NISTIR 89-4133

EVSIM - AN EVAPORATOR SIMULATION MODEL ACCOUNTING FOR REFRIGERANT AND ONE DIMENSIONAL AIR DISTRIBUTION

Piotr A. Domanski

U.S. DEPARTMENT OF COMMERCE
National Institute of Standards and Technology
Gaithersburg, MD 20899

U.S. DEPARTMENT OF COMMERCE
Robert A. Mosbacher, Secretary
NATIONAL INSTITUTE OF STANDARDS AND TECHNOLOGY
Raymond G. Kammer, Acting Director

NIST

NISTIR 89-4133

EVSIM - AN EVAPORATOR SIMULATION MODEL ACCOUNTING FOR REFRIGERANT AND ONE DIMENSIONAL AIR DISTRIBUTION

Piotr A. Domanski

U.S. DEPARTMENT OF COMMERCE
National Institute of Standards and Technology
Gaithersburg, MD 20899

Sponsored by:
U.S. Department of Energy
Office of Building and Community Systems
Conservation and Renewable Energy
Washington, DC

August 1989

U.S. DEPARTMENT OF COMMERCE
Robert A. Mosbacher, Secretary
NATIONAL INSTITUTE OF STANDARDS AND TECHNOLOGY
Raymond G. Kammer, Acting Director

ABSTRACT

This report describes a computer model, EVSIM, of a refrigerant-to-air heat exchanger of the type used in residential air conditioning as an evaporator. The model provides performance predictions of a one-slab or two-slab evaporator for a given refrigerant enthalpy at the coil inlet, saturation temperature and superheat at the coil outlet, and at imposed one dimensional air mass flow distribution over the coil face.

The model accounts for air distribution and for complex refrigerant circuitry designs by simulating refrigerant distribution. Performance of the coil is calculated employing a tube-by-tube scheme. Performance of each tube is evaluated individually based on individual air and refrigerant mass flow rates and their respective thermodynamic states assigned for each tube. The modelling effort emphasis was on the local thermodynamic phenomena which were described by fundamental heat transfer equations and equations of state relationships among material properties.

This report includes a User's Guide and a listing written in FORTRAN 77. Due to the detailed algorithms and tube-by-tube performance evaluation scheme, mini and main frame computers are best suited for simulation studies using EVSIM. Nevertheless, the model converges on an IBM AT compatible machine within 2-6 minutes when simulating a single slab evaporator.

Key words: air conditioner; coil; evaporator; heat exchanger; modeling; simulation

ACKNOWLEDGEMENTS

This project was sponsored by the U.S Department of Energy with M. McCabe and D. Abramson monitoring the program.

The principal investigator of the program at NIST was Dr. D.A. Didion.

TABLE OF CONTENTS

DISCLAIMER

In view of the presently accepted practice of the building industry in the United States and the structure of the computer software used in this project, common U.S. units of measurement have been used in this report. In recognition of the United States as a signatory to the General Conference of Weights and Measures, which gave official status to the SI system of units in 1960, appropriate conversion factors have been provided in the table below. The reader interested in making further use of the coherent system of SI units is referred to: NBS SP330, 1972 Edition, 'The International System of Units,' or E380-72, ASTM Metric Practice Guide (American National Standard Z210.1).

METRIC CONVERSION FACTORS

Length	1 inch (in) = 25.4 millimeters (mm) 1 foot (ft) = 0.3048 meter (m)
Area	1 ft^2 = 0.092903 m^2
Volume	1 ft^3 = 0.028317 m^3
Temperature	F = 9/5 C + 32
Temperature Interval	1°F = 5/9°C or K
Mass	1 pound (lb) = 0.453592 kilogram (kg)
Mass Per Unit Volume	1 lb/ft^3 = 16.0185 kg/m^3
Energy	1 Btu = 1.05506 Kilojoules (kJ)
Specific Heat	1 Btu/[(lb)(°F)] = 4.1868 kJ/[(kg)(K)]
Gallon	1 gallon = 0.0037854 m^3

