging to approximately three atmospheres. Flow rates and initial temperatures were controlled to provide a range of combinations. Results and discussions of this experimental work was published (Coban, 1986). The principle observations made in the experiments were that of the entry and exit temperatures of the two working fluids. It is against selected cases from this work that the output of this modelling is compared.

Figures 9 through 11 show comparisons to three experimental cases studied by Coban conducted on a 3.0 m (column length) by .305 m (column radius)

DCHX which dispersed n-Pentane into water with initial droplet radii of 2.0 mm. The comparisons cover a variety of flow rates, pressures, and starting temperatures. The results of the simulations are depicted by the solid line profiles. The inlet and outlet temperatures are known from the experiments and are shown as symbols.

Figure 9 shows a close correlation with experimental data is achieved. The known conditions of the water are the inlet and outlet temperatures of 85°C and 71°C and for the n-pentane, 41°C and 71°C, respectively. The simulation of a system operating under the same conditions results in entry and exit temperatures identical to the experimentally observed results.

Figure 10 shows a comparison to observed results to the spray-column operating at a narrower inlet temperature differential. It is seen here that the simulation diverges somewhat at the exit temperatures. The n-pentane did not rise to the same observed value. The simulation prediction was that there was incomplete vaporization while experimental observations show completion of vaporization. Though the divergence appears significant, it is noted that vaporization in the simulation was nearing completion, at which time the temperature would rise as rapidly as it did in Figure 9. The comparison in Figure 10 suggests some refinement could be made to the transfer coefficient expressions.

Figure 11 shows a comparison where the inlet temperature differential increases to a spread of 85°C to 31°C. The results of this simulation show a reasonable correlation to the known data, indicating that the heat transfer algorithm is fairly well constructed.

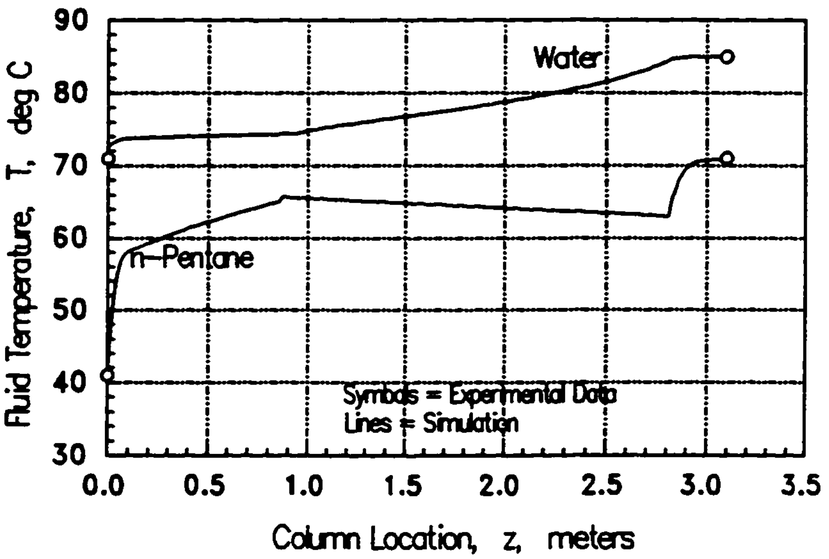

Figure 9 Comparison of Simulation Results to Experimental Data. (Pressure = 260 kPa, Water Rate = 1.94 kg/s, n-Pent Rate = .21 kg/s.) (Data from Coban, 1986).

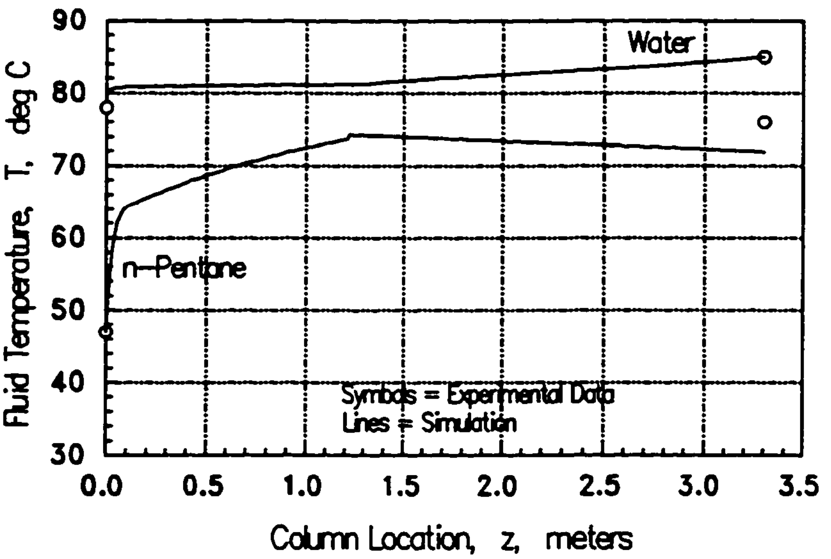

Figure 10 Comparison of Simulation Results to Experimental Data. (Press = 329 kPa, Water Rate = 2.11 kg/s, n-Pent Rate = .12 kg/s.) (Data from Coban, 1986).

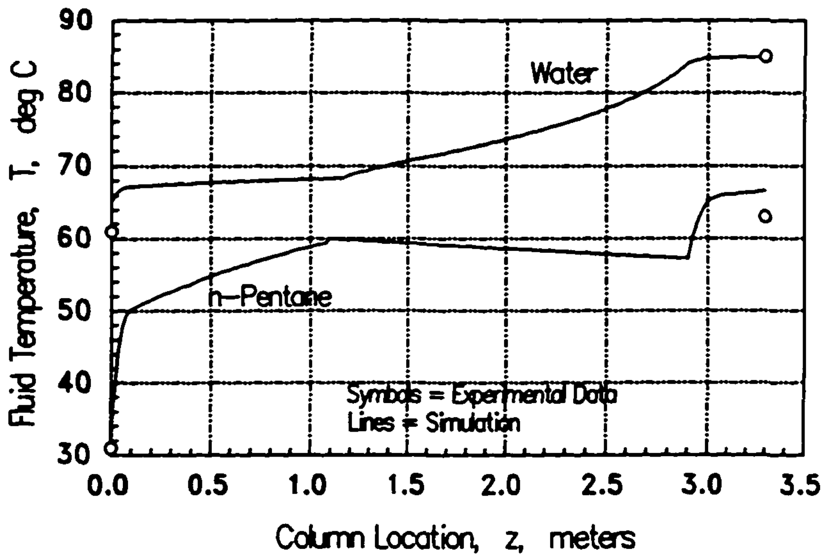

Figure 11 Comparison of Simulation Results to Experimental Data. (Press = 225 kPa, Water Rate = 1.52 kg/s, n-Pent Rate = .25 kg/s.) (Data from Coban, 1986.)

Individual droplet/bubble rise regimes are discernable in the profiles. The expected rapid temperature rise occurring in the first few centimeters of column corresponds to the high circulation condition present at droplet injection and initial rise. Following that, the attenuated rise occurring in the first meter corresponds to the stable liquid rise regime. This continues until the dispersed fluid reaches its saturation temperature, at which time phase change occurs. It is seen that during phase change, there is a decrease in dispersed fluid temperature. This is the result of the reduction in hydrostatic pressure seen by the droplet/bubbles as they rise through the column. The reduction in pressure lowers the saturation temperature. The vaporization process hastens as the process nears completion of the phase change. This is attributed to increased size and circulation effects of the vapor cap on the bubble. Upon completion of vaporization, the bubble rises rapidly and circulation once again decays.

In each of the three figures, the continuous fluid temperature is more or less unremarkable since water has a higher specific heat and remains at a constant state. Generally, it is of less interest, though no less important, than the working dispersed fluid.

Details of the dispersed droplet/bubble characteristics such as droplet radius, quality and holdup can be of interest but are generally difficult to obtain in experimental work. Simulation offers some predictability. Figures 11 to 17 depict a variety of key parameters for the three-phase baseline situation of Figure 9.

Figure 12 depicts the values of the heat transfer coefficient for the droplet/bubble rising through the column. The transfer coefficient profile confirms the temperature profile. The circulation influence which decays rapidly shows up at the outset, following droplet injection. This is followed by a lower stabilized value for the stable rise regime until the onset of vaporization. The weighting of the Tochitani expression and the Fan and Tsuchiya expression during phase change is seen in the saddle-shaped portion of the profile. The Tochitani expression is quite high at the onset of phase change and decreases. The Fan and Tsuchiya prediction begins to pick up when droplet/bubble quality reaches approximately 0.5 and then increases as the vapor cap grows until vaporization is complete. This elevated value then undergoes a final decay as the bubble rises to its exit.

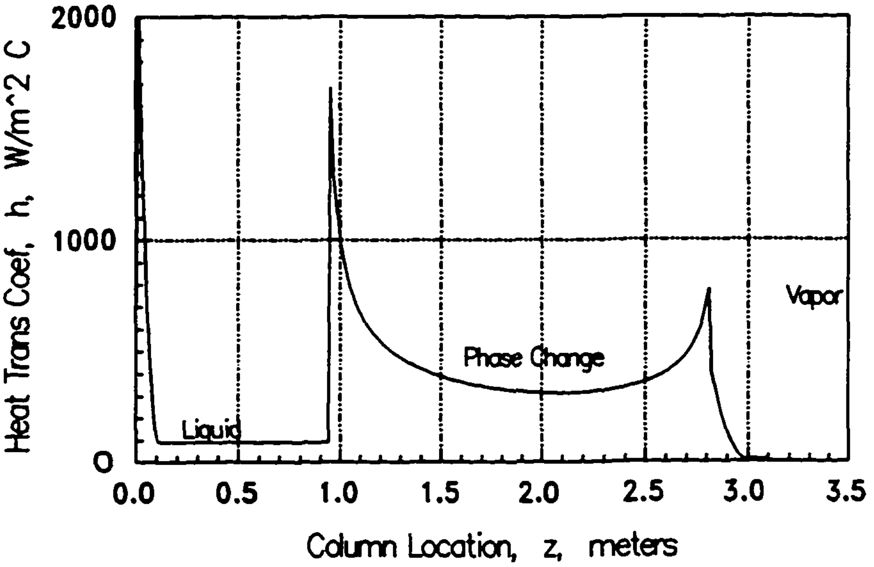

Figure 12 Droplet/Bubble Heat Transfer Coefficient Profile for Simulation Shown in Figure 8.

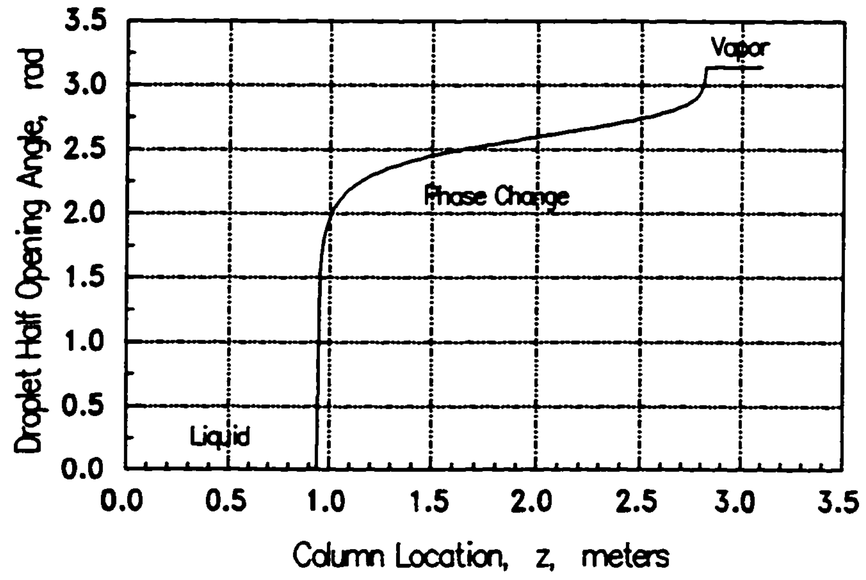

Figure 13 Droplet/Bubble Half Opening Angle Profile for Simulation Shown in Figure 8.

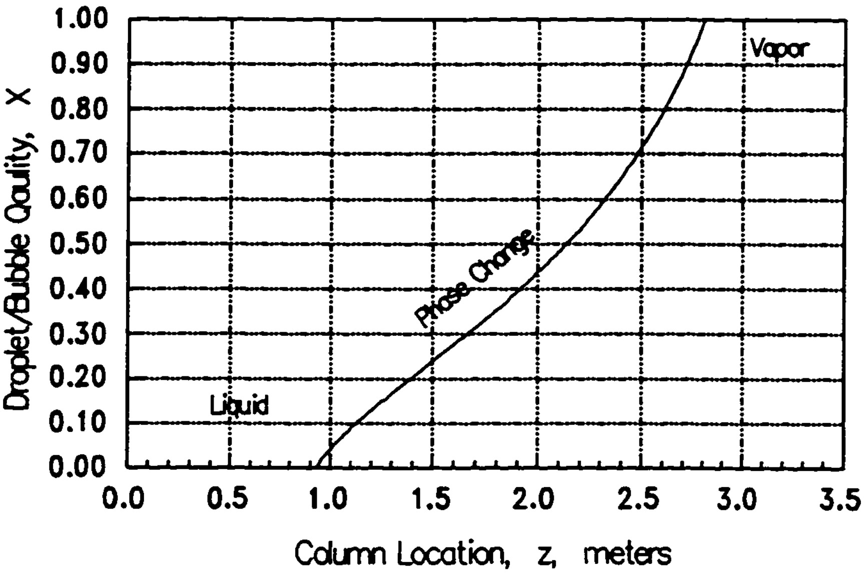

Figure 14 Droplet/Bubble Quality Profile for Simulation Shown in Figure 8.

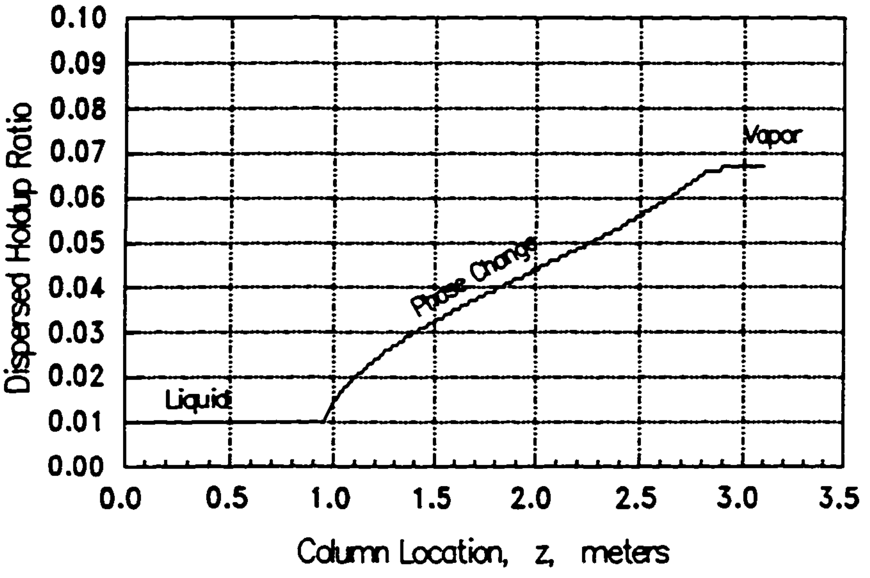

Figure 15 Dispersed Fluid Holdup Profile for Simulation Shown in Figure 8.

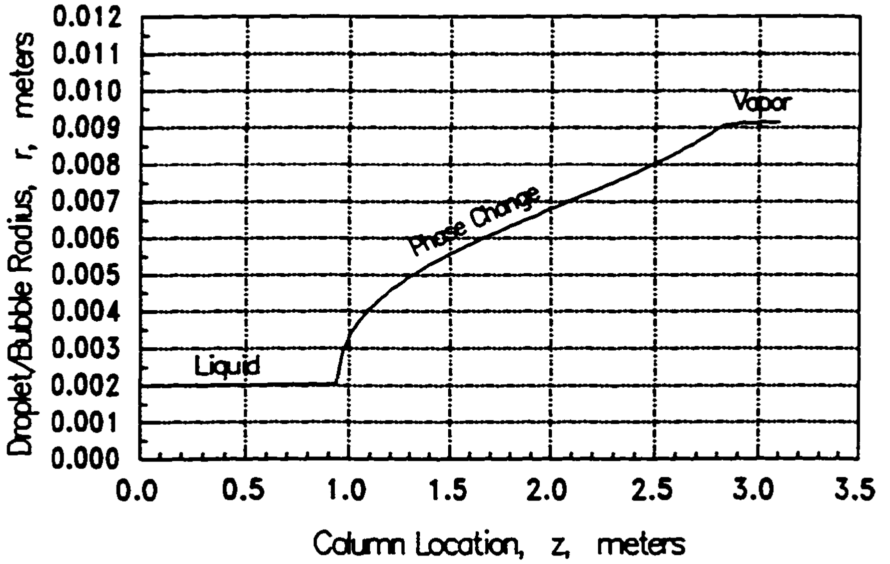

Figure 16 Droplet/Bubble Radius Profile for Simulation Shown in Figure 8.

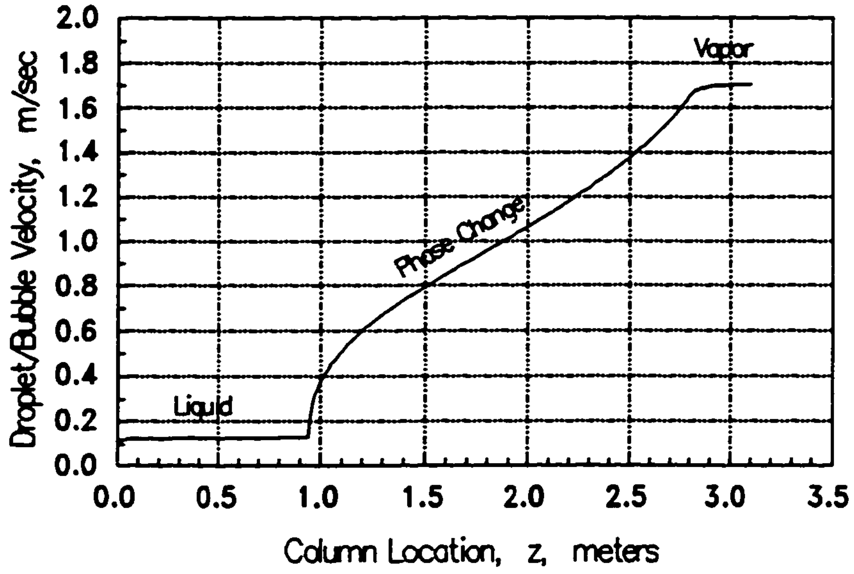

Figure 17 Droplet/Bubble Velocity Profile for Simulation Shown in Figure 8.

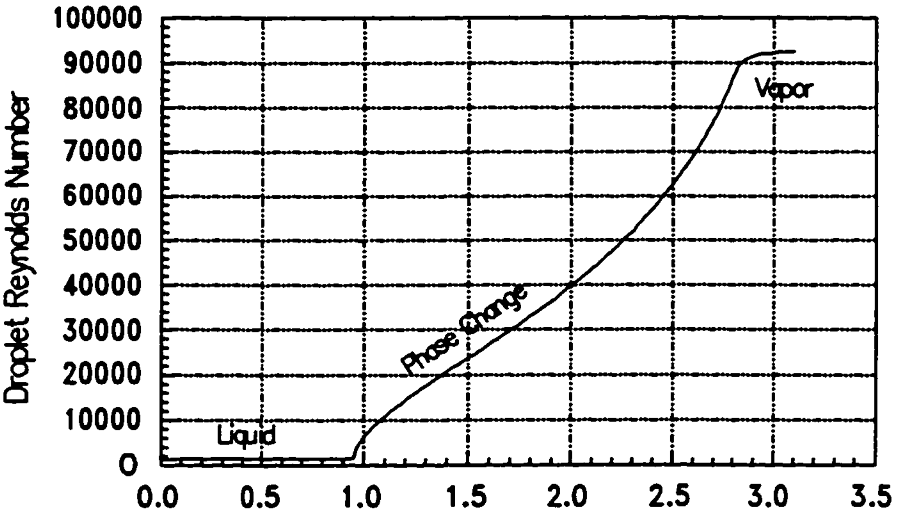

Figure 18 Droplet/Bubble Reynolds Number Profile for Simulation Shown in Figure 8.

Figures 13 and 14 show the droplet/bubble half opening angle and dispersed fluid quality. The profile clearly shows the progression from liquid to vapor through the phase change.

Figure 15 shows the holdup. As expected, this holdup is only a small fraction during the liquid rise. Upon vaporization, the void fraction increases dramatically. The importance of the holdup is, of course, its relationship to column flooding.

Figure 16 indicates the growth in the droplet/bubble radius. A comparison with Figures 13 and 14 reaffirms the progression through phase change from liquid to vapor, indicating a growth from 2 mm radius specified at injection to slightly more that 9 mm at completion of phase change.

Figures 17 and 18 depict droplet/bubble velocity and Reynolds number respectively. Like the previously depicted parameters, the increase in velocity at approximately the 1.0 meters location signals the onset of phase change with completion at 2.6 meters. Of note is the order of magnitude difference between the rise of the most of the parameters and that of the Reynolds number. This difference is attributed to the fact that a one-order increase in the velocity and droplet/bubble radius combines to a two-order of magnitudes increase in the Reynolds number.

The preceding profiles illustrate the potential for using simulation modeling for studying parameter behavior and trends. While it is acknowledged that experimental results are less challengeable, acquisition remains difficult and for some parameters are still unachievable. Simulation allows for exploration of the effects by and on key parameters.

Parametric Sensitivities

The simulation model, like its experimental counterpart, functions on the input of eight key parameters. They are two fluid flow rates, two fluid inlet temperatures, dispersed fluid initial droplet/bubble radius, and three device parameters - length, radius and operating pressure. One could include the fluid types as an input since fluid selection inherently creates a variety of thermodynamic and thermophysical property combinations, bringing the list of key parameters to ten. Including fluid type creates considerably more variation since each fluid has a unique density-viscosity-conductivity-specific heat combination. In this work, the list is limited to the first eight parameters stated above.

Initial Droplet Radius

Control of the dispersed fluid droplet size at injection has a marked effect on DCHX performance. Figure 19 shows the effect of varying the initial droplet radius on the duty of a water/n-pentane system. Primarily, the influence is caused by the reduction in overall surface area for a given fluid flow rate. This is seen in a simple analysis of the surface area and volume formulae for a sphere. Surface area of an individual droplet increases proportionally with R^2, whereas droplet volume increases with R^3. A doubling of the radius results in four times as much surface area on a droplet (which might first appear to be good for heat transfer) but results in one eighth the number of total droplets containing eight times as much of the fluid. The net result is a reduction in duty on the order of one half.

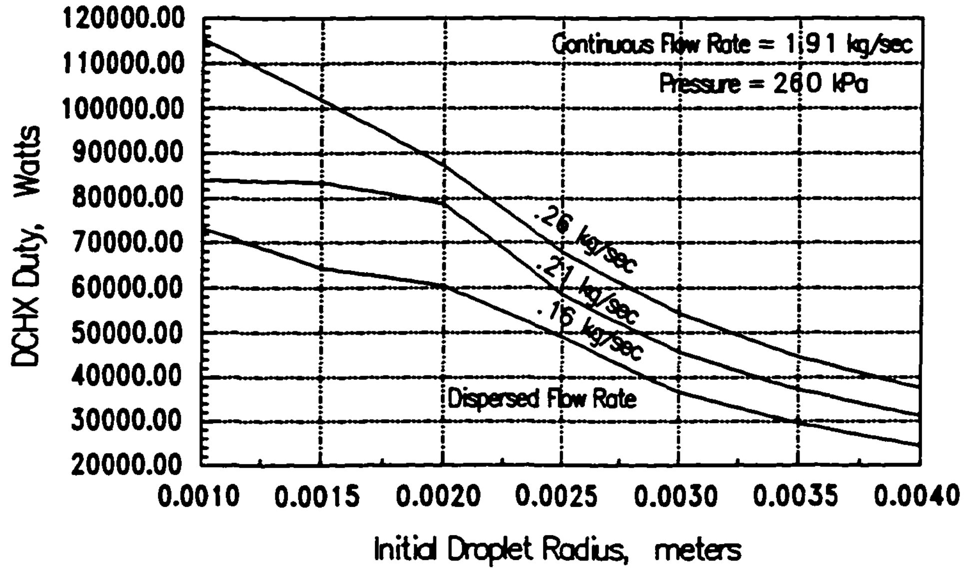

Figure 19 Sensitivity of DCHX Duty to Initial Droplet Radius.

Hydrodynamic factors associated with increasing or decreasing droplet size affect the duty as well. Larger droplets react to the viscosity of the counterflowing continuous fluid, causing a reduction in rise velocity. While this increases residence time in the DCHX, the more dominant influence is that convection processes are reduced, causing a loss of duty as droplet size increases.

A further compounding effect is that in a system where phase change barely finishes such as the examples shown in Figures 9 through 11, a reduction in heat transfer early on in dispersed fluid rise can result in incomplete vaporization before the fluid exits. This causes an even larger loss of overall duty since a significant portion of the heat transfer occurs during the phase change.

Generally, it is desirable to have as small a droplet size as is practical at injection to generate the highest duty. The limitation to this is that small size droplets may be difficult to achieve if one desires a large dispersed flow rate. Nozzle construction is a key issue.

Column Length

The effects of column length on DCHX duty are less pronounced but no less interesting. Generally, longer columns result in higher duty. The principle reason is the increased residence time during which the fluids are in longer contact. A secondary factor is the change in hydrostatic pressure exerted on the fluid in the lower portion of the column. It is commonly known that system duty increases with increased operating pressure because of *better* fluid properties. Increased column length increases the pressure at the bottom of a column, where the dispersed fluid enters.

The trend in duty versus column length for a water/n-pentane two phase system is shown in Figure 20. Note that the increased duty is not linear but tapers off as column length passes a certain length. This is due to the dispersed droplets/bubbles completing vaporization at some length less than the total length of the column. In these cases, the fluid merely completes its travel to the top of the column exchanging little or no heat once it has vaporized and reached the continuous fluid's temperature. Figure 20 suggests that a given system has an optimum length beyond which there is a diminishing return and which would require a cost-duty tradeoff analysis.

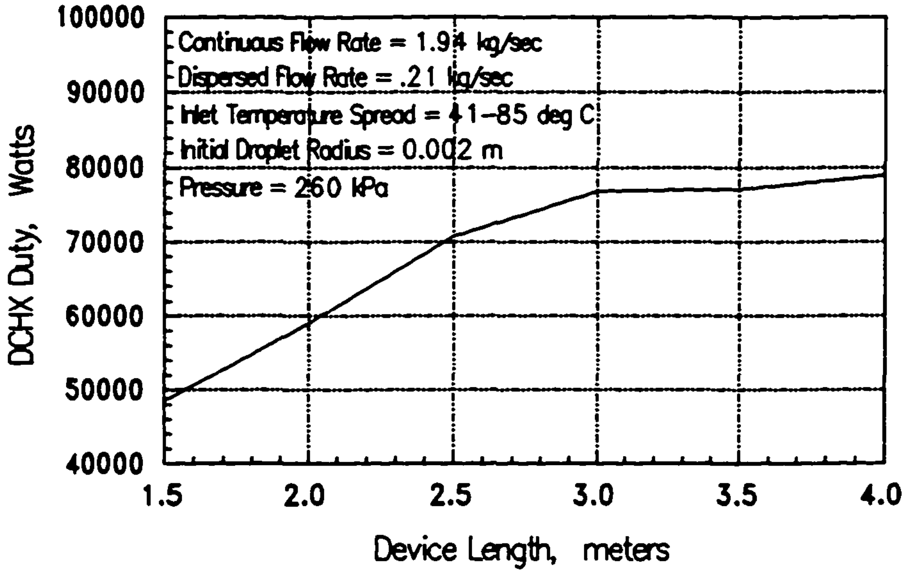

Figure 20 Sensitivity of DCHX Duty to Column Length.

Fluid Flow Rates

The most complex factors affecting DCHX performance are the individual fluid flow rates and their inlet temperatures (discussed in the next section). As mentioned in Chapter 1, control of fluid flow rates can be used to achieve one of several desired outcomes - temperature control of one fluid or overall system duty. A discussion of temperature control and optimized flow rates follows in later sections. Here a discussion is made about the effect of flow rates on overall duty. Figure 21 depicts the results of simulations over a range of dispersed and continuous fluid flow rates in a water/n-pentane system. In general, the trend follows that for a given flow rate of continuous fluid, the higher the

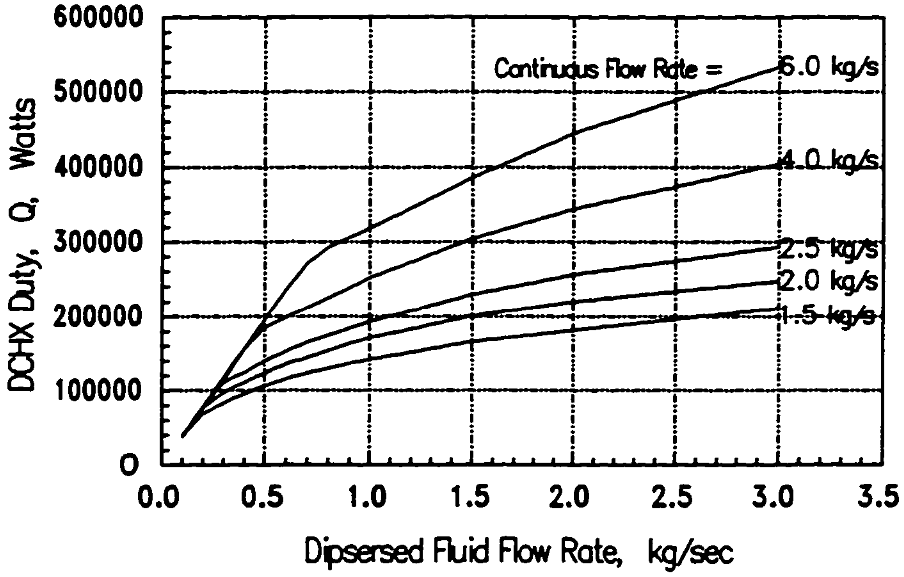

Figure 21 Sensitivity of DCHX Duty to Fluid Flow Rates.

dispersed fluid (n-pentane) flow rate is, the higher the duty is. However, it is seen in the figure that the increase is not linear and duty tends to level off as the dispersed flow increases. The reason for this is linked to the completion (or lack of completion) of the phase change. Note that for flows ranging between approximately 0.1 and 0.7 kg/sec, the rate of increase is nearly identical over the range of reference continuous flow rates. The rise follows a particular gradient until the duty breaks off of this gradient, fanning out to an asymptotic leveling. The sharp coincidental rise at the lower dispersed flow rates corresponds with the cases where phase change occurs and is completed. The point at which each profile *breaks* off from this common gradient is the point at which phase change fails to complete. Since a large amount of the heat transfer occurs during the phase change, an incomplete change shows up quickly in a curbing of duty. The leveling off of duty is evident in each of the five referenced continuous fluid flow rates.

Dispersed fluid flow rates greater than the rate at which duty levels off have the effect of making the three-phase system appear more and more like one with only two phases. The higher flow rates require a higher extraction of heat during the preheating section, which elevates the location at which phase change begins. Given the relatively inflexible amount of time in which phase change occurs (Figures 9, 10, and 11), a delay in the onset of phase change directly results in the loss of some portion of the change. The higher the dispersed flow rate, the more of the change is lost. The system still experiences an increase of duty with increased dispersed flow rate but at a significantly reduced gradient. This suggests that any three phase system should be operated at or near the point at which the phase change process can adequately be completed. It should be noted that in the this an other analyses, flooding conditions (one fluid hydrodynamicially overpowering the other) were not addressed. While the coded model accounts for the condition when it occurs, tested ranges of parameters were all within *safe* operating conditions.

Inlet Temperature

An interesting aspect influencing DCHX performance is the fluids' inlet temperatures and inlet temperature differential. It is noted, though, that this is an aspect over which one often has no control. In geothermal applications, for example, a system is restricted to using heated underground water which will be fixed at some temperature determined by natural forces. If control is feasible, the dispersed fluid should be introduced as close as possible to its vaporization temperature. Figure 22 depicts the duty of the three phase water/n-pentane system. Similar to Figure 20, a range of dispersed fluid flow rates is examined against a reference set of temperatures. An overall inlet spread of 50°C is maintained but distributed over a dispersed fluid inlet temperatures range between 20° to 60°C and corresponding continuous fluid inlet temperatures of 70° to 110°C. As in the flow rate effects discussed in the previous section, here

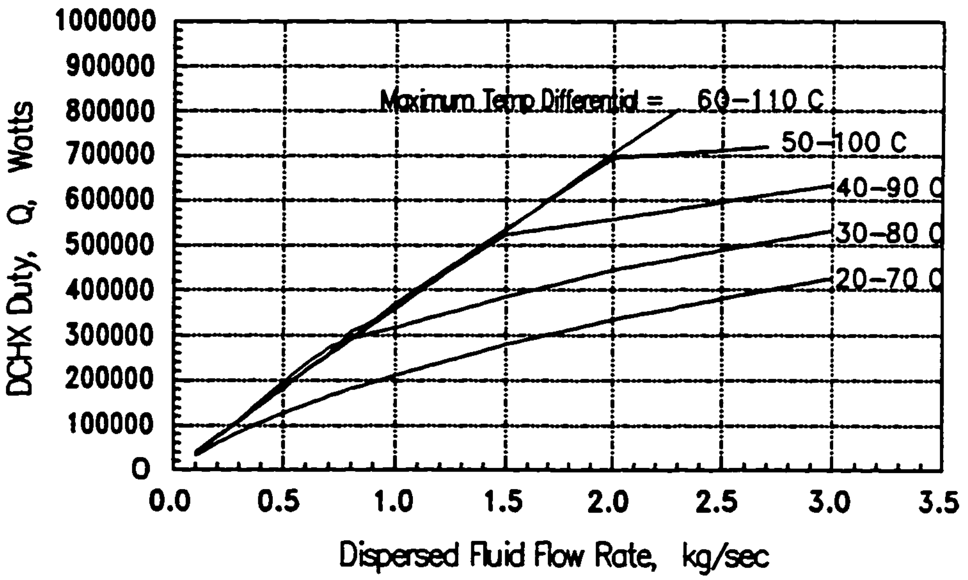

Figure 22 Sensitivity of DCHX Duty to Fluid Inlet Temperatures and Spread.

again there is a breaking off and fanning out of the duty curve. The trend is again attributed to the phase change portion. In the lower temperature spreads, the n-pentane spends an inordinate amount of time preheating before reaching vaporization. Vaporization temperature occurs at approximately 62°C for the operating pressure of 260 kPa used for this comparison. With a water inlet temperature of only 70°C, the dispersed fluid spends little time in the phase change portion. As the spread moves up to the last tested range of 60°-110°C condition, less and less time is spent preheating, resulting in more time spent in phase change. The result of this is the notable increase in duty shown in Figure 22.

Reversing Fluid Roles

A final aspect affecting DCHX performance is the selection of the roles the fluids take on in a system. Thus far, discussions have focused on n-pentane/water. Runs involving water and commercial heat transfer fluids or oils (examples of two-phase applications) show that duty can increase or decrease depending on whether the transfer fluids or oils are used as the dispersed or continuous fluid.

Table 2 lists the results of six cases where fluid combinations were tried and then their roles were reversed for comparison. It is seen that there is a mix of results ranging from as an increas of 37% to a loss of 47% in duty when the transfer fluids or oils were switched to the role of the continuous fluid. The apparent factor responsible is the fluids' viscosity ratio which affects droplet's velocity while in the system. Table 3 lists pertinent properties of the six transfer fluids used in the test cases. All were similar except for viscosity and subsequently the viscosity ratio with water. In the three cases where duty increased with switching the transfer fluid to the continuous role, the fluid's viscosities were of

TABLE 1 Device Output as a Result of Reversing Roles of Fluids

Dispersed Fluid	Continuous Fluid	Total Device Output- W	
Syltherm® 800	H_2O	171,620	
H_2O	Syltherm® 800	235,420	37% Inc
Therminol® 44	H_2O	230,381	
H_2O	Therminol® 44	270,738	17% Inc
Transformer Oil	H_2O	223,526	
H_2O	Transformer Oil	228,428	2% Inc
TM-1 Oil	H_2O	202,753	
H_2O	TM-1 Oil	201,358	1% Dec
Therminol® 55	H_2O	246,286	
H_2O	Therminol® 55	219,785	11% Dec
MS-20 Oil	H_2O	260,889	
H_2O	MS-20 Oil	138,686	47% Dec

Notes

1. Flow rates are at optimum for fluid combination with H_2O inlet temperature at 22°C and transfer fluid/oil inlet temperature at 100°C, column length of 3.0 m, column radius of .5642 m, initial droplet radius of 2 mm, and operating pressure of 101.3 kPa.

TABLE 2 Approximate Fluid Properties in DCHX Application.

	Syltherm® 800	Therminol® 44	Transformer Oil
Density kg/m^3	892	885	847
Viscosity Pa-s	.0008	.0018	.0060
Specific Heat J/kg-K	1701	2074	1939
Conductivity W/m-K	.1031	.1357	.1051
Fluid-H_2O Viscosity Ratio	1.18	2.83	9.23

	TM-1 Oil	Therminol® 55	MS-20 Oil
Density kg/m^3	857	848	864
Viscosity Pa-s	.0048	.0123	.1120
Specific Heat J/kg-K	.1702	2100	2196
Conductivity W/m-K	.1139	.1234	.1305
Fluid-Water Viscosity Ratio	7.35	18.92	172.1

approximately the same magnitude as that of the water, having ratios ranging from 1.18 to 9.23. In the three cases where duty decreased the fluids were significantly more viscous than water having a fluid-to-water visocity ratio ranging from 7.35 to 172.1. The conclusion drawn from this is that the less viscous fluid should be brought in as the continuous fluid. The reason is that the less viscous the continuous fluid, the higher the droplet rise velocity which results in higher convection. Some overlap of the trend is seen in the two marginally close cases of transformer oil/water and TM-1 oil/water, suggesting that the pivoting value of viscosity ratio lies approximately in the 7.0 to 10.0 range. Fluid combinations falling in this range would have to be examined individually to determine the best selection of roles.

Dual Utility Optimization

The discussion in Chapter 1 on DCHX optimization highlighted that various forms of optimization exist for DCHX. A system may be optimized for maximum duty, target temperature, or dual utility of fluids. To maximize on the first two, one need only supply an ample amount of flow of the non-critical fluid (the fluid of non-limited supply) for maximum duty or iterate on the flow rate until a specific target temperature in the fluid temperature profile is achieved. The third method requires the use of the Dual Utility Optimization characteristic presented in Chapter 2.

As highlighted in Chapter 1, classical measures such as the LMTD and effectiveness can fail to identify when optimized flow rates are achieved. Figures

23, 24, and 25 demonstrate this failure and show that the use of the DUO characteristic achieves the desired outcome. Figures 23 and 24 depict the behavior of the DUO characteristic when device length and dispersed droplet radius are varied over a range of values in a two-phase liquid-liquid DCHX using water and Therminol-44® - a commercially produced heat transfer fluid. In these two particular cases, the DUO correlates closely with the traditional definition of effectiveness. Reliance on classical effectiveness would be sufficient if length and droplet radius were the only unknown parameters.

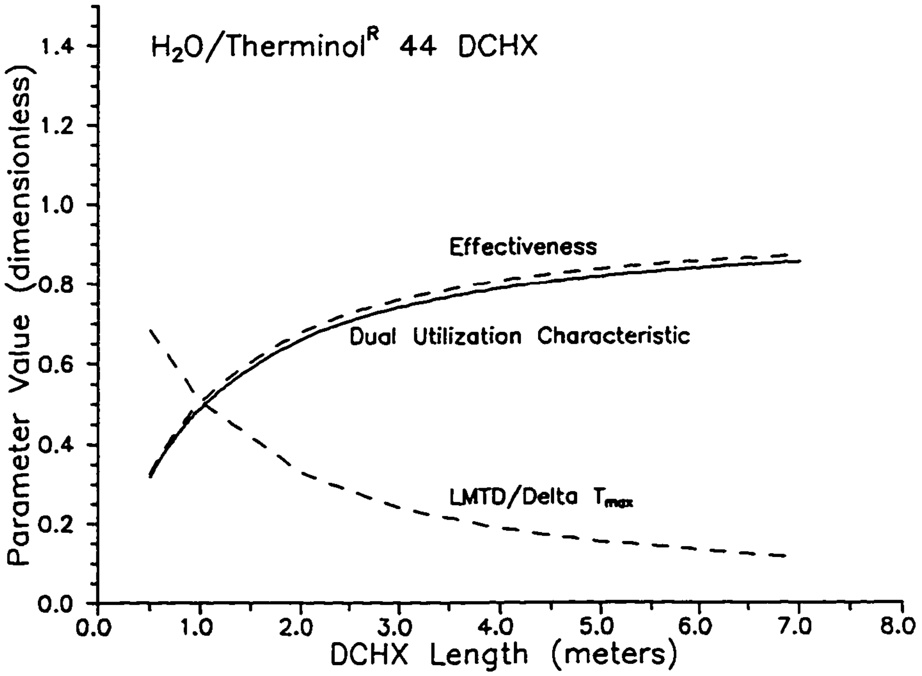

Figure 23 Comparison of Dual Utility Indicator to Effectiveness and LMTD as a Sensitivity Indicator for Column Length.

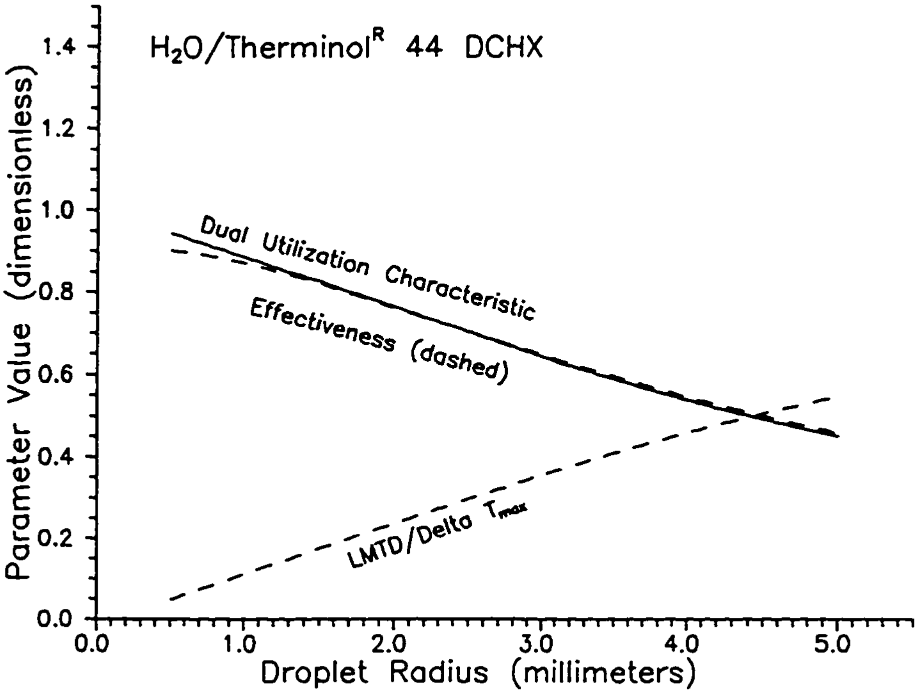

Figure 24 Comparison of Dual Utility Indicator to Effectiveness and LMTD as a Sensitivity Indicator for Initial Droplet Radius.

A more complex situation develops when the parameter in question is fluid flow rate as Figure 25 illustrates. It is noted that variations in device length and droplet radius are not linked to the dual utility problem per se but rather to the maximum duty output. For dual utility, properly balanced flow rates are required. Figure 25 shows differing trends in the profiles for the classical LMTD and effectiveness measures and based on effectiveness alone, one would conclude that an unoptimized condition occurred at a continuous fluid (water) flow rate of approximately 1.0 kg/sec in the evaluated system.

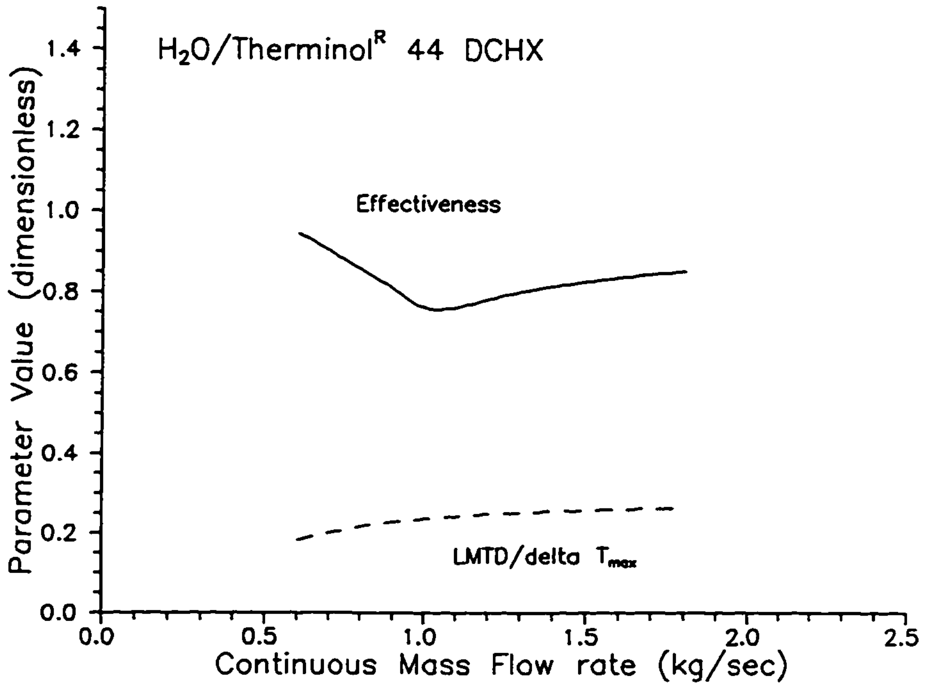

Figure 25 Effectiveness and LMTD Sensitivity to Continuous Fluid Flow Rate. Dispersed Fluid Flow Rate = 2.0 Kg/Sec.

In Figure 26, a plot of the DUO characteristic is added. The result reveals that an optimized condition is reached at the 1.0 kg/sec flow rate. In reaching a peak, the measure indicates that for the referenced flow rate of 2.0 kg/sec of heat transfer fluid, a 1.0 kg/sec flow rate of water will yield temperature profiles that are both parallel and at as close of an approach as is possible for the given inlet temperatures, droplet radii, length, etc., specified for this system. This is confirmed by examining the resulting fluid temperature profiles shown in Figure 27. By using the developed DUO characteristic indicator, the dual utility condition can be found for two-phase systems.

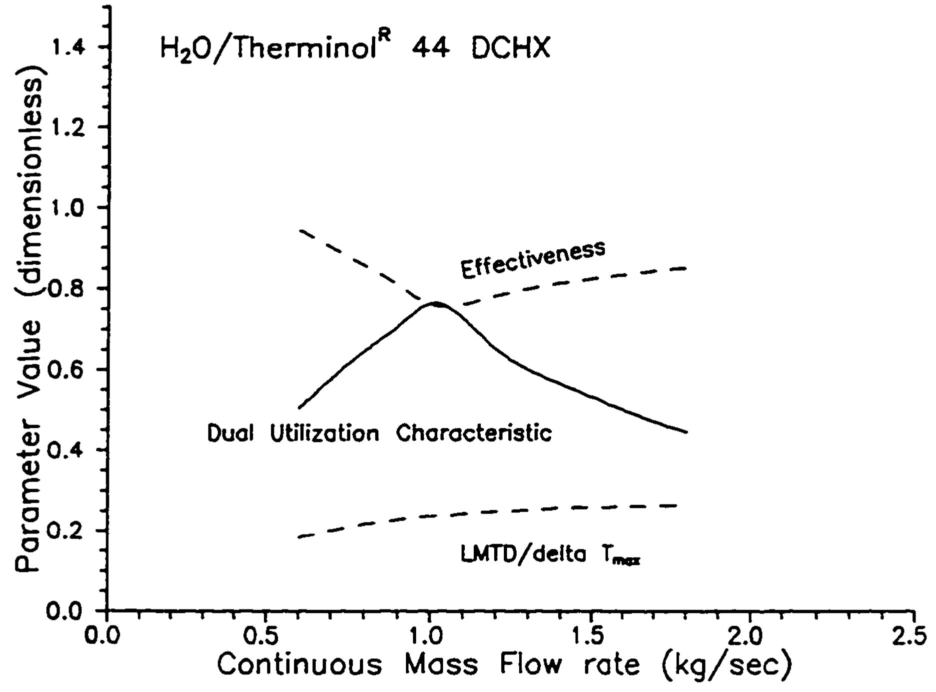

Figure 26 Comparison of Dual Utility Indicator to Effectiveness and LMTD as a Sensitivity Indicator for Continuous Fluid Flow Rate.

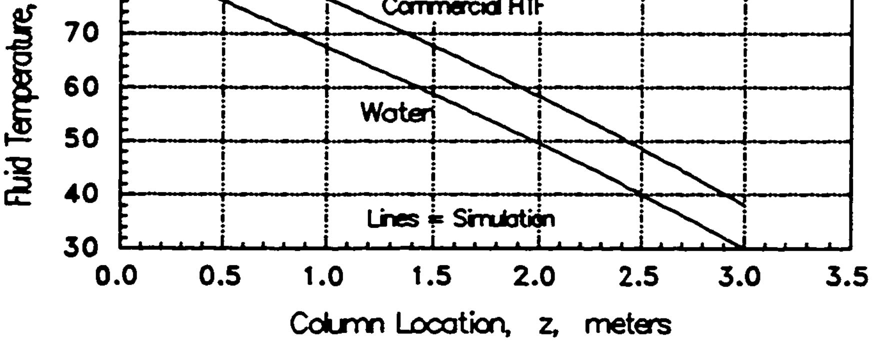

Figure 27 Temperature Profile for Liquid-Liquid DCHX Optimized for Dual Utility.

Referencing the correlation of Figure 23 and the stated observation that device length is generally unrelated to the dual utility condition, Figure 28 is presented, confirming that the optimized dual utility flow rates are unaffected by device length. The peak in the DUO indicator remains fixed at the 1.0 kg/sec rate. Device length has the overall effect of improving on the situation - bringing the profiles even closer - once the proper balance is found. In effect, the DUO exclusively identifies the flow rate balance and then can be used to fine tune other parameters.

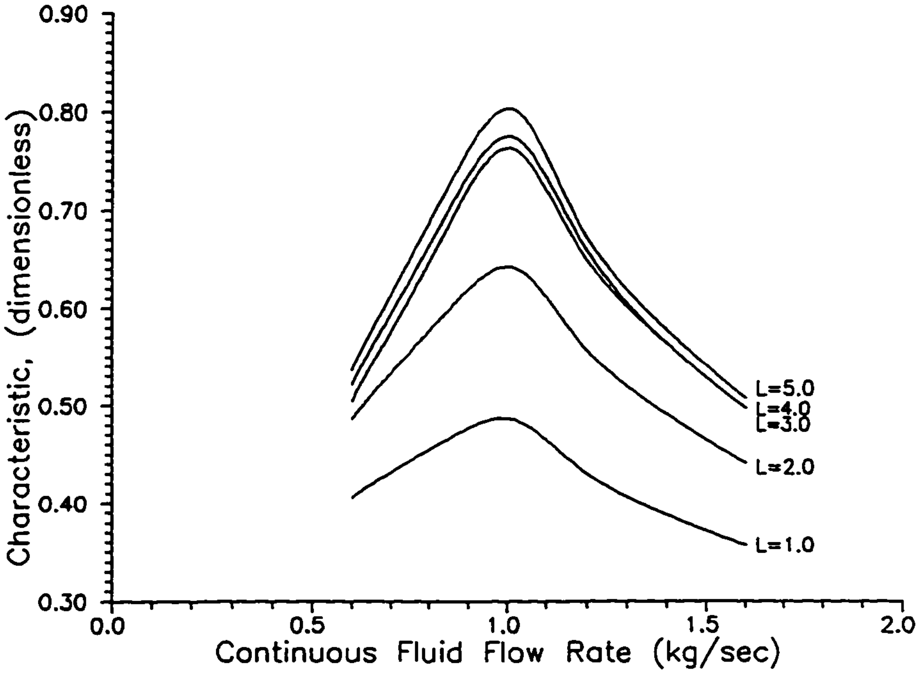

Figure 28 Combined Effects of Balanced Flow Rate and DCHX Length on Dual Utility Optimization. (1.0 m ≤ DCHX Length ≤ 5.0 m)

Simulation Based Designs for a Gas-Fired, Direct-Contact Water Heater

The practicality of the one-dimensional simulation was tested by employing the model to design a DCHX for industrial water heating. The designs focused on dispersing exhaust gases from the combustion of natural gas in air through a flow of water at tap temperature. Direct contact between the exhaust gases and water would cause the water to heat up according the flows being supplied. Using the model to test variations on two types of candidate systems resulted in designs which were presented at an international conference. Synopses of the designs are given in Appendix III which condenses formally published work (Brickman, Sethi and Boehm, 1994).

A secondary purpose of employing the model for the designs was to test its useability by a person not as familiar with direct-contact processes. Encoding the input file proved to be a reasonably simple task as did interpreting ensuing results. The model was used in an interactive fashion, with the *non-expert* subject (second author of the cited paper) testing and improving *best guess* parameters. The final designs were arrived upon following a thorough exploration of parameter variations. It was generally felt that the program was built at the right level of knowledge that would be expected of field engineers.

Two-Dimensional Sieve Tray Analysis

Two-dimensional modelling is useful in analyzing one particular form of DCHX enhancement, namely the addition of sieve trays in the preheating section. While fully two-dimensional modelling of an entire DCHX configured with half a dozen or more trays would be formidable, the modelling of a single module (the space between two consecutive trays) is more readily accomplished with results offering insight to the sieve tray enhancement.

The primary mechanism at work in sieve tray enhancement is the increased convection due to increased relative flow of continuous fluid around the droplets, resulting in a higher heat transfer for a given length of the DCHX. Increased relative flow is achieved by increasing the nominal continuous fluid input flow and forcing it through alternating by-pass gaps on the sides of a tray. This creates a cross flow through which the droplet must rise. The cross flow adds a horizontal velocity component, resulting in a higher relative flow around the droplets. It should be noted that forcing the flow to *switchback* through an array of properly sized and spaced trays is crucial to the enhancement as increasing the continuous flow rate has negligible effect by itself in an unmodified column.

Convection around an individual droplet is calculated from the stable rise expressions (characteristic of the preheating section) given in Chapter 2 since sieve trays are intended for use in preheating. If consecutive sieve trays are placed in the flow with alternating by-pass gaps, a cross flow is created in the continuous fluid. Dispersed fluid droplets enter this module through perforations in the sieve tray and rise up through the cross flow, are swept in the direction of the cross flow and caught under the next higher tray. The degree to which droplets are swept sideways depends primarily on the continuous fluid flow rate and to a lesser degree on the size of the by-pass gap and vertical spacing of the trays. At typical continuous fluid (water) flow rates, the hydrodynamic flow field can be approximately considered to be a potential flow field with minimal influence by the boundary influence. This assumption is defended by the fact that the bulk of interest lies in the heat transfer aspects of droplets in the field and not the micro analysis of an individual droplet's path in or out of a boundary layer.

The two-dimensional flow field is found by numerical application of the expressions described in Chapter 2. The result is an output grid of horizontal and vertical velocities. For each droplet starting position, a path is traced by the application of simple law of motion. With the paths known, a final iteration is made to determine droplet heat transfer referencing the changing velocity pattern of the two-dimensional space. From this, droplet efficiencies are calculated. It is found that a wide range of droplet patterns result when the continuous fluid flow rates are varied in a tray module.

Figures 29 through 35 show droplet traces through a sieve tray module with an aspect ratio of 0.5 and a by-pass gap of 20% of the column's cross sectional opening of 0.61 meters (column diameter). As stated in Chapter 2, grid spacing for the field was 20 x 20 and for the flow rates tested, boundary layer effects were did not appear. The reference continuous flow rate was 1.91 kg/sec. Figure 29 shows the result of the flow passing through one set of sieve trays. It is seen that there is little effect on a droplet's horizontal path due to the continuous flow being constricted through the by-pass gap. Generally, the droplet's vertical velocity remains constant except near the by-pass gap, where there is a slowing of the droplet. Note that Figure 29 shows a droplet being released in the lower by-pass gap. This trace is included to illustrate that a droplet reaching that location from the module just below would continue on up through the restriction into the module. This is an approximate continuation of the first trace on the left.

The effects of the cross flow become more apparent when the flow rate is increased in multiples. Figures 30 through 35 show droplet traces when the continuous flow rate is doubled, tripled, etc. Of interest are the runs which show traces for flow rates multiplied by 2, 3, 4, 5, and 6, respectively. In each succeeding run, more and more side sweep appears. In the latter runs, there is a significant amount of *capturing* of the droplets released near the exiting by-pass gap. The run shown in Figure 35 (10x reference rate) is presented to illustrate an extreme case where there is an inordinately high amount of capture, a condition that would be undesirable.

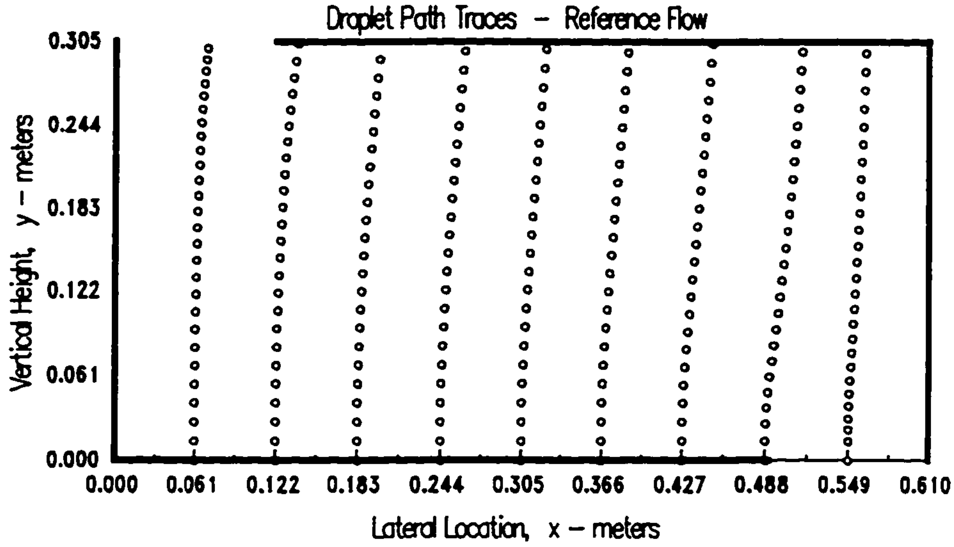

Figure 29 Droplet Path Traces - n-Pentane/H_2O. Reference Continuous Flow Rate = 1.91 kg/sec, Droplet Radius = 2 mm.

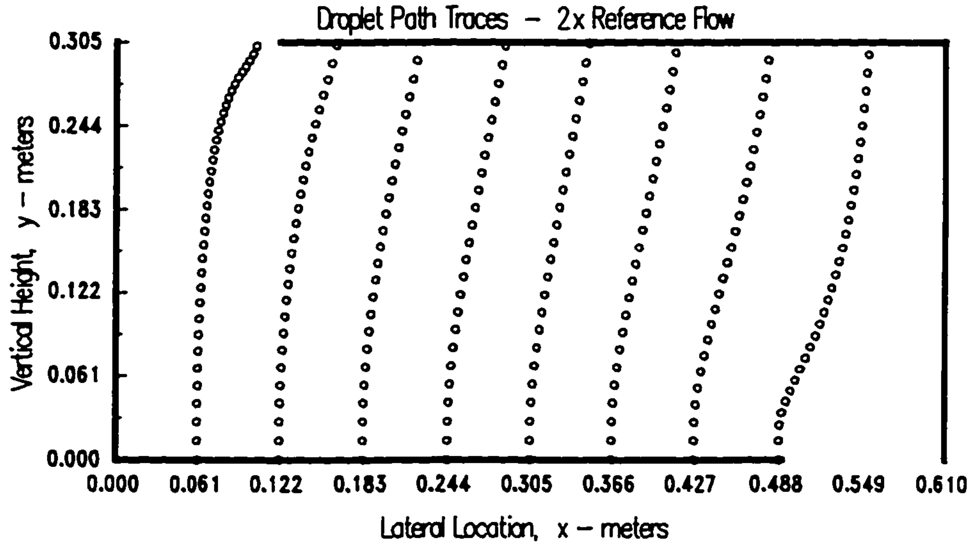

Figure 30 Droplet Path Traces - n-Pentane/H_2O. Continuous Flow Field Velocities = 2 x Reference, Droplet Radius = 2 mm.

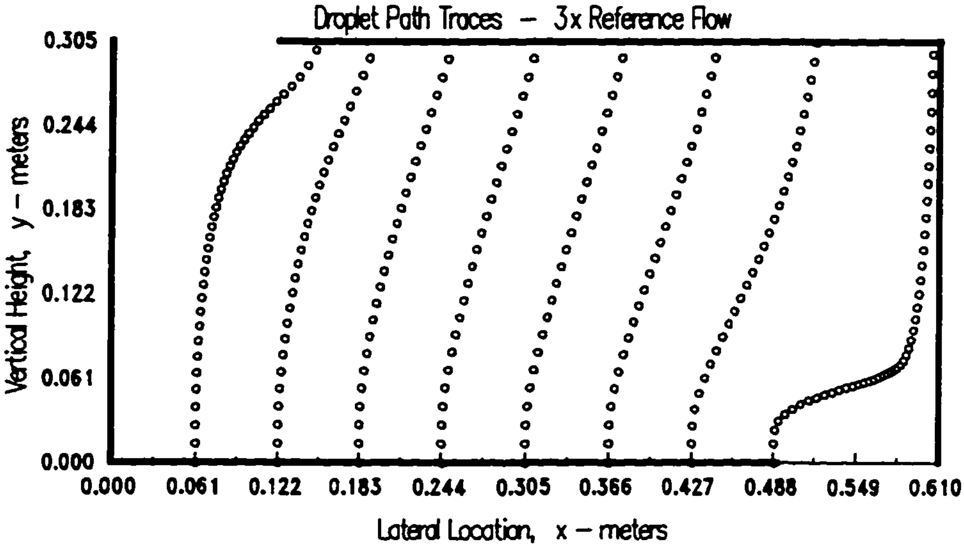

Figure 31 Droplet Path Traces - n-Pentane/H_2O. Continuous Flow Field Velocities = 3 x Reference, Droplet Radius = 2 mm.

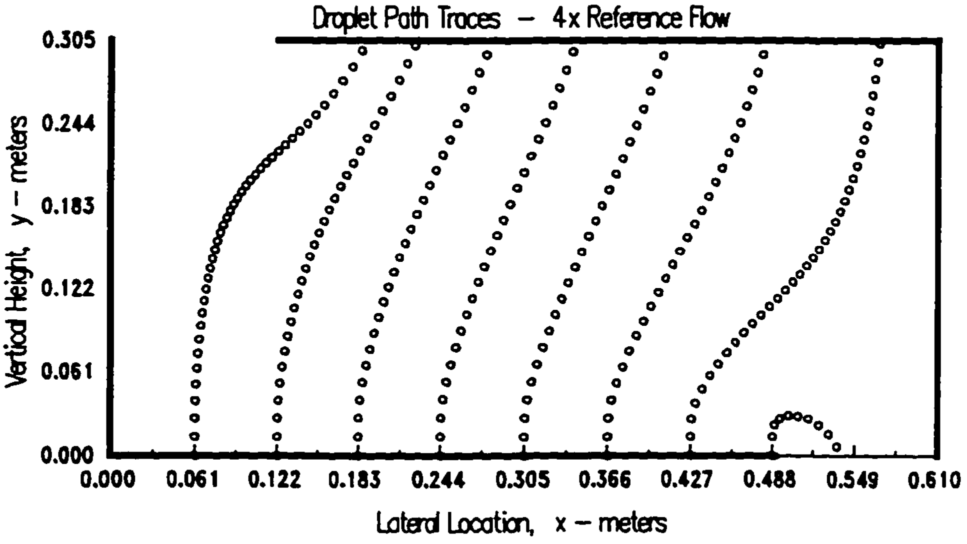

Figure 32 Droplet Path Traces - n-Pentane/H_2O. Continuous Flow Field Velocities = 4 x Reference, Droplet Radius = 2 mm.

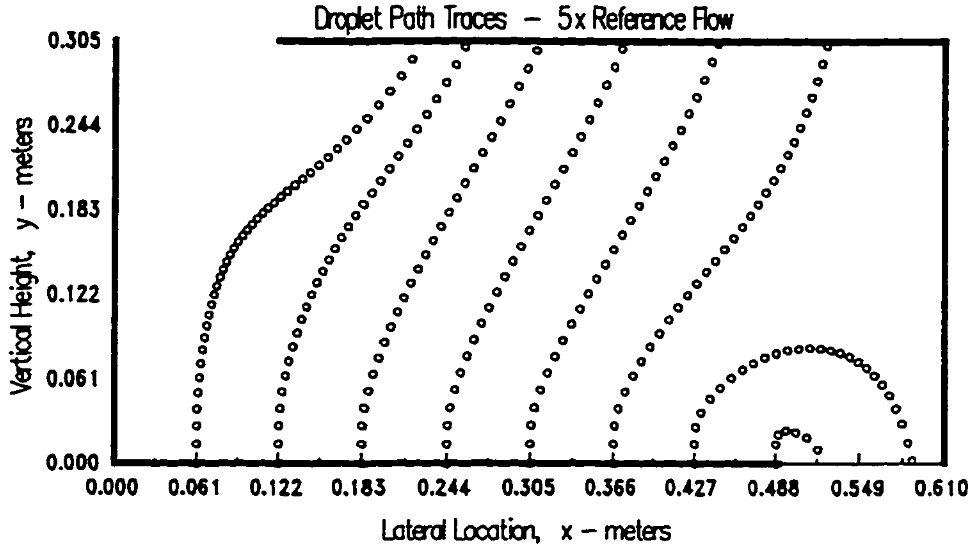

Figure 33 Droplet Path Traces - n-Pentane/H_2O. Continuous Flow Field Velocities = 5 x Reference, Droplet Radius = 2 mm.

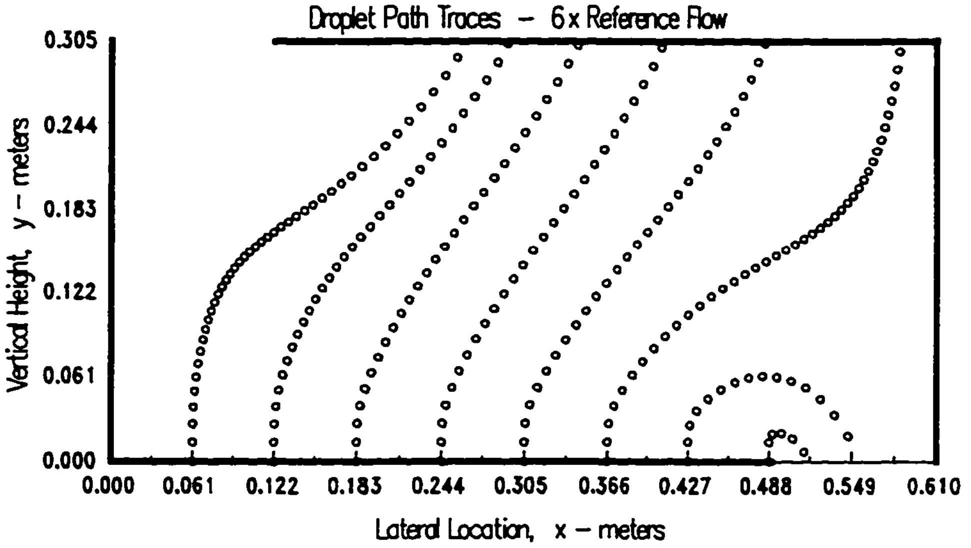

Figure 34 Droplet Path Traces - n-Pentane/H_2O. Continuous Flow Field Velocities = 6 x Reference, Droplet Radius = 2 mm.

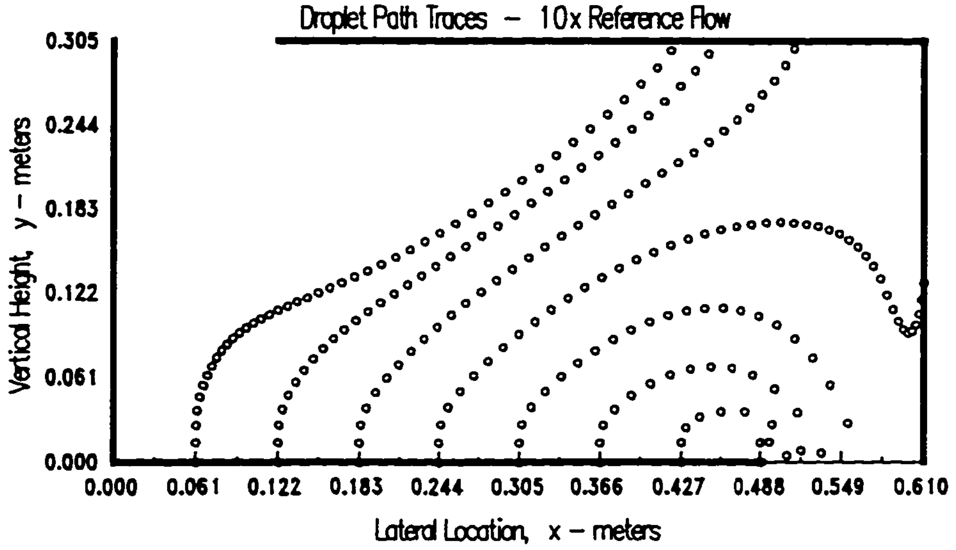

Figure 35 Droplet Path Traces - n-Pentane/H_2O. Continuous Flow Field Velocities = 10 x Reference, Droplet Radius = 2 mm.

As stated, the flow field and path traces allow for the calculation of heat transfer to a droplet due to a continuously updated relative velocity vector. The heat gained - evidenced by the droplets rise in temperature - is compared to the maximum gain possible in the module, and a droplet efficiency is found. Figure 36 shows efficiencies for individual droplet traces referenced to their release location on the lower tray. The figure shows an increase in efficiency as the flow rate is increased and that the highest gain is found in the traces closest to the by-pass gap where, as one would expect, the flow is locally contracting or expanding. Note the sharp fall off in efficiency for trace locations near the lower by-pass gap. The efficiencies of these traces are calculated to show the loss resulting from a droplet being captured by the continuous flow and being swept under the lower tray.

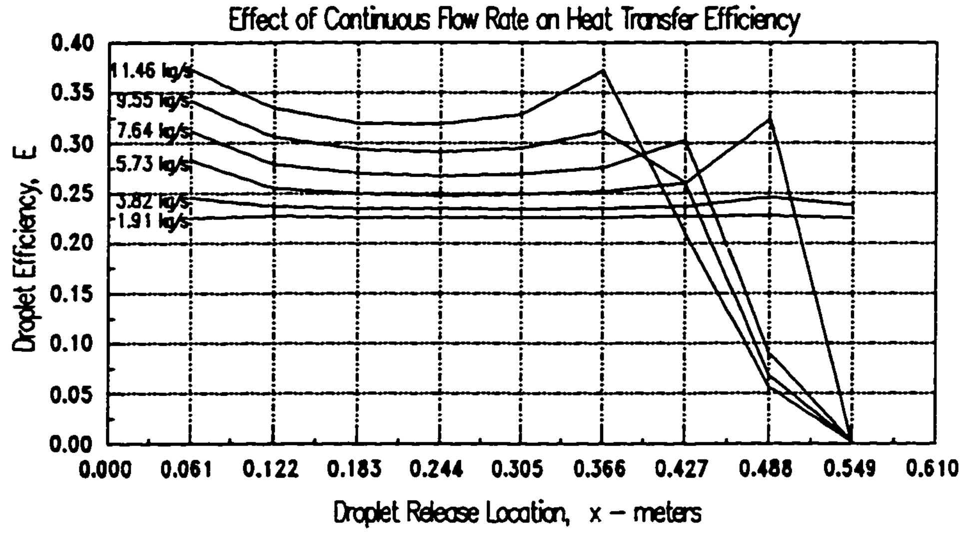

Figure 36 Droplet Effectiveness as a Function of Release Location and Continuous Flow Rate.

Droplet efficiency can be averaged for those traces that traverse the module without capture. These averages are plotted to reveal the benefit of creating the cross flow condition by sieve tray. Figure 37 shows that the average efficiency can be raised from approximately .22 to .35. This improvement is attributed to two factors - (1) the increased relative flow past the droplet due to the establishment of a cross flow and (2) the increase residence time particularly noticeable in the traces rising from the ends of the lower tray which experience a greater vertical component. The latter accounts for the saddle shape of the curves in Figure 36.

The saddle shape of Figure 36 suggests a design consideration for constructing sieve trays. Perforations made in the trays could be of greater density at the edges to exploit the higher local efficiency of these locations. Further it is seen that trays should have a non-perforated margin along the edge that protrudes into the flow to prevent droplet capture by the flow at this location.

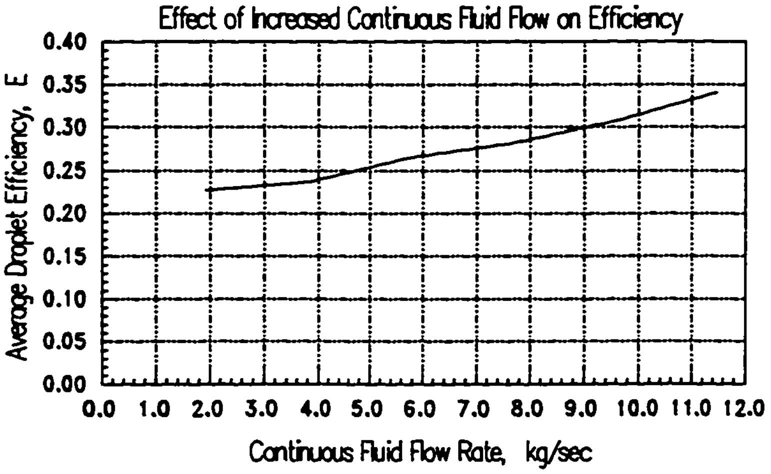

Figure 37 Effect of Continuous Flow Rate on Average Droplet Efficiency. (Only droplets traversing module are included in average.)

An alternate use of sieve trays is the strategic placement of a single tray for the purpose of droplet size control. As discussed earlier in this chapter, droplet size, volume, and population greatly affect the heat transfer. Reducing the size (radius) of droplet a factor of two creates a population four times as large, thus increasing the heat transfer.

A scenario for this approach differs from the two-dimensional analysis just presented. There is negligible droplet growth during the preheating stage of a three-phase system when the dispersed fluid is in liquid form. Bubble growth occurs during phase change, so the first consideration is the placement of the tray at a physical location in that region. If, in a particular application in which nominal liquid droplet radius was 2.0 mm and vapor bubble growth was from 2.0 mm to 9.0 mm, one wished to reset the growing bubble back to a 2.0 mm bubble when it reached a interim size of 5.0, the key question would be where to place the tray. Once a tray is positioned and the steady state heat transfer profile changes, the effect cascades through the entire column altering the preheating time and shifting the column location where vaporization (and bubble growth) begins. The result is that tray placement based on a unmodified column is no longer correct and a new location must be tried and revaluated, which leads to an experimentally endless process.

Numerically, tray placement is a substantially easier process. The simulation code can be modified to search for the placement compatible with the specified boundary and initial conditions. Figures 38 and 39 show comparisons of the heat transfer and droplet/bubble radius for an open spray column and one configured with a single sieve tray. The boost in heat transfer is quite evident at the 2.0 meter location in Figure 39 and corresponds to the resetting of the droplet size from 5.0 mm to 2.0 mm in Figure 39 and the accompanying increase in bubble population.

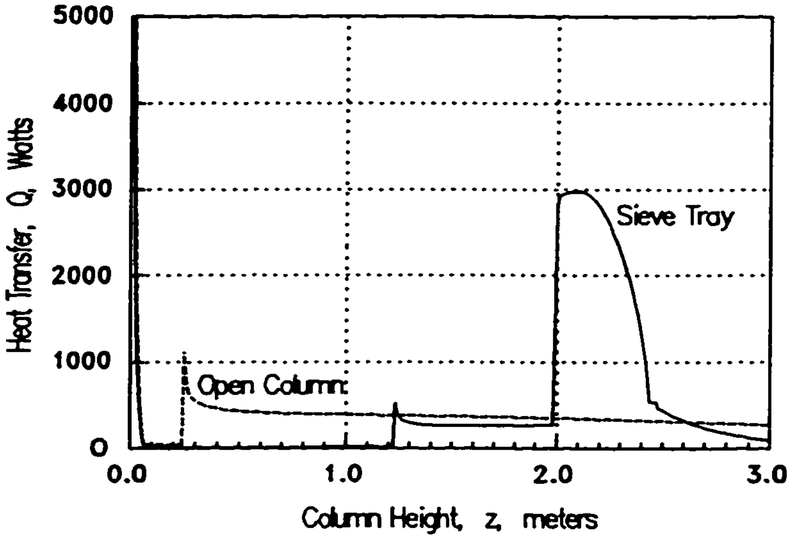

Figure 38 Comparison of Column-Wise Heat Transfer Between an Open Spray-Column and the Addition of a Single Sieve Tray Located in Vapor Rise Section of DCHX.

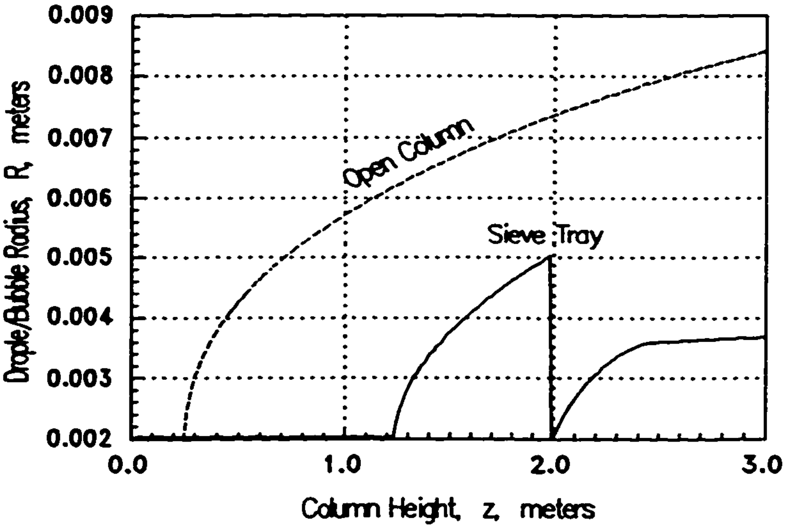

Figure 39 Effect of Radius Control by a Single Sieve Tray in Vapor Rise Section of DCHX.

It is seen in Figure 39 that bubble growth began at approximately 0.25 meters in the 3.0 meter column and that a bubble size of 5.0 mm was reached at approximately 0.8 meters. However, droplet growth appears quite different in a modified column, with onset of vaporization delayed until 1.2 meters and a 5.0 mm bubble reached at 2.0 meters. The large extraction of heat associated with the resetting of the bubble size reduces the continuous fluid temperature downstream, which in turn slows the preheating time. This causes the shift in events in the profiles, illustrating the synergism of DCHX system.

CHAPTER 4

CONCLUSIONS AND RECOMMENDATIONS

A robust one-dimensional model simulating the responses in a spray-column direct-contact heat exchanger is developed. The model represents a significant improvement over previous modelling works in that it incorporates an array of heat transfer expressions that account for the varying droplet/bubble regimes. The model also incorporates improvements that correct apparent errors in previous modelling. Comparisons are made to a limited amount of experimental data and parametric sensitivities are explored as a means of validating the work. Generally, there is good agreement with known data, suggesting a qualitative model that approaches quantitative status.

A method for determining a specific form of system optimization - dual utility - is presented. The success in determining optimization depends on the use of a new indicator developed in this work. The dual utility optimization (DUO) indicator is shown to be uniquely effective in identifying properly balanced flow rates, column dimensions, etc.

One-dimensional mechanisms are expanded into an analysis of sieve tray enhancement. By determining the two-dimensional flow field that would result from adding sieve trays on alternate sides of a column, the influence of cross flow on droplets is examined and found to increase heat transfer efficiency due to increased relative flow around an individual droplet. An improvement on the order of 40% is possible by increasing the continuous fluid flow rate in a cross flow field by a factor of five. The key element is shown to be the cross flow, since increasing flow rate in an unmodified column has markedly less influence.

Findings of this work add to the understanding of this type of heat exchanger. Key elements are as follows.

1. It is critical to the modelling that each of the regimes that droplets/bubbles experience be separately modelled to insure accuracy and faithfulness to the physics of the system. Such a treatise of the regimes has not been found in other modelling attempts.

2. Parametric sensitivities reveal that when using dispersed fluids that will undergo phase change, significant increases in system duty are achieved by introducing the dispersed fluid at or near its boiling temperature. Whether or not this is possible without some kind of artificial preheating must be evaluated for each application. Gains on the order of 100% are shown possible by properly introducing a dispersed fluid near its boiling temperature vis a vis some lower amount.

3. It has traditionally been held that increasing flow rates increases system duty. This trend is supported in this work but suggests that the rate of duty increase is not linear with fluid flow rates but rather tapers off. For design, this tapering off must be evaluated against the cost (i.e. pumping costs) of increased

flow rates.

4. One form of optimization is now readily identified through the development of a customized indicator that identifies when two fluids are being equally utilized. This ability is of benefit when both working fluids are in limited supply or costs of supplying the fluids is a factor.

5. The addition of sieve trays to enhance the preheating portion of a DCHX can improve droplet heat transfer efficiency when accompanied by increased continuous fluid flow rate. The increased relative flow around a droplet increases convection, which improves heat transfer efficiency of a droplet over a given distance.

6. Selective use of a sieve tray in the vapor bubble rise portion as a means to control (reduce) bubble size is effective but has a cascading effect on the heat transfer profile of a DCHX column. Bubble size reduction creates more surface area due to a larger number of bubbles. A heat transfer *burst* occurs as a result. However, this burst lowers the temperature profile of the continuous fluid downstream, resulting in longer preheating time.

Direct-contact mechanisms are complex, and continued research would be beneficial. The following areas are suggested.

1. Continued research in droplet/bubble heat transfer is needed to develop either additional expressions or a single more universally applicable expression for determining droplet convection in a wider range of fluids (Prandtl numbers), temperatures, and Reynolds numbers . Existing expressions are still somewhat

restrictive or are unproven over wide ranging conditions.

2. Further work in optimization should be pursued, especially a study of the influence of switching the roles of the fluids. A cursory examination using the one-dimensional model revealed that in some cases simply switching the continuous fluid to the dispersed increased duty, though this was not a consistent event. It is suspected that a link exists between system duty and a mix of the fluids' viscosity ratio and thermophysical properties. Rise velocity directly affects convection around the droplet, and a correlation may be found to help determine which fluid should be dispersed into the other.

3. Expanded two-dimensional work to include extension to whole system modeling and alternate forms of enhancements such as packed beds, agitators, etc. should be pursued. Novel approaches must be found to prevent the tedious and computationally expensive task of individual droplet tracking. Both experimental and computational pursuits are required for validation and completeness. It is known that enhancements are of benefit, but defining useful expressions is needed if these types of heat exchangers are to see increased use.

4. An inevitable area of research will be injection nozzle design since as stated in the literature, injection circulation is a key regime in the heat transfer process. Since a number of injection methods are possible, increasing the knowledge base in this area will be important in eventual designs.

APPENDIX I

DEVELOPMENT OF VAPOR BUBBLE RISE REGIME HEAT TRANSFER COEFFICIENT

Following a similar analysis as that of Coban and Boehm (1986), an expression for the heat transfer coefficient for the vapor bubble rise regime is found. The development differs from the Coban and Boehm work in that an alternative expression from Carslaw & Jaeger (1959) is used as a starting point. This alternative start expresses the temperature profile of a sphere due to heat transfer from a *linearly varying continuous medium* as opposed to the expression chosen by Coban and Boehm, which assumes a linearly changing surface temperature. It will is shown that ending results are similar for the cases where internal conduction resistance is much greater than the convective resistance but that the new result will also work for those few cases where this condition is not true. This analysis begins with the Carslaw & Jaeger expression for the temperature profile inside a sphere exposed to a linearly changing continuous fluid.

$$T(r,t) = \gamma\left[t+\frac{r^2RH-R^2(2+RH)}{6\alpha RH}\right] + \frac{2R^2H\gamma}{\alpha r}\sum_{n=1}^{\infty}\frac{e^{-\alpha\lambda_n^2t}\sin r\lambda_n}{\lambda_n^2\{R^2\lambda_n^2+RH(RH-1)\}\sin R\lambda_n} \qquad \text{I-1}$$

Neglecting the oscillatory transient portion

$$T(r,t) = \gamma\left[t + \frac{r^2RH-R^2(2+RH)}{6\alpha RH}\right] \qquad \text{I-2}$$

Of particular note are the terms, H and γ, in the above expression. The Carslaw and Jaeger reference defines H as the ratio of the exterior convection coefficient to the conduction coefficient of the vapor inside the bubble or h_o/k_v. γ is the coefficient of the time variable describing the linearly changing temperature of the medium surrounding the bubble. While it is important, γ will be eliminated in the final steps of the development. Other terms of note are α ($k/\rho c_p$), the well known thermal diffusion coefficient, R, the physical radius of the bubble, and lower case r, the radial location variable for the sphere. The reader is directed to Carslaw and Jaeger for a complete description of the terms contained in the transient portion of equation I-1. No detail is given since this portion is dropped in this analysis.

The development is continued by finding the average interior temperature for use in a transfer relation. The average is used because it is assumed that a bubble will be sufficiently small and exhibit some mixing and thus will be of homogeneous temperature. The average is found by

$$T_{ave} = \frac{\int_{Vol} T(r,t)\,dV}{V} = \frac{\int_0^R \gamma\left[t + \frac{r^2RH-R^2(2+RH)}{6\alpha RH}\right]4\pi r^2 dr}{\frac{4}{3}\pi R^3} \qquad \text{I-3}$$

which reduces to

$$T_{ave} = \gamma\left[t - \frac{R}{3\alpha H} - \frac{R^2}{15\alpha}\right] \qquad \text{I-4}$$

A temperature differential for use in a *total* convective expression for the bubble is found by

$$\Delta T = T_c - T_{ave} = \gamma t - \gamma\left[t - \frac{R}{3\alpha H} - \frac{R^2}{15\alpha}\right] = \gamma\left[\frac{R}{3\alpha H} + \frac{R^2}{15\alpha}\right] \qquad \text{I-5}$$

The temperature gradient at the bubble's surface is derived from equation I-2. The gradient is used to set up a conduction expression at the surface.

$$\frac{dT}{dr}\bigg|_R = \gamma\left[\frac{r}{3\alpha}\right]_R = \frac{\gamma R}{3\alpha} \qquad \text{I-6}$$

A relationship is now established, equating the surface conduction into the sphere to the overall transfer into the interior of the sphere by convection. (Note the implication made that the heat transfer process is steady state, confirmed by the elimination of the time variable from the gradient and differential expressions. This relationship is

$$\dot{Q} = k_v A_B \frac{dT}{dr}\bigg|_R = h_B A_B \Delta T \qquad \text{I-7}$$

or

$$\dot{Q} = k_v A_B \frac{\gamma R}{3\alpha} = h_B A_B \gamma\left[\frac{R}{3\alpha \frac{h_o}{k_v}} + \frac{R^2}{15\alpha}\right] \qquad \text{I-8}$$

Thus an expression for the overall transfer coefficient which accounts for the outside convection and interior conduction of a bubble in the vapor bubble rise regime, h_{vbr} (h_B in the above expression), can be defined as

$$h_{vbr} = \frac{5h_o k_v}{5k_v + Rh_o} \qquad \text{I-9}$$

This expression is similar to the expression used by Coban and Boehm in that it represents the series combination of the exterior convection resistance and the internal conduction resistance. By noting that a separation of terms yields

$$h_{vbr} = \left[\frac{1}{h_o} + \frac{R}{5k_v}\right]^{-1} \qquad \text{I-10}$$

the individual resistances are revealed. The last term is the same "*internal convection*" term derived by Coban and Boehm.

The h_o term is the convection coefficient acting between the continuous medium and the exterior surface of the bubble. A suitable expression for this term is described in Chapter 2.

It is of note to observe that if $5k_v/R << h_o$, then the heat transfer will be dominated by the $5k_v/R$ term

$$h_{vbr} = \frac{5k_v}{R} \; ; \quad \left(\frac{5k_v}{R} << h_o\right) \qquad \text{I-11}$$

which is the Coban and Boehm result. In selecting the alternative starting expression, a result is reached which essentially duplicates the Coban and Boehm result for those cases where the internal conduction resistance is considerably greater than the exterior resistance, which is true in a majority of cases. The improvement in this version is that the external convection resistance is retained for those cases when this condition is not met, making this a more universal expression.

APPENDIX II

DUAL UTILIZATION OPTIMIZATION (DUO) INDICATOR

A measure of whether the temperature profiles of a DCHX are those of a system exhibiting the condition of optimized dual utility can be constructed from a measure of two key characteristics of the profiles - the degree of parallelism and closeness of approach. Parallelism is determined by subtracting the temperature difference at both ends of the DCHX column. If the differentials are equal, the difference will be zero, indicating perfectly parallel inlet and outlet conditions. Conversely, nonparallel profiles retain some value. To convert this measure to one that ranges between 0 and 1 and retains a convention of 0=worst case/1=best case, the absolute value of the differential is divided by the maximum difference possible and subtracted from 1. This results in an indicator for parallelism which becomes a component of the final DUO indicator and is given as follows.

$$\left(1 - \left|\frac{\Delta T_1 - \Delta T_2}{\Delta T_{\max}}\right|\right) \qquad \text{II-1}$$

The second component - closeness of approach - is constructed to determine the closeness of approach of the temperature profiles using the commonly known log mean temperature difference (LMTD). The LMTD is divided by the maximum possible temperature difference, yielding a value bounded by 0 and 1. In this manner, the closer the approach is, the smaller the LMTD derivative becomes, approaching a value of 0 in the limit. The result is then subtracted from 1 to convert to the convention of 0=non optimum/1=optimum as follows

$$\left(1 - \frac{LMTD}{\Delta T_{\max}}\right) \qquad \text{II-2}$$

The two above expressions are multiplied together to form the DUO indicator as such

$$DUO = \left(1 - \frac{LMTD}{\Delta T_{\max}}\right)\left(1 - \left|\frac{\Delta T_1 - \Delta T_2}{\Delta T_{\max}}\right|\right) \qquad \text{II-3}$$

Like its components, the DUO indicator ranges between 0 and 1 with 1 being the ideal case. It should be noted, though, that the so called *ideal* case - coincidental profiles - is an impossibility since if there is no temperature differential, then there is no heat transfer. Rather, the usefulness of the indicator lies in the location of its peak value as a particular parameter - say, flow rate - is varied, indicating that the value of the parameter has been found for dual optimized fluid utility. Further discussion of this indicator and results of its use are given in Chapters 2 and 3.

APPENDIX III

DIRECT CONTACT, GAS-FIRED WATER HEATER

Two designs are presented for the adaptation of a spray-column direct contact heat exchanger for producing hot water in commercial applications from combustion exhaust. The designs are developed utilizing the numerical simulation of two and three-phase spray-columns described in this dissertation to demonstrate the utility of the numerical simulation and to show that such systems are feasible and substantially less expensive than the classical boiler/shell-and-tube designs.

Overview

Water heating from products of combustion is a frequently applied industrial process. Principally, this can be accomplished through the use of direct-contact heat exchangers (DCHXs), the general advantages being DCHXs are (a) a more efficient, (b) exhibit almost no fouling, and (c) have lower capital costs.

Employing the developed one-dimensional model results in the creation of of two separate system designs - an *on-demand (OD)* and *thermal reservoir (TR)*. Simulation analysis of the OD design resulted in a system that permits an *on-demand* draw of water heated from a tap condition of 22°C to 49°C (72°F to 120°F). The design is made on a water flow rate per unit area of column cross section and can be sized to meet a particular need. The second system, the *thermal reservoir (TR)*, is one that requires the spray-column to double as a thermal storage tank for a quantity of water heated beyond the required use temperature and then mixed with tap condition water in the proper ratio. This design is especially suited for situations where demand flow rate is high but not necessarily continuous and the system can run during off hours or overnight generating the hot reserve supply.

In both cases, the use of a DCHX shows to be highly efficient, having an effectiveness (ϵ) of 1.0. The only limitation is that the water must come in direct contact with combustion exhaust rendering it non-potable and must, therefore, be limited to such as linen cleaning, leather tanning, or industrial applications.

On-Demand Design

The *OD* design is desirable due to the flexible for reasons stated. The design allows for a one pass through on an as-needed basis only and is useful for moderate temperature and flow rate applications. It is the simplest and most true to the counter-flow spray-column DCHX concept. Tap water enters at the top of a spray-column and is heated by a rising flow of hot combustion gases. The exiting water is heated to the desired temperature during the single pass through the spray-column. Such a design is achievable for a variety of output temperatures and flow rates. Several concerns arise in specifying a design for an on-demand system. First, an adequate temperature rise must occur in one pass through. Second, this rise must occur for a reasonable flow rate of water. Natural gas, which is primarily methane, when combusted with air in the proper ratio, can produce an exhaust gas at 1500°C. The combustion products retain the high percentage of nitrogen from the air therefore exhibiting nearly the same properties as air. Bringing the hot gas in contact with water results in a highly effective transfer leading to two possible outcomes. Either a small flow of water can experience a large increase in temperature or a large flow of water can undergo a small or moderate change in temperature or some combination of the two in accordance with the respective mass rate and thermal capacitance of the fluids. A optimal balance is, of course, the desired goal.

Achieving a usable temperature rise with a substantial flow requires a high combustion gas mass flow which can lead to flooding; the condition where the spray-column ceases to support the counterflow of fluids. At an operating pressure of one atmosphere, simulations show that flooding is likely when the dispersed-to-continuous-mass-flow-rate ratio approaches .15 for the combustion gas temperature stated. This is due to the large difference in fluid densities.

The exact ratio at which flooding occurs is influenced by the initial dispersed gas bubble diameter. The smaller the bubble size, the more efficient the heat transfer is. Typically, diameters ranging from 2 mm to 20 mm are desirable. Small bubble sizes mean more bubbles per unit volume and therefore more total bubble surface area in contact with the continuous fluid for heat transfer. However, total bubble drag also becomes high which leads to flooding in the column. It is necessary to introduce the combustion gas bubble at a size that precludes flooding but yet is within the accepted limits of the spherical bubble assumption. A design inlet bubble diameter of 16 mm is used.

Table 1 lists the parameters for an *OD* design for a spray-column DCHX operating at a nominal pressure of one atmosphere. These parameters are listed on a per square meter of column cross sectional area. To increase or decrease the output flow requirements, one adjusts the DCHX radius while holding all other parameters constant. The mass flow rate of 0.075 kg/sec of water equates to 272 liters/hr or 72 gal/hr. The system is capable of 9400 watts of transfer when operating at atmospheric pressure which will raise the flow of water from 22°C to 49°C (72°F to 120°F), a rise of 27°C (48°F). The effectiveness of this system in extracting heat from combustion gas is 1.0 based on the inlet conditions of the gas properties. The actual transfer of heat is slightly less due to changes in properties as the gas cools. Further, the heat transfer to the water differs from the heat transfer from gas due to the small amount of heat lost in the mass transfer effects (evaporation of water to the gas bubbles).

Figures 2 through 4 depict profiles of key fluid parameters near the bottom of the column. As mentioned, for the specified dispersed inlet bubble radius, nearly all heat transfer occurs in the first few centimeters of the column just as the combustion gas enters. If the inlet bubble size is allowed to increase the transfer stretches out further along the column height. Figure 5 depicts the trend of control volume transfer for a range of larger bubble sizes. The limitations associated with large bubble diameters (i.e. not retaining a spherical shape and/or

Table III-1 **Parameter Specification (Per Unit Cross Section) for On-Demand Spray-Column DCHX Water Heater.**

Parameter	Value
Dispersed Fluid	Methane/Air Combust. Products
Flow Rate	.005 kg/sec
Inlet Temp	1500°C
Bubble Dia.	1.6 cm
Continuous Fluid	Water
Flow Rate	.075 kg/sec
Inlet Temp	22°C
DCHX Radius	.564 m
DCHX Length	1.0 m
Operating Press.	101.325 kPa
Actual Heat Transfer Rate to Water	8500 watts
Max Transfer Rate	9400 watts
$\epsilon_{\text{(based on inlet cond.)}}$	≈ 1.0
Water Outlet Temperature	49°C
LMTD	87.6°C
Volumetric Heat Transfer Coeff.	96.9 W/m^3 °C

possible agglomeration and/or breakup are acknowledged and trends are shown for comparative purposes.

A second and perhaps more preferable method exists for increasing the flow rate and/or temperature rise - that is by increasing the operating pressure. Increasing the pressure increases the density of the inlet exhaust gas which reduces the bubble volume and surface area. This eases flooding tendencies thereby allowing higher combustion gas flow rates which in turn introduces more heat to the system. There is relatively little effect on effectiveness, thus, duty increases proportionally with the pressure.

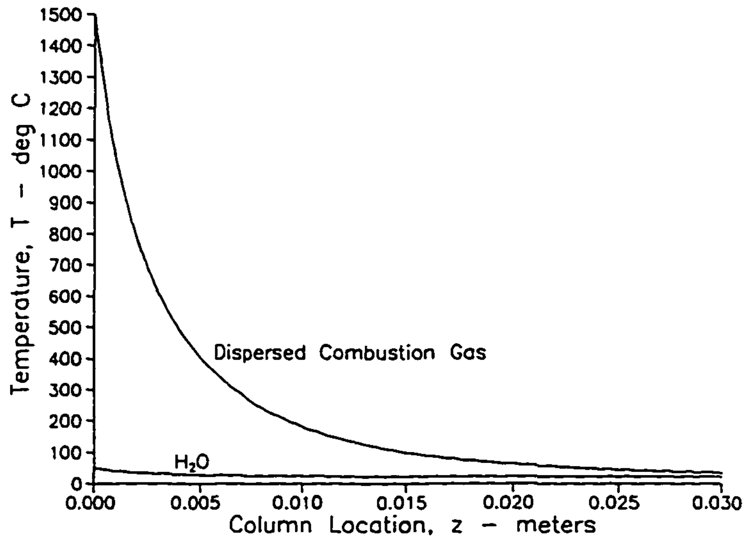

Figure III-1 Temperature Profiles for Bottom Portion of the *On-Demand* System Described in Table III-1.