LIST OF TABLES

LIST OF FIGURES

LIST OF SYMBOLS

A = flow cross sectional area

ANNUL = length fraction of the tube with flow quality up to .85

A_f = fin surface area

$A_{p,i}$ = pipe inside area

$A_{p,m}$ = pipe mean surface area

A_o = pipe total outside surface area

Bo $= \dfrac{Q}{G \cdot i_{fg}}$, boiling number

C_p = heat capacity at constant pressure

D = diameter

d = indentation diameter of Vickers microhardness test, 25g load

FPI = number of fins per inch

F_i = fraction of refrigerant mass flow rate flowing through a given branch of refrigerant circuit

F_j = enhancement multiplier for lanced fins

f = friction factor

G = mass flux

Gz $= \dfrac{Re \cdot Pr \cdot D_H}{N \cdot S_1}$, Graetz number

g = gravitational acceleration

g_c = 32.2 ($ft \cdot lb/lb_f \cdot s^2$), dimensional constant

h = thermal conductance

$h_{c,o}$ = convection heat transfer coefficient at the exterior surface

$h_{D,o}$ = air-side mass transfer coefficient

h_i = inside tube heat transfer coefficient

h_{tf} = thermal conductance of the pipe-to-fin contact

I = tube expansion interference

i = enthalpy

i_{fg} = latent heat of evaporation

J = 778.17 ($lb_f \cdot ft/Btu$), mechanical equivalent of heat

j $= \dfrac{h \cdot Pr^{2/3}}{G \cdot C_{p,a}}$, j-factor

K_f $= \dfrac{J \cdot i_{fg} \cdot \Delta x}{L}$, Pierre's boiling number

k = thermal conductivity

L = length

Le $= \dfrac{h_{c,o}}{h_{D,o} \cdot C_{p,a}}$, Levis number

l_s = width of a strip in a lanced fin

M $= \left(\dfrac{2 \cdot h}{k_f \cdot t} \right)^{0.5}$, fin parameter, or molecular weight

m = mass flow rate

N = number of tube depth rows

n_s = number of strips in the enhanced zone area in a lanced fin

P = pressure

Q = heat transfer rate, or coil capacity

Pr $= \dfrac{\mu \cdot C_p}{k}$, Prandtl number

R = equivalent fin tip radius

Re $= \dfrac{G \cdot D}{\mu}$, Reynolds number

R_i = resistance to flow offered by a given branch refrigerant circuit

R' = condensation rate per unit width of a fin

r = $0.5 \cdot D_o$, outside radius of a tube

S_l = tube spacing in air flow direction (depth row pitch)

S_p = fin pattern depth for a wavy fin, per Figure 5

S_s = length of a strip of a lanced fin

S_t = tube spacing normal to air flow

s = spacing between adjacent fins

T = temperature

t = temperature or fin thickness

U = overall heat transfer coefficient

V = velocity

v = specific volume

w_a = humidity ratio of air

w_w = humidity ratio of saturated air

XDRY = length fraction of the tube with flow quality within the range 0.85 - 1.00

x = refrigerant flow quality

x_p = thickness of pipe wall

y = distance from the wall or tube pitch per Figure 4

δ = average thickness of condensate film

μ = absolute viscosity

ρ = density

ϕ = fin efficiency

$$\phi_s = \frac{\text{strip area}}{\text{total fin area}}$$

1. INTRODUCTION

This report describes a computer model of a refrigerant-to-air heat exchanger of the type used as an evaporator in almost all residential air-conditioning applications. A typical design is shown in Figure 1. The refrigerant flows inside tubes arranged in a certain circuit while the air passes outside the finned tubes. The air flows through the heat exchanger, evaporates the refrigerant, and cools down in the process. If the temperature of the air drops below the dew point temperature, water vapor contained in the air stream condenses on the heat exchanger surface.