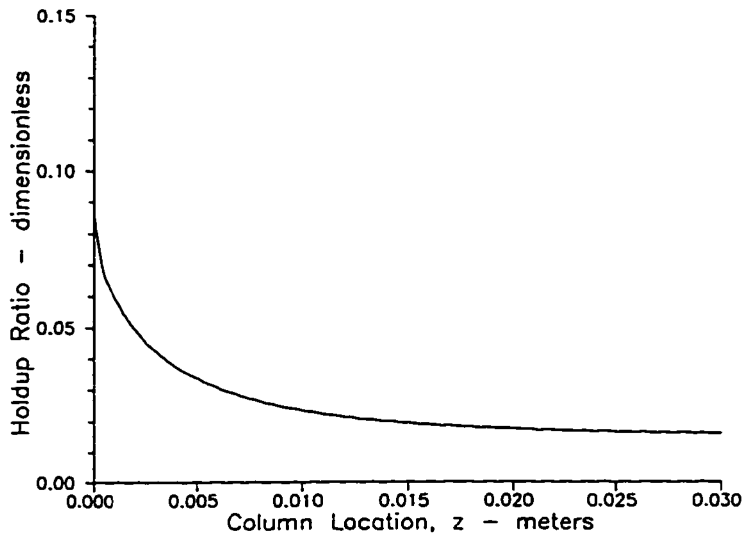

Figure III-2 Holdup Ratio for Bottom Portion of the *OD* System Described in Table III-1.

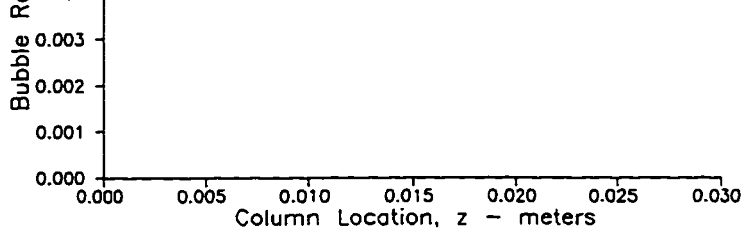

Figure III-3 Dispersed Fluid Bubble Radius for Botto m Portion of the *OD* System Described in Table III-1.

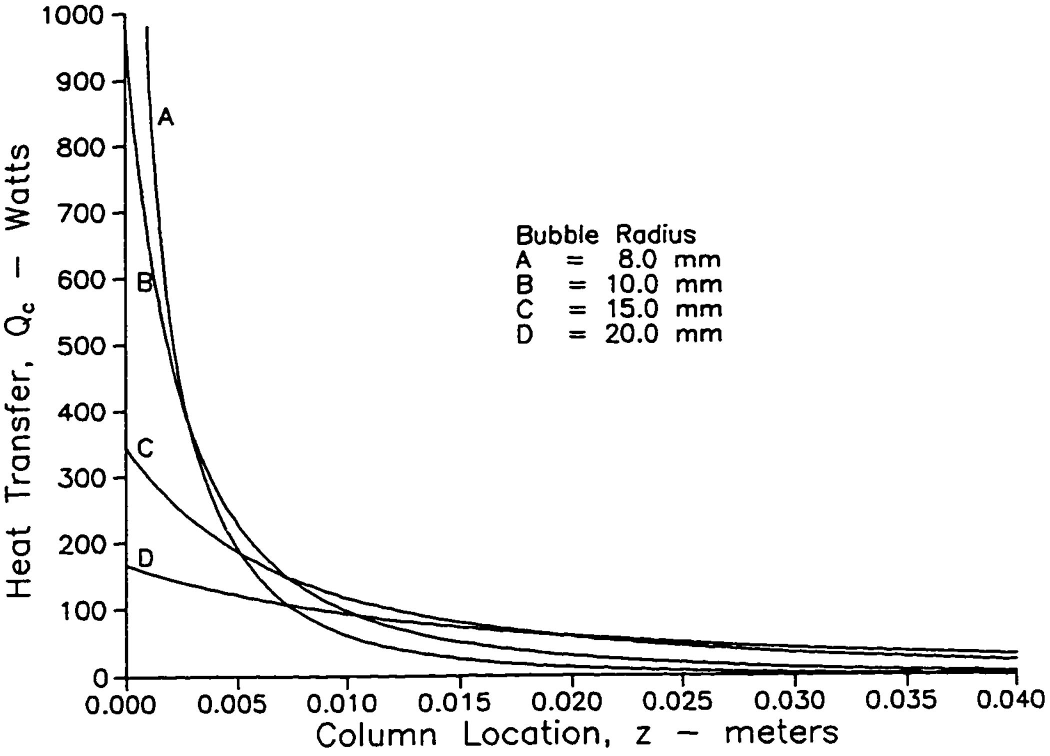

Figure III-4 Control Volume Heat Transfer Profile for *OD* System Described in Table III-1.

Thermal Reservoir Design

Consider an application that requires hot water flow rates of 2.0 kg/sec (≈1900 gal/hr). If the *OD* design spray-column is sized for this flow, the column flow area would have to be 27 m^2 equating to a diameter of 2.9 m. For such a case, an *OD* system may not be practical since the physical size may not be easily accommodated. An alternative approach is possible with a system that continuously recirculates water through a spray-column which doubles as a *thermal reservoir*. This allows a small system to transfer heat continuously, overshooting the design temperature in favor of overheating the water. The overheated supply is mixed with cooler tap water in the proper ratio to supply the higher desired flow rate at the required temperature. A necessary condition for the *TR* design is that the demand is not continuous. The off-demand time is used to build the thermal reservoir up to the higher temperature for use during a limited period of draw. The size of a *TR* system can still be large depending on the amount of water stored but is considerably smaller than a proportionally sized *OD* system. The large size results in very small water flow velocities inside the column - usually on the order of a few millimeters per second - making this design similar to the familiar well-stirred reactor with a small flow of the bulk fluid. The reservoir design generally is selected for continuous heating to higher temperatures to meet high demand periods of shorter duration.

Table III-2 Parameter Specification for Thermal Reservoir Spray-Column DCHX Water Heater.

Parameter	Value
Dispersed Fluid	Methane/Air Combust. Products
Flow Rate	.045 kg/sec
Inlet Temp	1500°C
Bubble Dia.	3.0 cm
Continuous Fluid	Water
Flow Rate	1.6 kg/sec
Inlet Temp	22°C[†] - 84°C[‡]
DCHX Radius	1.1 m
DCHX Length	5.5 m
Operating Press.	101.325 kPa
Actual Heat Transfer Rate to Water	77,000[†] - 67,500[‡] watts
Max Transfer Rate Possible	84,900[†] - 81300[‡] watts
$\epsilon_{\text{(based on inlet cond.)}}$	≈ 1.0
Water Temp Rise per pass	12°C[†] - 10°C[‡]
LMTD	94.2°C[†] - 117.3°C[‡]
Volumetric Heat Transfer Coefficient	39.2 W/m^3 °C[†] - 27.5 W/m^3 °C[‡]

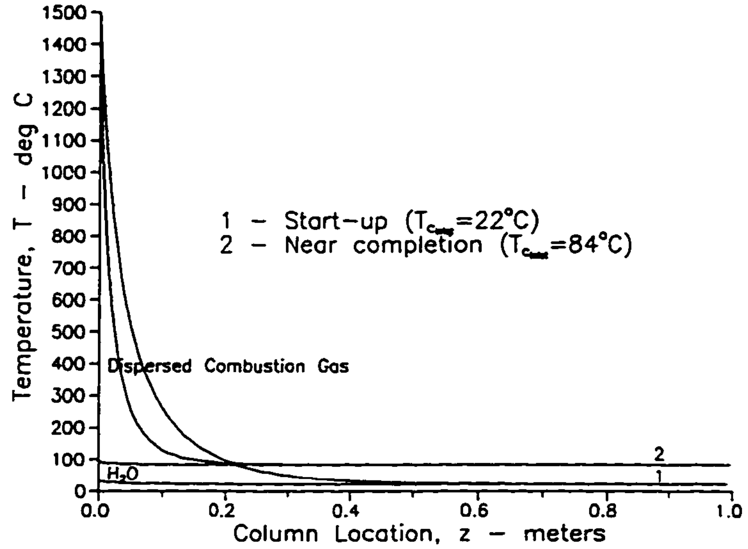

Figure III-5 Temperature Profiles for *Thermal Reservoir* System Described in Table III-2. 1-System Start-Up, 2-Near Completion.

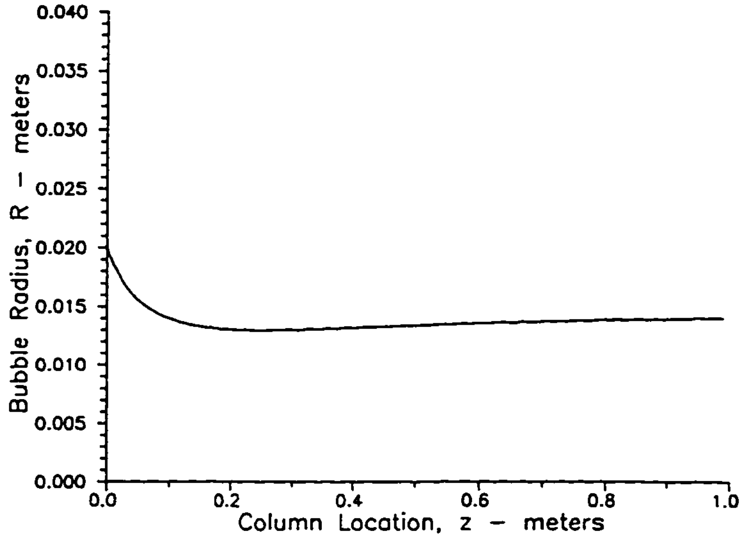

Figure III-6 Typical Profile of Dispersed Bubble Radius for *TR* System Described in Table III-2.

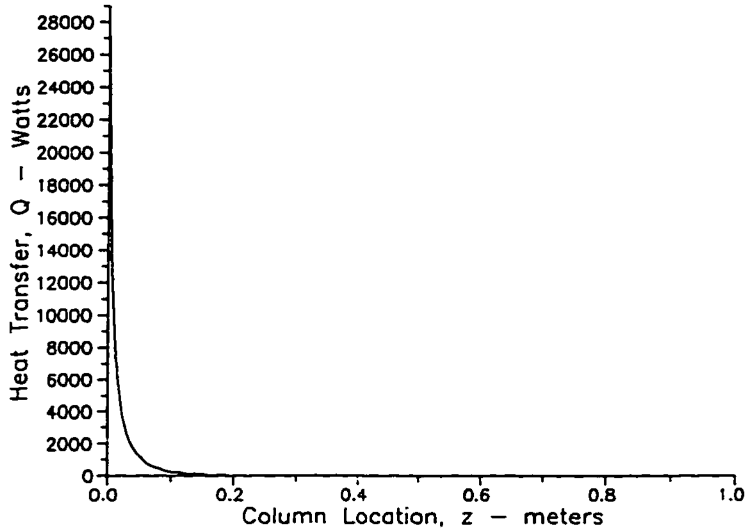

Figure III-7 Control Volume Heat Transfer for *TR* System Described in Table III-2. (Start-up conditions.)

Table 2 lists the parameters of a spray-column system that meets one particular case study in which a user had a requirement for 57000 liters of hot water per day but operated only one 8 hour shift. This system recirculates and heats 21000 liters (5500 gals) of water to 95°C (203°F) which when mixed with 36000 liters (9500 gals) of tap water at 22°C (72°F) will produce 57000 liters (15000 gals) of water at 49°C (120°F). The system operates at a duty of 77,000.

As in the *OD* design, combustion gas is introduced at 1500°C. Water passing through the spray column at the specified parameters will gain an average of 11°C per pass. Exit water is reintroduced at the top of the spray column where is mixes into the bulk flow. Eventually (approximately one day), the total bulk temperature will reach an elevated temperature of 95°C. The 11°C gain holds nearly constant with minimal loss of effectiveness as the mixed temperature entry water rises to 95°C. This is due to the high temperature of the inlet gas. The max ideal transfer to water starting at 22°C is nearly equal to the ideal transfer for water starting at 84°C since there is less than a 5% difference in the maximum gas/water temperature differential. The ideal transfer holds between 84,900 - 81,300 watts (start - near completion). The effectiveness of extracting heat from the dispersed fluid is again constant at approximately 1.0 with (as stated earlier) the effectiveness of heat transfer into the water being slightly less due to property changes and mass transfer.

Figures 6, 7, and 8 depict typical parametric profiles for the *TR* system. The inlet bubble diameter given in Table 2 is listed at 3.0 cm. It is acknowledged that some assumptions may break down for this large of a bubble and these effects are accounted for. Figure 7 shows the reduction in the bubble size as it rises and cools to a higher density. Flooding potential only occurs near the dispersed gas inlet and then ceases to be a risk as the bubble shrinks. As in the first design, Figure 8 indicates that a bulk of the heat transfer occurs within the first few percent of length.

System Costs

A cost analysis for the *TR* design given has been done and compared to a comparably sized boiler/shell and tube system. Spray-column direct contact heat exchangers have inherently lower costs due primarily to less complexity of the systems and the elimination of a substantial amount of material (i.e. numerous tubes) from the exchanger portion. Estimated cost of a *TR* system is approximately one fifth of a conventional boiler/shell and tube system.

The total cost of a *TR* system is approximately $6000 (1993 U.S. dollars). The cost of a comparable conventional system consisting of a boiler, shell and tube heat exchanger, 3750 liter (1000 gal.) storage vessel, and associated components is estimated at $33,600 (1993 U.S. dollars) indicating that the cost of the DCHX is less than 20% that of the conventional system.

APPENDIX IV

CURVE FIT FLUID PROPERTY EXPRESSIONS

Curve fits of fluid property data were constructed for use in the one-dimensional code. Fluid properties come from a number of sources including Coban (1986) and references such as Vargaftik (1975) and the *1989 ASHRAE Fundamentals Handbook* in the case of naturally occurring or common gases and liquids and promotional pamphlets and brochures for some of the commercially produced heat transfer fluids. This appendix comprises is a compilation of expressions used.

(Unless otherwise noted, units are $T(°C)$, $c_p(J/kg\text{-}K)$, $\rho(kg/m^3)$, $k(W/m\text{-}k)$, $\mu(Pa\text{-}s)$, *and results are for standard pressure of 101.325 kPa.)*

Therminol® 44 *liquid*
TH44

For $-45 < T < 260$
$c_p = 2.47558T + 1895.91$
$\rho = 940.03 - 0.75776T - 1.2075\text{x}10^{-4}T^2$
$k = 1.4566\text{x}10^{-1} - 1.2663\text{x}10^{-4}T - 1.5176\text{x}10^{-7}T^2$

For $-45 < T < 10$
$\mu = .0168827\exp(-.0876271T)$

For $10 < T < 260$
$$\mu = .014207 - 5.84755\text{x}10^{-4}T + 1.1979\text{x}10^{-5}T^2 - 1.409\text{x}10^{-7}T^3 + 9.8367\text{x}10^{-10}T^4 - 4.01099\text{x}10^{-12}T^5 + 8.80257\text{x}10^{-15}T^6 - 8.0192\text{x}10^{-18}T^7$$

Therminol® 55 - *liquid*
TH55
For $0 < T < 343$
$c_p = 3.38436T + 1862.95$
$\rho = 895.12 - 0.66735T - 6.1756 \text{x} 10^{-6}T^2$
$k = .130426 - 8.3889 \text{x} 10^{-5}T - 1.17235 \text{x} 10^{-7}T^2$

For $0 < T < 38$
$\mu = .301559\exp(-.0713115T)$

For $38 < T < 343$
$\mu = .104613 - 3.9908 \text{x} 10^{-3}T + 7.0359 \text{x} 10^{-5}T^2 - 7.1588 \text{x} 10^{-7}T^3 + 4.5138 \text{x} 10^{-9}T^4$
$- 1.7912 \text{x} 10^{-11}T^5 + 4.3535 \text{x} 10^{-14}T^6 - 5.9175 \text{x} 10^{-17}T^7 + 3.444 \text{x} 10^{-20}T^8$

Therminol® 60 - *liquid*
TH60
For $-50 < T < 343$
$c_p = 1517.5 + 3.7058T$
$\rho = 1007.9 - 0.64663T - 6.8946 \text{x} 10^{-5}T^2$
$k = .13363 - 7.6511 \text{x} 10^{-5}T - 3.6863 \text{x} 10^{-9}T^2$

For $-50 < T < 10$
$\mu = .0233656\exp(-.0841699T)$

For $10 < T < 343$
$\mu = .019873 - 8.1667 \text{x} 10^{-4}T + 1.7149 \text{x} 10^{-5}T^2 - 2.0926 \text{x} 10^{-7}T^3 + 1.554 \text{x} 10^{-9}T^4$
$- 7.1014 \text{x} 10^{-12}T^5 + 1.9461 \text{x} 10^{-14}T^6 - 2.9291 \text{x} 10^{-17}T^7 + 1.8593 \text{x} 10^{-20}T^8$

Therminol® 66 - *liquid*
TH66
For $-18 < T < 371$
$c_p = 1514.7 + 3.5326T$
$\rho = 1020.3 - 0.673361T - 2.7852 \text{x} 10^{-5}T^2$
$k = .120973 - 5.08537 \text{x} 10^{-5}T - 1.06489 \text{x} 10^{-7}T^2$

For $-18 < T < 49$
$\mu = .928583\exp(-.0891155T)$
ELSE
For $49 < T < 371$
$\mu = 14.3448T^{-1.79827}$

Therminol® 88 - *liquid*
TH88
For $149 < T < 482$.
$c_p = 2.1984T + 1627.2$
$\rho = -1.2077x10^{-4}T^2 - 0.7389T + 1121.2$
$k = .139621 - 7.9469x10^{-5}T - 2.8337x10^{-9}T^2$
$\mu = 3.6914T^{-2.01174}$

Therminol® VP-1 - *liquid*
MTHVP1
For $15 < T < 404$
$c_p = 1502.4 + 2.7206T$
$\rho = 1071.2 - 0.62931T - 7.8203x10^{-2}T^2$
$k = .143285 - 1.53527x10^{-4}T + 2.03924x10^{-8}T^2$
$\mu = .00792082 - 2.0877x10^{-4}T + 2.9702x10^{-6}T^2 - 2.6742x10^{-8}T^3$
$+ 1.5734x10^{-10}T^4 - 5.9706x10^{-13}T^5 + 1.3963x10^{-15}T^6$
$- 1.8196x10^{-18}T^7 + 1.0071x10^{-21}T^8$

Dowtherm® SR-1 - *liquid*
DTSR1
For $0 < T < 125$
$c_p = 3204.1 + 3.8701T$
$\rho = 1082.5 - 0.33961T - 2.4148x10^{-3}T^2$
$k = .363 + 9.09344x10^{-4}T - 3.81547x10^{-6}T^2$
$\mu = .00733697 - 1.23288x10^{-4}T + 5.53221x10^{-7}T^2$

Dowtherm® 4000 - *liquid*
DT4000
For $4 < T < 177$
$c_p = 3169.2 + 3.9563T$
$\rho = 1096.7 - 0.491T - 1.5049x10^{-3}T^2$
$k = .362926 + 9.2063x10^{-4}T - 3.91495x10^{-6}T^2$
$\mu = .00746199 - 1.53974x10^{-4}T + 1.4021x10^{-6}T^2 - 2.825x10^{-9}T^3$

Dowtherm® 4000 (30% Glycol) - *liquid*
DT4030
For $4 < T < 176$
$c_p = 3741.2 + 1.5305T$
$\rho = 1061.6 - .346849T - .00193194T^2$
$k = .437696 + 1.24698x10^{-3}T - 4.8875x10^{-6}T^2$
$\mu = -8.3167x10^{-4}\ln(T) + .00441$

Dowtherm® 4000 (40% Glycol)- *liquid*
DT4040

For $4 < T < 176$
$c_p = 3533.9 + 2.8283T - .00219331T^2$
$\rho = 1071.6 - .34685T - .00193194T^2$
$k = .403952 + .001046T - 4.1122\text{x}10^{-6}T^2$
$\mu = -.00116578\ln(T) + .00609$

Dowtherm® 4000 (50% Glycol) - *liquid*
DT4050

For $4 < T < 176$
$c_p = 3321.4 + 3.49T - 1.1506\text{x}10^{-3}T^2$
$\rho = 1092.4 - .5401T - .0011855T^2$
$k = .3719 + 8.4555\text{x}10^{-4}T - 3.3368\text{x}10^{-6}T^2$
$\mu = -.00161257\ln(T) + .0083384$

Dowtherm® 4000 (60% Glycol) - *liquid*
DT4060

For $4 < T < 176$
$c_p = 3103.8 + 4.4285T - .0020395T^2$
$\rho = 1112.9 - .754322T$
$k = .339419 + 7.60199\text{x}10^{-4}T - 2.8656\text{x}10^{-6}T^2$
$\mu = .0323917T^{-.866}$

Dowfrost® - *liquid*
DWFRST

For $4 < T < 121$
$c_p = 3455.9 + 3.8394T$
$\rho = 1054.1 - 0.43901T - 2.5735\text{x}10^{-3}T^2$
$k = .349979 + 7.25542\text{x}10^{-4}T - 4.00611\text{x}10^{-6}T^2$
$\mu = .0152155 - 2.79327\text{x}10^{-4}T + 1.30927\text{x}10^{-6}T^2$

Dowfrost® HD - *liquid*
DWFHD

For $4 < T < 177$
$c_p = 3297.7 + 5.0675T - 4.9202\text{x}10^{-3}T^2$
$\rho = 1071.4 - 0.61005T - 9.05\text{x}10^{-4}T^2$
$k = .34947 + 8.504\text{x}10^{-4}T - 6.3946\text{x}10^{-6}T^2 + 1.1496\text{x}10^{-7}T^3$
$\mu = .0155255 - 3.5542\text{x}10^{-4}T + 2.765\text{x}10^{-6}T^2 - 7.0061\text{x}10^{-9}T^3$

Dowfrost® HD (30% Glycol) - *liquid*
DFHD30
For $4 < T < 162$
$c_p = 3894.6 + 1.7059T - 2.2976\text{x}10^{-3}T^2$
$\rho = 1052.6 - .587151T - 6.4261\text{x}10^{-4}T^2$
$k = .4205 + .0012295T - 5.4564\text{x}10^{-6}T^2$
$\mu = ½(.004767\exp(-.018319T) + .018137T^{-.780412})$

Dowfrost® HD (40% Glycol) - *liquid*
DFHD40
For $4 < T < 162$
$c_p = 3736.9 + 2.3371T - 2.11822\text{x}10^{-3}T^2$
$\rho = 1062.3 - .51924T - .00142626T^2$
$k = .38 + 9.759\text{x}10^{-4}T - 4.0706\text{x}10^{-6}T^2$
$\mu = ½(.0077474\exp(-.020585T) + .0352158T^{-.880378})$

Dowfrost® HD (50% Glycol) - *liquid*
DFHD50
For $4 < T < 162$
$c_p = 3496.9 + 3.5702T - .0020608T^2$
$\rho = 1073 - .757688T$
$k = .34087 + 8.2641\text{x}10^{-4}T - 3.3728\text{x}10^{-6}T^2$
$\mu = ½(.0117985\exp(-.022392T) + .0601634T^{-.952745})$

Dowfrost® HD (60% Glycol) - *liquid*
DFHD60
For $4 < T < 162$
$c_p = 3257 + 4.80134T - 1.9952\text{x}10^{-3}T^2$
$\rho = 1072.3 - .519239T - .00142626T^2$
$k = .30883 + 6.3162\text{x}10^{-4}T - 2.6656\text{x}10^{-6}T^2$
$\mu = ½(.0194357\exp(-.025182T) + .123987T^{-1.07719})$

Paratherm® NF - *liquid*
PARANF
For $10 < T < 316$
$c_p = 1810.9 + 3.5594T$
$\rho = 729.65 - 0.33504T$
$k = .13431 - 7.0226\text{x}10^{-5}T$
$\mu = 4.40572T^{-1.60423}$

Paratherm® HE - *liquid*
PARAHE
For $10 < T < 316$
$c_p = 1876.8 + 4.1316T$
$\rho = 746.1 - 0.63119T$
$k = 0.13455 - 7.4724\text{x}10^{-5}T$
$\mu = 795.75T^{-2.70403}$

CALFLO AF - *liquid*
CFAF
For $40 < T < 316$
$c_p = 1714.9 + 3.8688T$
$\rho = 862.89 - 0.63225T + 4.0234\text{X}10^{-4}T^2$
$k = .13807 - 5.4571\text{x}10^{-5}T + 1.7361\text{x}10^{-9}T^2$
$\rho = 39.7783T^{-1.99202}$

CALFLO FG - *liquid*
CFFG
For $40 < T < 288$
$c_p = 1803.8 + 3.3423T$
$\rho = 854.12 - 0.273305T - 7.5461\text{x}10^{-4}T^2$
$k = .13913 - 8.5789\text{x}10^{-5}T + 9.1377\text{x}10^{-8}T^2$
$\mu = 3.62695T^{-1.55491}$

CALFLO HTF - *liquid*
CFHTF
For $40 < T < 316$
$c_p = 1845.8 + 3.2776T$
$\rho = 871.0 - 0.58587T + 2.2249\text{x}10^{-4}T^2$
$k = .13907 - 5.4571\text{x}10^{-5} + 1.7361\text{x}10^{-9}T^2$
$\mu = 50.8759T^{-2.00572}$

Syltherm® 800 - *liquid*
SYLTH8
For $-73 < T < 260$
$c_p = 1574.5 + 1.7067T$
$\rho = 955.89 - 0.81738T - 4.8042\text{x}10^{-4}T^2$
$k = .10792 - 6.52567\text{x}10^{-5}T + 1.74916\text{x}10^{-9}T^2$

For $-73 < T < 10$

$\mu = .00220438 - 3.9223\text{x}10^{-5}T - 4.6422\text{x}10^{-7}T - 2.5722\text{x}10^{-8}T^3$

For $10 < T < 260$

$\mu = .00202134 - 3.07134\text{x}10^{-5}T + 2.34941\text{x}10^{-7}T^2 - 8.7194\text{x}10^{-10}T^3$
$+ 1.22576\text{x}10^{-12}T^4$

Thermalane® 550 - *liquid*
THM550

For $17 < T < 316$
$c_p = 1989.5 + 3.3681T$
$\rho = 852.91 - 0.67118T$
$k = .14486 - 7.4074\text{x}10^{-5}T$

For $-17 < T < 4.5$
$\mu = .184015\exp(-.0602736T)$

For $4.5 < T < 316$

$\mu = .174439 - 8.1538\text{x}10^{-3}T + 1.8709\text{x}10^{-4}T^2 - 2.5991\text{x}10^{-6}T^3 + 2.3307\text{x}10^{-8}T^4$
$- 1.37384\text{x}10^{-10}T^5 + 5.27456\text{x}10^{-13}T^6 - 1.2669\text{x}10^{-15}T^7$
$+ 1.72579\text{x}10^{-18}T^8 - 1.01648\text{x}10^{-21}T^9$

Thermalane® 600 - *liquid*
THM600

For $-17 < T < 327$
$c_p = 2094.7 + 3.1495T$
$\rho = 835.58 - 0.65347T$
$k = .15629 - 6.196\text{x}10^{-5}T$

For $-17 < T < 4.5$
$\mu = .182273\exp(-.0732021T)$

For $4.5 < T < 327$

$\mu = .170328 - 8.7649\text{x}10^{-3}T + 2.1606\text{x}10^{-4}T^2 - 3.1393\text{x}10^{-6}T^3 + 2.8835\text{x}10^{-8}T^4$
$- 1.716\text{x}10^{-10}T^5 + 6.5874\text{x}10^{-13}T^6 - 1.5717\text{x}10^{-15}T^7 + 2.1173\text{x}10^{-18}T^8$
$- 1.2293\text{x}10^{-21}T^9$

Thermalane® 800 - *liquid*
THM800

For $-17 < T < 371$

$c_p = 2184.5 + 3.2144T - 9.6370\text{x}10^{-4}T^2$

$\rho = 2184.5 + 3.2144^{T} - 9.6370\text{x}10^{-4}T^2$

$k = .1529 - 1.0318\text{x}10^{-4}T - 7.4183\text{x}10^{-9}T^2$

$\mu = .126335 - 6.0631\text{x}10^{-3}T + 1.1617\text{x}10^{-4}T^2 - 1.134\text{x}10^{-6}T^3 + 6.1506\text{x}10^{-9}T^4$
$\quad - 1.8752\text{x}10^{-11}T^5 + 3.00614\text{x}10^{-14}T^6 - 1.971\text{x}10^{-17}T^7$

Thermex® - *liquid*
THRMX

(Temperature limits unknown)

$c_p = 2.83089T + 1528.32$

$\rho = -.938571T + 1089.29$

$k = -1.3333\text{x}10^{-4}T + .14319$

$\mu = .27911T^{-1.23818}$

Air - *gas*
AIR

For $-173 < T < 2726$

$c_p = 1005.28 + 5.3251\text{x}10^{-3}T + 6.337\text{x}10^{-4}T^2 - 8.1579\text{x}10^{-7}T^3$
$\quad + 4.9426\text{x}10^{-10}T^4 - 1.5492\text{x}10^{-13}T^5 + 2.3265\text{x}10^{-17}T^6$

$k = 0.0254297 + 8.3037\text{x}10^{-5}T - 9.6375\text{x}10^{-8}T^2 + 1.286\text{x}10^{-10}T^3$
$\quad - 7.3051\text{x}10^{-14}T^4 + 1.5801\text{x}10^{-17}T^5$

$\mu = 1.71833\text{x}10^{-5} + 5.0923\text{x}10^{-8}T - 3.8357\text{x}10^{-11}T^2 + 2.8467\text{x}10^{-14}T^3$
$\quad - 9.9701\text{x}10^{-18}T^4 + 1.3181\text{x}10^{-21}T^5$

For $-173 < T < 26$

$\rho = 1.26941 - 4.886\text{x}10^{-3}T + 2.5977\text{x}10^{-5}T^2 + 1.8185\text{x}10^{-7}T^3 + 1.7867\text{x}10^{-9}T^4$

For $26 < T < 2225$

$\rho = 1.26214 - 4.1154\text{x}10^{-3}T + 1.011\text{x}10^{-5}T^2 - 1.6588\text{x}10^{-8}T^3 + 1.7341\text{x}10^{-11}T^4$
$\quad - 1.1284\text{x}10^{-14}T^5 + 4.3989\text{x}10^{-18}T^6 - 9.3648\text{x}10^{-22}T^7 + 8.3467\text{x}10^{-26}T^8$

For $2225 < T < 2726$

$\rho = 0.350535 - 1.313\text{x}10^{-4}T + 1.6274\text{x}10^{-8}T^2$

Carbon Dioxide [CO_2] - *gas*
CO2

For -53 < T < 1526
$c_p = 827.967 + 0.99954T - 8.4763\text{x}10^{-4}T^2 + 3.8487\text{x}10^{-7}T^3 - 7.1375\text{x}10^{-11}T^4$
$\rho = 1.9713 - 5.8952\text{x}10^{-3}T + 9.8244\text{x}10^{-6}T^2 - 7.5549\text{x}10^{-9}T^3 + 2.0916\text{x}10^{-12}T^4$
$k = 1.4448\text{x}10^{-2} + 8.3527\text{x}10^{-5}T - 1.3133\text{x}10^{-8}T^2$
$\mu = 1.4589\text{x}10^{-5} + 3.8088\text{x}10^{-8}T - 6.2015\text{x}10^{-12}T^2$

Hydrogen [H_2] - *gas*
H2

For -173 < T < 1726
$c_p = 14183.4 + 5.5081T - .037213T^2 + 1.2355\text{x}10^{-4}T^3 - 2.2182\text{x}10^{-7}T^4$
$\quad + 2.3445\text{x}10^{-10}T^5 - 1.4451\text{x}10^{-13}T^6 + 4.77946\text{x}10^{-17}T^7 - 6.51397\text{x}10^{-21}T^8$
$\rho = .087964 - 3.0046\text{x}10^{-4}T + 1.247\text{x}10^{-6}T^2 - 6.5073\text{x}10^{-9}T^3 + 2.4773\text{x}10^{-11}T^4$
$\quad - 5.7169\text{x}10^{-14}T^5 + 8.0088\text{x}10^{-17}T^6 - 6.8746\text{x}10^{-20}T^7 + 3.5342\text{x}10^{-23}T^8$
$\quad - 9.9886\text{x}10^{-27}T^9 + 1.19331\text{x}10^{-30}T^{10}$
$k = .167244 + 4.9794\text{x}10^{-4}T - 3.0927\text{x}10^{-7}T^2 + 2.6862\text{x}10^{-10}T^3$
$\quad - 6.87395\text{x}10^{-14}T^4$
$\mu = 8.26065\text{x}10^{-6} + 2.0427\text{x}10^{-8}T - 6.8867\text{x}10^{-12}T^2 + 1.7348\text{x}10^{-15}T^3$

Nitrogen [N_2] - *gas*
N2

For -173 < T < 1026
$c_p = 1039.9 - .033984T + 6.0155\text{x}10^{-4}T^2 - 5.3704\text{x}10^{-7}T^3 + 1.4479\text{x}10^{-10}T^4$
$\rho = 1.2826 - 4.5158\text{x}10^{-3}T + 1.0512\text{x}10^{-5}T^2 - 5.0297\text{x}10^{-8}T^3 + 4.2596\text{x}10^{-10}T^4$
$\quad - 1.8193\text{x}10^{-12}T^5 + 3.9745\text{x}10^{-15}T^6 - 4.6671\text{x}10^{-18}T^7 + 2.8145\text{x}10^{-21}T^8$
$\quad - 6.8517\text{x}10^{-25}T^9$
$k = .023643 + 7.4477\text{x}10^{-5}T - 3.9978\text{x}10^{-8}T^2 + 2.1437\text{x}10^{-11}T^3$
$\mu = 1.6277\text{x}10^{-5} + 4.7868\text{x}10^{-8}T - 2.9288\text{x}10^{-11}T^2 + 1.1255\text{x}10^{-14}T^3$

Oxygen [O_2] - *gas*
O2

For -173 < T < 1026
$c_p = 913.86 + .090997T + .0013194T^2 - 3.1674\text{x}10^{-6}T^3 + 3.00295\text{x}10^{-9}T^4$
$\quad - 1.03672\text{x}10^{-12}T^5$
$\rho = 1.4662 - 5.1602\text{x}10^{-3}T + 1.1947\text{x}10^{-5}T^2 - 5.7867\text{x}10^{-8}T^3 + 4.9429\text{x}10^{-10}T^4$
$\quad - 2.114\text{x}10^{-12}T^5 + 4.6193\text{x}10^{-15}T^6 - 5.4242\text{x}10^{-18}T^7 + 3.2709\text{x}10^{-21}T^8$
$\quad - 7.9618\text{x}10^{-25}T^9$
$k = .0235842 + 7.58894\text{x}10^{-5}T - 1.43399\text{x}10^{-8}T^2$
$\mu = 1.90145\text{x}10^{-5} + 5.8246\text{x}10^{-8}T - 4.16376\text{x}10^{-11}T^2 + 2.14079\text{x}10^{-14}T^3$

Ammonia [NH_3] - *gas*
NH3

For $26 < T < 306$
$c_p = 2134.76 + .538392T + .0063678T^2 - 9.9898\text{x}10^{-6}T^3$
$\rho = .73673 - .00209815T + 2.8379\text{x}10^{-6}T^2$
$\mu = 9.1404\text{x}10^{-6} + 3.80243\text{x}10^{-8}T - 9.0094\text{x}10^{-12}T^2$
$k = .022197 \quad 8.65576\text{x}10^{-5}T + 3.232\text{x}10^{-7}T^2 - 5.4597\text{x}10^{-10}T^3$

MS-20 Oil - *liquid*
MS20

(Temperature limits unknown)
$c_p = 3.09412T + 1980.44$
$\rho = -0.56681T + 903.679$
$k = 8.57848\text{x}10^{-8}T^2 - 1.19191\text{x}10^{-4}T + .136087$
$\mu = 4331.33T^{-2.69759}$

Transformer Oil (Standard 982-68) - *liquid*
TRAN

(Temperature limits unknown)
$c_p = -1.5597\text{x}10^{-9}T^6 + 3.6213\text{x}10^{-7}T^5 - 1.7345\text{x}10^{-5}T^4 - 1.4121\text{x}10^{-3}T^3$
$+ .12049T^2 + 3.04729T + 1616.47$
$\rho = -.62105T + 891.364$
$k = -8.19549\text{x}10^{-5}T + .112297$
$\mu = .0406485\exp(-.032348T)$

TM-1 (VTU-M3-11-62) - *liquid*
TM1

For $-50 < T < 100$
$c_p = 1.20508T + 1618.48$
$\rho = -0.63744T + 902.499$
$k = -9.53055\text{x}10^{-5}T + .120687$
$\mu = .382997T^{-1.08805}$

VM-4 Oil (Standard 7903-56) - *liquid*
VM4

For $-30 < T < 100$
$c_p = -2.3592\text{x}10^{-9}T^6 + 5.4339\text{x}10^{-7}T^5 - 3.7219\text{x}10^{-5}T^4 + 3.022\text{x}10^{-4}T^3$
$+ 0.048831T^2 +.23315T + 1457.74$
$\rho = -0.5915T + 915.492$
$k = -8.94004\text{x}10^{-5}T + .129112$
$\mu = 61.3841T^{-1.85956}$

KhF22 Oil (Standard 5546-66) - *liquid*
KHF22

(Temperature limits unknown)

$c_p = -1.58271\times10^{-6}T^4 + 6.52884\times10^{-5}T^3 + 0.012453T^2 + .241957T + 1638.96$

$\rho = -0.72984T + 1009.52$

$k = -1.48464\times10^{-4}T + .157694$

$\mu = 2.57617T^{-1.30322}$

Engine Oil (Unused) - *liquid*
ENG

For $0 < T < 156$

$c_p = 1798.42 + 4.06543T + .001455T^2$

$\rho = 899.55 - .5926T$

$k = .147 - 7.18122\times10^{-5}T - 7.0708\times10^{-7}T^2 + 3.6242\times10^{-9}T^3$

$\mu = 3.8022 - .31428T + .013304T^2 - 3.5893T^3 + 6.4967\times10^{-6}T^4 - 7.9143\times10^{-8}T^5 + 6.3585\times10^{-10}T^6 - 3.2142\times10^{-12}T^7 + 9.2318\times10^{-15}T^8 - 1.146\times10^{-17}T^9$

Ethylene Glycol [$C_2H_4(OH)_2$] - *liquid*
ETHYLG

For $0 < T < 100$

$c_p = 2293.78 + 4.4801T$

$\rho = 1129.76 - .23264T - .026345T^2 + 4.5083\times10^{-4}T^3 - 2.3342\times10^{-6}T^4$

$k = .24194 + 3.205\times10^{-4}T + 2.4239\times10^{-6}T^2 - 4.34\times10^{-9}T^3 - 1.1914\times10^{-9}T^4 - 8.8208\times10^{-12}T^5$

$\mu = .06442 - .0041299T + 1.4896\times10^{-4}T^2 - 3.2665\times10^{-6}T^3 + 4.1721\times10^{-8}T^4 - 2.8171\times10^{-10}T^5 + 7.4264\times10^{-13}T^6$

Glycerin [$C_3H_5(OH)_3$] - *liquid*
GLYCRN

For $0 < T < 46$

$c_p = 2260.92 + 5.8326T + .01288T^2$

$\rho = 1275.52 - .5666T - 7.2682\times10^{-4}T^2$

$k = .282065 + 3.4417\times10^{-4}T - 1.0215\times10^{-5}T^2 + 1.0936\times10^{-7}T^3$

$\mu = 10.457 - .98243T + .03866T^2 - 7.1174\times10^{-4}T^3 + 5.0052\times10^{-6}T^4$

Mercury (Hg) - *liquid*
HG

For $0 < T < 326$
$c_p = 1403.48 - .3996T + 7.7106\text{x}10^{-4}T^2$
$\rho = 13592.9 - 2.4009T$
$k = 8.1777 + .013656T - 6.49055\text{x}10^{-6}T^2$
$\mu = .0016815 - 6.15448\text{x}10^{-6}T + 1.96029\text{x}10^{-8}T^2 - 2.4526\text{x}10^{-11}T^3$

n-Pentane - *liquid*
PENT

For $0 < T < 200$
c_p = *Determined by subroutine (Riemer, 1976b)*
ρ = *Determined by subroutine (Riemer, 1976b)*
$k = .120500001 - 2.19166518\text{x}10^{-4}T - 1.4583602\text{x}10^{-6}T^2 + 4.1667695\text{x}10^{-8}T^3$
$- 4.16677821\text{x}10^{-10}T^4$
$\mu = 2.8306484\text{x}10^{-4} - 4.7037877\text{x}10^{-6}T + 3.9457157\text{x}10^{-7}T^2$
$- 2.20408288\text{x}10^{-8}T^3 + 5.8042066\text{x}10^{-10}T^4 - 7.813455978\text{x}10^{-12}T^5$
$+ 5.23570269\text{x}10^{-14}T^6 - 1.38655465\text{x}10^{-16}T^7$

n-Pentane - *vapor*
PENT

For $0 < T < 200$
c_p = *Determined by subroutine (Riemer, 1976b)*
ρ = *Determined by subroutine (Riemer, 1976b)*
$k = .0117000943 + .0974954646\text{x}10^{-3}T + 1.2506797\text{x}10^{-7}T^2$
$- 3.20993285\text{x}10^{-10}T^3$
$\mu = 49.3702925\text{x}10^{-7} + .938695079\text{x}10^{-7}T - .0151413279\text{x}10^{-7}T^2$
$+ 1.380067486\text{x}10^{-11}T^3 - 4.01657705\text{x}10^{-14}T^4$

Butane - *liquid*
BUT

(Temperature limits unknown)
c_p = *Determined by subroutine (Riemer, 1976b)*
ρ = *Determined by subroutine (Riemer, 1976b)*
$k = -7.81255\text{x}10^{-10}T^4 + 2.60419\text{x}10^{-7}T^3 - 3.21878\text{x}10^{-5}T^2$
$+ .00149585T + .0749995$
$\mu = -8.41033\text{x}10^{-14}T^5 + 3.28754\text{x}10^{-11}T^4 - 5.00885\text{x}10^{-9}T^3$
$+ 3.7359\text{x}10^{-7}T^2 - 1.47048T + .000376398$

Butane - ***vapor***
BUT

(Temperature limits unknown)
c_p = *Determined by subroutine (Riemer, 1976b)*
ρ = *Determined by subroutine (Riemer, 1976b)*
$k = -4.3704\times10^{-11}T^3 + 1.01398\times10^{-7}T^2 + .000108024T + .01310114$
$\mu = -7.04548\times10^{-18}T^5 + 6.15458\times10^{-15}T^4 - 2.03319\times10^{-12}T^3$
$+ 2.99428\times10^{-10}T^2 + 5.54933\times10^{-9}T + 7.46943\times10^{-6}T$

Isopentane - ***liquid***
ISOP

(Temperature limits unknown)
c_p = *Determined by subroutine (Riemer, 1976b)*
ρ = *Determined by subroutine (Riemer, 1976b)*
$k = -8.88886\times10^{-8}T^3 + 4.63094\times10^{-6}T^2 - .000298849T + .11906$
$\mu = -1.66661\times10^{-10}T^3 + 1.99997\times10^{-8}T^2 - 2.78333\times10^{-6}T + .000272$

Isopentane - ***vapor***
ISOP

(Temperature limits unknown)
c_p = *Determined by subroutine (Riemer, 1976b)*
ρ = *Determined by subroutine (Riemer, 1976b)*
$k = .014023604 + .1208272\times10^{-3}T + 1.08600139\times10^{-7}T^2 - 1.9940388\times10^{-8}T^3$
$+ 4.904895\times10^{-10}T^4 - 4.92943376\times10^{-12}T^5 + 2.2305489\times10^{-14}T^6$
$- 3.7421512\times10^{-17}T^7$
$\mu = -2.09459\times10^{-12}T^2 + 2.224021\times10^{-8}T + 6.38969\times10^{-6}$

Isobutane - ***liquid***
ISOB

(Temperature limits unknown)
c_p = *Determined by subroutine (Riemer, 1976b)*
ρ = *Determined by subroutine (Riemer, 1976b)*
$k = .10545730 - .5897775\times10^{-3}T + .27657500\times10^{-5}T^2$
$\mu = 1.55136\times10^{-10}T^3 - 6.16567\times10^{-9}T^2 - 2.09543\times10^{-6}T + .000214$

Isobutane - *vapor*
ISOB

(Temperature limits unknown)
c_p = *Determined by subroutine (Riemer, 1976b)*
ρ = *Determined by subroutine (Riemer, 1976b)*
$k = .013813870 + .1088360\text{x}10^{-3}T + .3069733333\text{x}10^{-06}T^2$
$-.334880\text{x}10^{-08}T^3 + .1116266667\text{x}10^{-10}T^4$
$\mu = -4.35675\text{x}10^{-14}T^4 + 1.22251\text{x}10^{-11}T^3 - 1.22875\text{x}10^{-9}T^2$
$+ 7.04506\text{x}10^{-8}T + 6.17173\text{x}10^{-6}$

Heptane - *liquid*
HEP

For $-90 < T < 20$
c_p = *Determined by subroutine (Riemer, 1976b)*
ρ = *Determined by subroutine (Riemer, 1976b)*
$k = .13420 - 2.7167\text{x}10^{-4}T + 4.8701\text{x}10^{-8}T^2$

For $-90 < T < 20$
$\mu = 5.2757\text{x}10^{-4} - 7.8221\text{x}10^{-6}T + 5.2534\text{x}10^{-8}T^2 + 1.9301\text{x}10^{-9}T^3$
$+ 5.5358\text{x}10^{-11}T^4$

Heptane - *vapor*
HEP

For $-90 < T < 400$
c_p = *Determined by subroutine (Riemer, 1976b)*
ρ = *Determined by subroutine (Riemer, 1976b)*
$k = 8.8553\text{x}10^{-3} + 9.2249\text{x}10^{-5}T + 7.1772\text{x}10^{-8}T^2$
$\mu = 5.3255\text{x}10^{-6} + 2.0043\text{x}10^{-8}T + 2.6577\text{x}10^{-13}T^2$

Methane - *liquid*
METH

For $-173 < T < 83$
c_p = *Determined by subroutine (Riemer, 1976b)*
ρ = *Determined by subroutine (Riemer, 1976b)*
$k = .23242 + 4.6197\text{x}10^{-3}T + 4.2448\text{x}10^{-5}T^2 + 9.6463\text{x}10^{-8}T^3$
$\mu = 6.7496\text{x}10^{-5} + 1.4941\text{x}10^{-6}T + 1.1267\text{x}10^{-8}T^2$

Methane - *vapor*
METH

For $-173 < T < 83$
c_p = *Determined by subroutine (Riemer, 1976b)*
ρ = *Determined by subroutine (Riemer, 1976b)*
$k = .030875 + 1.3364\text{x}10^{-4}T + 9.8656\text{x}10^{-8}T^2$
$\mu = 1.025\text{x}10^{-5} + 3.2756\text{x}10^{-8}T - 1.8745\text{x}10^{-11}T^2$

Ethylene - *liquid*
ETHYL

For $-165 < T < 225$
c_p = *Determined by subroutine (Riemer, 1976b)*
ρ = *Determined by subroutine (Riemer, 1976b)*
$k = 8.1027\text{x}10^{-2} - 1.2232\text{x}10^{-3}T - 1.2078\text{x}10^{-6}T^2$
$\mu = -2.3585\text{x}10^{-3} - 6.3783\text{x}10^{-5}T - 5.527\text{x}10^{-7}T^2 - 1.6643\text{x}10^{-9}T^3$

Ethylene - *vapor*
ETHYL

For $-165 < T < 225$
c_p = *Determined by subroutine (Riemer, 1976b)*
ρ = *Determined by subroutine (Riemer, 1976b)*
$k = 1.6976\text{x}10^{-2} + 1.2134\text{x}10^{-4}T + 1.2177\text{x}10^{-7}T^2$
$\mu = 9.4809\text{x}10^{-6} + 3.2084\text{x}10^{-8}T - 1.1608\text{x}10^{-11}T^2$

Freon® R-11 Refrigerant - *liquid*
F11

For $0 < T < 150$
c_p = *Determined by subroutine (Riemer, 1976c)*
ρ = *Determined by subroutine (Riemer, 1976c)*
$k = 9.5768\text{x}10^{-2} - 2.4018\text{x}10^{-4}T + 4.7302\text{x}10^{-8}T^2$
$\mu = 5.6689\text{x}10^{-4} - 6.5163\text{x}10^{-6}T + 3.436\text{x}10^{-8}T^2 - 7.393\text{x}10^{-11}T^3$

Freon® R-11 Refrigerant - *vapor*
F11

For $0 < T < 150$
c_p = *Determined by subroutine (Riemer, 1976c)*
ρ = *Determined by subroutine (Riemer, 1976c)*
$k = 8.1527\text{x}10^{-3} + 2.3475\text{x}10^{-5}T + 1.8628\text{x}10^{-7}T^2$
$\mu = 1.0179\text{x}10^{-5} + 3.0692\text{x}10^{-8}T + 9.3987\text{x}10^{-11}T^2$

Freon® R-13 Refrigerant - *liquid*
F13

For $0 < T < 100$
c_p = *Determined by subroutine (Riemer, 1976c)*
ρ = *Determined by subroutine (Riemer, 1976c)*
$k = 5.3061\text{x}10^{-2} - 2.9112\text{x}10^{-4}T + 1.7371\text{x}10^{-6}T^2$
$\mu = 1.0132\text{x}10^{-4} - 3.6831\text{x}10^{-7}T + 2.8835\text{x}10^{-8}T^2$

Freon® R-13 Refrigerant - *vapor*
F13

For $0 < T < 100$
c_p = *Determined by subroutine (Riemer, 1976c)*
ρ = *Determined by subroutine (Riemer, 1976c)*
$k = 1.3605\text{x}10^{-2} + 1.5381\text{x}10^{-4}T + 7.7252\text{x}10^{-7}T^2$
$\mu = 1.4922\text{x}10^{-5} + 1.163\text{x}10^{-7}T + 1.2104\text{x}10^{-9}T^2 + 8.2369\text{x}10^{-12}T^3$

Freon® R-22 Refrigerant - *liquid*
F22

For $-20 < T < 60$
c_p = *Determined by subroutine (Riemer, 1976c)*
ρ = *Determined by subroutine (Riemer, 1976c)*
$k = 9.6174\text{x}10^{-2} - 4.2368\text{x}10^{-4}T + 3.4954\text{x}10^{-7}T^2$
$\mu = 2.1019\text{x}10^{-4} - 2.2755\text{x}10^{-6}T + 1.0823\text{x}10^{-8}T^2$

Freon® R-22 Refrigerant - *vapor*
F22

For $-20 < T < 60$
c_p = *Determined by subroutine (Riemer, 1976c)*
ρ = *Determined by subroutine (Riemer, 1976c)*
$k = 9.5013\text{x}10^{-3} + 6.0705\text{x}10^{-5}T - 1.8442\text{x}10^{-8}T^2$
$\mu = 1.18\text{x}10^{-5} + 4.0\text{x}10^{-8}T$

Freon® R-23 Refrigerant - *liquid*
F23

For $-40 < T < 10$
c_p = *Determined by subroutine (Riemer, 1976c)*
ρ = *Determined by subroutine (Riemer, 1976c)*
$k = 7.4817\text{x}10^{-2} - 7.547\text{x}10^{-4}T - 1.4837\text{x}10^{-6}T^2$
$\mu = 9.0309\text{x}10^{-5} - 1.6701\text{x}10^{-6}T + 4.2279\text{x}10^{-9}T^2$

Freon® R-23 Refrigerant - *vapor*
F23

For $-40 < T < 10$
c_p = *Determined by subroutine (Riemer, 1976c)*
ρ = *Determined by subroutine (Riemer, 1976c)*
$k = 1.7685\text{x}10^{-2} + 2.1117\text{x}10^{-4}T + 1.2942\text{x}10^{-6}T^2$
$\mu = 1.5063\text{x}10^{-5} + 1.3885\text{x}10^{-7}T + 1.657\text{x}10^{-9}T^2$

Freon® R-113 Refrigerant - *liquid*
F113

For $0 < T < 170$
c_p = *Determined by subroutine (Riemer, 1976c)*
ρ = *Determined by subroutine (Riemer, 1976c)*
$k = 8.1947\text{x}10^{-2} - 1.7689\text{x}10^{-4}T + 1.3415\text{x}10^{-8}T^2$
$\mu = 9.4573\text{x}10^{-4} - 1.1889\text{x}10^{-5}T + 7.0746\text{x}10^{-8}T^2 - 1.6326\text{x}10^{-10}T^3$

Freon® R-113 Refrigerant - *vapor*
F113

For $0 < T < 170$
c_p = *Determined by subroutine (Riemer, 1976c)*
ρ = *Determined by subroutine (Riemer, 1976c)*
$k = 7.7373\text{x}10^{-3} + 4.3806\text{x}10^{-5}T + 5.8518\text{x}10^{-8}T^2$
$\mu = 8.1603\text{x}10^{-6} + 8.025\text{x}10^{-8}T - 6.2726\text{x}10^{-10}T^2 + 2.3745\text{x}10^{-12}T^3$

Freon® R-114 Refrigerant - *liquid*
F114

For $-40 < T < 115$
c_p = *Determined by subroutine (Riemer, 1976c)*
ρ = *Determined by subroutine (Riemer, 1976c)*
$k = 7.0594\text{x}10^{-2} - 2.389\text{x}10^{-4}T + 4.0843\text{x}10^{-9}T^2$
$\mu = 4.9155\text{x}10^{-4} - 6.5554\text{x}10^{-6}T + 4.3739\text{x}10^{-8}T^2 - 1.2737\text{x}10^{-10}T^3$

Freon® R-114 Refrigerant - *vapor*
F114

For $-40 < T < 115$
c_p = *Determined by subroutine (Riemer, 1976c)*
ρ = *Determined by subroutine (Riemer, 1976c)*
$k = 8.1576\text{x}10^{-5}T + .00882409$
$\mu = 1.0578\text{x}10^{-5} + 3.2039\text{x}10^{-8}T + 1.2712\text{x}10^{-10}T^2$

APPENDIX V

ONE-DIMENSIONAL SIMULATION: PROGRAM USAGE

The program consists of the main execution code containing all required subroutines, one input file, and seven output files. The files are named

dcx.f - FORTRAN code.
dxin.dat - Input file.
dxout.dat - Output file containing summary of DCHX operation.
dxoutd.dat - Output file containing dispersed fluid profile
dxoutd1.dat - Output file (Additional dispersed fluid parameters)
dxoutd2.dat - Output file (Additional dispersed fluid parameters)
dxoutc.dat - Output file containing continuous fluid profile.
dxoutc1.dat - Output file (Additional continuous fluid parameters)
dxoutfh.dat - Output file containing pertinent DCHX fluid characteristics parameters

The input file is required for execution. Output files are created by the program and overwritten with each consecutive use. The input file must contain the parameters explained in the paragraphs below. All parameters are entered and reported in S.I. units.

Input File - [dxin.dat]

The input file is where the user specifies system size, fluid parameters, and start-up conditions. The data contained in the file must be entered one item per line in the following order.

Line #1 - Coded name of the dispersed fluid chosen. This code is found in Table V-1. The code must be in capital letters and occupy the first six spaces of the line. Codes with less than six letters/numbers must have fill spaces following them to account for a total of six spaces in the field.

Line #2 - Dispersed fluid mass flow rate (kg/sec). Generally this is a decimal quantity.

Line #3 - Dispersed fluid inlet temperature (°C)

Line #4 - Dispersed fluid initial droplet/bubble radius (meters). Generally this is a small value such as 1 or 2 mm which would be entered as 0.001 or 0.002.

Line #5 - Coded name of the continuous fluid chosen. This code is found in Table V-1. The code must be in capital letters and occupy the first six spaces of the line. Codes with less than six letters/numbers must have fill spaces following them to account for a total of six spaces in the field.

Line #6 - Continuous fluid mass flow rate (kg/sec). Generally this is a decimal quantity.

Line #7 - Continuous fluid inlet temperature (°C)

Line #8 - Continuous fluid quality (dimensionless value of either 0.0 or 1.0). A value of 0.0 indicates a liquid, a value of 1.0 indicates a vapor. The program does not permit the continuous fluid to change state and will terminate if this parameter changes within the program.

Line #9 - DCHX column radius (meters)

Line #10 - DCHX column height (meters)

Line #11 - DCHX operating pressure (kPa). This is the nominal operating pressure at the top of the column. The pressure at the bottom and throughout the length of the column will be adjusted for hydrostatic effects.

Line #12 - Dispersed fluid nozzle inlet area (meters). It is assumed that the dispersed fluid is being introduced through a spray nozzle. This quantity represents the sum of the areas of the inlet holes of the nozzle device. The input value will be evaluated and change this parameter to match the resulting dispersed fluid velocity near the nozzle. The new value will be listed in the output.

Line #13 - Number of individual control volumes the column is divided into for the Runge-Kutta solution. Control volume is a disc shaped volume equal to the cross sectional area of the column times the height of the increment. The maximum number is 2000. The minimum must be a reasonable amount (generally 100-200 minimum) to achieve accurate results. The number of control volumes bears directly on the run time. Values less than 300 run reasonably fast. Values above 500 are substantially more expensive if multiple runs are planned. *(Control volumes should be at least as large as the largest anticipated droplet/bubble size.)*

Line #14

First entry - Fluid enthalpy tolerance (J/Kg°C). Tolerance to which program iterates before a solution is reported. This parameter applied to the fluid which is the target of the Runge-Kutta shooting method (i.e. heavy fluid introduced at top of DCHX column). Program calculates enthalpy for the inlet conditions and then references each shooting method iteration. Once solution values fall within the tolerance, the program completes final portions and prints out a solution profile. Generally a value of 0.1 or 0.01 should be used. Smaller values will increase required CPU time.

Second entry - Fluid temperature min/max difference tolerance (°C). Tolerance of bracketed values of shooting method guesses. If subsequent guesses of the starting temperature of the target fluid changes by less than this parameter then program declares a solution is reached. Generally a value of 0.1 or 0.01 should be used. Smaller values will increase required CPU time.

Line #15

First Entry - Search acceleration index. This parameter governs the way the program searches for temperature guesses in the Runge-Kutta methodology. The values may be 0,1,2, or 3. Generally a value of 0 is used which is sets the search method to a bisection of the temperature interval. For certain combinations of fluids, a coarse initial guess can erroneously eliminate large portions of the interval and prohibit precise solutions. An indication that this is happening occurs in the temperature profiles of one of the fluids. The iterated temperature for the known inlet condition differs from the temperature given in the input file by some value. To correct this occurrence the index is incremented by 1 and the run retried. This is repeated until the inlet temperature from the iterated solution once again agrees with the input value. The maximum value of this index is 3 which represents a slower *shooting guess* sequence. A setting of 3 can double the CPU time required for a run.

Second Entry - Column heat loss coefficient. A value of 1.0 indicates no loss of column heat to the outside environment. A value of 1.1 (for example) indicates a 10% loss of heat from the continuous fluid to the outside environment. A value of 1.15 would mean a 15% loss, etc.

Program Compiling

The code is written in FORTRAN 77. All subroutines needed to run the program are written into the code. No outside library routines are required. Compiling of the program can be done per normal procedures.

Output Files

Results of the DCHX design code are contained in seven output files - dxout.dat, dxoutd.dat, dxoutd1.dat, dxoutd2.dat, dxoutc.dat, dxoutc1.dat, and dxoutfh.dat. The primary output file is "dxout.dat". This file contains a record of the key input parameters from the dxin.dat input file and resulting DCHX general performance . This is also where any error messages or limit violations are recorded. This should be the first file reviewed at the completion of a simulation run. An example of the output is shown in the appendix. In all cases, the total output of the DCHX is given. In addition, some analysis is performed. For cases where the dispersed fluid undergoes a phase change along the length of the column, the Log Mean Temperature Difference and volumetric heat transfer coefficient are calculated and recorded. For cases where the dispersed fluid stays as either a liquid or vapor, the effectiveness is given in accordance with the effectiveness-NTU method of calculation. Additional qualifying information is given about the direction of the heat transfer and the numerical iterations required to run the input specifications.

The remaining six output files contain profile data referenced to the control volume location along the length of the tested DCHX column. Output files containing the letter "d", "d1", and "d2" refer to the dispersed fluid results, "c" and "c1" is for the continuous fluid results. An additional file, "fh" hold generalized data concerning the fluid dynamic parameters and heat transfer coefficients. Reynolds number, Peclect number and drag coefficient are examples of fluid dynamic parameters of interest.

All output is in S.I. units with these units specified at the top of each column where applicable. Examples of the files are contained in this appendix.

Error Treatment

It is likely that during the course of implementing and testing combinations of parameters, the code will fail and return error messages that state mathematical errors have occurred. It is difficult to anticipate all combinations and extremes of the parameters and build in "fixes" or advisory messages. Therefore, it is important that the user be at least somewhat skilled in the engineering aspects of direct contact heat exchangers to understand where the potential source of error lies. Below are guidelines in to aid in trouble shooting an error occurrence.

1. Continuous or dispersed fluid is at a temperature outside of the range of

the data that went into constructing the curve fits of the thermodynamic and thermophysical properties. The user must adjust parameters such that the fluid operates at a higher or lower temperature.

2. The fluids selected are not buoyant enough to each other.

3. The dispersed fluid nozzle area is too small or too large for the flow rate specified. This causes entry of the fluid to act as either a jet or ooze instead of normal buoyant flow. Adjust this value up by a factor of 2 to 10 as a first attempt at overcoming a mathematical error.

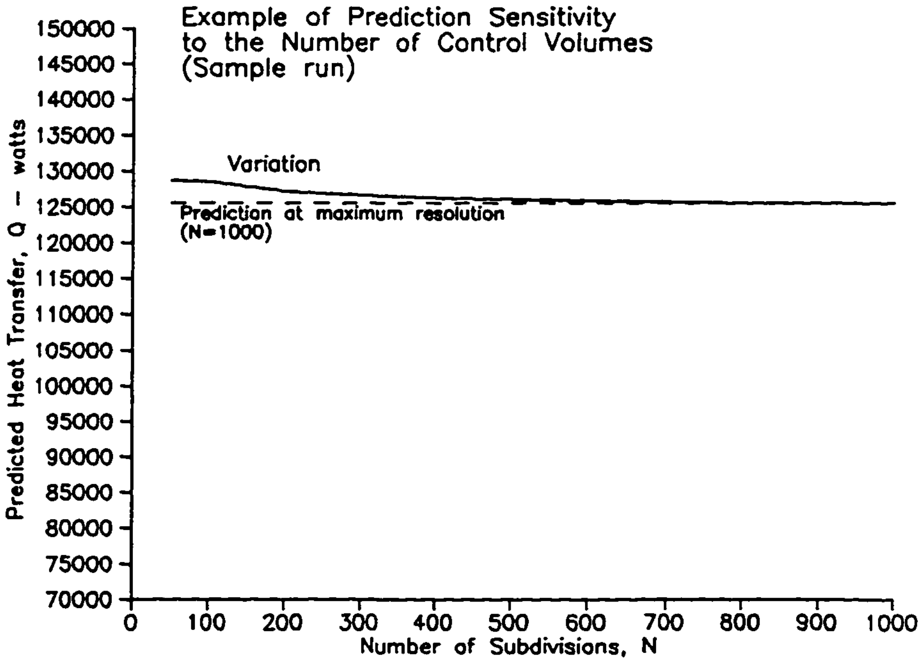

Figure V-1 Numerical Sensitivity to Control Volume Resolution.

Table V-1 Fluids Listing

Fluid/Gas	Code used to specify fluid in input file	Temperature Range °C	Notes
1. Water	H20	0 < T < 700	
Hydrocarbons			
2. Butane	BUT	Limits unknown	
3. Ethylene	ETHYL	-165 < T < 225	
4. Heptane	HEP	-90 < T < 400	
5. Isobutane	ISOB	Limits unknown	
6. Isopentane	ISOP	Limits unknown	
7. Methane	METH	-173 < T < 83	
8. Pentane	PENT	0 < T < 200	
Refrigerants			
9. R-11	F11	0 < T < 150	
10. R-13	F13	0 < T < 100	
11. R-22	F22	-20 < T < 60	
12. R-23	F23	-40 < T < 10	
13. R-113	F113	0 < T < 170	
14. R-114	F114	-40 < T < 115	
Heat Transfer Fluids			
15. Therminol® 44	TH44	-45 < T < 260	
16. Therminol® 55	TH55	0 < T < 343	
17. Therminol® 60	TH60	-59 < T < 343	
18. Therminol® 66	TH66	-18 < T < 371	
19. Therminol® 88	TH88	149 < T < 482	
20. Therminol® VP-1	MTHVP1	15 < T < 404	
21. DOWTHERM SR-1	DWTHSR1	0 < T < 125	
22. DOWTHERM 4000	DT4000	4 < T < 177	
23. DOWTHERM 4000, 30% Glycol	DT4030	4 < T < 176	
24. DOWTHERM 4000, 40% Glycol	DT4040	4 < T < 176	
25. DOWTHERM 4000, 50% Glycol	DT4050	4 < T < 176	
26. DOWTHERM 4000, 60% Glycol	DT4060	4 < T < 176	
27. DOWFROST	DWFRST	4 < T < 121	
28. DOWFROST HD	DWFHD	4 < T < 177	
29. DOWFROST HD, 30% Glycol	DFHD30	4 < T < 162	

30. DOWFROST HD, 40% Glycol	DFHD40	4 < T < 162	
31. DOWFROST HD, 50% Glycol	DFHD50	4 < T < 162	
32. DOWFROST HD, 60% Glycol	DFHD60	4 < T < 162	
33. THERMALANE® 550	THM550	-17 < T < 316	
34. THERMALANE® 600	THM600	-17 < T < 327	
35. THERMALANE® 800	THM800	-17 < T < 371	
36. Paratherm NF	PARANF	10 < T < 316	
37. Paratherm HE	PARAHE	10 < T < 316	
38. CALFLO AF	CFAF	40 < T < 316	
39. CALFLO FG	CFFG	40 < T < 288	
40. CALFLO HTF	CFHTF	40 < T < 316	
41. SYLTHERM® 800	SYLTH8	-73 < T < 260	
42. Thermex®	THRMX		
Miscellaneous Oils			
43. Engine Oil (Unused)	ENG	0 < T < 156	
44. KhF22 Oil (Standard 5546-66)	KHF22	Limits unknown	
45. MS-20 Oil	MS20	Limits unknown	
46. Transformer Oil (Standard 982-68)	TRAN	Limits unknown	
47. VM-4 Oil (Standard 7903-56)	VM4	-30 < T < 100	
48. TM-1 (VTU-M3-11-62)	TM1	-50 < T < 100	
Gases			
49. Air	AIR	-173 < T < 2726	
50. Ammonia (NH_3)	NH3	26 < T < 306	
51. Carbon Dioxide (CO_2)	CO2	-53 < T < 1526	
52. Hydrogen (H_2)	H2	-173 < T < 1726	
53. Nitrogen (N_2)	N2	-173 < T < 1026	
54. Oxygen (O_2)	O2	-173 < T < 1026	
Miscellaneous Fluids			
55. Ethylene Glycol [$C_2H_4(OH)_2$]	ETHYLG	0 < T < 100	
56. Glycerin [$C_3H_5(OH)_3$]	GLYCRN	0 < T < 46	
57. Mercury (Hg)	HG	0 < T < 326	

Sample Input File - dxin.dat

```
PENT            DISPERSED FLUID
0.21              Dispersed flow rate (kg/sec)
41.0              Dispersed inlet temperature (°C)
0.002             Initial dispersed droplet radius (m)
H2O             CONTINUOUS FLUID
1.91              Continuous flow rate (kg/sec)
85.0              Continuous inlet temperature (°C)
0.0               Continuous fluid quality
.305            DCHX radius (m)
3.0             DCHX length (m)
260.0           DCHX pressure (kPa)
.1              DCHX dispersed fluid nozzle inlet area (m²)
150             Numerical resolution
0.01 0.01       Enthalpy and Temperature check tolerance
0 1.1 1         Search accel. index, Heat loss coef., Progress
flag
```

```
 DATAFILE:  dxout.dat

 Input Values

     Heat Exchanger Length:     3.00 meters
     Heat Exchanger Radius:     0.50 meters
Nominal Operating Pressure:   260.00 kPa

                    Dispersed Fluid    Continuous Fluid
         Fluid:        PENT               H2O
Mass Flow Rate:        0.1200             1.9100   kg/sec
   Inlet Temp:        41.00              85.00     deg C

Disp. Fld. Droplet/Bubble Radius:  .002000 meters
        Continuous Fluid Quality:  0.

 DCHX Parameter Profile


  Fluid Flow Configuration:  1
 Temperature Configuration:  1
Total Device Heat Trans  D:     39125. Watts
Total Device Heat Trans  C:    -47341. Watts

Dispersed Fluid Temp @ z=0:    41.0 deg C
Dispersed Fluid Temp @ z=N:    73.0 deg C
Continuous Fluid Temp @z=0:    77.6 deg C
Continuous Fluid Temp @z=N:    85.0 deg C

  Log Mean Temp Difference:    22.09   deg C
               Parallelism:    0.0000
  Optimal Fluid Usage Ind.:    0.0000
 Volumetric Trans Coeff  D:      751.8285 W/m3 K
 Volumetric Trans Coeff  C:      909.7124 W/m3 K

              Prandtl # (D):  **********
              Prandtl # (C):     2.1524

Adjusted Dispersed Fluid Nozzle Area:  0.001


 HEAT TRANSFER IS FROM CONTINUOUS FLUID TO DISPERSED FLUID


CONVERGENCE OCCURRED IN  20 ITERATIONS
```

```
DATAFILE:  dxoutd.dat
DISPERSED FLUID RESULTS
DCHX Configuration = 1
Dispersed Fluid:  PENT
Idx    Loo  Vel(a) Vel(r)   Temp Holdup HtXfer   Qual  D- rad     Pop
 i      Z    Udz    Ud       Td    Phi   QcvD     Xd     r        nB
        m    m/s    m/s      C            W       -      m
150  3.000  1.717  1.719    73.0 0.014      8. 1.0000 .00917      69.
149  2.980  1.716  1.718    72.9 0.014      9. 1.0000 .00917      69.
148  2.960  1.715  1.718    72.9 0.014      9. 1.0000 .00916      69.
147  2.940  1.714  1.717    72.9 0.014      9. 1.0000 .00916      69.
146  2.920  1.713  1.716    72.8 0.014      9. 1.0000 .00916      69.
145  2.900  1.713  1.715    72.8 0.014      9. 1.0000 .00915      69.
144  2.880  1.712  1.714    72.8 0.014      9. 1.0000 .00915      70.
143  2.860  1.711  1.713    72.7 0.014      9. 1.0000 .00915      70.
142  2.840  1.710  1.713    72.7 0.014      9. 1.0000 .00914      70.
141  2.820  1.709  1.712    72.7 0.014      9. 1.0000 .00914      70.
140  2.800  1.709  1.711    72.6 0.014      9. 1.0000 .00914      70.
139  2.780  1.708  1.710    72.6 0.014      9. 1.0000 .00913      70.
138  2.760  1.707  1.709    72.5 0.014      9. 1.0000 .00913      70.
137  2.740  1.706  1.709    72.5 0.014      9. 1.0000 .00913      70.
136  2.720  1.705  1.708    72.5 0.014      9. 1.0000 .00913      70.
135  2.700  1.705  1.707    72.4 0.014      9. 1.0000 .00912      70.
134  2.680  1.704  1.706    72.4 0.014      9. 1.0000 .00912      70.
133  2.660  1.703  1.705    72.4 0.014      9. 1.0000 .00912      70.
132  2.640  1.702  1.705    72.3 0.014      9. 1.0000 .00911      70.
131  2.620  1.701  1.704    72.3 0.014      9. 1.0000 .00911      70.
130  2.600  1.701  1.703    72.3 0.014      9. 1.0000 .00911      70.

 .     .      .      .       .     .        .     .      .         .
 .     .      .      .       .     .        .     .      .         .
 .     .      .      .       .     .        .     .      .         .
 .     .      .      .       .     .        .     .      .         .
 .     .      .      .       .     .        .     .      .         .
 .     .      .      .       .     .        .     .      .         .
 .     .      .      .       .     .        .     .      .         .
 .     .      .      .       .     .        .     .      .         .

 23  0.460  0.137  0.140    64.9 0.002     55. 0.0000 .00203     867.
 22  0.440  0.137  0.140    64.7 0.002     56. 0.0000 .00203     867.
 21  0.420  0.137  0.140    64.5 0.002     57. 0.0000 .00203     868.
 20  0.400  0.137  0.139    64.3 0.002     57. 0.0000 .00203     869.
 19  0.380  0.137  0.139    64.1 0.002     58. 0.0000 .00203     870.
 18  0.360  0.137  0.139    63.9 0.002     59. 0.0000 .00203     870.
 17  0.340  0.137  0.139    63.6 0.002     60. 0.0000 .00203     871.
 16  0.320  0.136  0.139    63.4 0.002     61. 0.0000 .00203     872.
 15  0.300  0.136  0.139    63.2 0.002     62. 0.0000 .00203     873.
 14  0.280  0.136  0.139    62.9 0.002     63. 0.0000 .00202     874.
 13  0.260  0.136  0.139    62.7 0.002     64. 0.0000 .00202     875.
 12  0.240  0.136  0.138    62.5 0.002     65. 0.0000 .00202     876.
 11  0.220  0.136  0.138    62.2 0.002     66. 0.0000 .00202     877.
 10  0.200  0.136  0.138    62.0 0.002     67. 0.0000 .00202     877.
  9  0.180  0.135  0.138    61.7 0.002     68. 0.0000 .00202     878.
  8  0.160  0.135  0.138    61.5 0.002     69. 0.0000 .00202     879.
  7  0.140  0.135  0.138    61.2 0.002     71. 0.0000 .00202     880.
  6  0.120  0.135  0.137    61.0 0.002     72. 0.0000 .00202     881.
  5  0.100  0.135  0.137    60.7 0.002     73. 0.0000 .00202     883.
  4  0.080  0.134  0.137    60.0 0.002    210. 0.0000 .00202     885.
  3  0.060  0.133  0.136    58.5 0.002    510. 0.0000 .00202     892.
  2  0.040  0.132  0.134    55.4 0.002   1046. 0.0000 .00202     905.
  1  0.020  0.129  0.131    49.5 0.002   2105. 0.0000 .00201     923.
  0  0.000  0.136  0.138    41.0 0.002    170. 0.0000 .00200     877.
```

```
DATAFILE:  dxoutd1.dat
DISPERSED FLUID PARAMETERS
DCHX Configuration = 1
Dispersed Fluid:  PENT
                                        Thermal    Cond
Idx   Loo    Density     Viscosity     Capacit    Coeff  Enthalpy
 i     Z      rho           mu           Cp         k        Hd
       m      kg/m3        Pa-s        J/kg K    W/m K    J/kg K
150  3.000     6.248   0.79507E-05    1955.0   0.01948   -1178.2
149  2.980     6.254   0.79495E-05    1955.0   0.01948   -1178.3
148  2.960     6.260   0.79484E-05    1954.9   0.01947   -1178.3
147  2.940     6.267   0.79473E-05    1954.8   0.01947   -1178.4
146  2.920     6.273   0.79462E-05    1954.8   0.01947   -1178.5
145  2.900     6.279   0.79451E-05    1954.7   0.01946   -1178.6
144  2.880     6.286   0.79440E-05    1954.7   0.01946   -1178.7
143  2.860     6.292   0.79428E-05    1954.6   0.01945   -1178.7
142  2.840     6.298   0.79417E-05    1954.5   0.01945   -1178.8
141  2.820     6.305   0.79406E-05    1954.5   0.01944   -1178.9
140  2.800     6.311   0.79395E-05    1954.4   0.01944   -1179.0
139  2.780     6.317   0.79383E-05    1954.4   0.01944   -1179.1
138  2.760     6.324   0.79372E-05    1954.3   0.01943   -1179.1
137  2.740     6.330   0.79360E-05    1954.2   0.01943   -1179.2
136  2.720     6.336   0.79349E-05    1954.2   0.01942   -1179.3
135  2.700     6.343   0.79337E-05    1954.1   0.01942   -1179.4
134  2.680     6.349   0.79326E-05    1954.1   0.01941   -1179.5
133  2.660     6.355   0.79314E-05    1954.0   0.01941   -1179.5
132  2.640     6.362   0.79303E-05    1953.9   0.01941   -1179.6
131  2.620     6.368   0.79291E-05    1953.9   0.01940   -1179.7
130  2.600     6.375   0.79280E-05    1953.8   0.01940   -1179.8

 .     .        .           .            .         .         .
 .     .        .           .            .         .         .
 .     .        .           .            .         .         .
 .     .        .           .            .         .         .
 .     .        .           .            .         .         .
 .     .        .           .            .         .         .
 .     .        .           .            .         .         .
 .     .        .           .            .         .         .

 22  0.440   578.310   0.15578E-03    2486.9   0.10420   -1525.9
 21  0.420   578.524   0.15604E-03    2486.1   0.10427   -1526.5
 20  0.400   578.740   0.15631E-03    2485.2   0.10434   -1527.0
 19  0.380   578.961   0.15657E-03    2484.4   0.10441   -1527.5
 18  0.360   579.184   0.15685E-03    2483.5   0.10448   -1528.1
 17  0.340   579.411   0.15712E-03    2482.6   0.10455   -1528.6
 16  0.320   579.642   0.15740E-03    2481.7   0.10463   -1529.2
 15  0.300   579.876   0.15768E-03    2480.8   0.10470   -1529.7
 14  0.280   580.114   0.15797E-03    2479.8   0.10478   -1530.3
 13  0.260   580.355   0.15826E-03    2478.9   0.10485   -1530.9
 12  0.240   580.600   0.15856E-03    2477.9   0.10493   -1531.5
 11  0.220   580.849   0.15886E-03    2477.0   0.10501   -1532.1
 10  0.200   581.102   0.15916E-03    2476.0   0.10508   -1532.7
  9  0.180   581.359   0.15947E-03    2475.0   0.10516   -1533.3
  8  0.160   581.620   0.15979E-03    2474.0   0.10524   -1533.9
  7  0.140   581.885   0.16011E-03    2473.0   0.10532   -1534.6
  6  0.120   582.154   0.16043E-03    2472.0   0.10541   -1535.2
  5  0.100   582.428   0.16076E-03    2471.0   0.10549   -1535.9
  4  0.080   583.086   0.16156E-03    2468.5   0.10569   -1537.5
  3  0.060   584.653   0.16347E-03    2462.7   0.10615   -1541.3
  2  0.040   587.767   0.16737E-03    2451.6   0.10704   -1548.9
  1  0.020   593.644   0.17565E-03    2432.1   0.10863   -1563.3
  0  0.000   601.951   0.19100E-03    2408.0   0.11076   -1583.9
```

```
DATAFILE:  dxoutd2.dat
DISPERSED DROPLET/BUBBLE INFO
Dispersed Fluid:  PENT
 i     Z         hT       hS       hTS        hST       hBB         h
150  3.000      0.0      0.0      0.0     19233.2     10.6       10.6
149  2.980      0.0      0.0      0.0     19230.1     10.6       10.6
148  2.960      0.0      0.0      0.0     19226.9     10.6       10.6
147  2.940      0.0      0.0      0.0     19223.8     10.6       10.6
146  2.920      0.0      0.0      0.0     19220.7     10.6       10.6
145  2.900      0.0      0.0      0.0     19217.6     10.6       10.6
144  2.880      0.0      0.0      0.0     19214.4     10.6       10.6
143  2.860      0.0      0.0      0.0     19211.3     10.6       10.6
142  2.840      0.0      0.0      0.0     19208.2     10.6       10.6
141  2.820      0.0      0.0      0.0     19205.1     10.6       10.6
140  2.800      0.0      0.0      0.0     19201.9     10.6       10.6
139  2.780      0.0      0.0      0.0     19198.8     10.6       10.6
138  2.760      0.0      0.0      0.0     19195.7     10.6       10.6
137  2.740      0.0      0.0      0.0     19192.5     10.6       10.6
136  2.720      0.0      0.0      0.0     19189.4     10.6       10.6
135  2.700      0.0      0.0      0.0     19186.3     10.6       10.6
134  2.680      0.0      0.0      0.0     19183.2     10.6       10.6
133  2.660      0.0      0.0      0.0     19180.0     10.6       10.6
132  2.640      0.0      0.0      0.0     19176.9     10.6       10.6
131  2.620      0.0      0.0      0.0     19173.8     10.6       10.6
130  2.600      0.0      0.0      0.0     19170.6     10.6       10.6

  .     .         .        .        .          .         .          .
  .     .         .        .        .          .         .          .
  .     .         .        .        .          .         .          .
  .     .         .        .        .          .         .          .
  .     .         .        .        .          .         .          .
  .     .         .        .        .          .         .          .
  .     .         .        .        .          .         .          .
  .     .         .        .        .          .         .          .

 25  0.500   2494.1    101.4     97.4     19233.1      0.0       97.4
 24  0.480   2493.6    101.5     97.5     19233.1      0.0       97.5
 23  0.460   2493.1    101.5     97.6     19233.1      0.0       97.6
 22  0.440   2492.6    101.6     97.7     19233.1      0.0       97.7
 21  0.420   2492.1    101.7     97.7     19233.1      0.0       97.7
 20  0.400   2491.5    101.8     97.8     19233.1      0.0       97.8
 19  0.380   2491.0    101.9     97.9     19233.1      0.0       97.9
 18  0.360   2490.4    102.0     98.0     19233.1      0.0       98.0
 17  0.340   2489.8    102.1     98.1     19233.1      0.0       98.1
 16  0.320   2489.3    102.2     98.2     19233.1      0.0       98.2
 15  0.300   2488.7    102.3     98.2     19233.1      0.0       98.2
 14  0.280   2488.1    102.4     98.3     19233.1      0.0       98.3
 13  0.260   2487.5    102.5     98.4     19233.1      0.0       98.4
 12  0.240   2486.9    102.6     98.5     19233.1      0.0       98.5
 11  0.220   2486.2    102.7     98.6     19233.1      0.0       98.6
 10  0.200   2485.6    102.8     98.7     19233.1      0.0       98.7
  9  0.180   2484.9    102.9     98.8     19233.1      0.0       98.8
  8  0.160   2484.3    103.0     98.9     19233.1      0.0       98.9
  7  0.140   2483.6    103.1     98.9     19233.1      0.0       98.9
  6  0.120   2482.9    103.1     99.0     19233.1      0.0       99.0
  5  0.100   2482.1    103.2     99.1     19233.1      0.0       99.1
  4  0.080   2480.1    103.4     99.3     19233.1      0.0      274.9
  3  0.060   2475.6    103.9     99.7     19233.1      0.0      616.0
  2  0.040   2466.8    104.9    100.6     19233.1      0.0     1089.1
  1  0.020   2455.5    107.6    103.1     19233.1      0.0     1733.0
  0  0.000   2503.9    121.6    116.0     19233.1      0.0      116.0
```

```
DATAFILE:  dxoutc.dat
CONTINUOUS FLUID RESULTS
DCHX Configuration = 1
Continuous Fluid:  H2O
Idx   Loc    Vel       Temp   HtXfer          Press
 i     Z     Uc        Tc     QcvC    1-Phi    P
             m/s       C      W                kPa
150  3.000 -0.00255   85.0    -10.  0.9858   231.67
149  2.980 -0.00255   85.0    -10.  0.9858   231.86
148  2.960 -0.00255   85.0    -10.  0.9858   232.05
147  2.940 -0.00255   85.0    -10.  0.9858   232.23
146  2.920 -0.00255   85.0    -10.  0.9858   232.42
145  2.900 -0.00255   85.0    -10.  0.9858   232.61
144  2.880 -0.00255   85.0    -10.  0.9858   232.80
143  2.860 -0.00255   85.0    -10.  0.9858   232.98
142  2.840 -0.00255   85.0    -11.  0.9858   233.17
141  2.820 -0.00255   85.0    -11.  0.9858   233.36
140  2.800 -0.00255   85.0    -11.  0.9858   233.55
139  2.780 -0.00255   85.0    -11.  0.9858   233.73
138  2.760 -0.00255   85.0    -11.  0.9858   233.92
137  2.740 -0.00255   85.0    -11.  0.9859   234.11
136  2.720 -0.00255   85.0    -11.  0.9859   234.30
135  2.700 -0.00255   85.0    -11.  0.9859   234.48
134  2.680 -0.00255   85.0    -11.  0.9859   234.67
133  2.660 -0.00255   85.0    -11.  0.9859   234.86
132  2.640 -0.00255   85.0    -11.  0.9859   235.05
131  2.620 -0.00255   85.0    -11.  0.9859   235.23
130  2.600 -0.00255   85.0    -11.  0.9859   235.42

 .     .       .        .       .      .        .
 .     .       .        .       .      .        .
 .     .       .        .       .      .        .
 .     .       .        .       .      .        .
 .     .       .        .       .      .        .
 .     .       .        .       .      .        .
 .     .       .        .       .      .        .
 .     .       .        .       .      .        .

 23  0.460 -0.00250   78.7    -66.  0.9981   255.61
 22  0.440 -0.00250   78.7    -67.  0.9981   255.80
 21  0.420 -0.00250   78.7    -68.  0.9981   255.99
 20  0.400 -0.00250   78.7    -69.  0.9981   256.19
 19  0.380 -0.00250   78.7    -71.  0.9981   256.38
 18  0.360 -0.00250   78.7    -72.  0.9981   256.57
 17  0.340 -0.00250   78.6    -73.  0.9981   256.76
 16  0.320 -0.00250   78.6    -74.  0.9981   256.95
 15  0.300 -0.00250   78.6    -75.  0.9981   257.14
 14  0.280 -0.00250   78.6    -76.  0.9981   257.33
 13  0.260 -0.00250   78.6    -78.  0.9981   257.52
 12  0.240 -0.00250   78.6    -79.  0.9981   257.71
 11  0.220 -0.00250   78.6    -80.  0.9981   257.90
 10  0.200 -0.00250   78.6    -81.  0.9981   258.09
  9  0.180 -0.00250   78.6    -83.  0.9981   258.28
  8  0.160 -0.00250   78.5    -84.  0.9981   258.47
  7  0.140 -0.00250   78.5    -85.  0.9981   258.66
  6  0.120 -0.00250   78.5    -87.  0.9981   258.86
  5  0.100 -0.00250   78.5    -88.  0.9981   259.05
  4  0.080 -0.00250   78.5   -254.  0.9981   259.24
  3  0.060 -0.00250   78.4   -617.  0.9980   259.43
  2  0.040 -0.00250   78.3  -1266.  0.9980   259.62
  1  0.020 -0.00250   78.0  -2547.  0.9980   259.81
  0  0.000 -0.00250   77.6   -206.  0.9981   260.00
```

```
DATAFILE:  dxoutc1.dat
CONTINUOUS FLUID PARAMETERS
DCHX Configuration = 1
Continuous Fluid:  H2O
                                              Thermal    Cond
Idx    Loc     Density      Viscosity        Capacit     Coeff  Enthalpy
 i      Z        rho           mu              Cp          k       Hd
        m       kg/m3         Pa-s           J/kg K     W/m K   J/kg K
150   3.000    968.704    0.32794E-03     4199.1   0.67218     356.0
149   2.980    968.705    0.32795E-03     4199.1   0.67218     356.0
148   2.960    968.706    0.32795E-03     4199.1   0.67218     356.0
147   2.940    968.707    0.32795E-03     4199.1   0.67218     356.0
146   2.920    968.708    0.32796E-03     4199.1   0.67218     356.0
145   2.900    968.709    0.32796E-03     4199.1   0.67218     356.0
144   2.880    968.710    0.32796E-03     4199.1   0.67218     356.0
143   2.860    968.711    0.32796E-03     4199.1   0.67218     356.0
142   2.840    968.712    0.32797E-03     4199.1   0.67218     356.0
141   2.820    968.713    0.32797E-03     4199.1   0.67218     356.0
140   2.800    968.714    0.32797E-03     4199.1   0.67218     356.0
139   2.780    968.715    0.32798E-03     4199.1   0.67218     356.0
138   2.760    968.716    0.32798E-03     4199.1   0.67217     356.0
137   2.740    968.717    0.32798E-03     4199.1   0.67217     356.0
136   2.720    968.718    0.32798E-03     4199.1   0.67217     355.9
135   2.700    968.720    0.32799E-03     4199.1   0.67217     355.9
134   2.680    968.721    0.32799E-03     4199.1   0.67217     355.9
133   2.660    968.722    0.32799E-03     4199.1   0.67217     355.9
132   2.640    968.723    0.32800E-03     4199.1   0.67217     355.9
131   2.620    968.724    0.32800E-03     4199.1   0.67217     355.9
130   2.600    968.725    0.32800E-03     4199.1   0.67217     355.9

 .      .         .            .             .         .          .
 .      .         .            .             .         .          .
 .      .         .            .             .         .          .
 .      .         .            .             .         .          .
 .      .         .            .             .         .          .
 .      .         .            .             .         .          .
 .      .         .            .             .         .          .
 .      .         .            .             .         .          .

 22   0.440    972.700    0.35489E-03     4193.5   0.66830     329.6
 21   0.420    972.706    0.35493E-03     4193.5   0.66830     329.6
 20   0.400    972.712    0.35497E-03     4193.5   0.66829     329.5
 19   0.380    972.718    0.35501E-03     4193.5   0.66828     329.5
 18   0.360    972.724    0.35505E-03     4193.5   0.66828     329.4
 17   0.340    972.731    0.35510E-03     4193.5   0.66827     329.4
 16   0.320    972.737    0.35514E-03     4193.5   0.66826     329.3
 15   0.300    972.743    0.35519E-03     4193.5   0.66826     329.3
 14   0.280    972.750    0.35523E-03     4193.5   0.66825     329.3
 13   0.260    972.756    0.35528E-03     4193.4   0.66824     329.2
 12   0.240    972.763    0.35533E-03     4193.4   0.66824     329.2
 11   0.220    972.770    0.35538E-03     4193.4   0.66823     329.1
 10   0.200    972.777    0.35543E-03     4193.4   0.66822     329.1
  9   0.180    972.784    0.35548E-03     4193.4   0.66822     329.0
  8   0.160    972.791    0.35553E-03     4193.4   0.66821     329.0
  7   0.140    972.798    0.35558E-03     4193.4   0.66820     328.9
  6   0.120    972.806    0.35563E-03     4193.4   0.66819     328.9
  5   0.100    972.813    0.35569E-03     4193.4   0.66819     328.8
  4   0.080    972.831    0.35582E-03     4193.4   0.66817     328.7
  3   0.060    972.874    0.35614E-03     4193.3   0.66812     328.4
  2   0.040    972.959    0.35678E-03     4193.2   0.66803     327.8
  1   0.020    973.119    0.35800E-03     4193.0   0.66786     326.8
  0   0.000    973.348    0.35976E-03     4192.7   0.66761     325.2
```

```
DATAFILE:  dxoutfh.dat
FLUID AND HEAT TRANSFER VALUES
DCHX Configuration = 1
               Reynolds    Peclect    Drag         Ht Xfer
Idx    Loc     Number      Number     Coeff        Coeff
        Z       Re          Pe         Cd            h
        m                                         W/m2 K
150   3.000    93075.     190767.   0.283E+00       10.6
149   2.980    92999.     190614.   0.283E+00       10.6
148   2.960    92924.     190461.   0.283E+00       10.6
147   2.940    92849.     190308.   0.283E+00       10.6
146   2.920    92773.     190156.   0.283E+00       10.6
145   2.900    92698.     190004.   0.283E+00       10.6
144   2.880    92623.     189852.   0.283E+00       10.6
143   2.860    92548.     189699.   0.283E+00       10.6
142   2.840    92473.     189548.   0.283E+00       10.6
141   2.820    92398.     189396.   0.283E+00       10.6
140   2.800    92323.     189244.   0.283E+00       10.6
139   2.780    92248.     189093.   0.283E+00       10.6
138   2.760    92174.     188942.   0.283E+00       10.6
137   2.740    92099.     188790.   0.283E+00       10.6
136   2.720    92024.     188639.   0.283E+00       10.6
135   2.700    91950.     188488.   0.283E+00       10.6
134   2.680    91875.     188338.   0.283E+00       10.6
133   2.660    91801.     188187.   0.283E+00       10.6
132   2.640    91726.     188036.   0.283E+00       10.6
131   2.620    91652.     187886.   0.283E+00       10.6
130   2.600    91578.     187735.   0.283E+00       10.6

 .      .         .          .          .             .
 .      .         .          .          .             .
 .      .         .          .          .             .
 .      .         .          .          .             .
 .      .         .          .          .             .
 .      .         .          .          .             .
 .      .         .          .          .             .
 .      .         .          .          .             .

 23   0.460     1552.       3460.   0.433E+00       97.6
 22   0.440     1551.       3456.   0.433E+00       97.7
 21   0.420     1549.       3453.   0.434E+00       97.7
 20   0.400     1547.       3450.   0.434E+00       97.8
 19   0.380     1545.       3446.   0.434E+00       97.9
 18   0.360     1544.       3442.   0.434E+00       98.0
 17   0.340     1542.       3439.   0.434E+00       98.1
 16   0.320     1540.       3435.   0.434E+00       98.2
 15   0.300     1538.       3432.   0.434E+00       98.2
 14   0.280     1536.       3428.   0.435E+00       98.3
 13   0.260     1534.       3424.   0.435E+00       98.4
 12   0.240     1532.       3420.   0.435E+00       98.5
 11   0.220     1530.       3416.   0.435E+00       98.6
 10   0.200     1528.       3412.   0.435E+00       98.7
  9   0.180     1526.       3408.   0.435E+00       98.8
  8   0.160     1524.       3404.   0.436E+00       98.9
  7   0.140     1522.       3400.   0.436E+00       98.9
  6   0.120     1519.       3395.   0.436E+00       99.0
  5   0.100     1514.       3390.   0.436E+00       99.1
  4   0.080     1502.       3379.   0.437E+00      274.9
  3   0.060     1479.       3352.   0.438E+00      616.0
  2   0.040     1444.       3300.   0.441E+00     1089.1
  1   0.020     1508.       3225.   0.440E+00     1733.0
  0   0.000     1495.       3377.   0.000E+00      116.0
```