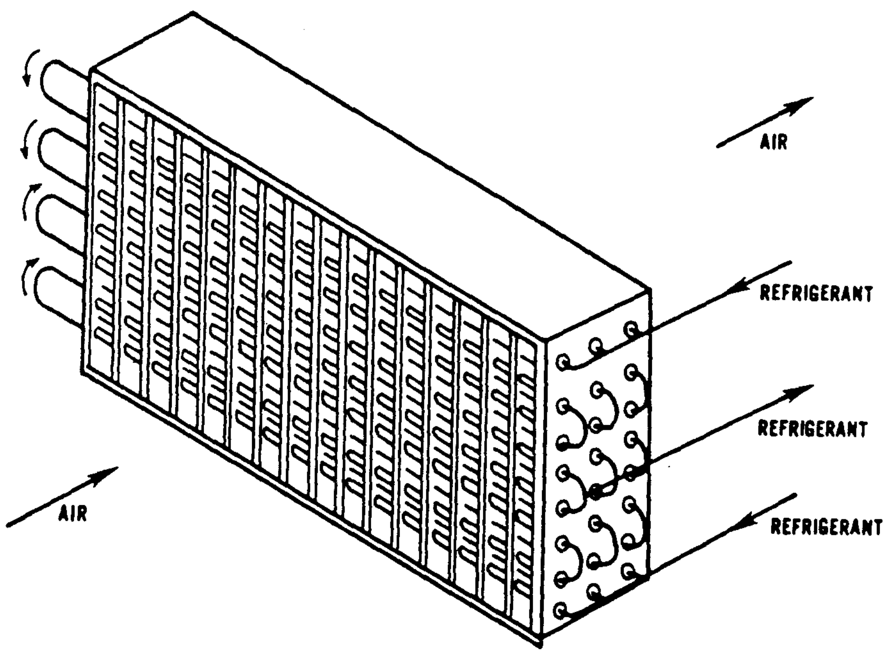


Figure 1. A schematic of a single-slab evaporator.

During flow through the heat exchanger the temperature of the air gradually decreases while the temperature of the refrigerant remains approximately constant as long as the refrigerant is in the two-phase state. Once refrigerant is fully evaporated, its temperature increases and the temperature difference between the refrigerant and air decreases. Air-conditioning systems are designed to operate at a small refrigerant superheat at the evaporator outlet. Capacity of the evaporator may be significantly reduced if the outlet superheat is a result of mixing of highly superheated vapor and two-phase flows leaving different circuits of the coil [1].

A number of design features impact the performance of an evaporator coil. Flat, wavy, lanced or louvered fins may be used on the air side. Tubes with a smooth inside surface are most popular but enhanced surfaces are also available. Different materials may be used; fins are normally made from aluminum while tubes are manufactured from aluminum or copper. Other significant design features are staggering pattern of tubes, tube diameter, number of tube depth rows, fin pitch and thickness, design of refrigerant circuitry, and coil configuration. The last two design aspects affect refrigerant and air distributions.

Manufacturing technique may also impact the heat exchanger performance. For example, contact resistance between a tube and fin may depend on the tool quality; outside surface treatment may promote better drainage of condensate.

The above description indicates that detailed modeling of an evaporator coil may be quite involved. Indeed, the complexity of all the processes associated with heat transfer between the air and refrigerant in a cross-flow, finned tube heat exchanger requires simplifying assumptions to be made during formulation of the model. However, a more detailed, fundamentally justified model has better chance to correctly predict performance of the modeled hardware.

A simple and popular way to determine coil capacity depends on utilizing performance catalogs of major manufacturers [2,3], or performance correlations developed by fitting the catalog data. Capacity data are presented in the form of charts for specific designs specified by the tube pattern, tube diameter, tube and fin materials and the shape, spacing and thickness of fins. These charts are usually developed for 45°F refrigerant saturation temperature, and 80°F dry bulb and 67°F wet bulb temperature of air. The convenience and quickness of using coil catalogs comes at the price of a few disadvantages; refrigerant circuity can be accounted for only in a rudimentary fashion, and extrapolation from the conditions specified in the catalog to other operating conditions may be questionable.