APPENDIX VI

PROGRAM LISTING FOR ONE-DIMENSIONAL SIMULATION

```
C     PROGRAM TO SIMULATE A DIRECT CONTACT HEAT EXCHANGER
C     RICHARD BRICKMAN - DECEMBER 1995
C
      REAL  UD(0:2001),UC(0:2001),TD(0:2001),TC(0:2001),
     #     HUR(0:2001),DENSD(0:2001),DENSC(0:2001),HD(0:2001),
     #     HC(0:2001),R(0:2001),QCV(0:2001),XD(0:2001),P(0:2001),
     #     XC(0:2001),CKC(0:1),VISC(0:1),BET(0:2001),H(0:2001),
     #     HS(0:2001),HT(0:2001),ALPHC(0:2001),VISCD(0:2001),
     #     RYNLD(0:2001),COEFD(0:2001),PEC(0:2001),HCB(0:2001),
     #     CPPC(0:2001),CPPD(0:2001),CKKC(0:2001),CKKD(0:2001),
     #     WG(0:2001),HM(0:2001),DIFF(0:2001),ALPHD(0:2001),
     #     VISCC(0:2001),QCV2(0:2001),HSTAG(0:2001),BPOP(0:2001),
     #     HBB(0:2001),HST(0:2001),UDZ(0:2001)
      INTEGER TCNFIG
      COMMON/HHH/H,HS,HT,HCB,HBB,HST
      COMMON/ARRY/UDZ,UD,UC,TD,TC,P,HUR,DENSD,DENSC,HD,HC,R,QCV,
     #     DZ,XD,BET,CPPC,CPPD,CKKD,CKKC,PEC,VISCD,VISC,RYNLD,
     #     BPOP,VISCC,COEFD,WG,HM,DIFF,QCV2,HSTAG
      COMMON/SCLR/MRD,MRC,TEMPD,TEMPC,R0,PRESS,CQUAL,ICNFIG,TCNFIG,
     #     N,DNZAREA,DXVOLUME,DXLENGTH,DXRADIUS,DFTYPE,CFTYPE
      COMMON/SCLR2/ETA
C
      REAL  MRD,MRC,KP(4),KHD(4),KHC(4),KPC,KHDC,KHCC,NB,MC,MD,GMM,
     #     MT,IMT,MB
      REAL NU0,NUC
      CHARACTER*6,DFTYPE
      CHARACTER*6,CFTYPE
C
      IF(IFLAGON.EQ.1)PRINT*,'STARTING'
      PI=3.141592654
      G=9.81
      IF(IFLAGON.EQ.1)PRINT*,'FIRST CALCULATIONS'
C
C     DATA INPUT
C           Data is entered into program through an input file that
C               must accompany this coding.
C
      OPEN (UNIT=1, FILE='dxin.dat', STATUS='OLD')
      OPEN (UNIT=2, FILE='dxoutd.dat', STATUS='UNKNOWN')
      OPEN (UNIT=3, FILE='dxoutc.dat', STATUS='UNKNOWN')
      OPEN (UNIT=4, FILE='dxoutfh.dat', STATUS='UNKNOWN')
      OPEN (UNIT=5, FILE='dxoutd1.dat', STATUS='UNKNOWN')
      OPEN (UNIT=6, FILE='dxoutc1.dat', STATUS='UNKNOWN')
      OPEN (UNIT=7, FILE='dxout.dat', STATUS='UNKNOWN')
      OPEN (UNIT=8, FILE='dxoutmt.dat', STATUS='UNKNOWN')
      OPEN (UNIT=9, FILE='dxoutd2.dat', STATUS='UNKNOWN')
      OPEN (UNIT=10, FILE='xport.dat', STATUS='UNKNOWN')
```

```
      WRITE(2,*)'DATAFILE:  dxoutd.dat'
      WRITE(3,*)'DATAFILE:  dxoutc.dat'
      WRITE(4,*)'DATAFILE:  dxoutfh.dat'
      WRITE(5,*)'DATAFILE:  dxoutdl.dat'
      WRITE(6,*)'DATAFILE:  dxoutcl.dat'
      WRITE(7,*)'DATAFILE:  dxout.dat'
      WRITE(8,*)'DATAFILE:  dxoutmt.dat'
      WRITE(9,*)'DATAFILE:  dxoutd2.dat'
C
      IF(IFLAGON.EQ.1)PRINT*,'READING INPUT DATA'
      READ(1,6000)DFTYPE
      READ(1,*)MRD
      READ(1,*)TEMPD
      READ(1,*)RO
      READ(1,6000)CFTYPE
      READ(1,*)MRC
      READ(1,*)TEMPC
      READ(1,*)CQUAL
      CALL FLUIDCAT(DFTYPE,DFCAT,RD)
      CALL FLUIDCAT(CFTYPE,CFCAT,RC)
 6000 FORMAT(A6)
      IF(DFCAT.EQ.3) THEN
         WRITE(7,*)DFTYPE,'UPDATE REQUIRED IN PROGRAM TO RUN THIS
     # DISPERSED FLUID.'
         GOTO 9998
      ELSEIF(CFCAT.EQ.3) THEN
         WRITE(7,*)CFTYPE,'UPDATE REQUIRED IN PROGRAM TO RUN THIS
     # CONTINUOUS FLUID.'
         GOTO 9998
      ENDIF
C
      READ(1,*)DXRADIUS
      READ(1,*)DXLENGTH
      READ(1,*)PRESS
      READ(1,*)DNZAREA
      READ(1,*)N
      READ(1,*)HTOL,TTOL
      READ(1,*)IAF,ETA
      IFLAGON=0
C
      IF(IAF.LE.0) ACCELF=1.E20
      IF(IAF.EQ.2) ACCELF=1./.382
      IF(IAF.EQ.3) ACCELF=1./.618
      IF(IAF.GT.3) ACCELF=2.
      IF(N.GT.2000) THEN
         WRITE(7,*)'NUMBER OF DIVISIONS OF COLUMN CANNOT EXCEED 2000'
         GOTO 9998
      ENDIF
C
      CALL PPRNT
C
      IF(IFLAGON.EQ.1)PRINT*,'INITIALIZING'
      MD=ABS(MRD)
      MC=ABS(MRC)
      IX=CQUAL
C
C
      DZ=DXLENGTH/N
      DXAREA=PI*DXRADIUS**2
      IF(DNZAREA.GT.DXAREA) THEN
         WRITE(7,*)'NOZZLE AREA EXCEEDS COLUMN CROSS SECTION AREA'
         DNZAREA=DXAREA/5.
```

```
      ENDIF
      DXVOLUME=DXAREA*DXLENGTH
      XCCHK=0
C
C     Configuration Check
      IF(IFLAGON.EQ.1)PRINT*,'CHECKING CONFIGURATION'
C
      TGIVEN=TEMPC
      PGIVEN=PRESS
      IF(CFCAT.EQ.1) THEN
         CALL SIFLUID('TP',CFTYPE,TGIVEN,PGIVEN,XCZ,SCZ,HCZ,VCZ,CUZ,
     #        TCZ,PCZ,SLCZ,HLCZ,VLCZ,ULCZ,SVCZ,HVCZ,VVCZ,UVCZ)
         CALL QUALCHCK(XCZ)
      ELSEIF (CFCAT.EQ.2) THEN
         CALL FLTHCAP(CFTYPE,TGIVEN,PGIVEN,CP1,RHOCZ,XCZ,ITRNG)
         CALL TRNGCHK(ITRNG,tgiven)
         VCZ=1./RHOCZ
      ENDIF
      TGIVEN=TEMPD
      PGIVEN=PRESS
      IF(DFCAT.EQ.1) THEN
         CALL SIFLUID('TP',DFTYPE,TGIVEN,PGIVEN,XDZ,SDZ,HDZ,VDZ,DUZ,
     #        TDZ,PDZ,SLDZ,HLDZ,VLDZ,ULDZ,SVDZ,HVDZ,VVDZ,UVDZ)
         CALL QUALCHCK(XDZ)
      ELSEIF (DFCAT.EQ.2) THEN
         CALL FLTHCAP(DFTYPE,TGIVEN,PGIVEN,CP1,RHODZ,XDZ,ITRNG)
         CALL TRNGCHK(ITRNG,tgiven)
         VDZ=1./RHODZ
      ENDIF
C
      IF(VDZ/VCZ.GT.1.) THEN
         ICNFIG=1
      ELSE
         ICNFIG=2
      ENDIF
C
      IF(TEMPC.GT.TEMPD) THEN
         TCNFIG=1
      ELSE
         TCNFIG=2
      ENDIF
C
      IPASS=1
C
C     INITIAL VALUE CALCULATIONS
      IF(IFLAGON.EQ.1)PRINT*,'INITIAL VALUE CALCULATIONS'
C
      IF(ICNFIG.EQ.1) THEN
         TGIVEN=TEMPD
c         PGIVEN=PRESS+G/VCZ*DXLENGTH/1000
         PGIVEN=PRESS
         IF(DFCAT.EQ.1) THEN
            CALL SIFLUID('TP',DFTYPE,TGIVEN,PGIVEN,XDZ,SDZ,HDZ,VDZ,
     #           DUZ,TDZ,PDZ,SLDZ,HLDZ,VLDZ,ULDZ,SVDZ,HVDZ,VVDZ,UVDZ)
            CALL QUALCHCK(XDZ)
            HD(0)=HDZ
            TD(0)=TDZ
            DENSD(0)=1./VDZ
         ELSEIF(DFCAT.EQ.2) THEN
            CALL FLTHCAP(DFTYPE,TGIVEN,PGIVEN,CP1,RHODZ,XDZ,ITRNG)
            CALL TRNGCHK(ITRNG,tgiven)
            HD(0)=0
```

```
            TD(0)=TEMPD
            VDZ=1/RHODZ
            DENSD(0)=RHODZ
         ENDIF
         XD(0)=XDZ
         MRC=-ABS(MRC)
         MRD=ABS(MRD)
         R(0)=R0
      ELSE
         TGIVEN=TEMPC
C         PGIVEN=PRESS+G/VCZ*DXLENGTH/1000
         PGIVEN=PRESS
         IF(CFCAT.EQ.1) THEN
            CALL SIFLUID('TP',CFTYPE,TGIVEN,PGIVEN,XCZ,SCZ,HCZ,VCZ,CUZ,
     #           TCZ,PCZ,SLCZ,HLCZ,VLCZ,ULCZ,SVCZ,HVCZ,VVCZ,UVCZ)
            CALL QUALCHCK(XCZ)
            HC(0)=HCZ
            TC(0)=TCZ
            DENSC(0)=1./VCZ
         ELSEIF(CFCAT.EQ.2) THEN
            CALL FLTHCAP(CFTYPE,TGIVEN,PGIVEN,CP1,RHOCZ,XCZ,ITRNG)
            CALL TRNGCHK(ITRNG,tgiven)
            HC(0)=0
            TC(0)=TEMPC
            VCZ=1/RHOCZ
            DENSC(0)=RHOCZ
         ENDIF
         XC(0)=XCZ
         MRD=-ABS(MRD)
         MRC=ABS(MRC)
         RN=R0
         DENSCG=DENSC(0)
      ENDIF
C
C
C     SHOOTING METHOD INITIALIZATION
      IF(IFLAGON.EQ.1)PRINT*,'INITIALIZE SHOOTING METHOD'
C       Bisection using the continuous and dispersed fluid
C       input temperatures as the starting upper and lower limits.
C
      ICOUNT=0
C
      IF(TEMPC.GT.TEMPD) THEN
         TUPPER=TEMPC
         TLOWER=TEMPD
      ELSE
         TUPPER=TEMPD
         TLOWER=TEMPC
      ENDIF
      ITLIM=1
C
C  BEGIN ITERATION FOR SHOOTING METHOD
      IF(IFLAGON.EQ.1)PRINT*,'BEGIN SHOOTING ITERATION'
C
 11   IF(IFLAGON.EQ.1)PRINT*,ICOUNT,'-',I-1
      ICOUNT=ICOUNT+1
      PRINT*,'ITERATION ',ICOUNT
 1    IF(XCCHK.EQ.1) GOTO 2
      IF(ABS(TUPPER-TLOWER).LE.TTOL) THEN
         IF(ICNFIG.EQ.1)THEN
            WRITE(*,*)'COULD NOT FIND SUITABLE INITIAL TC(0)'
            WRITE(7,*)'COULD NOT FIND SUITABLE INITIAL TC(0)'
```

```
         ELSE
            WRITE(*,*)'COULD NOT FIND SUITABLE INITIAL TD(0)'
            WRITE(7,*)'COULD NOT FIND SUITABLE INITIAL TD(0)'
         ENDIF
         GOTO 9998
      ENDIF
 2    IF(ICNFIG.EQ.1) THEN
            TC(0)=(TUPPER+TLOWER)/2.
         ELSE
            TD(0)=(TUPPER+TLOWER)/2.
      ENDIF
C
C     Iteration dependent initial values
      IF(IFLAGON.EQ.1)PRINT*,'ITERATION DEPENDENT INITIAL VALUE CALC'
C
      IF(ICNFIG.EQ.1) THEN
c         P(0)=PRESS+(MD*DENSD(0)+MC/VCZ)/(MC+MD)*DXLENGTH*G/1000.
         P(0)=PRESS
         TGIVEN=TC(0)
         PGIVEN=P(0)
         IF(CFCAT.EQ.1) THEN
            CALL SIFLUID('TP',CFTYPE,TGIVEN,PGIVEN,XC(0),SC,HC(0),VC,
     #           UIC,TC(0),PC,SLC,HLC,VLC,ULC,SVC,HVC,VVC,UVC)
            CALL QUALCHCK(XC(0))
         ELSEIF(CFCAT.EQ.2) THEN
            CALL FLTHCAP(CFTYPE,TGIVEN,PGIVEN,CP1,DENSC(0),XC(0),ITRNG)
            CALL TRNGCHK(ITRNG,tgiven)
         ENDIF
         IF(CFCAT.EQ.2) GOTO 5
         IF(XCCHK.EQ.1) GOTO 5
         IF(XC(0).NE.CQUAL)THEN
            IF(TCNFIG.EQ.1) THEN
               PGIVEN=P(0)
               XGIVEN=CQUAL
               CALL SIFLUID('PX',CFTYPE,PGIVEN,XGIVEN,XCX,SCX,HCX,
     #              VCX,UICX,TLOWER,PCX,SLCX,HLCX,VLCX,ULCX,SVCX,
     #              HVCX,VVCX,UVCX)
               GOTO 1
            ELSE
               PGIVEN=P(0)
               XGIVEN=CQUAL
               CALL SIFLUID('PX',CFTYPE,PGIVEN,XGIVEN,XCX,SCX,HCX,
     #              VCX,UICX,TUPPER,PCX,SLCX,HLCX,VLCX,ULCX,SVCX,
     #              HVCX,VVCX,UVCX)
               GOTO 1
            ENDIF
         ENDIF
 5       xcchk=1
         IF(CFCAT.EQ.1) DENSC(0)=1./VC
         IF(CFCAT.EQ.2) HC(0)=0.0
      ELSE
         P(0)=PRESS
c         P(0)=PRESS+(MC*DENSC(0)+MD/VDZ)/(MD+MC)*G*DXLENGTH/1000.
         TGIVEN=TD(0)
         PGIVEN=P(0)
         IF(DFCAT.EQ.1) THEN
            CALL SIFLUID('TP',DFTYPE,TGIVEN,PGIVEN,XD(0),SD,HD(0),VD,
     #           UID,TD(0),PD,SLD,HLD,VLD,ULD,SVD,HVD,VVD,UVD)
            CALL QUALCHCK(XD(0))
            DENSD(0)=1./VD
            IF(VLD*VVD.NE.0) THEN
               RHODL=1./VLD
```

```
               RHODV=1./VVD
            ENDIF
         ELSEIF(DFCAT.EQ.2) THEN
            CALL FLTHCAP(DFTYPE,TGIVEN,PGIVEN,CP1,DENSD(0),XD(0),ITRNG)
            CALL TRNGCHK(ITRNG,tgiven)
            HD(0)=0.0
         ENDIF
      ENDIF
C
C     GUESS OF DISPERSED FLUID VELOCITY AT LOCATION Z=0
C
      IF(ICOUNT.EQ.1) THEN
         UDZ(0)=MRD/DENSD(0)/DNZAREA
      ELSE
         DO 92 IUI=10,1,-1
            IF(UDZ(IUI).NE.0.) THEN
               UDZ(0)=UDZ(IUI)
               GOTO 91
            ENDIF
 92      CONTINUE
 91      DNZAREA=MRD/DENSD(0)/UDZ(0)
      ENDIF
      PHI=MRD/DENSD(0)/UDZ(0)/DXAREA
      HUR(0)=PHI
      IF(ICNFIG.EQ.2) R(0)=RN*(1/DENSD(0)/VDZ)**(1./3.)
      UC(0)=MRC/(1-HUR(0))/DXAREA/DENSC(0)
      UDPR=0.
      UCPR=0.
      IF(XD(0).GE.1.) IXDREF=1
      IF((2*R(0)).LT.DZ) THEN
         NB0=DXAREA*DZ*PHI/(4./3.*PI*R(0)**3.)
      ELSE
         CVIN=2*R(0)/DZ
         NB0=DXAREA*DZ*CVIN*PHI/(4./3.*PI*R(0)**3.)
      ENDIF
      NB=NB0
      BPOP(0)=NB0
      WG(0)=0.0
      MB=DENSD(0)*4./3.*PI*R(0)**3
      DMB=0
C
C  SOLUTION OF DIFFERENTIAL EQUATIONS AND Z (LOCATION) DEPENDENT
C  VARIABLES ALONG LENGTH OF COLUMN
C
      IF(IFLAGON.EQ.1)PRINT*,'MARCHING FOR SOLUTION'
      DO 20 I=0,N
C
C  THERMAL CAPACITANCE OF CONTINUOUS FLUID (Cp=dH/dT)
C
         TGIVEN=TC(I)
         PGIVEN=P(I)
         IF(CFCAT.EQ.1) THEN
            CALL SIFLUID('TP',CFTYPE,TGIVEN,PGIVEN,XC2,SC2,HC2,VC2,UC2,
     #           TC2,PC2,SLC2,HLC2,VLC2,ULC2,SVC2,HVC2,VVC2,UVC2)
            TGIVEN=TC(I)-.0001
            PGIVEN=P(I)
            CALL SIFLUID('TP',CFTYPE,TGIVEN,PGIVEN,XC1,SC1,HC1,VC1,UC1,
     #           TC1,PC1,SLC1,HLC1,VLC1,ULC1,SVC1,HVC1,VVC1,UVC1)
            CPC=(HC2-HC1)/.0001*1000
         ELSEIF(CFCAT.EQ.2) THEN
            CALL FLTHCAP(CFTYPE,TGIVEN,PGIVEN,CPC,RHOC,XC1,ITRNG)
            CALL TRNGCHK(ITRNG,tgiven)
```

```
      ENDIF
      CPPC(I)=CPC
C
C  THERMAL CAPACITANCE OF DISPERSED FLUID (Cp=dH/dT)
C   (Generally not needed but found for completeness)
C
      TGIVEN=TD(I)
      PGIVEN=P(I)
      IF(DFCAT.EQ.1) THEN
         CALL SIFLUID('TP',DFTYPE,TGIVEN,PGIVEN,XD2,SD2,HD2,VD2,UD2,
     #         TD2,PC2,SLD2,HLD2,VLD2,ULD2,SVD2,HVD2,VVD2,UVD2)
         TGIVEN=TD(I)-.0001
         PGIVEN=P(I)
         CALL SIFLUID('TP',DFTYPE,TGIVEN,PGIVEN,XD1,SD1,HD1,VD1,UD1,
     #         TD1,PD1,SLD1,HLD1,VLD1,ULD1,SVD1,HVD1,VVD1,UVD1)
         CPD=(HD2-HD1)/.0001*1000.
      ELSEIF(DFCAT.EQ.2) THEN
         CALL FLTHCAP(DFTYPE,TGIVEN,PGIVEN,CPD,RHOD,XD1,ITRNG)
         CALL TRNGCHK(ITRNG,tgiven)
      ENDIF
      CPPD(I)=CPD
C
C  BETA
C
      if(iflagon.eq.1)print*,'Beta'
      IF(XD(I).GT.0.0.AND.XD(I).LT.1.0) THEN
         CALL BBLMIX(RHODL,RHODV,XD(I),BETA)
      ELSEIF(XD(I).LE.0.0) THEN
         BETA=0.0
      ELSEIF(XD(I).GE.1.0) THEN
         BETA=PI
      ENDIF
      IF(BETA.GT.PI) BETA=PI
      BET(I)=BETA
C
      IF(XD(I).GE.1.AND.XD(I-1).LT.1.)IXDREF=I
C
C  PROPERTY EVALUATION
C
      TGIVEN=TD(I)
      PGIVEN=P(I)*1000.
      CALL PROPTY(DFTYPE,TGIVEN,PGIVEN,CKLD,CKVD,VISLD,VISVD,
     #     SIGD,ITRNG)
      CALL TRNGCHK(ITRNG,tgiven)
      TGIVEN=TC(I)
      PGIVEN=P(I)*1000.
      CALL PROPTY(CFTYPE,TGIVEN,PGIVEN,CKC(0),CKC(1),
     #     VISC(0),VISC(1),SIGC,ITRNG)
      CALL TRNGCHK(ITRNG,tgiven)
      VISCC(I)=VISC(IX)
      CKKD(I)=CKLD*(1.-XD(I))+CKVD*XD(I)
      CKKC(I)=CKC(IX)
C
C  ALPHA OF CONTINUOUS AND DISPERSED FLUID
C
      ALPHAC=CKC(IX)/DENSC(I)/CPC
      ALPHC(I)=ALPHAC
      ALPHAD=CKKD(I)/DENSD(I)/CPD
      ALPHD(I)=ALPHAD
C
C  DISPERSED FLUID VISCOSITY
C
```

```
         VISD=VISLD*(1-XD(I))+VISVD*XD(I)
         VISCD(I)=VISD
C
         IF(I.EQ.0) THEN
            R(I)=R0
            IREF=I
            UCPR=0.
            UDPR=0.
            PHI=MRD/DENSD(I)/DXAREA/UDZ(I)
            HUR(I)=PHI
            UC(I)=MRC/DENSC(I)/(1-PHI)/DXAREA
            UD(I)=UDZ(I)-UC(I)
            REC=DENSC(I)*2.*R(I)*ABS(UD(I))/VISC(IX)
         ELSE
C
C BUBBLE RADIUS
C
            R(I)=R(IREF)*(DENSD(IREF)/DENSD(I))**(1./3.)
            IF(WG(I).NE.0.) THEN
               VOLBUB=4./3.*PI*R(I)**3+WG(I)*MB/DENSCG
               R(I)=(VOLBUB*3./4./PI)**(1./3.)
            ENDIF
C
C REYNOLDS NUMBER
C
            if(iflagon.eq.1)print*,'Reynolds'
            REC=DENSC(I)*2.*R(I)*ABS(UD(I-1))/VISC(IX)
C
C DRAG COEFFICIENT
C
            if(iflagon.eq.1)print*,'CD'
            CD=(24./REC+6./(1+REC**0.5)+0.4)*((1+2*VISC(IX)/3/VISD)
     #         /(1+VISC(IX)/VISD))
            COEFD(I)=CD
C
C DISPERSED FLUID RELATIVE VELOCITY
C
            if(iflagon.eq.1)print*,'Velocity'
            GOTO 5677
            DELTARHO=ABS(DENSD(i)-DENSC(i))
            aMORTON=G*VISC(ix)**4*DELTARHO/DENSC(i)**2/SIGC**3
            EOTVOS=G*DELTARHO*R(i)**2/SIGC
            HHHH=4./3.*EOTVOS*aMORTON**(-.149)*(VISD/VISC(ix))**(-.14)
            IF(HHHH.LE.59.3) THEN
               aJJJJ=.94*HHHH**.757
            ELSE
               aJJJJ=3.42*HHHH**.441
            ENDIF
            UD(i)=VISC(ix)/DENSC(i)/2./R(I)*aMORTON**(-.149)
     #           *(aJJJJ-.857)
            GOTO 5678
C
C     This part not used anymore in lieu of above new velocity
C     calculation.
 5677       D=(R(I)**2+R(IREF)**2)/2/R(I)/R(0)
            TPC=TC(I)+273.15
            TPD=TD(I)+273.15
            IF(ICNFIG.EQ.1) THEN
               u1=1.1547
               u2=((1-densd(i)/densc(i)*(r(IREF)/r(i))**3)*(2*r(i)
     #           *g/cd))**0.5
               u3=D**(5./6.-D/tpc)
```

```
               u4=((tpc**2+tpd**2)/2/tpc/tpd)**D
               u5=(cpc*visc(iX)/ckc(iX))**(r(IREF)/1.6/r(i))
               UD(I)=u1*u2*u3/u4/u5
            ELSE
               IF(DENSD(I).GT.DENSC(I)) THEN
                  UD(I)=-ABS((4./3.*(DENSD(I)-DENSC(I))/DENSD(I)
     #                 *2*R(I)*G/CD))**0.5
               ELSE
                  UD(I)=ABS((4./3.*(DENSD(I)-DENSC(I))/DENSD(I)
     #                 *2*R(I)*G/CD))**0.5
               ENDIF
            ENDIF
 5678       UDZ(I)=UD(I)+UC(I-1)
C
C DISPERSED FLUID VELOCITY GRADIENT
C
            UDPR=(UDZ(I)-UDZ(I-1))/DZ
C
C HOLDUP RATIO
C
            PHI=MRD/DENSD(I)/DXAREA/UDZ(I)
            HUR(I)=PHI
C
C CONTINUOUS FLUID VELOCITY AND GRADIENT
C
            UC(I)=MRC/DENSC(I)/(1-HUR(I))/DXAREA
            UCPR=(UC(I)-UC(I-1))/DZ
C
         ENDIF
         RYNLD(I)=REC
         PEC(I)=ABS(UD(I))*2*R(I)/ALPHAC
         PRANDTLD=VISD*CPD/CKKD(I)

C
C DROPLET/BUBBLE POPULATION
C
         IF(2*R(I).GT.DZ) THEN
            NB=DXAREA*2.*R(I)*PHI/(4./3.*PI*R(I)**3)
         ELSE
            NB=DXAREA*DZ*PHI/(4./3.*PI*R(I)**3)
         ENDIF
         BPOP(I)=NB
C
C HEAT TRANSFER COEFFICIENT
C
C-Tochitani et al.
            if(iflagon.eq.1)print*,'Tochitani'
         HTH=CKC(IX)/2/R(I)*(ABS(UD(I))*2*R(I)/ALPHAC)**(1./3.)
     #        *(0.466194077*(PI-BETA+.5*SIN(2.*BETA))**(2./3.))
C
         IF(XD(I).LE.0.0)  THEN
C
C-Sideman
            if(iflagon.eq.1)print*,'Sideman'
            HSH=.00375*ABS(UD(I))*2*R(I)/ALPHAC*(1./(1.+VISC(IX)/
     #           VISLD))*CKLD/2./R(I)
C-Tochitani/Sideman Series Combination
            HHTS=HTH*HSH/(HTH+HSH)
C-Jacobs (Sideman) High Circulation and Decay - Weighted
            if(iflagon.eq.1)print*,'Jacobs(Sideman)'
C            HJAC=50+.0085*REC*PRANDTLD**.7*CKC(IX)/2./R(I)
            VKV=1.-1.45*(2.+3.*VISD/VISC(IX))/(1.+(DENSD(I)*VISD/
```

```
     #           DENSC(I)/VISC(IX))**.5)*REC**(-.5)
            HJAC=(2.+1.13*(PEC(I)*VKV)**.5)*CKC(IX)/2./R(I)
            HJT=HTH*HJAC/(HTH+HJAC)
            DDNORM=(I-IREF)*DZ/(2.*R(IREF))
            IF(DDNORM.LE.30) THEN
               HH=HTJ*(1.-DDNORM/30.)**2
C               HH=(0.5*HHTS+0.5*HJAC*(1.-DDNORM/10.)**2)
               IF(HH.LT.HHTS) HH=HHTS
            ELSE
               HH=HHTS
            ENDIF
            HCBN=0.
            HBBR=0.
            HFT=0.
         ELSEIF(XD(I).GT.0.0) THEN
C                                                   Coban Modification
            if(iflagon.eq.1)print*,'Coban modification'
            HCBN=5.*CKVD/R(I)
     #           *(1.-0.466194077*(PI-BETA+.5*SIN(2.*BETA))**(2./3.))
C
C                                           Brickman/Boehm Modification
C                                                 Steinberger & Treybal
            if(iflagon.eq.1)print*,'Brickman/Boehm'
            PR=PEC(I)/REC
            CALL H2OBETA(TC(I),CBeta)
            GR=9.81*CBeta*ABS(TD(I)-TC(I))*(2*R(I))**3/(VISC(IX)/
     #           DENSC(I))
            NU0=2.+.569*(GR*PR)**.25
            NUC=NU0+.347*REC**.62*PR**.31
            HSTR=NUC*CKC(IX)/2./R(I)
            HBBR=5.*HSTR*CKVD/(5.*CKVD+R(I)*HSTR)
     #           *(1.-0.466194077*(PI-BETA+.5*SIN(2.*BETA))**(2./3.))
C
C                                                 Mass Transfer Analogy
C                                                 (Fan/Tsuchiya, Ch 9)
            CALL DIFFUSEC(CFTYPE,DFTYPE,TC(I),P(I),ZD)
            if(iflagon.eq.1)print*,'Fan/Tsuchiya'
            EC=2.
            IF(BETA.LT.(PI/2.)) THEN
               BH=2.*R(I)*SIN(BETA)/EC
            ELSE
               BH=2.*R(I)/EC
            ENDIF
            DE=BH*EC**.6667
            ZPE=DE*UD(I)/ZD
            SH=1.79*(3.*EC**2+4.)**.6667/(EC**2+4.)*PEC(I)**.5
            SC=VISD/DENSD(I)/ZD
            SHNU=SH*(PR/SC)**(1./3.)
            HFT=SHNU*CKVD/(2*R(I))
            IF (XD(I).LT.1.0) THEN
               HH=(1-XD(i))*HTH+XD(i)*HFT
            ELSE
               DDNORM2=(I-IXDREF)*DZ/(2.*R(IXDREF))
               IF(DDNORM2.LE.10) THEN
                  HH=(.5*HBBR+.5*HFT*(1.-DDNORM2/10.)**2)
                  IF(HH.LT.HBBR) HH=HBBR
               ELSE
                  HH=HBBR
               ENDIF
            ENDIF
            HSH=0.
            HHTS=0.
```

```
            HJAC=0.
            HCUL=0.
         ENDIF
         HS(I)=HSH
         HT(I)=HTH
         HCB(I)=HCBN
         HST(I)=HSTR
         HBB(I)=HBBR
         H(I)=HH
 9090    CONTINUE
C
C HEAT TRANSFER
C
         if(iflagon.eq.1)print*,'HTC-1'
         IF ((2*R(I)).LE.DZ) THEN
            SURFACE=4*PI*R(I)**2
         ELSE
            SURFACE=2*PI*R(I)*DZ
         ENDIF
         if(iflagon.eq.1)print*,'HTC-2'
         QDJ=HH*(TC(I)-TD(I))*SURFACE*NB
         QCV(I)=QDJ
         QCJ=-QDJ*ETA
         QCV2(I)=QCJ
C
         IF(I.EQ.N) GOTO 20
C
C  CHECK FOR FLOODING CONDITION
C
         IF(IFLAGON.EQ.1)PRINT*,'Flood Check'
         IF((UC(I)*UDZ(I)).GE.0) THEN
            PRINT*,'*************************************************'
            PRINT*,'******** FLOODING HAS OCCURRED IN COLOMN ********'
            PRINT*,'*************************************************'
            PRINT*,'ITERATION: ',ICOUNT
            PRINT*,'LOCATION    ',I,'/',N,'   Z= ',I*DZ,' m'
            PRINT*,'DISPERSED FLUID FLOW RATE  = ',MRD,' kg/sec'
            PRINT*,'CONTINUOUS FLUID FLOW RATE = ',MRC,' kg/sec'
            PRINT*,'DISPERSED FLUID VELOCITY   = ',UDZ(I),' m/sec'
            PRINT*,'RELATIVE FLUID VELOCITY    = ',UD(I),' m/sec'
            PRINT*,'CONTINUOUS FLUID VELOCITY  = ',UC(I),' m/sec'
            PRINT*,'HOLDUP                     = ',HUR(I)
            GOTO 9999
         ENDIF
C
C  CHECK FOR DISPERSED FLUID HOLDUP REVERSAL
C
         IF(IFLAGON.EQ.1)PRINT*,'Holdup Check'
         IF(HUR(I).GT.0.95) THEN
            PRINT*,'*************************************************'
            PRINT*,'******** HOLDUP RATIO HAS EXCEEDED 0.95  ********'
            PRINT*,'*************************************************'
            PRINT*,'ITERATION: ',ICOUNT
            PRINT*,'LOCATION    ',I,'/',N,'   Z= ',I*DZ/N,' m'
            PRINT*
            PRINT*,' DISPERSED FLUID FLOW RATE = ',MRD,' kg/sec'
            PRINT*,'CONTINUOUS FLUID FLOW RATE = ',MRC,' kg/sec'
            PRINT*
            PRINT*,' DISPERSED FLUID HOLDUP    = ',HUR(I)
            PRINT*
            GOTO 9999
         ENDIF
```

```
C
         DENSDJ=DENSD(I)
         DENSCJ=DENSC(I)
C
C  RUNGE-KUTTA SOLUTION
C   4th order
c
         IF(IFLAGON.EQ.1)PRINT*,'Runge-Kutta'
         DO 10 J=1,4
         IF(IFLAGON.EQ.1)PRINT*,'RK-K values'
            KP(J)=dz*(-MRD/DXAREA*UDPR-MRC/DXAREA*UCPR
     #           -(DENSCJ*(1-PHI)+DENSDJ*PHI)*G)
            KHD(J)=dz*(QDJ/DZ/MRD)
            KHC(J)=dz*(QCJ/DZ/MRC)
C
            IF(J.EQ.1.OR.J.EQ.2) THEN
               PPI=P(I)+.5*KP(J)/1000.
               HDI=HD(I)+.5*KHD(J)/1000.
               HCI=HC(I)+.5*KHC(J)/1000.
            ELSEIF(J.EQ.3.) THEN
               PPI=P(I)+KP(J)/1000.
               HDI=HD(I)+KHD(J)/1000.
               HCI=HC(I)+KHC(J)/1000.
            ELSEIF(J.GT.3) THEN
               GOTO 10
            ENDIF
C
         IF(IFLAGON.EQ.1)PRINT*,'RK-Dispersed density'
            IF(DFCAT.EQ.1) THEN
               PGIVEN=PPI
               HGIVEN=HDI
               CALL SIFLUID ('PH',DFTYPE,PGIVEN,HGIVEN,XDI,SDI,HDI,
     #              VDI,UDI,TDI,PDI,SDLI,HDLI,VDLI,UDLI,SDVI,
     #              HDVI,VDVI,UDVI)
               CALL QUALCHCK(XDI)
               DENSDJ=1./VDI
               IF(VDLI*VDVI.NE.0) THEN
                  RHODL=1./VDLI
                  RHODV=1./VDVI
               ENDIF
            ELSEIF(DFCAT.EQ.2) THEN
               CALL FLTHCAP(DFTYPE,TD(I),P(I),CPDI,RHOD1,XDI,ITRNG)
               CALL TRNGCHK(ITRNG,td(i))
               TDI=TD(I)+(HDI-HD(I))/CPDI
               CALL FLTHCAP(DFTYPE,TDI,P(I),CPD1,DENSDJ,XDI,ITRNG)
               CALL TRNGCHK(ITRNG,tdi)
            ENDIF
         IF(IFLAGON.EQ.1)PRINT*,'RK-Continuous density'
            IF(CFCAT.EQ.1) THEN
               PGIVEN=PPI
               HGIVEN=HCI
               CALL SIFLUID ('PH',CFTYPE,PGIVEN,HGIVEN,XCI,SCI,HCI,
     #              VCI,UCI,TCI,PCI,SCLI,HCLI,VCLI,UCLI,SCVI,
     #              HCVI,VCVI,UCVI)
               CALL QUALCHCK(XCI)
               DENSCJ=1./VCI
            ELSEIF(CFCAT.EQ.2) THEN
               CALL FLTHCAP(CFTYPE,TC(I),P(I),CPCI,RHOC1,XC1,ITRNG)
               CALL TRNGCHK(ITRNG,tc(i))
               TCI=TC(I)+(HCI-HC(I))/CPCI
               CALL FLTHCAP(CFTYPE,TCI,P(I),CPC1,DENSCJ,XC1,ITRNG)
               CALL TRNGCHK(ITRNG,tci)
```

```
            ENDIF
         IF(IFLAGON.EQ.1)PRINT*,'RK-Dispersed Properties'
            TGIVEN=TDI
            PGIVEN=PPI*1000.
            CALL PROPTY(DFTYPE,TGIVEN,PGIVEN,CKLD,CKVD,VISLD,VISVD,
     #           SIGD,ITRNG)
            CALL TRNGCHK(ITRNG,tgiven)
         IF(IFLAGON.EQ.1)PRINT*,'RK-Continuous Properties'
            TGIVEN=TCI
            PGIVEN=PPI*1000.
            CALL PROPTY(CFTYPE,TGIVEN,PGIVEN,CKC(0),CKC(1),
     #           VISC(0),VISC(1),SIGC,ITRNG)
            CALL TRNGCHK(ITRNG,tgiven)
            ALPHAC=CKC(IX)/DENSCJ/CPC
C
C BETA
C
            IF(IFLAGON.EQ.1)PRINT*,'RK-Beta'
            IF(XDI.GT.0.0.AND.XDI.LT.1.0) THEN
               CALL BBLMIX(RHODL,RHODV,XDI,BETA)
            ELSEIF(XDI.LE.0.0) THEN
               BETA=0.0
            ELSEIF(XDI.GE.1.0) THEN
               BETA=PI
            ENDIF
            IF(BETA.GT.PI) BETA=PI
c
c BUBBLE RADIUS
C
            IF(IFLAGON.EQ.1)PRINT*,'RK-radius'
            RJ=R(IREF)*(DENSD(IREF)/DENSDJ)**(1./3.)
            IF(WG(I).NE.0.) THEN
               VOLBUB=4./3.*PI*RJ**3+WG(I)*MB/DENSCG
               RJ=(VOLBUB*3./4./PI)**(1./3.)
            ENDIF
C
            IF(I.EQ.0.OR.I.EQ.IREF) THEN
               RECJ=DENSC(I)*2.*RJ*ABS(UD(I))/VISC(IX)
            ELSE
               RECJ=DENSC(I)*2.*RJ*ABS(UD(I-1))/VISC(IX)
            ENDIF
C
c     Tochitani et al. Coefficient
            IF(IFLAGON.EQ.1)PRINT*,'RK-Tochitani'
            HTJ=CKC(IX)/2/RJ*(ABS(UD(I))*2*RJ/ALPHAC)**(1./3.)
     #           *(0.466194077*(PI-BETA+.5*SIN(2.*BETA))**(2./3.))
            IF(XDI.LE.0.0)  THEN
c     Sideman Coefficient
               IF(IFLAGON.EQ.1)PRINT*,'RK-Sideman'
               HSJ=.00375*ABS(UD(I))*2*RJ/ALPHAC*(1./(1.+VISC(IX)/
     #              VISLD))*CKLD/2./RJ
c     Sideman/Tochitani series combination
               HHTSJ=HTJ*HSJ/(HTJ+HSJ)
c     Jacobs (Sideman) High Circulation with Decay
               IF(IFLAGON.EQ.1)PRINT*,'RK-Jacobs(Sideman)'
C               HJACJ=50.+.0085*RECJ*PRANDTLD**.7*CKC(IX)/2./RJ
               VKV=1.-1.45*(2.+3.*VISD/VISC(IX))/(1.+(DENSDJ*VISD/
     #              DENSCJ/VISC(IX))**.5)*RECJ**(-.5)
               HJACJ=(2.+1.13*(PEC(I)*VKV)**.5)*CKC(IX)/2./RJ
               HJTJ=HTJ*HJACJ/(HTJ+HJACJ)
               DDNORM=(I-IREF)*DZ/(2.*R(IREF))
               IF(DDNORM.LE.30) THEN
```

```
                  HHJ=HJTJ*(1.-DDNORM/30.)**2
C                  HHJ=(0.0*HHTSJ+1.0*HJACJ*(1.-DDNORM/10.)**2)
                  IF(HHJ.LT.HHTSJ) HHJ=HHTSJ
               ELSE
                  HHJ=HHTSJ
               ENDIF
               HCBNJ=0.
               HBBRJ=0.
            ELSE
c     Coban Modification
               HCBNJ=5.*CKVD/RJ*(1.-0.466194077*(PI-BETA
     #            +.5*SIN(2.*BETA))**(2./3.))
c     Brickman/Boehm Modification
c     Steinberger & Treybal Coefficient
               IF(IFLAGON.EQ.1)PRINT*,'RK-Brickman/Boehm'
               CALL H2OBETA(TC(I),CBeta)
               GR=9.81*CBeta*ABS(TD(I)-TC(I))*(2*RJ)**3/(VISC(IX)/
     #              DENSCJ)
               NU0=2.+.569*(GR*PR)**.25
               NUC=NU0+.347*RECJ**.62*PR**.31
               HSTRJ=NUC*CKC(IX)/2./RJ
               HBBRJ=5.*HSTRJ*CKVD/(5.*CKVD+R(I)*HSTRJ)*
     #              (1.-0.466194077*(PI-BETA+.5*SIN(2.*BETA))**(2./3.))
C
c     Transfer analogy (Fan/Tsuchiya, Ch 9)
c
               CALL DIFFUSEC(CFTYPE,DFTYPE,TC(I),P(I),ZD)
               IF(IFLAGON.EQ.1)PRINT*,'RK-Fan/Tsuchiya'
               EC=2.
               IF(BETA.LT.(PI/2.)) THEN
                  BH=2.*R(I)*SIN(BETA)/EC
               ELSE
                  BH=2.*R(I)/EC
               ENDIF
               DE=BH*EC**.6667
               ZPE=DE*UD(I)/ZD
               SH=1.79*(3.*EC**2+4.)**.6667/(EC**2+4.)
     #              *PEC(I)**.5
               SCJ=VISD/DENSDJ/ZD
               SHNU=SH*(PR/SCJ)**(1./3.)
               HFT=SHNU*CKVD/(2*R(I))
               HFT=SH*Ckvd/(2*R(I))
               IF (XD(I).LT.1.0) THEN
                  HHJ=(1-XD(i))*HTJ+XD(i)*HFT
               ELSE
                  IF(DDNORM2.LE.10) THEN
                     HHJ=(.5*HBBRJ+.5*HFT*(1.-DDNORM2/10.)**2)
                     IF(HHJ.LT.HBBRJ) HHJ=HBBRJ
                  ELSE
                     HHJ=HBBRJ
                  ENDIF
               ENDIF
               HSJ=0.
               HHTSJ=0.
               HJACJ=0.
               HCULJ=0.
            ENDIF
            IF(IFLAGON.EQ.1)PRINT*,'RK-HTC-1'
            IF((2*RJ).LE.DZ) THEN
               SURFJ=4*PI*RJ**2
            ELSE
               SURFJ=2*PI*RJ*DZ
```

```
            ENDIF
            QDJ=HHJ*(TCI-TDI)*SURFJ*NB
            QCJ=-QDJ*eta
C
            IF(IFLAGON.EQ.1)PRINT*,'Completed J=',j
 10      CONTINUE
         IF(IFLAGON.EQ.1)PRINT*,'End Runge-Kutta loop'
C
         KPC=(KP(1)+2*KP(2)+2*KP(3)+KP(4))/6.
         KHDC=(KHD(1)+2*KHD(2)+2*KHD(3)+KHD(4))/6.
         KHCC=(KHC(1)+2*KHC(2)+2*KHC(3)+KHC(4))/6.

C
         P(I+1)=P(I)+KPC/1000.
         HD(I+1)=HD(I)+KHDC/1000.
         HC(I+1)=HC(I)+ETA*KHCC/1000.
C
         IF(DFCAT.EQ.1) THEN
            PGIVEN=P(I+1)
            HGIVEN=HD(I+1)
            CALL SIFLUID('PH',DFTYPE,PGIVEN,HGIVEN,XD(I+1),SD,
      #          HD(I+1),VD,UID,TD(I+1),PD,SDL,HDL,VDL,UDL,SDV,
      #          HDV,VDV,UDV)
            CALL QUALCHCK(XD(I+1))
            DENSD(I+1)=1./VD
            IF(VDL*VDV.NE.0) THEN
               RHODL=1./VDL
               RHODV=1./VDV
            ENDIF
         ELSEIF(DFCAT.EQ.2) THEN
            CALL FLTHCAP(DFTYPE,TD(I),P(I+1),CPD,DENSD(I+1),XD(I+1),
      #          ITRNG)
            CALL TRNGCHK(ITRNG,td(i))
            TD(I+1)=TD(I)+(HD(I+1)-HD(I))*1000./CPD
            CALL FLTHCAP(DFTYPE,TD(I+1),P(I+1),CPD,DENSD(I+1),
      #          XD(I+1),ITRNG)
            CALL TRNGCHK(ITRNG,td(i+1))
         ENDIF
C
         IF(CFCAT.EQ.1) THEN
            PGIVEN=P(I+1)
            HGIVEN=HC(I+1)
            CALL SIFLUID('PH',CFTYPE,PGIVEN,HGIVEN,XC(I+1),SC,HC(I+1)
      #          ,VC,UIC,TC(I+1),PC,SCL,HCL,VCL,UCL,SCV,HCV,VCV,UCV)
            CALL QUALCHCK(XC(I+1))
            DENSC(I+1)=1./VC
         ELSEIF(CFCAT.EQ.2) THEN
            CALL FLTHCAP(CFTYPE,TC(I),P(I+1),CPC,DENSC(I+1),XC(I+1),
      #          ITRNG)
            CALL TRNGCHK(ITRNG,tc(i))
            TC(I+1)=TC(I)+(HC(I+1)-HC(I))*1000./CPC
            CALL FLTHCAP(CFTYPE,TC(I+1),P(I+1),CPC,DENSC(I+1),XC(I+1),
      #          ITRNG)
            CALL TRNGCHK(ITRNG,tc(i+1))
         ENDIF
         IF(XC(I+1).NE.CQUAL) THEN
            WRITE(7,*)'CONTINUOUS FLUID QUALITY CHANGE OF STATE'
            WRITE(7,*)'  CONT FLUID QUALITY= ',XC(I+1)
            WRITE(7,*)'TOO MUCH HEAT MAY BE TRANSFERING'
         ENDIF
C
C
```

```
         IF(IAF.EQ.1) ACCELF=ICOUNT
         IF(TCNFIG.EQ.1) THEN
            IF(ICNFIG.EQ.1) THEN
               IF(TC(I+1).GT.TEMPC) THEN
                  IF(ICOUNT.GT.2) TUPPER=TC(0)+(TUPPER-TC(0))/ACCELF
                  ITLIM=1
                  GOTO 11
               ENDIF
            ELSE
               IF(TD(I+1).LT.TEMPD) THEN
                  IF(ICOUNT.GT.2) TLOWER=TD(0)-(TD(0)-TLOWER)/ACCELF
                  ITLIM=1
                  GOTO 11
               ENDIF
            ENDIF
         ELSE
            IF(ICNFIG.EQ.1) THEN
               IF(TC(I+1).LT.TEMPC) THEN
                  IF(ICOUNT.GT.2) TLOWER=TC(0)-(TC(0)-TLOWER)/ACCELF
                  ITLIM=1
                  GOTO 11
               ENDIF
            ELSE
               IF(TD(I+1).GT.TEMPD) THEN
                  IF(ICOUNT.GT.2) TUPPER=TD(0)+(TUPPER-TD(0))/ACCELF
                  ITLIM=1
                  GOTO 11
               ENDIF
            ENDIF
         ENDIF
C
 20   CONTINUE
      ITLIM=0
      richard=1.0
C
C     END DIFFERENTIAL EQUATION SOLUTION RUN
C
      IF(IFLAGON.EQ.1)PRINT*,'END MARCHING:TESTING FOR CONVERGENCE'
C     TEST FOR CORRECT BOUNDARY CONDITION STARTING TEMPERATURE
C
      IF(ICNFIG.EQ.1) THEN
         IF (CFCAT.EQ.1) THEN
            IF(ABS(HC(N)-HCZ).LE.HTOL) THEN
               CALL PRNTRSLT(ICOUNT,IPASS)
               GOTO 9999
            ENDIF
         ELSEIF(CFCAT.EQ.2) THEN
            IF(ABS(TC(N)-TEMPC).LE.TTOL) THEN
               CALL PRNTRSLT(ICOUNT,IPASS)
               GOTO 9999
            ENDIF
         ENDIF
      ELSE
         IF(DFCAT.EQ.1) THEN
            IF(ABS(HD(N)-HDZ).LE.HTOL) THEN
               CALL PRNTRSLT(ICOUNT,IPASS)
               GOTO 9999
            ENDIF
         ELSEIF(DFCAT.EQ.2) THEN
            IF(ABS(TD(N)-TEMPD).LE.TTOL) THEN
               CALL PRNTRSLT(ICOUNT,IPASS)
               GOTO 9999
```

```
            ENDIF
         ENDIF
      ENDIF
C
      IF(TUPPER-TLOWER.LE.TTOL) THEN
         CALL PRNTRSLT(ICOUNT,IPASS)
         GOTO 9999
      ENDIF
C
C  ADJUST STARTING TEMPERATURE GUESS
      IF(IFLAGON.EQ.1)PRINT*,'ADJUSTING TEMPERATURE GUESS'
C
 9997 IF(IAF.EQ.1) ACCELF=ICOUNT
      IF(ICNFIG.EQ.1) THEN
         IF(HC(N).GT.HCZ) THEN
            IF(ICOUNT.GT.3) TUPPER=TC(0)+(TUPPER-TC(0))/ACCELF
         ELSE
            IF(ICOUNT.GT.3) TLOWER=TC(0)-(TC(0)-TLOWER)/ACCELF
         ENDIF
      ELSE
         IF(HD(N).GT.HDZ) THEN
            IF(ICOUNT.GT.3) TUPPER=TD(0)+(TUPPER-TD(0))/ACCELF
         ELSE
            IF(ICOUNT.GT.3) TLOWER=TD(0)-(TD(0)-TLOWER)/ACCELF
         ENDIF
      ENDIF
      GOTO 11
C
C
 9999 IF(IPASS.NE.1) GOTO 9998
C
C     LOCATION OF FUTURE 2nd PASS ITERATION WITH IMPROVED INITIAL
C     GUESSES
C
 9998 STOP
      END
c_______________________________________________
      SUBROUTINE FLUIDCAT(FTYPE,FCAT,R)
      CHARACTER*6 FTYPE
      FCAT=3
      R=0.0
C
      IF(FTYPE.EQ.'H2O   ') THEN
         FCAT=1
         R=461.48
      ELSEIF(FTYPE.EQ.'PENT  ') THEN
         FCAT=1
         R=115.24
      ELSEIF(FTYPE.EQ.'BUT   ') THEN
         FCAT=1
         R=143.05
      ELSEIF(FTYPE.EQ.'ISOB  ') THEN
         FCAT=1
         R=143.05
      ELSEIF(FTYPE.EQ.'ISOP  ') THEN
         FCAT=1
         R=115.24
      ELSEIF(FTYPE.EQ.'HEP   ') THEN
         FCAT=1
      ELSEIF(FTYPE.EQ.'ETHYL ') THEN
         FCAT=1
      ELSEIF(FTYPE.EQ.'METH  ') THEN
```

```
    FCAT=1
ELSEIF(FTYPE.EQ.'F11   ') THEN
    FCAT=1
ELSEIF(FTYPE.EQ.'F12   ') THEN
    FCAT=1
ELSEIF(FTYPE.EQ.'F13   ') THEN
    FCAT=1
ELSEIF(FTYPE.EQ.'F22   ') THEN
    FCAT=1
ELSEIF(FTYPE.EQ.'F23   ') THEN
    FCAT=1
ELSEIF(FTYPE.EQ.'F113  ') THEN
    FCAT=1
ELSEIF(FTYPE.EQ.'F114  ') THEN
    FCAT=1
ELSEIF(FTYPE.EQ.'TH44  ') THEN
    FCAT=2
ELSEIF(FTYPE.EQ.'TH55  ') THEN
    FCAT=2
ELSEIF(FTYPE.EQ.'TH60  ') THEN
    FCAT=2
ELSEIF(FTYPE.EQ.'TH66  ') THEN
    FCAT=2
ELSEIF(FTYPE.EQ.'TH88  ') THEN
    FCAT=2
ELSEIF(FTYPE.EQ.'CFAF  ') THEN
    FCAT=2
ELSEIF(FTYPE.EQ.'CFFG  ') THEN
    FCAT=2
ELSEIF(FTYPE.EQ.'CFHTF ') THEN
    FCAT=2
ELSEIF(FTYPE.EQ.'SYLTH8') THEN
    FCAT=2
ELSEIF(FTYPE.EQ.'THM550') THEN
    FCAT=2
ELSEIF(FTYPE.EQ.'THM600') THEN
    FCAT=2
ELSEIF(FTYPE.EQ.'THM800') THEN
    FCAT=2
ELSEIF(FTYPE.EQ.'DTSR1 ') THEN
    FCAT=2
ELSEIF(FTYPE.EQ.'DT4000') THEN
    FCAT=2
ELSEIF(FTYPE.EQ.'DT4030') THEN
    FCAT=2
ELSEIF(FTYPE.EQ.'DT4040') THEN
    FCAT=2
ELSEIF(FTYPE.EQ.'DT4050') THEN
    FCAT=2
ELSEIF(FTYPE.EQ.'DT4060') THEN
    FCAT=2
ELSEIF(FTYPE.EQ.'DFHD30') THEN
    FCAT=2
ELSEIF(FTYPE.EQ.'DFHD40') THEN
    FCAT=2
ELSEIF(FTYPE.EQ.'DFHD50') THEN
    FCAT=2
ELSEIF(FTYPE.EQ.'DFHD60') THEN
    FCAT=2
ELSEIF(FTYPE.EQ.'DWFHD ') THEN
    FCAT=2
ELSEIF(FTYPE.EQ.'DWFRST') THEN
```

```
         FCAT=2
      ELSEIF(FTYPE.EQ.'PARANF') THEN
         FCAT=2
      ELSEIF(FTYPE.EQ.'PARAHE') THEN
         FCAT=2
      ELSEIF(FTYPE.EQ.'MTHVP1') THEN
         FCAT=2
      ELSEIF(FTYPE.EQ.'THRMX ') THEN
         FCAT=2
      ELSEIF(FTYPE.EQ.'AIR   ') THEN
         FCAT=2
         R=287.03
      ELSEIF(FTYPE.EQ.'CO2   ') THEN
         FCAT=2
         R=188.91
      ELSEIF(FTYPE.EQ.'H2    ') THEN
         FCAT=2
      ELSEIF(FTYPE.EQ.'N2    ') THEN
         FCAT=2
      ELSEIF(FTYPE.EQ.'O2    ') THEN
         FCAT=2
      ELSEIF(FTYPE.EQ.'NH3   ') THEN
         FCAT=2
      ELSEIF(FTYPE.EQ.'MS20  ') THEN
         FCAT=2
      ELSEIF(FTYPE.EQ.'TRAN  ') THEN
         FCAT=2
      ELSEIF(FTYPE.EQ.'TM1   ') THEN
         FCAT=2
      ELSEIF(FTYPE.EQ.'VM4   ') THEN
         FCAT=2
      ELSEIF(FTYPE.EQ.'KHF22 ') THEN
         FCAT=2
      ELSEIF(FTYPE.EQ.'ENG   ') THEN
         FCAT=2
      ELSEIF(FTYPE.EQ.'ETHYLG') THEN
         FCAT=2
      ELSEIF(FTYPE.EQ.'GLYCRN') THEN
         FCAT=2
      ELSEIF(FTYPE.EQ.'HG    ') THEN
         FCAT=2
      ENDIF
C
 100  RETURN
      END
C___________________________________________________________
      SUBROUTINE DIFFUSEC(CFTYPE,DFTYPE,TC,P,D)
      REAL KTE(81),OMEGA(81)
      REAL KTE12,OMEGA12,M1,M2
      CHARACTER*6,CFTYPE
      CHARACTER*6,DFTYPE
      DATA(KTE(I),I=1,81)/.3,.35,.4,.45,.5,.55,.6,.65,.7,.75,.8,.85,.9,
     #.95,1.,1.05,1.1,1.15,1.2,1.25,1.3,1.35,1.4,1.45,1.5,1.55,1.6,
     #1.65,1.7,1.75,1.8,1.85,1.9,1.95,2.,2.1,2.2,2.3,2.4,2.5,2.6,2.7,
     #2.8,2.9,3.,3.1,3.2,3.3,3.4,3.5,3.6,3.7,3.8,3.9,4.,4.1,4.2,4.3,
     #4.4,4.5,4.6,4.7,4.8,4.9,5.,6.,7.,8.,9.,10.,20.,30.,40.,50.,60.,
     #70.,80.,90.,100.,200.,300./
      DATA(OMEGA(I),I=1,81)/2.662,2.476,2.318,2.184,2.066,1.966,1.877,
     #1.798,1.729,1.667,1.612,1.562,1.517,1.476,1.439,1.406,1.375,
     #1.346,1.32,1.296,1.273,1.253,1.233,1.215,1.198,1.182,1.167,1.153,
     #1.14,1.128,1.116,1.105,1.094,1.084,1.075,1.057,1.041,1.026,1.012,
     #.9996,.9878,.977,.9672,.9576,.949,.9406,.9328,.9256,.9186,.912,
```

```
#.9058,.8998,.8942,.8888,.8836,.8788,.874,.8694,.8652,.861,.8568,
#.853,.8492,.8456,.8422,.8124,.7896,.7712,.7556,.7424,.664,.6232,
#.596,.5756,.5596,.5464,.5352,.5256,.513,.4644,.436/
 D=0.0
 IF(CFTYPE.EQ.'ARGON ') THEN
     SIG1=3.542
     EK1=93.3
     M1=39.948
 ELSEIF(CFTYPE.EQ.'HEL   ') THEN
     SIG1=2.551
     EK1=10.22
     M1=4.0026
 ELSEIF(CFTYPE.EQ.'KRYP  ') THEN
     SIG1=3.655
     EK1=178.9
     M1=83.8
 ELSEIF(CFTYPE.EQ.'AIR   ') THEN
     SIG1=3.711
     EK1=78.6
     M1=29.0
 ELSEIF(CFTYPE.EQ.'METH  ') THEN
     SIG1=3.758
     EK1=148.6
     M1=16.0426
 ELSEIF(CFTYPE.EQ.'CO2   ') THEN
     SIG1=3.941
     EK1=195.2
     M1=44.0098
 ELSEIF(CFTYPE.EQ.'ETHYL ') THEN
     SIG1=4.163
     EK1=224.7
     M1=28.0536
 ELSEIF(CFTYPE.EQ.'ETH   ') THEN
     SIG1=4.443
     EK1=215.7
     M1=30.0694
 ELSEIF(CFTYPE.EQ.'PROPYL') THEN
     SIG1=4.678
     EK1=298.9
     M1=41.0725
 ELSEIF(CFTYPE.EQ.'BUT   ') THEN
     SIG1=4.687
     EK1=531.4
     M1=58.123
 ELSEIF(CFTYPE.EQ.'ISOB  ') THEN
     SIG1=5.278
     EK1=330.1
     M1=58.123
 ELSEIF(CFTYPE.EQ.'PENT  ') THEN
     SIG1=5.784
     EK1=341.1
     M1=72.1498
 ELSEIF(CFTYPE.EQ.'HEX   ') THEN
     SIG1=5.949
     EK1=399.3
     M1=86.1766
 ELSEIF(CFTYPE.EQ.'H2    ') THEN
     SIG1=2.827
     EK1=59.7
     M1=2.0158
 ELSEIF(CFTYPE.EQ.'H2O   ') THEN
     SIG1=2.641
```

```
         EK1=809.1
         M1=18.0152
      ELSEIF(CFTYPE.EQ.'N2    ') THEN
         SIG1=3.798
         EK1=71.4
         M1=28.0134
      ELSEIF(CFTYPE.EQ.'O2    ') THEN
         SIG1=3.467
         EK1=106.7
         M1=31.9988
      ELSEIF(CFTYPE.EQ.'H2S   ') THEN
         SIG1=3.623
         EK1=301.1
         M1=34.0818
      ENDIF
C
C
      IF(DFTYPE.EQ.'ARGON ') THEN
         SIG2=3.542
         EK2=93.3
         M2=39.948
      ELSEIF(DFTYPE.EQ.'HEL   ') THEN
         SIG2=2.551
         EK2=10.22
         M2=4.0026
      ELSEIF(DFTYPE.EQ.'KRYP  ') THEN
         SIG2=3.655
         EK2=178.9
         M2=83.8
      ELSEIF(DFTYPE.EQ.'AIR   ') THEN
         SIG2=3.711
         EK2=78.6
         M2=29.0
      ELSEIF(DFTYPE.EQ.'METH  ') THEN
         SIG2=3.758
         EK2=148.6
         M2=16.0426
      ELSEIF(DFTYPE.EQ.'CO2   ') THEN
         SIG2=3.941
         EK2=195.2
         M2=44.0098
      ELSEIF(DFTYPE.EQ.'ETHYL ') THEN
         SIG2=4.163
         EK2=224.7
         M2=28.0536
      ELSEIF(DFTYPE.EQ.'ETH   ') THEN
         SIG2=4.443
         EK2=215.7
         M2=30.0694
      ELSEIF(DFTYPE.EQ.'PROPYL') THEN
         SIG2=4.678
         EK2=298.9
         M2=41.0725
      ELSEIF(DFTYPE.EQ.'BUT   ') THEN
         SIG2=4.687
         EK2=531.4
         M2=58.123
      ELSEIF(DFTYPE.EQ.'ISOB  ') THEN
         SIG2=5.278
         EK2=330.1
         M2=58.123
      ELSEIF(DFTYPE.EQ.'PENT  ') THEN
```

```
         SIG2=5.784
         EK2=341.1
         M2=72.1498
      ELSEIF(DFTYPE.EQ.'HEX  ') THEN
         SIG2=5.949
         EK2=399.3
         M2=86.1766
      ELSEIF(DFTYPE.EQ.'H2   ') THEN
         SIG2=2.827
         EK2=59.7
         M2=2.0158
      ELSEIF(DFTYPE.EQ.'H2O  ') THEN
         SIG2=2.641
         EK2=809.1
         M2=18.0152
      ELSEIF(DFTYPE.EQ.'N2   ') THEN
         SIG2=3.798
         EK2=71.4
         M2=28.0134
      ELSEIF(DFTYPE.EQ.'O2   ') THEN
         SIG2=3.467
         EK2=106.7
         M2=31.9988
      ELSEIF(DFTYPE.EQ.'H2S  ') THEN
         SIG2=3.623
         EK2=301.1
         M2=34.0818
      ENDIF
C
      IF((EK1*EK2).LE.0.0) GOTO 100
      IF((SIG1*SIG2).EQ.0.0) GOTO 100
      TK=TC+273.15
      SIG12=.5*(SIG1+SIG2)
      KTE12=1./(SQRT(EK1*EK2)/TK)
      DO 10 I=80,1,-1
         IF(KTE(I).LE.KTE12) THEN
            OMEGA12=(KTE12-KTE(I))/(KTE(I+1)-KTE(I))*
     #           (OMEGA(I+1)-OMEGA(I))+OMEGA(I)
            GOTO 20
         ENDIF
 10   CONTINUE
      GOTO 100
C
C    CHAPMAN-ENSKOG THEORY FOR DIFFUSION COEFFICIENT
C
 20   D=1.86E-03*TK**1.5*(1./M1+1./M2)/(P/101.325)/SIG12**2/OMEGA12
      D=D/10000.
C
 100  RETURN
      END
C_______________________________________________
      SUBROUTINE PPRNT
      REAL  UD(0:2001),UC(0:2001),TD(0:2001),TC(0:2001),
     #      HUR(0:2001),DENSD(0:2001),DENSC(0:2001),HD(0:2001),
     #      HC(0:2001),R(0:2001),QCV(0:2001),XD(0:2001),P(0:2001),
     #      XC(0:2001),CKC(0:1),VISC(0:1),BET(0:2001),H(0:2001),
     #      HS(0:2001),HT(0:2001),ALPHC(0:2001),VISCD(0:2001),
     #      RYNLD(0:2001),COEFD(0:2001),PEC(0:2001),HCB(0:2001),
     #      CPPC(0:2001),CPPD(0:2001),CKKC(0:2001),CKKD(0:2001),
     #      WG(0:2001),HM(0:2001),DIFF(0:2001),ALPHD(0:2001),
     #      VISCC(0:2001),QCV2(0:2001),HSTAG(0:2001),BPOP(0:2001),
     #      HBB(0:2001),HST(0:2001),UDZ(0:2001)
```

```
      COMMON/HHH/H,HS,HT,HCB,HBB,HST
      COMMON/ARRY/UDZ,UD,UC,TD,TC,P,HUR,DENSD,DENSC,HD,HC,R,QCV,
     #     DZ,XD,BET,CPPC,CPPD,CKKD,CKKC,PEC,VISCD,VISC,RYNLD,
     #     BPOP,VISCC,COEFD,WG,HM,DIFF,QCV2,HSTAG
      COMMON/SCLR/MRD,MRC,TEMPD,TEMPC,R0,PRESS,CQUAL,ICNFIG,TCNFIG,
     #     N,DNZAREA,DXVOLUME,DXLENGTH,DXRADIUS,DFTYPE,CFTYPE
      REAL MRD,MRC
      INTEGER TCNFIG
      CHARACTER*6,DFTYPE
      CHARACTER*6,CFTYPE
      WRITE(7,*)' '
      WRITE(7,*)'Input Values'
      WRITE(7,7001) DXLENGTH,DXRADIUS,PRESS
 7001 FORMAT(/,'     Heat Exchanger Length:  ',F7.2,' meters',/,
     #     '     Heat Exchanger Radius:  ',F7.2,' meters',/,
     #     'Nominal Operating Pressure:  ',F7.2,' kPa')
      WRITE(7,7002)DFTYPE,CFTYPE,MRD,MRC,TEMPD,TEMPC
 7002 FORMAT(/,18X,'Dispersed Fluid',5X,'Continuous Fluid',/,
     #     '          Fluid:',7X,A6,15X,A6,/,
     #     'Mass Flow Rate:',5X,F8.4,13x,F8.4,'   kg/sec',/,
     #     '    Inlet Temp:',3X,F8.2,13x,F8.2,'     deg C')
      WRITE(7,7003)R0,CQUAL
 7003 FORMAT(/,'Disp. Fld. Droplet/Bubble Radius:  ',F7.6,' meters',/,
     #     '         Continuous Fluid Quality:  ',F2.0,/)
      RETURN
      END
C______________________________________________
      SUBROUTINE PPRINT(ICOUNT,IPASS)
      REAL  UD(0:2001),UC(0:2001),TD(0:2001),TC(0:2001),
     #     HUR(0:2001),DENSD(0:2001),DENSC(0:2001),HD(0:2001),
     #     HC(0:2001),R(0:2001),QCV(0:2001),XD(0:2001),P(0:2001),
     #     XC(0:2001),CKC(0:1),VISC(0:1),BET(0:2001),H(0:2001),
     #     HS(0:2001),HT(0:2001),ALPHC(0:2001),VISCD(0:2001),
     #     RYNLD(0:2001),COEFD(0:2001),PEC(0:2001),HCB(0:2001),
     #     CPPC(0:2001),CPPD(0:2001),CKKC(0:2001),CKKD(0:2001),
     #     WG(0:2001),HM(0:2001),DIFF(0:2001),ALPHD(0:2001),
     #     VISCC(0:2001),QCV2(0:2001),HSTAG(0:2001),BPOP(0:2001),
     #     HBB(0:2001),HST(0:2001),UDZ(0:2001)
      COMMON/HHH/H,HS,HT,HCB,HBB,HST
      COMMON/ARRY/UDZ,UD,UC,TD,TC,P,HUR,DENSD,DENSC,HD,HC,R,QCV,
     #     DZ,XD,BET,CPPC,CPPD,CKKD,CKKC,PEC,VISCD,VISC,RYNLD,
     #     BPOP,VISCC,COEFD,WG,HM,DIFF,QCV2,HSTAG
      COMMON/SCLR/MRD,MRC,TEMPD,TEMPC,R0,PRESS,CQUAL,ICNFIG,TCNFIG,
     #     N,DNZAREA,DXVOLUME,DXLENGTH,DXRADIUS,DFTYPE,CFTYPE
      COMMON/SCLR2/ETA
      REAL MRD,MRC,LMTD,MCPMIN
      INTEGER TCNFIG
        WRITE(7,*)'DCHX Parameter Profile'
        QXFRD=0.0
        QXFRC=0.0
        PRANDTLD=0.0
        PRANDTLC=0.0
        DO 10 I=0,N
           QXFRD=QXFRD+QCV(I)/ETA
           QXFRC=QXFRC+QCV2(I)
           PRANDTLD=PRANDTLD+CPPD(I)*VISCD(I)/CKKD(I)
           PRANDTLC=PRANDTLC+CPPC(I)*VISCC(I)/CKKC(I)
 10     CONTINUE
        PRANDTLD=PRANDTLD/N
        PRANDTLC=PRANDTLC/N
        LMTD=(ABS(TD(N)-TC(N))-ABS(TD(0)-TC(0)))/LOG(ABS(TD(N)-TC(N))/
```

```
     #     ABS(TD(0)-TC(0)))
      UUVD=ABS(QXFRD)/DXVOLUME/LMTD
      UUVC=ABS(QXFRC)/DXVOLUME/LMTD
      IF(XD(0).NE.XD(N)) GOTO 20
      IF(ICNFIG.EQ.1) THEN
         CPPDS=CPPD(0)
         CPPCS=CPPC(N)
      ELSE
         CPPDS=CPPD(N)
         CPPCS=CPPC(0)
      ENDIF
      IF((ABS(MRD)*CPPDS).LT.(ABS(MRC)*CPPCS)) THEN
         MCPMIN=ABS(MRD)*CPPDS
      ELSE
         MCPMIN=ABS(MRC)*CPPCS
      ENDIF
      DTMAX=ABS(TEMPC-TEMPD)
      QIDEAL=MCPMIN*DTMAX
      eD=ABS(QXFRD)/QIDEAL
      eC=ABS(QXFRC)/QIDEAL
      DTDT=abs((TD(0)-TC(0))/(TD(N)-TC(N)))
      DUOC=(1.-LMTD/DTMAX)*(1.-ABS(LOG(DTDT))*(LMTD/DTMAX))
      WRITE(7,*)
      WRITE(7,7010)ICNFIG,TCNFIG,QXFRD,QXFRC,TD(0),TD(N),TC(0),
     #     TC(N),LMTD,DTDT,DUOC,UUVD,UUVC,eD,eC,qideal,
     #     PRANDTLD,PRANDTLC
      print*,qxfrd
      GOTO 30
20    WRITE(7,*)
      WRITE(7,7020)ICNFIG,TCNFIG,QXFRD,QXFRC,TD(0),
     #     TD(N),TC(0),TC(N),LMTD,DTDT,DUOC,UUVD,UUVC,
     #     PRANDTLD,PRANDTLC
      print*,qxfrd
7010  FORMAT(/,' Fluid Flow Configuration:  ',I1,/,
     #        ' Temperature Configuration:  ',I1,//,
     #        'Total Device Heat Trans  D:  ',F10.0,' Watts',/,
     #        'Total Device Heat Trans  C:  ',F10.0,' Watts',//,
     #        'Dispersed Fluid Temp @ vz=0:  ',F6.1,' deg C',/,
     #        'Dispersed Fluid Temp @ z=N:  ',F6.1,' deg C',/,
     #        'Continuous Fluid Temp @z=0:  ',F6.1,' deg C',/,
     #        'Continuous Fluid Temp @z=N:  ',F6.1,' deg C',//,
     #        '  Log Mean Temp Difference:  ',F7.2, '   deg C',/,
     #        '               Parallelism:  ',f8.4,/,
     #        '  Optimal Fluid Usage Ind.:  ',f8.4,/,
     #        ' Volumetric Trans Coeff  D:  ',F12.4,' W/m3 K',/,
     #        ' Volumetric Trans Coeff  C:  ',F12.4,' W/m3 K',//,
     #        '     e-NTU Effectiveness D:  ',F8.3,/,
     #        '     e-NTU Effectiveness C:  ',F8.3,/,
     #        '        Max Ideal Transfer:  ',f10.0,//,
     #        '             Prandtl # (D):  ',f10.4,/,
     #        '             Prandtl # (C):  ',f10.4,/)
7020  FORMAT(/,' Fluid Flow Configuration:  ',I1,/,
     #        ' Temperature Configuration:  ',I1,/,
     #        'Total Device Heat Trans  D:  ',F10.0,' Watts',/,
     #        'Total Device Heat Trans  C:  ',F10.0,' Watts',//,
     #        'Dispersed Fluid Temp @ z=0:  ',F6.1,' deg C',/,
     #        'Dispersed Fluid Temp @ z=N:  ',F6.1,' deg C',/,
     #        'Continuous Fluid Temp @z=0:  ',F6.1,' deg C',/,
     #        'Continuous Fluid Temp @z=N:  ',F6.1,' deg C',//,
     #        '  Log Mean Temp Difference:  ',F7.2,'   deg C',/,
     #        '               Parallelism:  ',f8.4,/,
     #        '  Optimal Fluid Usage Ind.:  ',f8.4,/,
```

```
     #        ' Volumetric Trans Coeff  D:  ',F12.4,' W/m3 K',/,
     #        ' Volumetric Trans Coeff  C:  ',F12.4,' W/m3 K',//,
     #        '             Prandtl # (D):  ',f10.4,/,
     #        '             Prandtl # (C):  ',f10.4,/)
 7025  FORMAT('Adjusted Dispersed Fluid Nozzle Area: ',F6.3,//)
 30    WRITE(7,7025)DNZAREA
       IF(QXFRD.GT.QXFRC)THEN
          WRITE(7,*)'HEAT TRANSFER IS FROM CONTINUOUS FLUID TO
DISPERSE
     #D FLUID'
       ELSE
          WRITE(7,*)'HEAT TRANSFER IS FROM DISPERSED FLUID TO
CONTINUOU
     #S FLUID'
          ENDIF
          WRITE(7,7011)ICOUNT
 7011     FORMAT(//,'CONVERGENCE OCCURRED IN ',I3,' ITERATIONS')
       RETURN
       END
C_____________________________________________________
      SUBROUTINE PRNTRSLT(ICOUNT,IPASS)
      REAL  UD(0:2001),UC(0:2001),TD(0:2001),TC(0:2001),
     #   HUR(0:2001),DENSD(0:2001),DENSC(0:2001),HD(0:2001),
     #   HC(0:2001),R(0:2001),QCV(0:2001),XD(0:2001),p(0:2001),
     #   XC(0:2001),CKC(0:1),VISC(0:1),BET(0:2001),H(0:2001),
     #   HS(0:2001),HT(0:2001),ALPHC(0:2001),VISCD(0:2001),
     #   RYNLD(0:2001),COEFD(0:2001),PEC(0:2001),HCB(0:2001),
     #   CPPC(0:2001),CPPD(0:2001),CKKD(0:2001),CKKC(0:2001),
     #   WG(0:2001),HM(0:2001),DIFF(0:2001),VISCC(0:2001),
     #   QCV2(0:2001),HSTAG(0:2001),BPOP(0:2001),
     #   HBB(0:2001),HST(0:2001),UDZ(0:2001)
      REAL MRD,MRC,LMTD
      INTEGER TCNFIG
      COMMON/HHH/H,HS,HT,HCB,HBB,HST
      COMMON/ARRY/UDZ,UD,UC,TD,TC,P,HUR,DENSD,DENSC,HD,HC,R,QCV,
     #   DZ,XD,BET,CPPC,CPPD,CKKD,CKKC,PEC,VISCD,VISC,RYNLD,
     #   BPOP,VISCC,COEFD,WG,HM,DIFF,QCV2,HSTAG
      COMMON/SCLR/MRD,MRC,TEMPD,TEMPC,RO,PRESS,CQUAL,ICNFIG,TCNFIG,
     #   N,DNZAREA,DXVOLUME,DXLENGTH,DXRADIUS,DFTYPE,CFTYPE
      COMMON/SCLR2/ETA
      CHARACTER*6,CFTYPE
      CHARACTER*6,DFTYPE
      PRINT*,'PRINTING RESULTS'
      CALL PPRINT(ICOUNT,IPASS)
      WRITE(2,*)'DISPERSED FLUID RESULTS'
      WRITE(3,*)'CONTINUOUS FLUID RESULTS'
      WRITE(4,*)'FLUID AND HEAT TRANSFER VALUES'
      WRITE(5,*)'DISPERSED FLUID PARAMETERS'
      WRITE(6,*)'CONTINUOUS FLUID PARAMETERS'
      WRITE(9,*)'DISPERSED DROPLET/BUBBLE INFO'
      DO 1 I=2,6
 1    WRITE(I,*)'DCHX Configuration = ',ICNFIG
      WRITE(2,*)'Dispersed Fluid:  ',DFTYPE
      WRITE(5,*)'Dispersed Fluid:  ',DFTYPE
      WRITE(3,*)'Continuous Fluid:  ',CFTYPE
      WRITE(6,*)'Continuous Fluid:  ',CFTYPE
      WRITE(9,*)'Dispersed Fluid:  ',DFTYPE
      WRITE(2,*)'Idx   Loc   Vel(a) Vel(r)  Temp Holdup HtXfer  Qual  D-
     #rad    Pop'
      WRITE(2,*)' i     Z     Udz     Ud     Td    Phi    QcvD    Xd
     #r      nB'
      WRITE(2,*)'       m     m/s    m/s     C             W      -
```

```
      #m'
       WRITE(3,*)'Idx    Loc    Vel       Temp   HtXfer          Press'
       WRITE(3,*)' i      Z      Uc        Tc     QcvC    1-Phi     P'
       WRITE(3,*)'              m/s        C       W               kPa'
       WRITE(4,*)'             Reynolds   Peclect   Drag       Ht Xfer'
       WRITE(4,*)'Idx    Loc    Number    Number    Coeff       Coeff  '
       WRITE(4,*)'        Z      Re        Pe        Cd           h    '
       WRITE(4,*)'        m                                     W/m2 K '
       WRITE(5,*)'                                       Thermal    Cond'
       WRITE(5,*)'Idx    Loc    Density     Viscosity    Capacit    Coeff
Ent
      #halpy'
       WRITE(5,*)' i      Z       rho          mu          Cp          k

      #Hd'
       WRITE(5,*)'        m      kg/m3        Pa-s        J/kg K    W/m K
J
      #/kg K'
       WRITE(6,*)'                                       Thermal    Cond'
       WRITE(6,*)'Idx    Loc    Density     Viscosity    Capacit    Coeff
Ent
      #halpy'
       WRITE(6,*)' i      Z       rho          mu          Cp          k

      #Hd'
       WRITE(6,*)'        m      kg/m3        Pa-s        J/kg K    W/m K
J
      #/kg K'
       WRITE(8,*)'Idx    Loc    Mass Rat    M Xfer Coeff'
       WRITE(8,*)' i      Z        Wg           hM'
       WRITE(8,*)'        m                 kg/sec m2'
       WRITE(9,*)' i      Z        hT       hS       hTS        hST      hBB
      #     h'
C
       DO 10 I=N,0,-1
          Z=I*DZ
       WRITE(2,1002)I,Z,UDZ(I),UD(I),TD(I),HUR(I),QCV(I)/ETA,XD(I),R(I),
      #BPOP(I)
       WRITE(3,1003)I,Z,UC(I),TC(I),QCV2(I),1-HUR(I),P(I)
       WRITE(4,1004)I,Z,RYNLD(I),PEC(I),COEFD(I),H(I)
       WRITE(5,1005)I,Z,DENSD(I),VISCD(I),CPPD(I),CKKD(I),HD(I)
       WRITE(6,1006)I,Z,DENSC(I),VISCC(I),CPPC(I),CKKC(I),HC(I)
       WRITE(8,1008)I,Z,WG(I),HM(I)
       WRITE(9,1009)I,Z,hT(i),hS(i),hS(I)*hT(I)/(hS(I)+hT(I)),
      #     hST(I),hBB(I),H(I)
 10    CONTINUE
 1002  FORMAT(I4,1x,f6.3,2(1x,f6.3),1x,f6.1,1x,f5.3,1x,F6.0,1X,f6.4,
      #     1X,f6.5,1X,F7.0)
 1003  FORMAT(I4,1x,f6.3,1x,f8.5,1x,f6.1,1x,f7.0,2x,f6.4,1x,f8.2)
 1004  FORMAT(I4,1X,F6.3,2X,F7.0,2X,F8.0,2X,E9.3,2X,F7.1)
 1005  FORMAT(I4,1X,F6.3,2X,F8.3,2X,E12.5,2X,F6.1,2X,F7.5,2X,F7.1)
 1006  FORMAT(I4,1X,F6.3,2X,F8.3,2X,E12.5,2X,F6.1,2X,F7.5,2X,F7.1)
 1007  FORMAT(I4,1X,F5.2,4(1X,E10.3),1X,F6.4)
 1008  FORMAT(I4,1X,F6.3,2X,F8.6,4X,E10.3)
 1009  FORMAT(I4,1X,F6.3,1X,3(1X,F7.1),2X,2(1X,F7.1),3X,F7.1)
 1010  format(f6.3,2x,f9.1,3x,f8.5,3x,5(x,f8.5))
       qsum=0.0
       do 20 i=0,n
          qsum=qsum+qcv(i)
          write(10,2001)i*dz,h(i),td(i),tc(i),udz(i),hur(i),xd(i),
      #         r(i),rynld(i),bet(i)
 20    continue
```

```
 2001 format(f5.3,1x,f6.0,2(1x,f5.2),1x,f7.4,2(1x,f5.3),1x,f7.5,
     #    1x,f7.0,1x,f5.3)
      RETURN
      END
C-----------------------------------------------
      SUBROUTINE BBLMIX(RHODL,RHODV,XD,BETA)
      REAL M
      DIMENSION X(1)
      COMMON/BMIX/M,XQ
        XQ=XD
      M=RHODL/RHODV
      CALL NONEQN(1,X,.00001,.01,315,200,1.E-6)
      BETA=X(1)
      RETURN
      END
C-----------------------------------------------
        SUBROUTINE QUALCHCK(X)
        IF(X.LT.0.0) THEN
           X=0.0
        ELSEIF(X.GT.1.0)THEN
           X=1.0
        ENDIF
        RETURN
        END
C-----------------------------------------------
        SUBROUTINE TRNGCHK(ITR,TF)
        IF(ITR.EQ.1) THEN
           WRITE(*,*)'*********************************************'
           WRITE(*,*)'TEMPERATURE OUT OF RANGE:  One of the fluid'
           WRITE(*,*)'temperatures has exceeded the range limit in'
           WRITE(*,*)'one of the property routines.'
           WRITE(*,*)'*********************************************'
           WRITE(*,*)'FLUID TEMPERATURE=',TF
           WRITE(7,*)
           WRITE(7,*)
           WRITE(7,*)'*********************************************'
           WRITE(7,*)'TEMPERATURE OUT OF RANGE:  One of the fluid'
           WRITE(7,*)'temperatures has exceeded the range limit in'
           WRITE(7,*)'one of the property routines.'
           WRITE(7,*)'*********************************************'
           WRITE(7,*)'FLUID TEMPERATURE=',TF
           STOP
        ELSE
           RETURN
        ENDIF
        END
C-----------------------------------------------
      SUBROUTINE NONEQN (N,X,XXS,XINC,NINT,ITER,EPS)
      REAL M
      DIMENSION X(N)
      COMMON/BMIX/M,XQ
        XS=XXS
      CC=0.0
      DELX=0.0
      II=0
      NP=1
      A=F(XS)
      IF(A.NE.0.0 .AND. XS.NE. 0.0) GOTO 10
      II=II+1
      X(II)=0.0
      XS=XS+XINC
      A=F(XS)
```

```
  10    AA=A
      A=F(XS)
      CC=AMAX1(XS,CC)
      IF(NP.GT.NINT) GOTO 50
      NP=NP+1
      AAA=A*AA
      XS=XS+XINC
      IF (AAA) 20,40,10
  20    XS=XS-XINC
      XINCN=XINC/5.0
      I=1
  30    B=F(XS)
      CON=(F(XS+XINCN)-F(XS))/XINCN
      DELX=B/CON
      XS=XS-DELX
      DELX=ABS(DELX)
      ITER1=I
      IF(DELX.LT.EPS) GOTO 40
      CC=AMAX1(XS,CC)
      I=I+1
      IF(I.GT.ITER) GOTO 40
      GOTO 30
  40    II=II+1
      X(II)=XS
      XS=CC+XINC
      IF(II.GE.N) GOTO 50
      A=F(XS)
      IF(A.EQ.0.0) A=F(XS+XINC)
      XS=XS+XINC
      GOTO 10
  50    CONTINUE
      RETURN
      END
C______________________________________________
      FUNCTION F(BETA)
      REAL M
      COMMON/BMIX/M,XQ
      F=3*COS(BETA)-(COS(BETA))**3+2-(4*(1-XQ))/(1+XQ*(M-1))
      RETURN
      END
C______________________________________________
        SUBROUTINE GINTGRT(EC,E,G)
        DTH=3.14159/100.
        G=0.
        DO 1 I=1,100
           TH=I*DTH-.5*DTH
           G=G+SIN(TH)**3*(1.-(1.-1./EC**4)*SIN(TH)**2)**.5/
     #          (1.-E**2*SIN(TH)**2)**(5./2.)*DTH
 1      CONTINUE
        RETURN
        END
C______________________________________________
        SUBROUTINE H2OBETA(TC,BETA)
        T=TC+273.15
        BETA=-.0241253+.000199695*T-5.50632E-7*T**2
     #        +5.20287E-10*T**3
        RETURN
        END
C______________________________________________
        SUBROUTINE FLTHCAP(FTYPE,TF,PRESS,CP,RHO,Q,ITR)
        CHARACTER*6 FTYPE
        ITR=0
```

```
      CP=0.0
      RHO=0.0
      Q=0.0
      TLU=-300
      TLL=-300
      IF(FTYPE.EQ.'TH44  ') THEN
         TLL=-45
         TLU=260
         IF(TF.LT.TLL.OR.TF.GT.TLU) GOTO 100
         CP=2.47558*TF+1895.91
         RHO=940.03-0.75776*TF-1.2075E-4*TF**2
         Q=0.0
      ELSEIF(FTYPE.EQ.'TH55  ') THEN
         TLL=0
         TLU=343.
         IF(TF.LT.TLL.OR.TF.GT.TLU) GOTO 100
         CP=3.38436*TF+1862.95
         RHO=895.12-0.66735*TF-6.1756E-6*TF**2
         Q=0.0
      ELSEIF(FTYPE.EQ.'TH60  ') THEN
         TLL=-50.
         TLU=343.
         IF(TF.LT.TLL.OR.TF.GT.TLU) GOTO 100
         CP=1517.5+3.7058*TF
         RHO=1007.9-0.64663*TF-6.8946E-5*TF**2
         Q=0.0
      ELSEIF(FTYPE.EQ.'TH66  ') THEN
         TLL=-18.
         TLU=371.
         IF(TF.LT.TLL.OR.TF.GT.TLU) GOTO 100
         CP=1514.7+3.5326*TF
         RHO=1020.3-0.67361*TF-2.7852E-5*TF**2
         Q=0.0
      ELSEIF(FTYPE.EQ.'TH88  ') THEN
         TLL=149.
         TLU=482.
         IF(TF.LT.TLL.OR.TF.GT.TLU) GOTO 100
         CP=2.1984*TF+1627.2
         RHO=-1.2077E-4*TF**2-0.7389*TF+1121.2
         Q=0.0
      ELSEIF(FTYPE.EQ.'DTSR1 ') THEN
         TLL=0.
         TLU=125.
         IF(TF.LT.TLL.OR.TF.GT.TLU) GOTO 100
         CP=3204.1+3.8701*TF
         RHO=1082.5-0.33961*TF-2.4148E-3*TF**2
         Q=0.0
      ELSEIF(FTYPE.EQ.'DT4000') THEN
         TLL=4.
         TLU=177.
         IF(TF.LT.TLL.OR.TF.GT.TLU) GOTO 100
         CP=3169.2+3.9563*TF
         RHO=1096.7-0.491*TF-1.5049E-3*TF**2
         Q=0.0
      ELSEIF(FTYPE.EQ.'DT4030') THEN
         TLL=4.
         TLU=176.
         IF(TF.LT.TLL.OR.TF.GT.TLU) GOTO 100
         CP=3741.2+1.5305*TF
         RHO=1061.6-.346849*TF-.00193194*TF**2
         Q=0.0
      ELSEIF(FTYPE.EQ.'DT4040') THEN
```

```
         TLL=4.
         TLU=176.
         IF(TF.LT.TLL.OR.TF.GT.TLU) GOTO 100
         CP=3533.9+2.8283*TF-.00219331*TF**2
         RHO=1071.6-.34685*TF-.00193194*TF**2
         Q=0.0
      ELSEIF(FTYPE.EQ.'DT4050') THEN
         TLL=4.
         TLU=176.
         IF(TF.LT.TLL.OR.TF.GT.TLU) GOTO 100
         CP=3321.4+3.49*TF-1.1506E-3*TF**2
         RHO=1092.4-.5401*TF-.0011855*TF**2
         Q=0.0
      ELSEIF(FTYPE.EQ.'DT4060') THEN
         TLL=4.
         TLU=176.
         IF(TF.LT.TLL.OR.TF.GT.TLU) GOTO 100
         CP=3103.8+4.4285*TF-.0020395*TF**2
         RHO=1112.9-.754322*TF
         Q=0.0
      ELSEIF(FTYPE.EQ.'DWFHD ') THEN
         TLL=4.
         TLU=177.
         IF(TF.LT.TLL.OR.TF.GT.TLU) GOTO 100
         CP=3297.7+5.0675*TF-4.9202E-3*TF**2
         RHO=1071.4-0.61005*TF-9.05E-4*TF**2
         Q=0.0
      ELSEIF(FTYPE.EQ.'DFHD30') THEN
         TLL=4.
         TLU=162.
         IF(TF.LT.TLL.OR.TF.GT.TLU) GOTO 100
         CP=3894.6+1.7059*TF-2.2976E-3*TF**2
         RHO=1052.6-.587151*TF-6.4261E-4*TF**2
         Q=0.0
      ELSEIF(FTYPE.EQ.'DFHD40') THEN
         TLL=4.
         TLU=162.
         IF(TF.LT.TLL.OR.TF.GT.TLU) GOTO 100
         CP=3736.9+2.3371*TF-2.11822E-3*TF**2
         RHO=1062.3-.51924*TF-.00142626*TF**2
         Q=0.0
      ELSEIF(FTYPE.EQ.'DFHD50') THEN
         TLL=4.
         TLU=162.
         IF(TF.LT.TLL.OR.TF.GT.TLU) GOTO 100
         CP=3496.9+3.5702*TF-.0020608*TF**2
         RHO=1073-.757688*TF
         Q=0.0
      ELSEIF(FTYPE.EQ.'DFHD60') THEN
         TLL=4.
         TLU=162.
         IF(TF.LT.TLL.OR.TF.GT.TLU) GOTO 100
         CP=3257+4.80134*TF-1.9952E-3*TF**2
         RHO=1072.3-.519239*TF-.00142626*TF**2
         Q=0.
      ELSEIF(FTYPE.EQ.'DWFRST') THEN
         TLL=4.
         TLU=121.
         IF(TF.LT.TLL.OR.TF.GT.TLU) GOTO 100
         CP=3455.9+3.8394*TF
         RHO=1054.1-0.43901*TF-2.5735E-3*TF**2
         Q=0.0
```

```
      ELSEIF(FTYPE.EQ.'PARANF') THEN
         TLL=10.
         TLU=316.
         IF(TF.LT.TLL.OR.TF.GT.TLU) GOTO 100
         CP=1810.9+3.5594*TF
         RHO=729.65-0.33504*TF
         Q=0.0
      ELSEIF(FTYPE.EQ.'PARAHE') THEN
         TLL=10.
         TLU=316.
         IF(TF.LT.TLL.OR.TF.GT.TLU) GOTO 100
         CP=1876.8+4.1316*TF
         RHO=746.1-0.63119*TF
         Q=0.0
      ELSEIF(FTYPE.EQ.'MTHVP1') THEN
         TLL=15.
         TLU=404.
         IF(TF.LT.TLL.OR.TF.GT.TLU) GOTO 100
         CP=1502.4+2.7206*TF
         RHO=1071.2-0.62931*TF-7.8203E-2*TF**2
         Q=0.0
      ELSEIF(FTYPE.EQ.'CFAF  ') THEN
         TLL=40.
         TLU=316.
         IF(TF.LT.TLL.OR.TF.GT.TLU) GOTO 100
         CP=1714.9+3.8688*TF
         RHO=862.89-0.63225*TF+4.0234E-4*TF**2
         Q=0.0
      ELSEIF(FTYPE.EQ.'CFFG  ') THEN
         TLL=40.
         TLU=288.
         IF(TF.LT.TLL.OR.TF.GT.TLU) GOTO 100
         CP=1803.8+3.3423*TF
         RHO=854.12-0.27305*TF-7.5461E-4*TF**2
         Q=0.0
      ELSEIF(FTYPE.EQ.'CFHTF ') THEN
         TLL=40.
         TLU=316.
         IF(TF.LT.TLL.OR.TF.GT.TLU) GOTO 100
         CP=1845.8+3.2776*TF
         RHO=871.0-0.58587*TF+2.2249E-4*TF**2
         Q=0.0
      ELSEIF(FTYPE.EQ.'SYLTH8') THEN
         TLL=-73.
         TLU=260.
         IF(TF.LT.TLL.OR.TF.GT.TLU) GOTO 100
         CP=1574.5+1.7067*TF
         RHO=955.89-0.81738*TF-4.8042E-4*TF**2
         Q=0.0
      ELSEIF(FTYPE.EQ.'THM550') THEN
         TLL=-17.
         TLU=316.
         IF(TF.LT.TLL.OR.TF.GT.TLU) GOTO 100
         CP=1989.5+3.3681*TF
         RHO=852.91-0.67118*TF
         Q=0.0
      ELSEIF(FTYPE.EQ.'THM600') THEN
         TLL=-17.
         TLU=327.
         IF(TF.LT.TLL.OR.TF.GT.TLU) GOTO 100
         CP=2094.7+3.1495*TF
         RHO=835.58-0.65347*TF
```

```
      Q=0.0
   ELSEIF(FTYPE.EQ.'THM800') THEN
      TLL=-17.
      TLU=371.
      IF(TF.LT.TLL.OR.TF.GT.TLU) GOTO 100
      CP=2184.5+3.2144*TF-9.6370E-4*TF**2
      RHO=2184.5+3.2144*TF-9.6370E-4*TF**2
      Q=0.0
   ELSEIF(FTYPE.EQ.'THRMX ') THEN
      IF(TF.LT.TLL.OR.TF.GT.TLU) GOTO 100
      CP=2.83089*TF+1528.32
      RHO=-.938571*TF+1089.29
      Q=0.0
   ELSEIF(FTYPE.EQ.'AIR   ') THEN
      TLL=-173.
      TLU=2726.
      IF(TF.LT.TLL.OR.TF.GT.TLU) GOTO 100
      CP=1005.28+5.3251E-3*TF+6.337E-4*TF**2-8.1579E-7*TF**3
#          +4.9426E-10*TF**4-1.5492E-13*TF**5+2.3265E-17*TF**6
      IF(TF.LE.26.) THEN
         RHO=1.26941-4.886E-3*TF+2.5977E-5*TF**2+1.8185E-7*TF**3
#             +1.7867E-9*TF**4
      ELSEIF (TF.GT.26.0.AND.TF.LE.2225.) THEN
         RHO=1.26214-4.1154E-3*TF+1.011E-5*TF**2-1.6588E-8*TF**3
#             +1.7341E-11*TF**4-1.1284E-14*TF**5+4.3989E-18*TF**6
#             -9.3648E-22*TF**7+8.3467E-26*TF**8
      ELSEIF (TF.GT.2225.) THEN
         RHO=0.350535-1.313E-4*TF+1.6274E-8*TF**2
      ENDIF
      Q=1.0
   ELSEIF(FTYPE.EQ.'CO2   ') THEN
      TLL=-53.
      TLU=1526
      IF(TF.LT.TLL.OR.TF.GT.TLU) GOTO 100
      CP=827.967+0.99954*TF-8.4763E-4*TF**2+3.8487E-7*TF**3
#          -7.1375E-11*TF**4
      RHO=1.9713-5.8952E-3*TF+9.8244E-6*TF**2-7.5549E-9*TF**3
#          +2.0916E-12*TF**4
      Q=1.0
   ELSEIF(FTYPE.EQ.'H2    ') THEN
      TLL=-173.
      TLU=1726.
      IF(TF.LT.TLL.OR.TF.GT.TLU) GOTO 100
      CP=14183.4+5.5081*TF-.037213*TF**2+1.2355E-4*TF**3
#          -2.2182E-7*TF**4+2.3445E-10*TF**5-1.4451E-13*TF**6
#          +4.77946E-17*TF**7-6.51397E-21*TF**8
      RHO=.087964-3.0046E-4*TF+1.247E-6*TF**2-6.5073E-9*TF**3
#          +2.4773E-11*TF**4-5.7169E-14*TF**5+8.0088E-17*TF**6
#          -6.8746E-20*TF**7+3.5342E-23*TF**8-9.9886E-27*TF**9
#          +1.19331E-30*TF**10
      Q=1.0
   ELSEIF(FTYPE.EQ.'N2    ') THEN
      TLL=-173.
      TLU=1026.
      IF(TF.LT.TLL.OR.TF.GT.TLU) GOTO 100
      CP=1039.9-.033984*TF+6.0155E-4*TF**2-5.3704E-7*TF**3
#          +1.4479E-10*TF**4
      RHO=1.2826-4.5158E-3*TF+1.0512E-5*TF**2-5.0297E-8*TF**3
#          +4.2596E-10*TF**4-1.8193E-12*TF**5+3.9745E-15*TF**6
#          -4.6671E-18*TF**7+2.8145E-21*TF**8-6.8517E-25*TF**9
      Q=1.0
   ELSEIF(FTYPE.EQ.'O2    ') THEN
```

```
      TLL=-173.
      TLU=1026.
      IF(TF.LT.TLL.OR.TF.GT.TLU) GOTO 100
      CP=913.86+.090997*TF+.0013194*TF**2-3.1674E-6*TF**3
#          +3.00295E-9*TF**4-1.03672E-12*TF**5
      RHO=1.4662-5.1602E-3*TF+1.1947E-5*TF**2-5.7867E-8*TF**3
#          +4.9429E-10*TF**4-2.114E-12*TF**5+4.6193E-15*TF**6
#          -5.4242E-18*TF**7+3.2709E-21*TF**8-7.9618E-25*TF**9
      Q=1.0
   ELSEIF(FTYPE.EQ.'NH3   ') THEN
      TLL=-100
      TLU=406.
      IF(TF.LT.TLL.OR.TF.GT.TLU) GOTO 100
      CP=2134.76+.538392*TF+.0063678*TF**2-9.9898E-6*TF**3
      RHO=.73673-.00209815*TF+2.8379E-6*TF**2
      Q=1.0
   ELSEIF(FTYPE.EQ.'MS20  ') THEN
      CP=3.09412*TF+1980.44
      RHO=-0.56681*TF+903.679
      Q=0.0
   ELSEIF(FTYPE.EQ.'TRAN  ') THEN
      CP=-1.5597E-9*TF**6+3.6213E-7*TF**5-1.7345E-5*TF**4
#          -1.4121E-3*TF**3+.12049*TF**2+3.04729*TF+1616.47
      RHO=-.62105*TF+891.364
      Q=0.0
   ELSEIF(FTYPE.EQ.'TM1   ') THEN
      CP=(.00120508*TF+1.61848)*1000.
      RHO=-0.63744*TF+902.499
      Q=0.0
   ELSEIF(FTYPE.EQ.'VM4   ') THEN
      CP=-2.3592E-9*TF**6+5.4339E-7*TF**5-3.7219E-5*TF**4
#          +3.022E-4*TF**3+0.04831*TF**2+.23315*TF+1457.74
      RHO=-0.5915*TF+915.492
      Q=0.0
   ELSEIF(FTYPE.EQ.'KHF22 ') THEN
      CP=-1.58271E-6*TF**4+6.52884E-5*TF**3+0.012453*TF**2
#          +.241957*TF+1638.96
      RHO=-0.72984*TF+1009.52
      Q=0.0
   ELSEIF(FTYPE.EQ.'ENG   ') THEN
      TLL=0.
      TLU=156.
      IF(TF.LT.TLL.OR.TF.GT.TLU) GOTO 100
      CP=1798.42+4.06543*TF+.001455*TF**2
      RHO=899.55-.5926*TF
      Q=0.0
   ELSEIF(FTYPE.EQ.'ETHYLG') THEN
      TLL=0
      TLU=100.
      IF(TF.LT.TLL.OR.TF.GT.TLU) GOTO 100
      CP=2293.78+4.4801*TF
      RHO=1129.76-.23264*TF-.026345*TF**2+4.5083E-4*TF**3
#          -2.3342E-6*TF**4
      Q=0.0
   ELSEIF(FTYPE.EQ.'GLYCRN') THEN
      TLL=0
      TLU=46
      IF(TF.LT.TLL.OR.TF.GT.TLU) GOTO 100
      CP=2260.92+5.8326*TF+.01288*TF**2
      RHO=1275.52-.5666*TF-7.2682E-4*TF**2
      Q=0.0
   ELSEIF(FTYPE.EQ.'HG    ') THEN
```

```
           TLL=0
           TLU=326
           IF(TF.LT.TLL.OR.TF.GT.TLU) GOTO 100
           CP=1403.48-.3996*TF+7.7106E-4*TF**2
           RHO=13592.9-2.4009*TF
           Q=0.0
       ENDIF
       GOTO 1
 100   ITR=1
       GOTO 2
 1     IF(Q.EQ.1.0) RHO=RHO*PRESS/101.325
 2     RETURN
       END
C
C
C===============================================================
C===============================================================
C
      SUBROUTINE PROPTY(FTYPE,TF,PRC,CKL,CKV,VISL,VISV,SIG,ITR)
C     ************************************************************
C           THERMOPHYSICAL PROPERTIES PF
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           REFERENCE :  Tables of Thermophysical Properties
c           of Liquids and Gases,   N.B. Vargaftic
c     ***********************************************************
C           REFERENCE COBAN DISSERT FOR VARIABLE DEF.
        CHARACTER*6 FTYPE
C
        ITR=0
        VISV=0.
        VISL=0.
        CKV=0.
        CKL=0.
        SIG=0.
C
        IF(FTYPE.EQ.'H2O   ') THEN
      IF(PRC.LT.2.E06) THEN
      IF(TF.LT.100) THEN
C   P = 1 BAR
      A0=.5689999177E00
      A1=.2237388258E-02
      A2=-.4194219792E-04
      A3=.8461035643E-06
      A4=-.1064290936E-08
      A5=-.2195261464E-09
      A6=.2867647080E-11
      A7=-.1109E-13
      CK1=A0+A1*TF+A2*TF**2+A3*TF**3+A4*TF**4+A5*TF**5+
     #  A6*TF**6+A7*TF**7
C   P = 20 BARS
      A0=.5000276018E00
      A1=.1842722906E-01
```

```
      A2=-.1449807789E-02
      A3=.6254021227E-04
      A4=-.1490806953E-05
      A5=.1984199371E-07
      A6=-.1381372569E-09
      A7=.3915733022E-12
      CK2=A0+A1*TF+A2*TF**2+A3*TF**3+A4*TF**4+A5*TF**5+
     #  A6*TF**6+A7*TF**7
      CKL=CK1+(CK2-CK1)/(1.9E+06)*(PRC-.1E+06)
        CKV=0.0
      ELSE
C   P = 20 BARS
      A0=.5034354585E00
      A1=.1080602021E-01
      A2=-.4031109874E-03
      A3=.81871132370E-05
      A4=-.8981143005E-07
      A5=.5378607955E-09
      A6=-.1659310095E-11
      A7=.206279238E-14
      CK2=A0+A1*TF+A2*TF**2+A3*TF**3+A4*TF**4+A5*TF**5+
     #  A6*TF**6+A7*TF**7
      CKL=CK2
      CKV=0.0
      ENDIF
      ELSE
C   P=20 BARS
      A0=.5034354585E00
      A1=.1080602021E-01
        A2=-.4031109874E-03
      A3=.81871121370E-05
      A4=-.8981143005E-07
      A5=.5378607955E-09
      A6=-.165931009E-11
      A7=.20627923E-14
      CK1=A0+A1*TF+A2*TF**2+A3*TF**3+A4*TF**4+A5*TF**5+
     #  A6*TF**6+A7*TF**7
C   P = 40 BARS
      A0=.5719129275E00
      A1=.1915405391E-02
      A2=-.8020192384E-05
      A3=-.2277327713E-07
      A4=.4561862413E-09
      A5=-.3196354991E-11
      A6=.1089066733E-13
      A7=-.1478755907E-16
      CK2=A0+A1*TF+A2*TF**2+A3*TF**3+A4*TF**4+A5*TF**5+
     #  A6*TF**6+A7*TF**7
      CKL=CK1+(CK2-CK1)/(2.E+06)*(PRC-2.E+06)
      CKV=0.0
      ENDIF
      IF(PRC.LT.2.E06) THEN
      IF(TF.LT.100) THEN
C   P = 1 BAR
      A0=.1752498865E+05
      A1=-.5623342898E+03
      A2=.1293770770E+02
      A3=-.2316151658E00
      A4=.3092855426E-02
      A5=-.2800988617E-04
      A6=.1494852986E-06
      A7=-.3495565033E-09
```

```
      VIS1=A0+A1*TF+A2*TF**2+A3*TF**3+A4*TF**4+A5*TF**5+
     #  A6*TF**6+A7*TF**7
C    P = 20 BARS
      A0=.1751398227E+05
      A1=-.5621437186E+03
      A2=.1301016255E+02
      A3=-.2372883464E+00
      A4=.32728885154E-02
      A5=-.3108219012E-04
      A6=.1745751680E-06
      A7=-.4300887168E-09
      VIS2=A0+A1*TF+A2*TF**2+A3*TF**3+A4*TF**4+A5*TF**5+
     #  A6*TF**6+A7*TF**7
      VISL=VIS1+(VIS2-VIS1)/(1.9E+06)*(PRC-.1E+06)
      VISL=VISL*1.E-07
        VISV=0.0
      ELSE
C    P = 20 BARS
      A0=.4230490200E+04
      A1=.1884581715E+03
      A2=-.667610027E+01
      A3=.9206814826E-01
      A4=-.6904918772E-03
      A5=.2967543200E-05
      A6=-.6887934270E-08
      A7=.6712245273E-11
      VIS2=A0+A1*TF+A2*TF**2+A3*TF**3+A4*TF**4+A5*TF**5+
     #  A6*TF**6+A7*TF**7
      VISL=VIS2*1.E-07
      VISV=0.0
      ENDIF
      ELSE
      IF(TF.LT.100) THEN
C    P = 20 BARS
      A0=.17513982270E+05
      A1=-.5621437186E+03
      A2=.1301016255E+02
      A3=-.2372883464E+00
      A4=.3278885154E-02
      A5=-.3108219012E-04
      A6=.1745751680E-06
      A7=-.4300887168E-09
      VIS1=A0+A1*TF+A2*TF**2+A3*TF**3+A4*TF**4+A5*TF**5+
     #  A6*TF**6+A7*TF**7
C  P = 40 BARS
      A0=.1750199191E+05
      A1=-.5612118941E+03
      A2=.1295640067E+02
      A3=-.2346663975E+00
      A4=.3202648351E-02
      A5=.2986536004E-04
      A6=.1646732071E-06
      A7=-.3979925446E-09
      VIS2=A0+A1*TF+A2*TF**2+A3*TF**3+A4*TF**4+A5*TF**5+
     #  A6*TF**6+A7*TF**7
      VISL=VIS1+(VIS2-VIS1)/(2.E+06)*(PRC-2.E+06)
      VISL=VISL+1.E-07
        VISV=0.0
      ELSE
C    P = 20 BARS
      A0=.4230490200E+04
      A1=.1884581715E+03
```

```
      A2=-.667610027E+01
      A3=.9206814826E-01
      A4=-.6904918772E-03
      A5=.2967543200E-05
      A6=-.6887934270E-08
      A7=.6712245273E-11
      VIS1=A0+A1*TF+A2*TF**2+A3*TF**3+A4*TF**4+A5*TF**5+
     #   A6*TF**6+A7*TF**7
C   P = 40 BARS
      A0=.9542057174E+04
      A1=-.7263813086E+02
      A2=-.1240393524E+01
      A3=.2999460681E-01
      A4=-.2705631795E-03
      A5=.1284340755E-05
      A6=-.31852135E-08
      A7=.3261966188E-11
        VIS2=A0+A1*TF+A2*TF**2+A3*TF**3+A4*TF**4+A5*TF**5+
     # A6*TF**6+A7*TF**7
      VISL=VIS1+(VIS2-VIS1)/(2.E+06)*(PRC-2.E+06)
      VISL=VISL*1.E-07
        VISV=0.0
      ENDIF
        ENDIF
        SIG=75.50411188811198-8.46835E-02*TF-3.187966E-03*TF**2
     #   +5.076282E-05*TF**3-3.326923E-07*TF**4+6.025641E-10*TF**5
        SIG=SIG*1.E-03
      ELSEIF(FTYPE.EQ.'AIR    ') THEN
         TLL=-173.
         TLU=2726.
         IF(TF.LT.TLL.OR.TF.GT.TLU) GOTO 100
         VISV=1.71833E-5+5.0923E-8*TF-3.8357E-11*TF**2+2.8467E-14*
     #        TF**3-9.9701E-18*TF**4+1.3181E-21*TF**5
         VISL=0.0
         CKV=0.0254297+8.3037E-5*TF-9.6375E-8*TF**2+1.286E-10*TF**3
     #        -7.3051E-14*TF**4+1.5801E-17*TF**5
         CKL=0.0
         SIG=0.0
      ELSEIF(FTYPE.EQ.'CO2    ') THEN
         TLL=-53.
         TLU=1526.
         IF(TF.LT.TLL.OR.TF.GT.TLU) GOTO 100
         VISV=1.4589E-5+3.8088E-8*TF-6.2015E-12*TF**2
         VISL=0.
         CKV=1.4448E-2+8.3527E-5*TF-1.3133E-8*TF**2
         CKL=0.
         SIG=0.
      ELSEIF(FTYPE.EQ.'H2     ') THEN
         TLL=-173.
         TLU=1726.
         IF(TF.LT.TLL.OR.TF.GT.TLU) GOTO 100
         VISL=0.
         VISV=8.26065E-6+2.0427E-8*TF-6.8867E-12*TF**2+1.7348E-15*TF**3
         CKL=0.
         CKV=.167244+4.9794E-4*TF-3.0927E-7*TF**2+2.6862E-10*TF**3
     #        -6.87395E-14*TF**4
         SIG=0.
      ELSEIF(FTYPE.EQ.'N2     ') THEN
         TLL=-173.
         TLU=1026.
         IF(TF.LT.TLL.OR.TF.GT.TLU) GOTO 100
         VISL=0.
```

```
   VISV=1.6277E-5+4.7868E-8*TF-2.9288E-11*TF**2+1.1255E-14*TF**3
   CKL=0.
   CKV=.023643+7.4477E-5*TF-3.9978E-8*TF**2+2.1437E-11*TF**3
   SIG=0.
 ELSEIF(FTYPE.EQ.'O2    ') THEN
   TLL=-173.
   TLU=1026.
   IF(TF.LT.TLL.OR.TF.GT.TLU) GOTO 100
   VISL=0.
   VISV=1.90145E-5+5.8246E-8*TF-4.16376E-11*TF**2
#        +2.14079E-14*TF**3
   CKL=0.
   CKV=.0235842+7.58894E-5*TF-1.43399E-8*TF**2
   SIG=0.
 ELSEIF(FTYPE.EQ.'NH3   ') THEN
   TLL=26.
   TLU=306.
   IF(TF.LT.TLL.OR.TF.GT.TLU) GOTO 100
   VISL=0.
   VISV=9.1404E-6+3.80243E-8*TF-9.0094E-12*TF**2
   CKL=0.
   CKV=.022197+8.65576E-5*TF+3.232E-7*TF**2-5.4597E-10*TF**3
   SIG=0.
 ELSEIF(FTYPE.EQ.'PENT  ') THEN
 CKV=11.7000943+.0974954646*TF+1.2506797E-04*TF**2
#  -3.20993285E-10*TF**3
 CKV=CKV*1.E-03
 CKL=.120500001-2.19166518E-04*TF-1.4583602E-06*TF**2
#  +4.1667695E-08*TF**3-4.16677821E-10*TF**4
 VISL=.2830648416302071-4.7037877074119475E-03*TF
# +3.9457157378790382E-04*TF**2-2.2040828809048E-05*TF**3
# +5.80420663296576E-07*TF**4-7.8134559782743E-09*TF**5
# +5.2357026915869730E-11*TF**6-1.3865546463192681E-13*TF**7
 VISL=VISL*1.E-03
 VISV=49.3702925+.938695079*TF-.0151413279*TF**2
#  +1.38067486E-04*TF**3-4.01657705E-07*TF**4
 VISV=VISV*1.E-07
 SIG=18.200088-.104142117*TF-1.05600806E-03*TF**2
#  +5.42164101E-05*TF**3-7.87026653E-07*TF**4
 SIG=SIG*1.E-03
 ELSEIF(FTYPE.EQ.'BUT   ') THEN
 CKV=-4.3704E-11*TF**3+1.01398E-7*TF**2+.000108024*TF
# +.01310114
 CKL=-7.81255E-10*TF**4+2.60419E-7*TF**3-3.21878E-5*TF**2
# +.00149585*TF+.0749995
 VISL=-8.41033E-14*TF**5+3.28754E-11*TF**4-5.00885E-9*TF**3
# +3.7359e-7*TF**2-1.47048*TF+.000376398
 VISV=-7.04548e-18*TF**5+6.15458e-15*TF**4-2.03319e-12*TF**3
# +2.99428e-10*TF**2+5.54933e-9*TF+7.46943e-6
 SIG=5.30302E-11*TF**4+1.46969E-8*TF**3+1.52878E-6*TF**2
# -5.55336E-5*TF+.0157512
 ELSEIF(FTYPE.EQ.'ISOP  ') THEN
 VISL=-1.66661e-10*TF**3+1.99997E-8*TF**2-2.78333E-6*TF
# +.000272
 VISV=-2.09459E-12*TF**2+2.224021E-8*TF+6.38969E-6
 CKL=-8.88886E-8*TF**3+4.63094E-6*TF**2-.000298849*TF
#+.11906
 CKV=14.02360414148952+.1208272189021553*TF
# +1.0860013903942589E-04*TF**2-1.9940388279179968E-05*TF**3
# +4.9048950395836884E-07*TF**4-4.929433762331529E-09*TF**5
# +2.2305489177839495E-11*TF**6-3.7421511943571895E-14*TF**7
 CKV=CKV*1.E-03
```

```
      SIG=6.71858E-8*TF**2-.00010998*TF+.0171699
      ELSEIF(FTYPE.EQ.'ISOB  ') THEN
      VISL=1.55136E-10*TF**3-6.16567E-9*TF**2-2.09543E-6*TF
     # +.000214
      VISV=-4.35675E-14*TF**4+1.22251E-11*TF**3
     # -1.22875E-9*TF**2+7.04506E-8*TF+6.17173E-6
      SIG=6.78573E-8*TF**2-.000115583*TF+.0129523
C  PROBLEM  : CKISV.DAT
      A0=.13813870E+02
      A1=.1088360E+00
      A2=.3069733333E-03
      A3=-.334880E-05
      A4=.1116266667E-07
      CKV=A0+A1*TF+A2*TF**2+A3*TF**3+A4*TF**4
      CKV=CKV*1.E-03
C  PROBLEM  :CKESL
      A0=.10545730E+02
      A1=-.58977750E-01
      A2=.27657500E-03
      CKL=A0+A1*TF+A2*TF**2
      CKL=CKL*1.E-02
      ELSEIF (FTYPE.EQ.'HEP   ') THEN
         TLL=-90.
         TLU=400.
         IF(TF.LT.TLL.OR.TF.GT.TLU) GOTO 100
         CKL=.13420-2.7167E-4*TF+4.8701E-8*TF**2
         CKV=8.8553E-3+9.2249E-5*TF+7.1772E-8*TF**2
         IF(TF.LE.20.) THEN
            VISL=5.2757E-4-7.8221E-6*TF+5.2534E-8*TF**2+1.9301E-9*
     #           TF**3+5.5358E-11*TF**4
         ELSE
            VISL=5.2757E-4-7.8221E-6*20+5.2534E-8*20**2+1.9301E-9*
     #           20**3+5.5358E-11*20**4
         ENDIF
         VISV=5.3255E-6+2.0043E-8*TF+2.6577E-13*TF**2
         SIG=0.
      ELSEIF(FTYPE.EQ.'METH  ') THEN
         TLL=-173.
         TLU=83.
         IF(TF.LT.TLL.OR.TF.GT.TLU) GOTO 100
         CKL=.23242+4.6197E-3*TF+4.2448E-5*TF**2+9.6463E-8*TF**3
         CKV=.030875+1.3364E-4*TF+9.8656E-8*TF**2
         VISL=6.7496E-5+1.4941E-6*TF+1.1267E-8*TF**2
         VISV=1.025E-5+3.2756E-8*TF-1.8745E-11*TF**2
         SIG=0.
      ELSEIF(FTYPE.EQ.'ETHYL ') THEN
         TLL=-165.
         TLU=225.
         IF(TF.LT.TLL.OR.TF.GT.TLU) GOTO 100
         VISL=-2.3585E-3-6.3783E-5*TF-5.527E-7*TF**2-1.6643E-9*TF**3
         VISV=9.4809E-6+3.2084E-8*TF-1.1608E-11*TF**2
         CKL=8.1027E-2-1.2232E-3*TF-1.2078E-6*TF**2
         CKV=1.6976E-2+1.2134E-4*TF+1.2177E-7*TF**2
         SIG=0.
      ELSEIF(FTYPE.EQ.'MS20  ') THEN
        CKL=8.57848E-8*TF**2-1.19191E-4*TF+.136087
        CKV=0.0
        VISL=4331.33*TF**(-2.69759)
      VISV=0.0
      SIG=0.0
      ELSEIF(FTYPE.EQ.'TRAN  ') THEN
      CKL=-8.19549e-5*TF+.112297
```

```
 CKV=0.0
 VISL=.0406485*EXP(-.032348*TF)
 VISV=0.0
 SIG=0.0
 ELSEIF(FTYPE.EQ.'TM1   ') THEN
 CKL=-9.53055E-5*TF+.120687
 CKV=0.0
 VISL=.382997*TF**(-1.08805)
 VISV=0.0
 SIG=0.0
 ELSEIF(FTYPE.EQ.'VM4   ') THEN
 CKL=-8.94004E-5*TF+.129112
 CKV=0.0
 VISL=61.3841*TF**(-1.85956)
 VISV=0.0
 SIG=0.0
 ELSEIF(FTYPE.EQ.'KHF22 ') THEN
 CKL=-1.48464E-4*TF+.157694
 CKV=0.0
 VISL=2.57617*TF**(-1.30322)
 VISV=0.0
 SIG=0.0
 ELSEIF(FTYPE.EQ.'TH44  ') THEN
    TLL=-45.
    TLU=260.
    IF(TF.LT.TLL.OR.TF.GT.TLU) GOTO 100
    CKL=1.4566E-1-1.2663E-4*TF-1.5176E-7*TF**2
    IF(TF.LT.10.) THEN
       VISL=.0168827*EXP(-.0876271*TF)
    ELSEIF(TF.GE.10) THEN
       VISL=.014207-5.84755E-4*TF+1.1979E-5*TF**2-1.409E-7*TF**3
#        +9.8367E-10*TF**4-4.01099E-12*TF**5+8.80257E-15*TF**6
#        -8.0192E-18*TF**7
    ENDIF
    CKV=0.0
    VISV=0.0
    SIG=0.0
 ELSEIF(FTYPE.EQ.'TH55  ') THEN
    TLL=0.
    TLU=343.
    IF(TF.LT.TLL.OR.TF.GT.TLU) GOTO 100
    CKL=.130426-8.3889E-5*TF-1.17235E-7*TF**2
    CKV=0.0
    IF(TF.LT.38.) THEN
       VISL=.301559*EXP(-.0713115*TF)
    ELSEIF(TF.GE.38.) THEN
       VISL=.104613-3.9908E-3*TF+7.0359E-5*TF**2-7.1588E-7*TF**3
#           +4.5138E-9*TF**4-1.7912E-11*TF**5+4.3535E-14*TF**6
#           -5.9175E-17*TF**7+3.444E-20*TF**8
    ENDIF
    VISV=0.0
    SIG=0.0
 ELSEIF(FTYPE.EQ.'TH60  ') THEN
    TLL=-50.
    TLU=343.
    IF(TF.LT.TLL.OR.TF.GT.TLU) GOTO 100
    CKL=.13363-7.6511E-5*TF-3.6863E-9*TF**2
    CKV=0.0
    IF(TF.LT.10.) THEN
       VISL=.0233656*EXP(-.0841699*TF)
    ELSEIF(TF.GE.10.) THEN
       VISL=.019873-8.1667E-4*TF+1.7149E-5*TF**2-2.0926E-7*TF**3
```

```
#           +1.554E-9*TF**4-7.1014E-12*TF**5+1.9461E-14*TF**6
#           -2.9291E-17*TF**7+1.8593E-20*TF**8
         ENDIF
         VISV=0.0
         SIG=0.0
      ELSEIF(FTYPE.EQ.'TH66  ') THEN
         TLL=-18
         TLU=371
         IF(TF.LT.TLL.OR.TF.GT.TLU) GOTO 100
         CKL=.120973-5.08537E-5*TF-1.06489E-7*TF**2
         CKV=0.0
         IF(TF.LT.49) THEN
            VISL=.928583*EXP(-.0891155*TF)
         ELSE
            VISL=14.3448*TF**(-1.79827)
            ENDIF
         VISV=0.0
         SIG=0.0
      ELSEIF(FTYPE.EQ.'TH88  ') THEN
         TLL=149
         TLU=482
         IF(TF.LT.TLL.OR.TF.GT.TLU) GOTO 100
         CKL=.139621-7.9469E-5*TF-2.8337E-9*TF**2
         CKV=0.
         VISL=3.6914*TF**(-2.01174)
         VISV=0.
         SIG=0.
      ELSEIF(FTYPE.EQ.'CFAF  ') THEN
         TLL=40
         TLU=316.
         IF(TF.LT.TLL.OR.TF.GT.TLU) GOTO 100
         VISL=39.7783*TF**(-1.99202)
         CKL=.13807-5.4571E-5*TF+1.7361E-9*TF**2
         CKV=0.
         SIG=0.
      ELSEIF(FTYPE.EQ.'CFFG  ') THEN
         TLL=40
         TLU=288.
         IF(TF.LT.TLL.OR.TF.GT.TLU) GOTO 100
         VISL=3.62695*TF**(-1.55491)
         VISV=0.
         CKL=.13913-8.5789E-5*TF+9.1377E-8*TF**2
         CKV=0.
         SIG=0.
      ELSEIF(FTYPE.EQ.'CFHTF ') THEN
         TLL=40
         TLU=316.
         IF(TF.LT.TLL.OR.TF.GT.TLU) GOTO 100
         VISL=50.8759*TF**(-2.00572)
         VISV=0.
         CKL=.13907-5.4571E-5+1.7361E-9*TF**2
         CKV=0.
         SIG=0.
      ELSEIF(FTYPE.EQ.'SYLTH8') THEN
         TLL=-73
         TLU=260.
         IF(TF.LT.TLL.OR.TF.GT.TLU) GOTO 100
         IF(TF.LT.10) THEN
            VISL=.00220438-3.9223E-5*TF-4.6422E-7*TF-2.5722E-8*TF**3
         ELSE
            VISL=.00202134-3.07134E-5*TF+2.34941E-7*TF**2
#           -8.7194E-10*TF**3+1.22576E-12*TF**4
```

```
            ENDIF
         VISV=0.
         CKL=.10792-6.52567E-5*TF+1.74916E-9*TF**2
         CKV=0.
         SIG=0.
      ELSEIF(FTYPE.EQ.'PARAHE') THEN
         TLL=10.
         TLU=316.
         IF(TF.LT.TLL.OR.TF.GT.TLU) GOTO 100
         VISL=795.75*TF**(-2.70403)
         VISV=0.
         CKL=0.13455-7.4724E-5*TF
         CKV=0.
         SIG=0.
      ELSEIF(FTYPE.EQ.'PARANF') THEN
         TLL=10.
         TLU=316.
         IF(TF.LT.TLL.OR.TF.GT.TLU) GOTO 100
         VISL=4.40572*TF**(-1.60423)
         VISV=0.
         CKL=.13431-7.0226E-5*TF
         CKV=0.
         SIG=0.
      ELSEIF(FTYPE.EQ.'MTHVP1') THEN
         TLL=15.
         TLU=404.
         IF(TF.LT.TLL.OR.TF.GT.TLU) GOTO 100
         VISL=.00792082-2.0877E-4*TF+2.9702E-6*TF**2-2.6742E-8*TF**3
     $        +1.5734E-10*TF**4-5.9706E-13*TF**5+1.3963E-15*TF**6
     $        -1.8196E-18*TF**7+1.0071E-21*TF**8
         VISV=0.
         CKL=.143285-1.53527E-4*TF+2.03924E-8*TF**2
         CKV=0.
         SIG=0.
      ELSEIF(FTYPE.EQ.'THM550') THEN
         TLL=-17.
         TLU=316.
         IF(TF.LT.TLL.OR.TF.GT.TLU) GOTO 100
         CKL=1.4486E-1-7.4074E-5*TF
         CKV=0.
         IF(TF.LT.4.5) THEN
            VISL=.184015*EXP(-.0602736*TF)
         ELSE
            VISL=.174439-8.1538E-3*TF+1.8709E-4*TF**2-2.5991E-6*TF**3
     $        +2.3307E-8*TF**4-1.37384E-10*TF**5+5.27456E-13*TF**6
     $        -1.2669E-15*TF**7+1.72579E-18*TF**8-1.01648E-21*TF**9
         ENDIF
         VISV=0.
         SIG=0.
      ELSEIF(FTYPE.EQ.'THM600') THEN
         TLL=-17.
         TLU=327.
         IF(TF.LT.TLL.OR.TF.GT.TLU) GOTO 100
         IF(TF.LT.4.5) THEN
            VISL=.182273*EXP(-.0732021*TF)
         ELSE
            VISL=.170328-8.7649E-3*TF+2.1606E-4*TF**2-3.1393E-6*TF**3
     $           +2.8835E-8*TF**4-1.716E-10*TF**5+6.5874E-13*TF**6
     $           -1.5717E-15*TF**7+2.1173E-18*TF**8-1.2293E-21*TF**9
         ENDIF
         VISV=0.
         CKL=.15629-6.196E-5*TF
```

```
        CKV=0.
        SIG=0.
      ELSEIF(FTYPE.EQ.'THM800') THEN
        TLL=-17.
        TLU=371.
        IF(TF.LT.TLL.OR.TF.GT.TLU) GOTO 100
        VISL=.126335-6.0631E-3*TF+1.1617E-4*TF**2-1.134E-6*TF**3
     #        +6.1506E-9*TF**4-1.8752E-11*TF**5+3.00614E-14*TF**6
     #        -1.971E-17*TF**7
        VISV=0.
        CKL=.1529-1.0318E-4*TF-7.4183E-9*TF**2
        CKV=0.
        SIG=0.
      ELSEIF(FTYPE.EQ.'THRMX ') THEN
        CKL=-1.3333E-4*TF+.14319
        CKV=0.0
        VISL=.27911*TF**(-1.23818)
        VISV=0.0
        SIG=0.0
      ELSEIF(FTYPE.EQ.'DTSR1') THEN
        TLL=0.
        TLU=125.
        IF(TF.LE.TLL.OR.TF.GT.TLU) GOTO 100
        CKL=.363+9.09344E-4*TF-3.81547E-6*TF**2
        CKV=0.0
        VISL=.00733697-1.23288E-4*TF+5.53221E-7*TF**2
        VISV=0.0
        SIG=0.0
      ELSEIF(FTYPE.EQ.'DT4000') THEN
        TLL=4.
        TLU=177.
        IF(TF.LT.TLL.OR.TF.GT.TLU) GOTO 100
        VISL=.00746199-1.53974E-4*TF+1.4021E-6*TF**2-2.825E-9*TF**3
        VISV=0.
        CKL=.362926+9.2063E-4*TF-3.91495E-6*TF**2
        CKV=0.
        SIG=0.
      ELSEIF(FTYPE.EQ.'DT4030') THEN
        TLL=4.
        TLU=176.
        IF(TF.LT.TLL.OR.TF.GT.TLU) GOTO 100
        VISL=-8.3167E-4*LOG(TF)+.00441
        VISV=0.0
        CKL=.437696+1.24698E-3*TF-4.8875E-6*TF**2
        CLV=0.
        SIG=0.
      ELSEIF(FTYPE.EQ.'DT4040') THEN
        TLL=4.
        TLU=176.
        IF(TF.LT.TLL.OR.TF.GT.TLU) GOTO 100
        VISL=-.00116578*LOG(TF)+.00609
        VISV=0.
        CKL=.403952+.001046*TF-4.1122E-6*TF**2
        CKV=0.
        SIG=0.
      ELSEIF(FTYPE.EQ.'DT4050') THEN
        TLL=4.
        TLU=176.
        IF(TF.LT.TLL.OR.TF.GT.TLU) GOTO 100
        VISL=-.00161257*LOG(TF)+.0083384
        VISV=0.
        CKL=.3719+8.4555E-4*TF-3.3368E-6*TF**2
```

```
         CKV=0.
         SIG=0.
      ELSEIF(FTYPE.EQ.'DT4060') THEN
         TLL=4.
         TLU=176.
         IF(TF.LT.TLL.OR.TF.GT.TLU) GOTO 100
         VISL=.0323917*TF**(-.866)
         VISV=0.
         CKL=.339419+7.60199E-4*TF-2.8656E-6*TF**2
         CKV=0.
         SIG=0.
      ELSEIF(FTYPE.EQ.'DWFRST') THEN
         TLL=4.
         TLU=121.
         IF(TF.LT.TLL.OR.TF.GT.TLU) GOTO 100
         VISL=.0152155-2.79327E-4*TF+1.30927E-6*TF**2
         VISV=0.
         CKL=.349979+7.25542E-4*TF-4.00611E-6*TF**2
         CKV=0.
         SIG=0.
      ELSEIF(FTYPE.EQ.'DWFHD ') THEN
         TLL=4.
         TLU=176.
         IF(TF.LT.TLL.OR.TF.GT.TLU) GOTO 100
         VISL=.0155255-3.5542E-4*TF+2.765E-6*TF**2-7.0061E-9*TF**3
         VISV=0.
         CKL=.34947+8.504E-4*TF-6.3946E-6*TF**2+1.1496E-7*TF**3
         CKV=0.
         SIG=0.
      ELSEIF(FTYPE.EQ.'DFHD30') THEN
         TLL=4.
         TLU=162.
         IF(TF.LT.TLL.OR.TF.GT.TLU) GOTO 100
         VISL=.5*(.004767*EXP(-.018319*TF)+.018137*TF**(-.780412))
         VISV=0.
         CKL=.4205+.0012295*TF-5.4564E-6*TF**2
         CKV=0.
         SIG=0.
      ELSEIF(FTYPE.EQ.'DFHD40') THEN
         TLL=4.
         TLU=162.
         IF(TF.LT.TLL.OR.TF.GT.TLU) GOTO 100
         VISL=.5*(.0077474*EXP(-.020585*TF)+.0352158*TF**(-.880378))
         VISV=0.
         CKL=.38+9.759E-4*TF-4.0706E-6*TF**2
         CKV=0.
         SIG=0.
      ELSEIF(FTYPE.EQ.'DFHD50') THEN
         TLL=4.
         TLU=162.
         IF(TF.LT.TLL.OR.TF.GT.TLU) GOTO 100
         VISL=.5*(.0117985*EXP(-.022392*TF)+.0601634*TF**(-.952745))
         VISV=0.
         CKL=.34087+8.2641E-4*TF-3.3728E-6*TF**2
         CKV=0.
         SIG=0.
      ELSEIF(FTYPE.EQ.'DFHD60') THEN
         TLL=4.
         TLU=162.
         IF(TF.LT.TLL.OR.TF.GT.TLU) GOTO 100
         VISL=.5*(.0194357*EXP(-.025182*TF)+.123987*TF**(-1.07719))
         VISV=0.
```

```
    CKL=.30883+6.3162E-4*TF-2.6656E-6*TF**2
    CKV=0.
    SIG=0.
 ELSEIF(FTYPE.EQ.'F11   ') THEN
    TLL=0.
    TLU=150.
    IF(TF.LT.TLL.OR.TF.GT.TLU) GOTO 100
    CKL=9.5768E-2-2.4018E-4*TF+4.7302E-8*TF**2
    CKV=8.1527E-3+2.3475E-5*TF+1.8628E-7*TF**2
    VISL=5.6689E-4-6.5163E-6*TF+3.436E-8*TF**2-7.393E-11*TF**3
    VISV=1.0179E-5+3.0692E-8*TF+9.3987E-11*TF**2
    SIG=0.0
 ELSEIF(FTYPE.EQ.'F12   ') THEN
    TLL=0.
    TLU=100.
    IF(TF.LT.TLL.OR.TF.GT.TLU) GOTO 100
    CKL=-.00036907*TF+.07839
    CKV=4.8866E-13*TF**5+1.4967E-12*TF**4-3.415E-9*TF**3
#        +5.697E-8*TF**2+5.5499E-5*TF+.008341
    VISL=4.4004E-13*TF**4-2.1957E-10*TF**3+2.3831E-8*TF**2
#        -2.3225E-6*TF+.00026543
    VISV=4.6615E-14*TF**4+2.8129E-13*TF**3-1.31961E-10*TF**2
#        +4.7197E-8*TF+1.1779E-5
    SIG=0.0
 ELSEIF(FTYPE.EQ.'F13   ') THEN
    TLL=0.
    TLU=100.
    IF(TF.LT.TLL.OR.TF.GT.TLU) GOTO 100
    CKL=5.3061E-2-2.9112E-4*TF+1.7371E-6*TF**2
    CKV=1.3605E-2+1.5381E-4*TF+7.7252E-7*TF**2
    VISL=1.0132E-4-3.6831E-7*TF+2.8835E-8*TF**2
    VISV=1.4922E-5+1.163E-7*TF+1.2104E-9*TF**2+8.2369E-12*TF**3
    SIG=0.0
 ELSEIF(FTYPE.EQ.'F22   ') THEN
    TLL=-20.
    TLU=60.
    IF(TF.LT.TLL.OR.TF.GT.TLU) GOTO 100
    CKL=9.6174e-2-4.2368e-4*TF+3.4954e-7*TF**2
    CKV=9.5013E-3+6.0705E-5*TF-1.8442E-8*TF**2
    VISL=2.1019E-4-2.2755E-6*TF+1.0823E-8*TF**2
    VISV=1.18E-5+4E-8*TF
    SIG=0.0
 ELSEIF(FTYPE.EQ.'F23   ') THEN
    TLL=-40.
    TLU=10.
    IF(TF.LT.TLL.OR.TF.GT.TLU) GOTO 100
    CKL=7.4817E-2-7.547E-4*TF-1.4837E-6*TF**2
    CKV=1.7685E-2+2.1117E-4*TF+1.2942E-6*TF**2
    VISL=9.0309E-5-1.6701E-6*TF+4.2279E-9*TF**2
    VISV=1.5063E-5+1.3885E-7*TF+1.657E-9*TF**2
    SIG=0.0
 ELSEIF(FTYPE.EQ.'F113  ') THEN
    TLL=0.
    TLU=170.
    IF(TF.LT.TLL.OR.TF.GT.TLU) GOTO 100
    CKL=8.1947E-2-1.7689E-4*TF+1.3415E-8*TF**2
    CKV=7.7373E-3+4.3806E-5*TF+5.8518E-8*TF**2
    VISL=9.4573E-4-1.1889E-5*TF+7.0746E-8*TF**2-1.6326E-10*TF**3
    VISV=8.1603E-6+8.025E-8*TF-6.2726E-10*TF**2+2.3745E-12*TF**3
    SIG=0.0
 ELSEIF(FTYPE.EQ.'F114  ') THEN
    TLL=-40.
```

```
        TLU=115.
        IF(TF.LT.TLL.OR.TF.GT.TLU) GOTO 100
        CKL=7.0594E-2-2.389E-4*TF+4.0843E-9*TF**2
        CKV=8.1576E-5*TF+.00882409
        VISL=4.9155E-4-6.5554E-6*TF+4.3739E-8*TF**2-1.2737E-10*TF**3
        VISV=1.0578E-5+3.2039E-8*TF+1.2712E-10*TF**2
        SIG=0.0
      ELSEIF(FTYPE.EQ.'ENG   ') THEN
        TLL=0.
        TLU=156.
        IF(TF.LT.TLL.OR.TF.GT.TLU) GOTO 100
        VISL=3.8022-.31428*TF+.013304*TF**2-3.5893*TF**3
     #      +6.4967E-6*TF**4-7.9143E-8*TF**5+6.3585E-10*TF**6
     #      -3.2142E-12*TF**7+9.2318E-15*TF**8-1.146E-17*TF**9
        VISV=0.
        CKL=.147-7.18122E-5*TF-7.0708E-7*TF**2+3.6242E-9*TF**3
        CKV=0.
        SIG=0.
      ELSEIF(FTYPE.EQ.'ETHYLG') THEN
        TLL=0.
        TLU=100.
        IF(TF.LT.TLL.OR.TF.GT.TLU) GOTO 100
        VISL=.06442-.0041299*TF+1.4896E-4*TF**2-3.2665E-6*TF**3
     #      +4.1721E-8*TF**4-2.8171E-10*TF**5+7.4264E-13*TF**6
        VISV=0.
        CKL=.24194+3.205E-4*TF+2.4239E-6*TF**2-4.34E-9*TF**3
     #      -1.1914E-9*TF**4-8.8208E-12*TF**5
        CKV=0.
        SIG=0.
      ELSEIF(FTYPE.EQ.'GLYCRN') THEN
        TLL=0.
        TLU=46.
        IF(TF.LT.TLL.OR.TF.GT.TLU) GOTO 100
        VISL=10.457-.98243*TF+.03866*TF**2-7.1174E-4*TF**3
     #      +5.0052E-6*TF**4
        VISV=0.
        CKL=.282065+3.4417E-4*TF-1.0215E-5*TF**2+1.0936E-7*TF**3
        CKV=0.
        SIG=0.
      ELSEIF(FTYPE.EQ.'HG    ') THEN
        TLL=0.
        TLU=326.
        IF(TF.LT.TLL.OR.TF.GT.TLU) GOTO 100
        VISL=.0016815-6.15448E-6*TF+1.96029E-8*TF**2-2.4526E-11*TF**3
        VISV=0.
        CKL=8.1777+.013656*TF-6.49055E-6*TF**2
        CKV=0.
        SIG=0.
      ENDIF
      GOTO 5
 100   ITR=1
 5     RETURN
      END
C       _______________________
      SUBROUTINE SIFLUID(CYCLE,LFTYPE,FGIVEN,SGIVEN,QL,ENTR,ENTH,VOL,
     # U,TEMP,PRESS,SL,HL,VL,UL,SV,HV,VV,UV)
      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   ',
     #   'F21   ','F22   ','F23   ','F113  ','F114  '/
```

```
      DATA(ICARBN(I),I=1,12)/'METH  ','ETH    ','PROP  ','BUT    ',
     #   'PENT  ','HEX   ','HEPT  ','OCT    ','ETHYL ','PROPYL',
     #   'ISOB  ','ISOP  '/
      PARAMETER (M2=6)
      DO 100 I=1,9
      IF(LFTYPE.EQ.IFREON(I)) GOTO 400
  100 CONTINUE
      DO 200 I=1,12
      IF(LFTYPE.EQ.ICARBN(I)) GOTO 500
  200 CONTINUE
      IF(LFTYPE.EQ.'H2O   ') GOTO 300
c       WRITE(M2,9000) LFTYPE
        VOL=-9999
      RETURN
  300 LN=10
      NS=0
      NP=0
      CALL SIWATER(CYCLE,FGIVEN,SGIVEN,LN,NS,NP,QL,ENTR,ENTH,VOL,U,
     # TEMP,PRESS,SL,HL,VL,UL,SV,HV,VV,UV,NERROR)
      RETURN
  400 CALL SIFREON(CYCLE,FGIVEN,SGIVEN,LFTYPE,ENTH,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,
     # 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
EVAL
     #UATED')
      END
C
      SUBROUTINE WATER(CYCLE,FGIVEN,SGIVEN,CALLIN,NSTAT1,NPRINA,QUAL1,
     # ENTR1,ENTH1,VOL1,UINT1,TEMPF1,PRESS1,LSATS1,LSATH1,LSATV1,
     # 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,
     # TEMPK,TCRIT,PCRIT,TAUCRT,TGUESS,GAS,GAS1,CONVR,DPSDT,DGUESS,
     # DENSIT,DPDRHO,TEMPF,PRESS,UINT,ENTH,VOL,ENTR,DHDT,DHDRHO,DPDT,
     # SCAL,HCAL,DSDRHO,SJOULE,SKNOWN,ITERAT,PSAT,DENTOL,TTOLER,BTOLER,
     # PCAL,QUAL,PKNOWN,DSDT
      COMMON/PRINTA/NPRINT,NSTATE
      DATA TTOLER,BTOLER,CONVR/1.001,0.999,0.4299229/
      DATA TCRIT,PCRIT,TAUCRT,GAS1,GAS/374.136,3203.6,1.544912,
     #  66.9366,0.46151/
      DATA ITERAT/2000/
      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')GOTO 1930
      IF(CYCLE.EQ.'TX'.OR.CYCLE.EQ.'PX')GOTO 1720
      IF(CYCLE.EQ.'PS')GOTO 1100
      IF(CYCLE.EQ.'PH')GOTO 1330
C     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) GOTO 1200
      TEMPK=500.0
        IF(ENTR.GT.1.0580) DENSIT=1.0E-05
      IF(ENTR.GT.1.0580) GOTO 1550
      DENSIT=0.96
      GOTO 1630
 1200 CALL TSATUR
      CALL VAPDEN(NSTAT1)
      IF(ENTR/SCAL.GT.TTOLER) GOTO 1550
      CALL LIQDEN(NSTAT1)
      IF(ENTR/SCAL.LT.BTOLER) GOTO 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
      GOTO 2200
C ______________PRESSURE ENTHALPY_____________
 1330 PKNOWN=FGIVEN
      HKNOWN=SGIVEN
      PRESS=FGIVEN
      ENTH=SGIVEN
      IF(NPRINT.EQ.1) WRITE(M2,9060)
      IF(PRESS.LT.PCRIT) GOTO 1400
      TEMPK=500.0
      IF(ENTH.GT.902.5) DENSIT=1.0E-05
      IF(ENTH.GT.902.5) GOTO 1550
      DENSIT=0.96
      GOTO 1630
 1400 CALL TSATUR
      CALL VAPDEN(NSTAT1)
      IF(HKNOWN/HCAL.GT.TTOLER) GOTO 1550
      CALL LIQDEN(NSTAT1)
      IF(HKNOWN/HCAL.LT.BTOLER) GOTO 1630
      QUAL=(HKNOWN-LSATH)/(VSATH-LSATH)
      ENTR=(1.-QUAL)*LSATS + QUAL*VSATS
      UINT=(1.0-QUAL)*LSATU+QUAL*VSATU
      VOL=(1.0-QUAL)*LSATV + QUAL*VSATV
      CALL PRINT
      GOTO 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
      GOTO 2200
 1630 DENTOL=1.0E-05
      IF(CYCLE.EQ.'PH') GOTO 1680
      TEMPK=180.0*(5.4+9.630628*ENTR-2.640989*SQRT(ENTR)-0.10592*
     # ENTR**2)
      DENSIT=10.0/(62.4283*(0.12-.5276609*ENTR+.2878426*SQRT(ENTR)
     # +.4208526*ENTR**2))
      CALL PSFIND
      GOTO 1700
 1680 TEMPK=1.8*(512.0+1.370531*ENTH+4.737845*SQRT(ENTH)+4.540232E-04*
     # ENTH**2)
      DENSIT=1.E+04/(62.4283*(149.2-.185626*ENTH+2.671239*SQRT(ENTH)+
     # .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
      GOTO 2200
 1720 QUAL=SGIVEN
      IF(QUAL.GE.0.0) GOTO 1780
C     WRITE(M2,9020)
      QUAL=0.0
 1780 IF(QUAL.LE.1.0) GOTO 1810
c     WRITE (M2,9030)
      QUAL=1.0
C     _______________________PRESSURE QUALITY________________
 1810 IF(CYCLE.EQ.'TX')GOTO 1870
      PRESS =FGIVEN
      PKNOWN=FGIVEN
      IF(NPRINT.EQ.1) WRITE(M2,9070)
      IF(PRESS.LE.PCRIT*0.99) GOTO 1890
      NERROR=1
c     WRITE(M2,9100)
      RETURN
 1890 CALL TSATUR
      GOTO 2120
C     _____________TEMPERATURE-QUALITY __________________
 1870 TEMPF=FGIVEN
      IF(NPRINT.EQ.1) WRITE(M2,9080)
      IF(TEMPF.LE.705.0) GOTO 1900
      NERROR=1
c     WRITE(M2,9110)
      RETURN
 1900 TEMPK=(TEMPF+459.67)/1.8
      CALL PSATUR
      PRESS=PSAT
      GOTO 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) GOTO 1960
      IF(PRESS.GT.PCRIT)GOTO 2080
      GOTO 2035
 1960 CALL PSATUR
      IF(PRESS/PSAT.GT.TTOLER.OR.PRESS/PSAT.LT.BTOLER)GOTO 2030
c     WRITE(M2,9040)CALLIN,TEMPF,PRESS
      NERROR=1
      RETURN
 2030 IF(PRESS/PSAT.GT.TTOLER)GOTO 2080
 2035 QUAL=2.0
      DGUESS=1.E-05
      DENTOL=1.0E-10
      GOTO 2090
 2080 DGUESS=0.99
      DENTOL=1.E-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
      GOTO 2200
 2120 IF(ABS(QUAL).LT.0.000001) GOTO 2150
      CALL VAPDEN(NSTAT1)
      IF(ABS(QUAL-1.0).LT.0.000001) GOTO 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(750 DEG.F), WATER
TERMI
     2NATED')
      RETURN
      END
C    __________________________
      SUBROUTINE TSATUR
      IMPLICIT REAL(L)
      PARAMETER (M2=6)
      COMMON/WATER1/VSATS,VSATH,VSATU,VSATV,LSATS,LSATH,LSATU,LSATV,
     # TEMPK,TCRIT,PCRIT,TAUCRT,TGUESS,GAS,GAS1,CONVR,DPSDT,DGUESS,
     # DENSIT,DPDRHO,TEMPF,PRESS,UINT,ENTH,VOL,ENTR,DHDT,DHDRHO,DPDT,
     # SCAL,HCAL,DSDRHO,SJOULE,SKNOWN,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) GOTO 2330
      TSAT=TEMPK
 2320 CONTINUE
c     WRITE(M2,9000) ITERAT
 9000 FORMAT(1H ,'*** SATURATED TEMPERATURE SEARCH FAILED TO
     # CONVERGE IN',I5,' ITERATIONS, RESULTS MAY NO BE VALID***')
 2330 TSAT=TEMPK
      RETURN
      END
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.E-06
      DENTOL=1.E-12
      CALL DFIND(NSTAT1)
      VSATV=VOL
      VSATU=UINT
      VSATS=SCAL
      VSATH=HCAL
      RETURN
```

```
      END
C     ____________________
      SUBROUTINE LIQDEN(NSTAT1)
C  SET UP SEARCH FOR SATURATED LIQUID
      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
      END
C_________________________
      SUBROUTINE PRINT
C 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
c     WRITE(M2,9040) NSTATE
c     WRITE(M2,9050) TEMPF,PRESS
      IF(QUAL.LE.1.0.AND.QUAL.GE.0.0) GOTO 3280
c     IF(QUAL.GT.1.0) WRITE(M2,9060)
c     IF(QUAL.LT.0.0.AND.QUAL.GT.-2.5) WRITE(M2,9070)
c     IF(QUAL.LT.-2.5) WRITE(M2,9030)
      GOTO 3370
 3280   continue
c       WRITE(M2,9080) QUAL
c     WRITE(M2,9130)
c     WRITE(M2,9090) VSATU,LSATU
c     WRITE(M2,9100) VSATH,LSATH
c     WRITE(M2,9110) VSATS,LSATS
c     WRITE(M2,9120) VSATV,LSATV
 3370   continue
c       WRITE(M2,9140)
c     WRITE(M2,9090) UINT
c     WRITE(M2,9100) ENTH
c     WRITE(M2,9110) ENTR
c     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 EQUILIBRIUM',/,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 ,16X,'SATURATED',8X,'SATURATED',/,47X,'VAPOR',12X,
     1 'LIQUID',/,17X,60('*'))
 9140 FORMAT(1H ,/)
      RETURN
      END
C_____________________________
      SUBROUTINE PHFIND
C  NEWTON-RAPHSON ITERATION TO FIND DENSITY AND TEMPERATURE GIVEN
C   PRESSURE AND ENTHALPY FOR SUPERHEATED VAPOR AND COMPRESSED LIQUID
      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=1.001
      BT=.999
      DENT=1.0E-11
      DLAST=DENSIT
      TLAST=TEMPK
      CALL PROPER
      DO 3600 I=1,ITERAT
 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.
     1   ABS(TLAST-TEMPK).LT.1.0E-04)    RETURN
      DLAST=DENSIT
      TLAST=TEMPK
 3600 CONTINUE
c     WRITE(M2,9000) ITERAT
 9000 FORMAT(' ***TEMPERATURE-DENSITY SEARCH,GIVEN PRESSURE & ENTHALPY,
     1FAILED TO CONVERGE IN ',I5,' ITERATIONS, RESULTS MAY NOT BE
     2VALID')
      RETURN
      END
C    ___________________________
      SUBROUTINE PSFIND
C  NEWTON-RAPHSON ITERATIN TO FIND DENSITY AND TEMPERATURE
C  GIVEN PRESSURE
C  AND ENTROPY FOR SUPERHEATED VAPOR AND OCMPRESSED LIQUID
      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.E-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/
     1ENTR.LT.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
c     WRITE(M2,9000) ITERAT
 9000 FORMAT(' ***TEMPRATURE-DENSITY SEARCH GIVEN PRESSURE AND ENTROPY
     1FAILED TO CONVERGE IN ',I5,' ITERATIONS, RESULTS MAY NO BE VALID'
     2 )
      RETURN
      END
C
      SUBROUTINE PSATUR
C  COMPUTES SATURATION GIVEN SATURATION TEMPERATURE
      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 F(8)
      DATA(F(I),I=1,8)/-741.9242,-29.721,-11.55286,-0.8685635,
     # 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
     # -(TCRIT/TEMPK-TEMPC/TEMPK)*SUM2)
      RETURN
      END
C
      SUBROUTINE DFIND(NSTAT1)
C  NEWTON RAPHSON METHOD TO LOCATE DENSITY GIVEN PRESSURE
C  AND TEMPERATURE
      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.E-10) DENSIT=LASTD2/10.0
      CALL PROPER
```

```
      IF(PCAL/PRESS.GT.BTOLER.AND.PCAL/PRESS.LT.TTOLER.OR.
     # ABS(LASTD2-DENSIT).LT.DENTOL) RETURN
      LASTD2=DENSIT
 2560 CONTINUE
c     WRITE(M2,9000) ITERAT,PRESS,PCAL
 9000 FORMAT(' *** DENSITY SEARCH FAILED TO CONVERGE IN',I5,'
     1 ITERATIONS, RESULTS MAY NOT BE VALID***',/,
     2' DESIRED PRESSURE=',F10.3,' CALCULATED PRESSURE=',F10.3)
c     WRITE(M2,9001) NSTAT1
 9001 FORMAT(1H ,'STATE NUMBER',I3)
      RETURN
      END
C  ______________________________
      SUBROUTINE PROPER
C  COMPUTES PROPERTIES AND THEIR DERIVATIVES
      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,
     # -0.1564104,-6.3972405,-3.9661401,-0.69048554,-132.13917,
     # 7.7779182,-26.149751,-0.72546108,26.409282,15.453061,
     # 2.7407416,274.64632,-33.301902,65.326396,-9.2734289,
     # -47.740374,-29.142470,-5.1028070,-360.93828,-16.254622,
     # -26.181978,4.312584,56.32313,29.568796,3.9636085,342.18431,
     # -177.31074,0.0,0.0,0.0,0.0,0.0,-244.50042,127.48742,
     # 0.0,0.0,0.0,0.0,0.0,155.18535,137.46153,0.0,0.0,0.0,
     # 0.0,0.0,5.9728487,155.97836,0.0,0.0,0.0,0.0,0.0,
     # -410.30848,337.31180,-137.46618,6.7874983,136.87317,
     # 79.84797,13.041253,-416.0586,-209.88866,-733.96848,
     # 10.401717,645.8188,399.1757,71.531353 /
      DATA(C(I),I=1,8) /1857.065,3229.12,-419.465,36.6649,-20.5516,
     # 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) GOTO 2680
      DCON=1.0
      TAUCON=2.5
      GOTO 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
      END
C
      ____________________________
      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=.1450237743897283*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=.1450377*SGIVEN
      ELSEIF(CYCLE.EQ.'TX') THEN
      FGIVEN=FGIVEN*1.8+32.
      ELSEIF(CYCLE.EQ.'PS') THEN
      FGIVEN=.1450377*FGIVEN
      SGIVEN=SGIVEN/4.1868
      ELSEIF(CYCLE.EQ.'PH') THEN
      FGIVEN=.1450377*FGIVEN
      SGIVEN=SGIVEN/2.326
      ELSEIF(CYCLE.EQ.'PX') THEN
      FGIVEN=FGIVEN*.1450377
      ELSEIF(CYCLE.EQ.'TV') THEN
      FGIVEN=FGIVEN*1.8+32.
      SGIVEN=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/.1450377
      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
      SUBROUTINE CARBON(CYCLE,FGIVEN,SGIVEN,LFTYPE,H,S,V,TR,P,NPRIN,
     # 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/459.67,1.001,0.999,100,10.7335/
      DATA CONVR/0.185057/
      DATA(XISOB(I),I=1,27)/479907.0E-01,37264.0,8.58663,1.87890,
     #406763.0E05,101413.0E05,216863.0E02,853176.0E06,840860.0E08,
     #4.23987,7.11486,0.86401,9.079E-04,-2.073E-07,0.885E-10,
     #-0.241E-13,0.0,-776.11,6.872E-02,3.071E-04,0.0,0.0,0.0,
     #529.06,274.96,0.2373,58.12/
      DATA(PENT(I),I=1,27)/162185.0,51108.2,16.6070,2.44417,
     #135286.0E06,223931.0E05,388521.0E02,101769.0E07,
     #390860.0E08,7.06702,11.8593,
     #0.9179,1.511E-04,14.048E-07,-13.785E-10,4.560E-13,0.0,
     #-688.09,22.267E-02,-0.89E-04,5.018E-07,-3.048E-10,
     #0.641E-13,489.51,385.42,0.2007,72.146/
      DATA(PROP(I),I=1,27)/40066.4,18634.7,5.46248,
     #0.964762,274461.0E05,
     #796178.0E04,150520.0E02,453708.0E06,256053.0E08,2.01402,4.56182,
     #0.9353,26.158E-04,-56.478E-07,85.177E-10,-63.595E-13,18.451E-16,
     #-804.84,25.477E-02,-1.602E-04,5.160E-07,-2.110E-10,0.0,617.400,
     #206.13,0.3021,44.094/
      DATA(XMETH(I),I=1,27)/2574.89,7520.29,0.925404,0.723251,
     #437222.0E03,271092.0E03,47489.1,107737.0E05,301122.0E05,0.468828,
     #1.4860,1.7491,68.131E-04,-167.165E-07,244.251E-10,-180.176E-13,
     #52.320E-16,-1798.16,53.978E-02,-1.345E-04,0.676E-07,1.966E-10,
     #-1.037E-13,673.1,-116.43,0.62747,16.042/
      DATA(ETH(I),I=1,27)/22404.5,13439.3,3.11206,0.826059,681826.0E04,
     #295195.0E04,702189.0,257477.0E06,146819.0E08,0.90681,2.99656,
     #1.3021,23.761E-04,-32.627E-07,28.695E-10,-9.485E-13,0.0,-993.32,
     #30.668E-02,-1.427E-04,4.203E-07,-1.567E-10,0.0,709.8,90.03,
     #0.4218,30.068/
      DATA(PROPYL(I),I=1,27)/81880.4,6051.36,7.64114,0.114457,
     #29141.0E05,974762.0E04,54193.5E02,705921.0E06,341250.0E08,
     #1.36532,4.07919,1.2501,7.946E-04,-1.155E-07,-0.025E-10,0.054E-13,
     #0.0,376.86,9.868E-02,2.347E-04,0.295E-07,-0.196E-10,0.0,670.27,
     #197.4,0.3449,42.08/
      DATA(HEX(I),I=1,27)/434517.0,45333.1,29.4983,2.66233,318412.0E06,
     #526067.0E05,327460.0E02,552158.0E07,626433.0E09,9.70230,14.8720,
```

```
     #1.3746,-17.514E-04,35.508E-07,-21.907E-10,4.769E-13,0.0,
     #-618.38,.240E-02,4.148E-04,-.614E-07,
     #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
     #,228430.0E05,349220.0E02,142115.0E07,241326.0E08,6.16154,11.7384,
     #.7737,9.412E-04,-3.298E-07,2.306E-10,-.953E-13,.129E-16,
     #-709.28,7.482E-02,2.545E-04,.857E-07,
     #-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
     #,137436.0E05,364238.0E02,333159.0E06,230902.0E07,4.00985,7.54122,
     #0.9008,10.06E-04,-3.006E-07,0.846E-10,0.0,0.0,-739.77,22.389E-02,
     #-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,
     #615662.0E05,8351150.0,777123.0E07,636251.0E07,21.8782,24.7604,
     #1.3170,-17.863E-04,36.047E-07,-22.278E-10,4.856E-13,0.0,-596.40,
     #0.549E-02,3.989E-04,-0.587E-07,0.0,0.0,396.8,
     #512.85,0.1465,100.198/
      DATA(OCT(I),I=1 ,27)/131646.0,81690.6,10.5907,
     #4.86965,642053.0E06,
     #996546.0E05,185906.0E03,790575.0E07,346419.0E08,34.5124,21.9888,
     #.2641,-17.734E-04,35.827E-07,-22.131E-10,4.823E-13,0.0,-577.59,
     #0.612E-02,4.094E-04,-0.605E-07,0.0,0.0,364.
     #4,563.79,0.1284,114.224/
      DATA(ETHYL(I),I=1,27)/15978.1,12133.9,
     #2.62914,0.747945,409725.0E04
     #,163203.0E04,903550.0,517563.0E05,161706.0E05,0.589158,2.27971,
     #1.3982,23.551E-04,-34.268E-07,30.347E-10,-10.071E-13,0.0,928.25,
     #32.949E-02,-2.477E-04,4.768E-07,-1.711E-10,0.0,
     #742.0,49.82,0.5035,28.05/
      ERFLAG=2.
      IF(LFTYPE.EQ.'ISOB  ') GOTO 10
      IF(LFTYPE.EQ.'PENT  ') GOTO 20
      IF(LFTYPE.EQ.'ISOP  ') GOTO 30
      IF(LFTYPE.EQ.'METH  ') GOTO 40
      IF(LFTYPE.EQ.'ETH   ') GOTO 50
      IF(LFTYPE.EQ.'PROP  ') GOTO 60
      IF(LFTYPE.EQ.'HEX   ') GOTO 70
      IF(LFTYPE.EQ.'BUT   ') GOTO 80
      IF(LFTYPE.EQ.'HEPT  ') GOTO 90
      IF(LFTYPE.EQ.'OCT   ') GOTO 100
      IF(LFTYPE.EQ.'ETHYL ') GOTO 110
      IF(LFTYPE.EQ.'PROPYL') GOTO 120
c     WRITE(M2,9050)lftype
      RETURN
   10 CALL SETUP(XISOB,PCRIT,TCRIT,DCRIT)
      GOTO 180
   20 CALL SETUP(PENT,PCRIT,TCRIT,DCRIT)
      GOTO 180
   30 CALL SETUP(XISOP,PCRIT,TCRIT,DCRIT)
      GOTO 180
   40 CALL SETUP(XMETH,PCRIT,TCRIT,DCRIT)
      GOTO 180
   50 CALL SETUP(ETH,PCRIT,TCRIT,DCRIT)
      GOTO 180
   60 CALL SETUP(PROP,PCRIT,TCRIT,DCRIT)
      GOTO 180
   70 CALL SETUP(HEX,PCRIT,TCRIT,DCRIT)
      GOTO 180
   80 CALL SETUP(BUT,PCRIT,TCRIT,DCRIT)
      GOTO 180
```

```
   90 CALL SETUP(HEPT,PCRIT,TCRIT,DCRIT)
      GOTO 180
  100 CALL SETUP(OCT,PCRIT,TCRIT,DCRIT)
      GOTO 180
  110 CALL SETUP(ETHYL,PCRIT,TCRIT,DCRIT)
      GOTO 180
  120 CALL SETUP(PROPYL,PCRIT,TCRIT,DCRIT)
  180 IF(CYCLE.EQ.'TP')GOTO 200
      IF(CYCLE.EQ.'TX') GOTO 400
      IF(CYCLE.EQ.'PX') GOTO 600
      IF(CYCLE.EQ.'PS') GOTO 800
      IF(CYCLE.EQ.'PH') GOTO 1000
      IF(CYCLE.EQ.'TV') GOTO 1200
c     WRITE(M2,9000) CYCLE
      RETURN
  200 T=FGIVEN
      P=SGIVEN
      TR=T+ABST
      IF(T.GE.TCRIT) GOTO 220
      TCRITR=TCRIT+ABST
      CALL PRESAT(TR,PSAT,DL,DV,DCRIT,TCRITR,LFTYPE)
      IF(P/PSAT.GT.TTOLER.OR.P/PSAT.LT.BTOLER) GOTO 210
c     WRITE(M2,9010) T,P
      ERFLAG=1.
      RETURN
  210 IF(P/PSAT.GT.TTOLER) GOTO 240
  220 QUAL=2.0
      D=1.0E-06
      GOTO 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) GOTO 410
c     WRITE(M2,9020)QUAL
      RETURN
  410   IF(T.LE.TCRIT) GOTO 420
c        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)GOTO 610
c     WRITE(M2,9020) QUAL
      RETURN
  610 IF(P.LE.PCRIT) GOTO 620
```

```
C     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) GOTO 820
      TR=TCRIT*1.1
      DV=1.01*DCRIT
      GOTO 840
  820 CALL TEMSAT(P,TR,DL,DV,TCRIT,DCRIT)
      CALL SATC(TR,DL,DV,HL,SL,HV,SV)
      IF(S.LT.0.0) GOTO 830
      IF(S/SV.GT.TTOLER) GOTO 840
      IF(S/SL.LT.BTOLER) GOTO 860
      GOTO 835
  830 IF(S/SV.LT.BTOLER) GOTO 840
      IF(S/SL.GT.TTOLER) GOTO 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
      GOTO 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) GOTO 1020
      TR=TCRIT*1.1
      DV=1.01*DCRIT
      GOTO 1040
 1020 CALL TEMSAT(P,TR,DL,DV,TCRIT,DCRIT)
      CALL SATC(TR,DL,DV,HL,SL,HV,SV)
      IF(H.LE.0.0)GOTO 1030
      IF(H/HV.GT.TTOLER) GOTO 1040
      IF(H/HL.LT.BTOLER) GOTO 1060
      GOTO 1035
 1030 IF(H/HV.LT.BTOLER)GOTO 1040
      IF(H/HL.GT.TTOLER) GOTO 1060
 1035 QUAL=(H-HL)/(HV-HL)
      S=SV*QUAL+SL*(1.0-QUAL)
      VL=1.0/(DL*WM)
```

```
      VV=1.0/(DV*WM)
        if (qual.lt.0.0.and.qual.gt.-2.0) qual=0.0
      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
      GOTO 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,QUALL,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) GOTO 1240
      CALL PRESAT(TR,P,DL,DV,DCRIT,TCRITR,LFTYPE)
      IF(D/DL.LT.BTOLER) GOTO 1260
      IF(D/DV.GT.TTOLER) GOTO 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
      GOTO 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,
     1'AND THE PRESSURE',F10.4,'DEFINE A SATURATION STATE, CARBON
     2 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')
      END
C     ______________________________
      SUBROUTINE SETUP(GIVEN,PCRIT,TCRIT,DCRIT)
C  THIS SUBROUTINE SETS UP THE REQUIRED COEFFICIENTS FOR THE PARTICULAR
C  FLUID TO BE ANALYZED
      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
      END
C    ______________________________
      SUBROUTINE VFIND(T,P,D)
C  THIS SUBROUTINE FINDS THE DENSITY GIVEN TEMPERATURE AND PRESSURE FOR
C  COMPRESSED LIQUID AND SUPERHEATED VAPOR STATES
      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
c     WRITE(M2,9000) LIMIT
 9000 FORMAT(1H ,4X,'***VFIND FAILED TO CONVERGE IN',I6,'ITERATIONS')
      RETURN
      END
C
      SUBROUTINE ENTH(T,D,H,DHDT,DHDD)
C  THIS SUBROUTINE CALCULATES THE ENTHALPY GIVEN DENSITY AND
C  TEMPERATURE
      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)*
     1 D**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*
     1 C(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
      END
C    ______________________________
      SUBROUTINE ENTR(T,D,S,DSDT,DSDD)
C  THIS SUBROUTINE CALCULATES THE ENTROPY GIVEN DENSITY AND TEMPERATURE
      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
     #+C(10)*C(7)*D**5/(5.0*T**2)+2.0*C(5)*TERM3/(C(11)*T**3))/WM+CS(1)
     #+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
     #+C(7)*D**2/T**3-0.4*C(10)*C(7)*D**5/T**3-6.0*C(5)*TERM3/
     #(C(11)*T**4))/WM+CS(2)+2.0*CS(3)*T+3.0*CS(4)*T**2+4.0*CS(5)*T**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)*
     #(D+C(11)*D**3)*EXP(-C(11)*D**2)/T**3)*CONVR/WM
      RETURN
      END
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
c     WRITE(M2,9000) LFTYPE
c     WRITE(M2,9010) T,P
      IF(QUAL.LE.1.0.AND.QUAL.GE.0.0) GOTO 3280
c     IF(QUAL.GT.1.0) WRITE(M2,9020)
c     IF(QUAL.LT.0.0) WRITE(M2,9030)
      GOTO 3370
 3280   continue
c       WRITE(M2,9040) QUAL
c     WRITE(M2,9050)
c     WRITE(M2,9070) HV,HL
c     WRITE(M2,9080) SV,SL
c     WRITE(M2,9060) VV,VL
 3370   continue
c       WRITE(M2,9090)
c     WRITE(M2,9070) H
c     WRITE(M2,9080) S
c     WRITE(M2,9060) V
 9000 FORMAT(1H1,4X,'***THERMODYNAMIC PROPERTIES OF ',A6,' USING
     #STARLING-BWR QUATION OF STATE***')
 9010 FORMAT(1H ,4X,'TEMPERATURE = ',F7.1,'DEG.F',/,5X,'PRESSURE    =
',
     #    F10.5,'PSIA')
 9020 FORMAT(1H ,4X,'SUPERHEATED VAPOR',//////)
 9030 FORMAT(1H ,4X,'COMPRESSED LIQUID',//////)
 9040 FORMAT(1H ,4X,'SATURATED LIQUID-VAPOR EQUILIBRIUM',/,5X,'QUALITY
=
     # ',F6.4)
 9050 FORMAT(1H ,//,35X,'SATURATED',8X,'SATURATED',/,35X,'VAPOR',12X,
     #    'LIQUID',/,5X,60('#'))
 9060 FORMAT(1H ,4X,'SPECIFIC VOLUME (CU.FT/LBM)',5X,F9.4,8X,F9.4,/////
     #    /)
 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
      END
C    ______________________________
      SUBROUTINE PRESAT(T,P,DL,DV,DCRIT,TCRITR,FTYPE)
C  THIS SUBROUTINE FINDS SATURATION PRESSURE AND SATURATED LIQUID AND
```

```
C  VAPOR DENSITIES GIVEN TEMPERATURE
      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  ') GOTO 10
      DL=DCRIT*(1.8+3.4*(1.0-TRED)-2.7*(1.0-TRED)**2.0)
      DV=DCRIT/(275.0*(1.0-TRED))
      GOTO 200
   10 IF(FTYPE.NE.'ISOP  ') GOTO 20
      DL=DCRIT*(1.8+3.4*(1.0-TRED)-2.7*(1.0-TRED)**2.0)
      DV=DCRIT/(225.0*(1.0-TRED))
      GOTO 200
   20 IF(FTYPE.NE.'PENT  ') GOTO 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)
      GOTO 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.
     #AND.PL/PV.GT.BTOLER) RETURN
      DLLAST=DL
      DVLAST=DV
 1000 CONTINUE
C     WRITE(M2,9000) LIMIT
C9000 FORMAT(                      )
      RETURN
      END
C   ___________________________________
      SUBROUTINE SATC(T,DL,DV,HL,SL,HV,SV)
C  THIS SUBROUTINE DETERMINES SATURATED LIQUID AND VAPOR PROPERTIES
      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
```

```
      END
C     ______________________________
      SUBROUTINE TEMSAT(P,T,DL,DV,TCRIT,DCRIT)
C   THIS SUBROUTINE FINDS THE SATURATION TEMPERATURE AND
C   SATURATED LIQUID AND
C   VAPOR DENSITIES GIVEN PRESSURE
      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.
     #AND.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(         )
C9000 FORMAT(         )
      RETURN
      END
C     ______________________________
      SUBROUTINE PSKNOW(P,S,T,D)
C   THIS SUBROUTINE FINDS DENSITY AND TEMPERATURE GIEN PRESSURE AND
C   ENTROPY
```

```
C FOR COMPRESSED LIQUID AND SUPERHEATED VAPOR STATES
      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.
     #AND.SCAL/S.GT.BTOLER)RETURN
      DLAST=D
      TLAST=T
 1000 CONTINUE
c       WRITE(M2,9000) LIMIT
 9000 FORMAT(1H ,4X,'***PSKNOW FAILED TO CONVERGE IN',I6,' ITERATIONS')
      RETURN
      END
C     ________________________________________
      SUBROUTINE PHKNOW(P,H,T,D)
C   THIS SUBROUTINE FINDS THE DENSITY AND TEMPERATURE GIVEN PRESSURE AND
C   ENTHALPY FOR COMPRESSED LIQUID AND SUPERHEATED VAPOR STATES
      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,LIMIT
      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.
     #AND.HCAL/H.GT.BT) RETURN
      DLAST=D
      TLAST=T
 1000 CONTINUE
c     WRITE(M2,9000)LIMIT
 9000 FORMAT(1H ,4X,'***PHKNOW FAILED TO CONVERGE IN',I6,' ITERATIONS')
      RETURN
      END
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)
     #*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
     #*TERM3/(R*T)+C(5)/(R*C(11)*T**3)*(3.0*C(11)*D+3.0*C(11)**2*D**3
     #-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)
     #-6.0*C(10)*TERM2*D**5/(5.0*R*T**2)-3.0*C(5)*TERM4/(C(11)*R*T**4))
      RETURN
      END
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*
     #EXP(-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
     #+3.00*C(5)*D**2*TERM4*EXP(-C(11)*D**2)/T**2-2.0*C(11)**2*D**6
     #*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)+D**3*(C(3)*R+C(7)/T**2)-D**6*C(10)*C(7)/T**2-2.0*C(5)*D**3
     #*TERM4*EXP(-C(11)*D**2)/T**3
      RETURN
      END
C    ________________________________________
      SUBROUTINE INVERT(XJACOB,F,DX)
C  THIS SUBROUTINE SOLVES THREE SIMULTANEOUS EQUATIONS
C  BY GUASSIAN ELIMINATION
      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)
      GOTO 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
      END
C    ____________________________________________
      SUBROUTINE FREON(CYCLE,FGIVEN,SGIVEN,FTYPE,H,S,V,T,PSIA,
     #  HV,SV,VV,HL,SL,VL,QUAL,NPRT)
C
C  THIS SUBROUTINE CALCULATES THE THERMODYNAMIC PROPERTIES OF FREONS
C  BY USING THE MARTIN-NOU EQUATION OF STATE.
```

```
C
      CHARACTER FTYPE*6,CYCLE*2
      PARAMETER (M2=6)
      DIMENSION F113(4,12),F114(4,12),F11(4,12),F12(4,12),F13(4,12),
     #F14(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-10,1.001,0.999/
      DATA((F113(I,J),I=1,4),J=1,12)/33.0655,-4330.98,-9.2635,
     #0.0020539,0.0,0.0,0.0,0.0,0.0,0.05728,0.0,417.4,122.872,
     #-4.035,0.002618,0.0,-0.0128,-0.0214,5.0E-05,0.0,-6.36E-05,0.0,
     #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,
     #0.0,0.0,25.198,-0.40552,0.1699,459.6,109.49,3.0,0.02781,0.0,
     #0.0,0.0,498.9/
      DATA ((F114(I,J),I=1,4),J=1,12)/27.071306,-5113.7021,-6.3086761,
     #6.913003E-04,0.78142111002,768.35,0.0,0.0,0.0,0.062780807,
     #5.914907E-03,294.35,36.32,-2.3856704,1.0801207E-03,-6.5643648,
     #0.021776,0.034055687,-5.3336494E-06,0.16366057,0.63649,
     #-3.857481E-04,0.0,0.0,6.7186,1.6017659E-06,6.2632341E-10,
     #-1.0165314E-05,1.97E-06,0.0,0.0,0.0,0.0175,3.49E-04,-1.67E-07,
     #0.0,0.0,25.33966211,-0.1151271756,0.15842,459.6,95.4,2.0,
     #0.027533,0.0,3.0,0.0,473.187/
      DATA((F11(I,J),I=1,4),J=1,12)/42.14702865,-4344.343807,
     #-12.84596753,0.004008372507,0.0313605356,862.07,0.0,0.0,0.0,
     #0.078117,0.00190,388.47,34.57,-3.126759,1.318523E-03,-35.769990,
     #57.63811,-0.025341,4.8751212E-05,1.220367,43.63220,1.687277E-03,
     #-1.805062E-06,0.0,-42.82356,-2.358930E-05,2.448303E-08,
     #-1.478379E-04,36.70663,1.057504E08,-9.472103E04,0.0,0.023815,
     #2.798823E-04,-2.123734E-07,5.999018E-11,-336.807030,50.5418,
     #-0.0918395,0.17219,459.6,112.080,1.0,0.028927,0.0,4.50,580.00,
     #639.50/
      DATA((F12(I,J),I=1,4),J=1,12)/39.88381727,-3436.632228,
     #-12.47152228,0.004730442442,0.0,0.0,0.0,0.0,0.0,0.088734,
     #0.00065093886,233.70,34.84,-3.409727134,0.0015943484B,
     #-56.7627671,0.02696,0.06023944654,-1.879618431E-05,1.311399084,
     #0.834921,-0.000548737007,0.0,0.0,6.02683,0.0,3.468834E-09,
     #-2.54390678E-05,-0.655549E-05,0.0,0.0,0.0,0.0080945,0.000332662,
     #-2.413896E-07,6.72363E-11,0.0,-39.55655122,-0.0165379361,0.1359,
     #459.6,78.86,2.0,0.02870,0.0,5.475,0.0,596.9/
      DATA((F13(I,J),I=1,4),J=1,12)/25.96797498,-2709.538217,
     #-7.172343913,0.00254515398,0.2803010913,546.00,0.0,0.0,0.0,
     #0.102728,0.0048,84.00,36.06996128,-3.083417,2.341695E-03,
     #-18.212643,0.01566,0.058854,-5.671268E-05,0.571958,1.110,
     #-1.026061E-03,1.338679E-06,0.0,6.665,5.290649E-06,-7.395111E-9,
     #-3.874233E-05,3.245E-05,7.378601E07,-7.435565E04,0.0,0.01602,
     #2.823E-04,-1.159E-07,0.0,0.0,20.911,-0.05676,0.08898,459.6,
     #45.271,2.0,0.0277239,0.0,4.00,625.00,561.30/
      DATA((F14(I,J),I=1,4),J=1,12)/20.7154389,-2467.505285,
     #-4.69017025,0.00064798076,0.770707795,424.0,0.0,0.0,0.0,
     #0.1219336,0.0015,-50.1,39.06,-2.162959,2.135114E-03,-18.941131,
     #69.56848907,4.404057E-03,1.282818E-05,0.539776,4.58661139,
     #1.921072E-04,-3.918263E-07,0.0,36.17166615,-4.481049E-06,
     #9.062318E-09,-4.836678E-05,-8.05898583,5.838823E07,-9.263923E04,
     #0.0,3.00559282E-02,2.37043352E-04,-2.85660077E-08,
     #-2.95338806E-11,0.0,86.102162,0.36172528,0.5284,459.6,100.636,
     #1.0,0.02560163,0.0,4.00,661.199997,543.16/
      DATA((F21(I,J),I=1,4),J=1,12)/42.7908,-4261.34,-13.0295,
     #0.0039851,0.0,0.0,0.0,0.0,0.0,0.10427,0.0,
     #812.9,116.37962,-7.316,0.00464210,
     #0.0,-0.03106808,-0.20382376,0.0003593,0.0,-0.0000501,0.0,0.0,0.0,
     #0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0427,0.000140,0.0,0.0,0.0,0.0,
     #0.0,0.1906,459.6,120.45,3.0,0.030675,0.0,0.0,0.0,750.0/
      DATA((F22(I,J),I=1,4),J=1,12)/29.35754453,
```

```
     #-3845.193152,-7.86103122,0.002190939044,
     #0.445746703,686.1,0.0,0.0,0.0,0.124098,
     #0.002,204.81,32.76,-4.353547,0.002407252,-44.066868,
     #54.6344093,-0.017464,7.62789E-05,
     #1.483763,36.74892,0.002310142,-3.605723E-06,
     #0.0,-22.2925657,-3.724044E-05,5.355465E-08,-1.845051E-04,
     #20.47328862,1.363387E08,-1.672612E05,0.0,0.02812836,2.255408E-04,
     #-6.509607E-08,0.0,257.341,62.4009,-0.0453335,0.16016,459.69,
     #91.329,1.0,0.030525,0.0,4.2,548.2,721.906/
      DATA((F23(I,J),I=1,4),J=1,12)/328.90853,-7952.76913,
     #-144.5142304,0.2421150182,0.0,0.0,0.0,
     #-2.128066524E-04,9.43495542E-08,0.153270,0.00125,
     #78.73,32.7758,-4.679499,3.472778E-03,-159.775232,63.37784,
     #-0.012475,7.733388E-05,5.941212,-25.30533,2.068042E-03,
     #-3.684238E-06,0.0,144.16182,-3.868546E-05,6.455643E-08,
     #-7.394214E-04,-106.13280,7.502357E07,-1.114202E05,0.0,0.07628087,
     #-7.561805E-06,3.9065696E-07,-2.454905E-10,0.0,0.0,0.0,0.1198,
     #459.6,60.77,1.0,0.030510,0.0,5.50,520.00,701.42/
      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    ') GOTO 10
      IF(FTYPE.EQ.'F114    ') GOTO 20
      IF(FTYPE.EQ.'F11     ') GOTO 30
      IF(FTYPE.EQ.'F12     ') GOTO 40
      IF(FTYPE.EQ.'F13     ') GOTO 50
      IF(FTYPE.EQ.'F14     ') GOTO 60
      IF(FTYPE.EQ.'F21     ') GOTO 70
      IF(FTYPE.EQ.'F22     ') GOTO 80
      IF(FTYPE.EQ.'F23     ') GOTO 90
c     WRITE(M2,9060) FTYPE
      RETURN
   10 DO 15 I=1,4
      DO 15 J=1,12
      Q(I,J)=F113(I,J)
   15 CONTINUE
      GOTO 350
   20 DO 25 I=1,4
      DO 25 J=1,12
      Q(I,J)=F114(I,J)
   25 CONTINUE
      GOTO 350
   30 DO 35 I=1,4
      DO 35 J=1,12
      Q(I,J)=F11(I,J)
   35 CONTINUE
      GOTO 350
   40 DO 45 I=1,4
      DO 45 J=1,12
      Q(I,J)=F12(I,J)
   45 CONTINUE
      GOTO 350
```

```
   50 DO 55 I=1,4
      DO 55 J=1,12
      Q(I,J)=F13(I,J)
   55 CONTINUE
      GOTO 350
   60 DO 65 I=1,4
      DO 65 J=1,12
      Q(I,J)=F14(I,J)
   65 CONTINUE
      GOTO 350
   70 DO 75 I=1,4
      DO 75 J=1,12
      Q(I,J)=F21(I,J)
   75 CONTINUE
      GOTO 350
   80 DO 85 I=1,4
      DO 85 J=1,12
      Q(I,J)=F22(I,J)
   85 CONTINUE
      GOTO 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') GOTO 400
      IF(CYCLE.EQ.'TX') GOTO 600
      IF(CYCLE.EQ.'PX') GOTO 700
      IF(CYCLE.EQ.'PS') GOTO 900
      IF(CYCLE.EQ.'PH') GOTO 1000
      IF(CYCLE.EQ.'TV') GOTO 1100
c     WRITE(M2,9000)
      RETURN
  400 TF=FGIVEN
      PSIA=SGIVEN
      T=TF+Q(1,11)
      IF(TF.LT.Q(4,3)) GOTO 430
      V=1.0/Q(1,4)
      GOTO 480
  430 CALL VPEQM(PSAT,T,1)
      IF(PSIA/PSAT.GT.TTOLER.OR.PSIA/PSAT.LT.BTOLER) GOTO 470
c     WRITE(M2,9010)
      RETURN
  470 IF(PSIA/PSAT.GT.TTOLER) GOTO 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) GOTO 610
c     WRITE(M2,9070) QUAL
      RETURN
  610 T=TF+Q(1,11)
      IF(TF.LE.Q(4,3)) GOTO 630
c     WRITE(M2,9020) TCRIT
      RETURN
  630 CALL VPEQM (PSIA,T,1)
      GOTO 800
  700 PSIA=FGIVEN
      QUAL=SGIVEN
      IF(QUAL.LE.1.0.AND.QUAL.GE.0.0) GOTO 710
c     WRITE(M2,9070) QUAL
      RETURN
  710 IF(PSIA.LE.Q(4,12)) GOTO 730
c     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)
      GOTO 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)) GOTO 910
      IF(S.LE.SCRIT) GOTO 905
      T=Q(4,3)+Q(1,11)
      V=1.0/Q(4,11)
      GOTO 940
 905    continue
c       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) GOTO 940
      IF(S/SL.LT.BTOLER) GOTO 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))GOTO 1020
      IF(H.GT.HCRIT) GOTO 1010
C     WRITE(M2,9050)
      RETURN
 1010 T=Q(1,11)+Q(4,3)+10.0
      V=1.0/Q(4,11)
      GOTO 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)GOTO 1040
      IF(H/HL.LT.BTOLER) GOTO 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)GOTO 1140
      CALL VPEQM(PSAT,T,1)
      CALL FLQDEN(VL,T,DVLDT)
      CALL VOLUME(T,PSAT,VV)
      IF(V/VL.LT.BTOLER) GOTO 1160
      IF(V/VV.GT.TTOLER)GOTO 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 ENTROP(T,VV,S,DSDT,DSDV)
      CALL ENTHAL(PSIA,T,VV,H,DHDT,DHDV)
      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, PX, PS, PH',//,5X,'FREON TERMINATED')
 9010 FORMAT(1H ,4X,'***ERROR*** TEMPERATURE AND PRESSURE ARE NOT
     #INDEPENDENT UNDER THE SATURATION DOME',//,5X,'FREON TERMINATED')
 9020 FORMAT(1H ,4X,'***ERROR*** A SATURATION STATE DOES NOT EXIST
     #FOR TEMPERATURES ABOVE THE CRITICAL TEMPERATURE',F10.3,'DEG.F',
     #//,5X,    'FREON TERMINATED')
 9030 FORMAT(1H ,4X,'***ERROR*** A SATURATION STATE DOES NOT EXIST FOR
     #PRESSURES ABOVE THE CRITICAL PRESSURE',F10.3,'PSIA',//,5X,
     #'FREON TERMINATED')
 9040 FORMAT(1H ,4X,'THE MARTIN-HOU EQUATION OF STATE IS NOT VALID IN
     #THIS 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
     #THIS REGION,2.E. P>PC AND H;HC',//,5X,'FREON TERMINATED')
 9060 FORMAT(1H ,4X,'***ERROR*** THE FLUID ',A6,' CAN NOT BE EVALUATED
     #USING THE SUBROUTINE FREON',//,5X,'FREON TERMINATED')
 9070 FORMAT(1H ,4X,'***ERROR*** A QUALITY OF ',F6.3,'IS NOT ALLOWED',
     #//,5X,'FREON TERMINATED')
      END
C    ____________________________________
      SUBROUTINE VPEQM(P1,T,M)
C  THIS SUBROUTINE CALCULATES THE SATURATION PRESSURE OR TEMPERATURE
C  GIVEN THE OTHER
      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) GOTO 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(FT)
      P2=10.0**LOGP
      IF(M.EQ.1) P1=P2
      DELP=P1-P2
      ERROR=DELP/P1
        IF(ABS(ERROR).LT..1D-06) GOTO 110
```

```
      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
c     WRITE(M2,115) LIMIT,P1,P2
  110 RETURN
  115 FORMAT(1H ,4X,'***VPEQM FAILED TO CONVERGE IN',I6,'ITERATIONS',
     # /,'DESIRED PRESSURE=',F10.3,'CALCULATED PRESSURE=',F10.3)
      END
C    ------------------------------------------------
      SUBROUTINE VOLUME(T,P,V)
C  THIS SUBROUTINE FINDS THE VAPOR DENSITY BY USE OF THE NEWTON-
C  RAPHSON ITERATION METHOD, GIVEN PRESSURE AND TEMPERATURE.
      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.
     #   1.0E-10) RETURN
      VLAST=V
 1000 CONTINUE
c     WRITE(M2,9000) LIMIT
 9000 FORMAT(11X,'***VOLUME FAILED TO CONVERGE IN ',I5,' ITERATIONS')
      RETURN
      END
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) GOTO 105
      IF(ABS(EKA).LE.DERROR) GOTO 105
      EKA=EXP(EKA)
      IF(ABS(Q(1,12)).LE.DERROR) GOTO 100
      PART=Q(1,12)*ALOG((Q(1,12)*EKA+1.0)/(Q(1,12)*EKA))
      GOTO 110
  100 PART=0.0
      GOTO 110
  105 EKA=0.0
      RX=0.0
      RZ=0.0
      PART=0.0
      TERMA=0.0
      TERMB=0.0
      GOTO 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-
     # 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)
     #  /(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)
     #  /(3.0*VMB3)+RZ)*(1.0+Q(2,12)*T/(Q(4,3)+Q(1,11)))*EKIT+Q(2,10)
      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*
     #   VMB4)+RZ
      PART 2=-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)/VMB5+TERMA)
      PART2=-Q(4,4)/VMB2-Q(4,5)/VMB3-Q(4,6)/VMB4-Q(4,7)/VMB5-TERMB
     #   +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
      END
C     _________________________
      SUBROUTINE ENTROP(T,V,S,DSDT,DSDV)
C  THIS SUBROUTINE COMPUTES THE ENTROPY OF THE VAPOR.
      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) GOTO 100
      IF(ABS(EKA).LT.DERROR) GOTO 100
      IF(ABS(Q(1,12)).LE.DERROR) GOTO 100
      EKA=EXP(EKA)
      PART=Q(1,12)*ALOG((Q(1,12)*EKA+1.0)/(Q(1,12)*EKA))
      GOTO 105
  100 RX=0.0
      RZ=0.0
      EKA=0.0
      PART=0.0
      TERMA=0.0
      TERMB=0.0
      GOTO 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)/
     #  (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) GOTO 115
      EKIT = EXP(EKIT)
      GOTO 120
```

```
  115 EKIT = 0.0
  120 ZZ=Q(2,12)**2.0/(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(1,10)
     #  /(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(3,7)/
     #  (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*VMB2)+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)/VMB5+TERMA)
      PART2=C*Q(2,12)*EKIT/TC*(-Q(4,4)/VMB2-Q(4,5)/VMB3-Q(4,6)/VMB4
     #    -Q(4,7)/VMB5-TERMB+Q(4,8)*Q(1,12)/(Q(1,12)*EKA+1.0))
      DSDV=PART1+PART2
  130 RETURN
      END
C  ______________________________-
      SUBROUTINE PRINTF(T,P,QUAL,V,H,S,VV,HV,SV,VL,HL,SL,FTYPE,NPRIN)
C  THIS SUBROUTINE PRINTS THE RESULTS OF ROUTINE FREON
      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
c     WRITE(M2,9000) FTYPE
c     WRITE(M2,9010) T,P
      IF(QUAL.LE.1.0.AND.QUAL.GE.0.0) GOTO 3280
c     IF(QUAL.GT.1.0) WRITE(M2,9020)
c     IF(QUAL.LT.0.0) WRITE(M2,9030)
      GOTO 3370
 3280   continue
c       WRITE(M2,9040) QUAL
c     WRITE(M2,9050)
c     WRITE(M2,9070) HV,HL
c     WRITE(M2,9080) SV,SL
c     WRITE(M2,9060) VV,VL
 3370   continue
c       WRITE(M2,9090)
c     WRITE(M2,9070) H
c     WRITE(M2,9080) S
c     WRITE(M2,9060) V
 9000 FORMAT(1H1,4X,'***THERMODYNAMIC PROPERTIES OF ',A6,' USING
     #   MARTIN-HOU EQUATION OF STATE***')
 9010 FORMAT(1H ,4X,'TEMPERATURE = ',F7.1,'DEG.F',/,5X,'PRESSURE    =
     # ',F10.5,'PSIA')
 9020 FORMAT(1H ,4X,'SUPERHEATED VAPOR',//////)
 9030 FORMAT(1H ,4X,'COMPRESSED LIQUID-APPROXIMATED BY SATURATED
     # LIQUID',//////)
 9040 FORMAT(1H ,4X,'SATURATED LIQUID-VAPOR EQUILIBRIUM',/,5X,'QUALITY
     # = ',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
      END
```

```
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(1,2))/T+Q(4,1)*XX)
      D2PDT2=P*(2.0*XX*(Q(2,1)+Q(1,2)*Q(2,2)*ALOG10(FT))/T3+
     # XX/T2*(0.43429448/FT)+(Q(3,1)-Q(1,2))/T2)
      RETURN
      END
C
   ___________________________________
      SUBROUTINE FLQDEN(VL,TR,DVLDT)
C  THIS SUBROUTINE COMPUTES THE SATURATED LIQUID SPECIFIC VOLUME GIVEN
C  TEMPERATURE USING ONE OF THREE EQUATONS
      COMMON/FREON1/Q(4,12),TTOLER,BTOLER,LIMIT,DERROR,NTYPE
      TC=Q(4,3)+Q(1,11)
  100 GOTO (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
     #    +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*
     #    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
      END
C
   _______________________
      SUBROUTINE SATN(PSIA,T,V,HV,SV,HL,SL,VL)
C  THIS SUBROUTINE COMPUTES THE SATURATED VAPOR-LIQUID PROPERTIES GIVEN
C  TEMPERATURE AND  SPECIFIC VOLUME
      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
      END
C     ______________________________
      SUBROUTINE SFIND(T,P,V,S)
C  THIS SUBROUTINE FINDS TEMPERATURE AND DENSITY GIVEN PRESSURE
C   AND ENTROPY
C  FOR SUPERHEATED VAPOR STATES.
      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.
     #  TTOLER.AND.SCAL/S.GT.BTOLER) GOTO 1050
      IF(ABS(T-TLAST).LT.1.0E-04.AND.ABS(V-VLAST).LT.1.0E-07) GOTO 1050
      VLAST=V
      TLAST=T
 1000 CONTINUE
c     WRITE(M2,9000) LIMIT
 1050 RETURN
 9000 FORMAT(' ',10X,'***SFIND FAILED TO CONVERGE IN ',I5,
     #' ITERATIONS')
      END
C     ______________________________
      SUBROUTINE COMPS(P,T,V,S,VL)
C  THIS SUBROUTINE FINDS THE TEMPERATURE AND DENSITY GIVEN
C  PRESSURE AND ENTROPY FOR COMPRESSED LIQUID STATES.
      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.
     # TTOLER.AND.SL/S.GT.BTOLER) GOTO 1050
      IF(ABS(T-TLAST).LT.1.0E-04.AND.ABS(V-VLAST).LT.1.0E-07) GOTO 1050
      VLAST=V
      TLAST=T
 1000 CONTINUE
c     WRITE(M2,9000) LIMIT
 1050 P=PCAL
      RETURN
 9000 FORMAT(' ',10X,'***COMPS FAILED TO CONVERGE IN ',I5,
     #' ITERATIONS')
      END
C   ______________________________________
      SUBROUTINE HFIND(T,P,V,H)
C  THIS SUBROUTINE FINS TEMPERATURE AND DENSITY GIVEN PRESSURE
C  AND ENTHALPY FOR SUPERHEATED VAPOR STATES.
      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.TTOLER.AND.HCAL/H.LT.
     # TTOLER.AND.HCAL/H.GT.BTOLER) GOTO 1050
      IF(ABS(T-TLAST).LT.1.0E-04.AND.ABS(V-VLAST).LT.1.0E-07) GOTO 1050
      VLAST=V
      TLAST=T
 1000 CONTINUE
c     WRITE(M2,9000) LIMIT
 1050 RETURN
 9000 FORMAT(' ',10X,'HFIND FAILED TO CONVERGE IN ',I5,' ITERATIONS')
      END
C   ______________________________________
      SUBROUTINE COMPH(P,T,V,H,VL)
C  THIS SUBROUTINE FINS THE TERMPERATURE AND DENSITY GIVEN
C  PRESSURE AND ENTHALPY FOR COMPRESSED LIQUID STATES
      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.
     #  TTOLER.AND.HL/H.GT.BTOLER) GOTO 1050
      IF(ABS(T-TLAST).LT.1.0E-04.AND.ABS(V-VLAST).LT.1.0E-07) GOTO 1050
      VLAST=V
      TLAST=T
 1000 CONTINUE
c     WRITE(M2,9000) LIMIT
 1050 P=PCAL
      RETURN
 9000 FORMAT(' ',10X,'***COMPH FAILED TO CONVERGE IN ',I5,
     #' ITERATIONS')
      END
C  _______________________________
      SUBROUTINE EOS(P,T,V,DPDT,DPDV,DPDVDT,D2PDT2)
C  THIS SUBROUTINE CALCULATES THE PRESSURE FOR VAPRO STATES AND
C  ITS DERIVATIVES GIVEN TEMPERATURE AND DENSITY
      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) GOTO 105
      IF(ABS(EKA).GT.30.0) GOTO 105
      EKA=EXP(EKA)
      EKA2=EXP(EKA2)
      GOTO 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) GOTO 120
      TERM(6)=(Q(2,8)+Q(3,8)*T+Q(4,8)*EKIT)/(EKA*(1.0+Q(1,12)*EKA))
  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)
  150 DPDV=DPDV+PART(I)
      IF(ABS(EKA).LE.DERROR) PART(6) =0.0
      IF(ABS(EKA).LE.DERROR) GOTO 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)*
     #  EKA2)**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) GOTO 100
      IF(EKA.GT.30.0) GOTO 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))
      GOTO 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+
     #    (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+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
      Z3=Q(3,6)-Q(2,12)*Q(4,6)*EKIT/TC
      Z4=Q(3,7)-Q(2,12)*Q(4,7)*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/
     #VMB6-Z5*Z6
  175 RETURN
      END
C     _________________________________
      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*0.01602
      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
```

BIBLIOGRAPHY

ASHRAE, *1989 Handbook: Fundamentals*, American Society of Heating, Refrigerating and Air Conditioning Engineers, Inc., 1989

Ay, H., Johnson, R., and Yang, W., "Direct-Contact Heat Transfer Between a Rising Dispersed Phase in a Counterflow Stream," *Numerical Heat Transfer*, Part A, Vol. 26, pp. 667-682, 1994.

Baker, M. J., "The British Gas High Temperature Direct Contact Water Heater," *Proceedings, Institution of Chemical Engineers Symposium, Innovation in Process Energy Utilisation*, Bath, U.K., 1987

Brickman, R., Sethi, H., and Boehm, R., "Analysis of a Direct Contact , Gas-Fired Water Heater," *Proceedings of the Tenth International Heat Transfer Conference*, Brighton, U.K., Vol. 4, pp. 357-362, 1994

Carslaw, H. S. and Jaeger, J. C., *Conduction of Heat in Solids*, Clarendon Oxford University Press, New York, 1959.

Coban, M., *A Numerical Modeling of Three-Phase Direct Contact Heat-Exchangers*, Ph.D. thesis, University of Utah, Salt Lake City, UT, 1986.

Coban, M. and Boehm, R., "Numerical and Experimental Modelling of Three Phase Direct Contact Heat Exchangers," Paper SA-08, *Proceedings ofthe 8th International Heat Transfer Conference*, Vol. 6, San Francisco, CA, 1986.

Culbreth, W. and Marschall, E., "Heat Transfer Phenomenon in the Region of Dispersed Phase Injection in a Direct-Contact Liquid-Liquid Heat Exchanger," *International Communications on Heat and Mass Transfer*, Pergamon Press Ltd., Vol. 10, pp. 211-223, 1983.

Cussler, E., *Diffusion Mass Transfer in Fluid Systems*, Cambridge University Press, 1984

Fan, L. and Tsuchiya, K., *Bubble Wake Dynamics in Liquids and Liquid-Solid Suspensions*, Butterworth-Heinemann, 1990.

Golafshani, M. and Jacobs, H., "Stability of a Direct-Contact Spray Column Heat Exchanger," *Journal of Heat Transfer*, Vol. 111, pp. 767-772, August, 1989.

Hong, K. and Saito, A., "Numerical Method for Direct Contact Melting in Transient Process," *International Journal of Heat and Mass Transfer*, Vol. 36, No. 8, pp. 2093-2103, 1993.

Jacobs, H., *Direct Contact Heat Transfer Studies on Sieve Tray Columns for Geothermal Applications*, D.O.E. report No. DE86-7470, U.S. Depart of Energy, Idaho Falls, ID, December, 1985.

Jacobs H., "Direct-Contact Heat Transfer for Process Technologies," *Journal of Heat Transfer*, Vol 110, pp. 1259-1270, 1988.

Johnson G. and Marschall, E., "A Comment on Liquid-Liquid Direct-Contact Heat Transfer Near the Dispersed Phase Injection," *Proceedings, National Heat Transfer Conference*, American Society of Mechanical Engineers Paper 85-HT-74, Denver, Colorado, 1985.

Kehat, E. and Letan, R., "The Role of Wakes in the Mechanism of Extraction in Spray Columns," A.I.Ch.E. Journal, Vol. 17, No. 4, pp. 984-990. July, 1971.

Kreith, F. and Boehm, R. F., *Direct Contact Heat Transfer*, Hemisphere Publishing, New York, 1988.

Letan, R. and Kehat, E., "Mixing Efffects in a Spray-Column Heat Exchanger," *A.I.Ch.E. Journal*, Vol. 11, No. 5, pp. 804-808, September, 1965.

Mansson, D. and Marschall, E., "An Experimental and Numerical Study of Heat Transfer to Liquid Drops Rising in a Viscous Liquid," *Proceedings of the 10th International Heat Transfer Conference*, G. Hewitt, Ed., Vol. 5, pp. 345-350, August, 1994.

Mines, G., Demuth, O., and Wiggins, D., *Thermal and Hydraulic Performance Tests of a Sieve-Tray Direct-Contact Heat Exchanger Vaporizing Pure and Mixed-Hydrocarbon Rankine-Cycle Working Fluids*, D.O.E. Report No. DE83-16428, U.S.D.O.E., Idaho Falls, ID, August, 1983.

Moresco, L. and Marschall, E. "Temperature Measurements in a Liquid-Liquid Direct-Contact Heat Exchanger," *American Institute of Chemical Engineers Symposium Series, Heat Transfer Conference*, San Diego, 1979.

Orlander, R., Oshmyansky, S., Nichols, K., and Werner, D., *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.

Raina, G., Wanchoo, R., and Grover, P., "Direct Contact Heat Transfer with Phase Change: Motion of Evaporating Droplets," *A.I.Ch.E. Journal*, Vol. 30, No. 5, pp. 835-837, 1984.

Riemer, D., Jacobs, H., and Boehm, R., *A Computer Program for Determining the Thermodynamic Properties of Water*, University of Utah Report No. UTEC 76-171, 1976.

Riemer, D., Jacobs, H., and Boehm, R., *A Computer Program for Determining the Thermodynamic Properties of Light Hydrocarbons*, University of Utah Report No. UTEC 76-211, 1976.

Riemer, D., Jacobs, H., and Boehm, R., *A Computer Program for Determining the Thermodynamic Properties of Freon Refrigerants*, University of Utah Report No. UTEC 76-212, 1976.

Roache, P. J., *Computational Fluid Dynamics*, Academic Press, 1972.

Sideman, S., "Direct Contact Heat Transfer Between Immiscible Liquids," *Advances in Chemical Engineering*, Drew, T., Hoopes, J. Jr., and Vermeulen, T., Eds., Vol 6, pp. 207-286, Academic Press, New York, 1966.

Sideman, S. and Gat, Y., "Direct Contact Heat Transfer with Change of Phase: Spray-Column Studies of a Three-Phase Heat Exchanger", *American Institute of Chemical Engineers Journal*, March, pp. 296-303, 1966.

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

Steinberger, R. and Treybal, R., *A.I.Ch.E. Journal*, Vol 6, pp. 227, 1963.

Summers, S. and Crowe, C., "One-Dimensional Numerical Model for a Spray Column Heat Exchanger," *A.I.Ch.E. Journal*, Vol. 37, No. 11, pp. 1673-1679, November, 1991.

Tadrist, L., Seguin, P., Santini, R., and Pantaloni, J., "Experimental and Numerical Study of Direct Contact Heat Exchangers," *International Journal of Heat and Mass Transfer*, Vol. 29, No. 6, pp. 1215-1227, 1985.

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

Tochitani, Y., Mori, Y. H., and Komotori, K., "Vaporization of Single Liquid Drops in an Immiscible Liquid, Part I: Forms and Motions of Vaporizing Drops," *Warme-und Stoffubertragung,* Vol 10, pp. 51-59, 1977.

Tochitani, Y., Mori, Y. H., and Komotori, K., "Vaporization of Single Liquid Drops in an Immiscible Liquid, Part II: Heat Transfer Characteristics," *Warme-und Stoffubertragung,* Vol 10, pp. 71-79, 1977.

Vargaftik, N. B., *Tables on the Thermophysical Properties of Liquids and Gases in Normal and Dissociated States,* Hemisphere Publishing, New York, 1975.

White, F. M., *Viscous Fluid Flow,* McGraw-Hill, New York, 1974.

White, F. M., *Viscous Fluid Flow,* McGraw-Hill, New York, 1991.