Computer based models which provide performance predictions through evaluation of heat transfer relationships are more versatile. Among models in the public domain, the evaporator model contained in the general heat pump model [4] was developed rather for system studies. It uses effectiveness vs. N_{tu} correlations and the assumption that the heat exchanger consists of equivalent parallel refrigerant circuits.

Another evaporator model, included in the heat pump model [5], is based on a tube-by-tube approach. Heat transfer in this model is evaluated for each tube independently based on the refrigerant temperature in the tube and the average air temperature for all tubes in a given row. The selection of tubes for performance evaluation is opposite to the refrigerant flow, i.e. from the outlet to the inlet. This backward scheme results in faster convergence of the vapor compression cycle simulation at the price of unrealistic imposition of the same refrigerant parameters at the outlet of each circuit. The refrigerant distribution in the model is estimated based on the circuitry layout and is maintained unchanged during coil simulation. The model assumes a uniform distribution of air and assigns the same air mass flow rate for each tube.

This report describes further development of the evaporator model presented in [5]. The new model, EVSIM, is also based on a tube-by-tube approach but has several new features. It can simulate performance of an evaporator coil with non-uniform, traverse to the tubes, one-dimensional air distribution. Refrigerant distribution is simulated based on the pressure drops in each circuit. A forward iteration scheme, from the coil inlet to outlet, is used which allows for more realistic modeling; refrigerant superheat at different circuit outlets may be different depending on the heat gained and mass flow in each circuit.

2. MODELING APPROACH

2.1 Tube-by-Tube Method

The model described in this report, EVSIM, is based on a tube-by-tube approach. Evaluation of performance for a single finned tube is the basic part of the model. Performance of each tube is analyzed separately one at a time. Each tube is associated with refrigerant parameters and specific air mass flow rate, inlet temperature and humidity.

Another important part of the model is the logic which assigns tubes for calculation in proper order, maintains inventory of air and refrigerant parameters for each tube, and maintains the convergence scheme.

Simulation starts with the refrigerant inlet tube of a given circuit and progresses consecutively to the following tubes until the outlet is reached. If the circuit splits, the model proceeds with calculations of one branch of the circuit and, when the exit is reached, returns to the split point to finish calculations on the remaining branches. Once calculations for the circuit are completed, the remaining circuits are calculated starting with the inlet tubes.

The selection scheme for tube calculations assures that refrigerant parameters are always known at the entrance of the tube; they are equal to the outlet parameters for the proceeding tube. At the outset of simulation the temperature and humidity ratio of air is known only for tubes in the first depth row and is estimated for other rows. These estimates are updated with new calculated values as calculations progress.

2.2 Air Distribution

The model can account for non-uniform air distribution between coil tubes. Each tube itself is assumed to have uniform air distribution over its length. Air distribution data have to be provided to the model in the form of values of air velocity at the coil face at discrete points on the center plane, perpendicular to coil tubes (specification of air velocity at the coil face is explained in Appendix A). From these data, the model derives velocity distribution for the face of the coil and determines the air mass flow rate associated with each tube in the first depth row.

Air mass flow rates associated with tubes past the first row are calculated based on mass flow rates associated with the preceding tubes. It is assumed that the general direction of the air flow through the slab is perpendicular to slab face and that air splits equally at each tube [1]. Thus a given tube is exposed to the air stream which consists of 50% of the air streams associated with the two closest neighbors in the proceeding row.

Air temperature and humidity at each tube past the first row is calculated by evaluating the mixing balances for air by standard psychometric equations. If the mixing process results in saturated air, the amount of condensed vapor is calculated and is assumed to collect on fins and drain. The mixing process is calculated by subroutine MIXAIR. The air distribution is assigned by subroutine DISTR2.

2.3 Refrigerant Distribution

Refrigerant distribution is governed in a coil by pressure drop. In this report, separate consideration has been given to single-slab and two-slab evaporators.

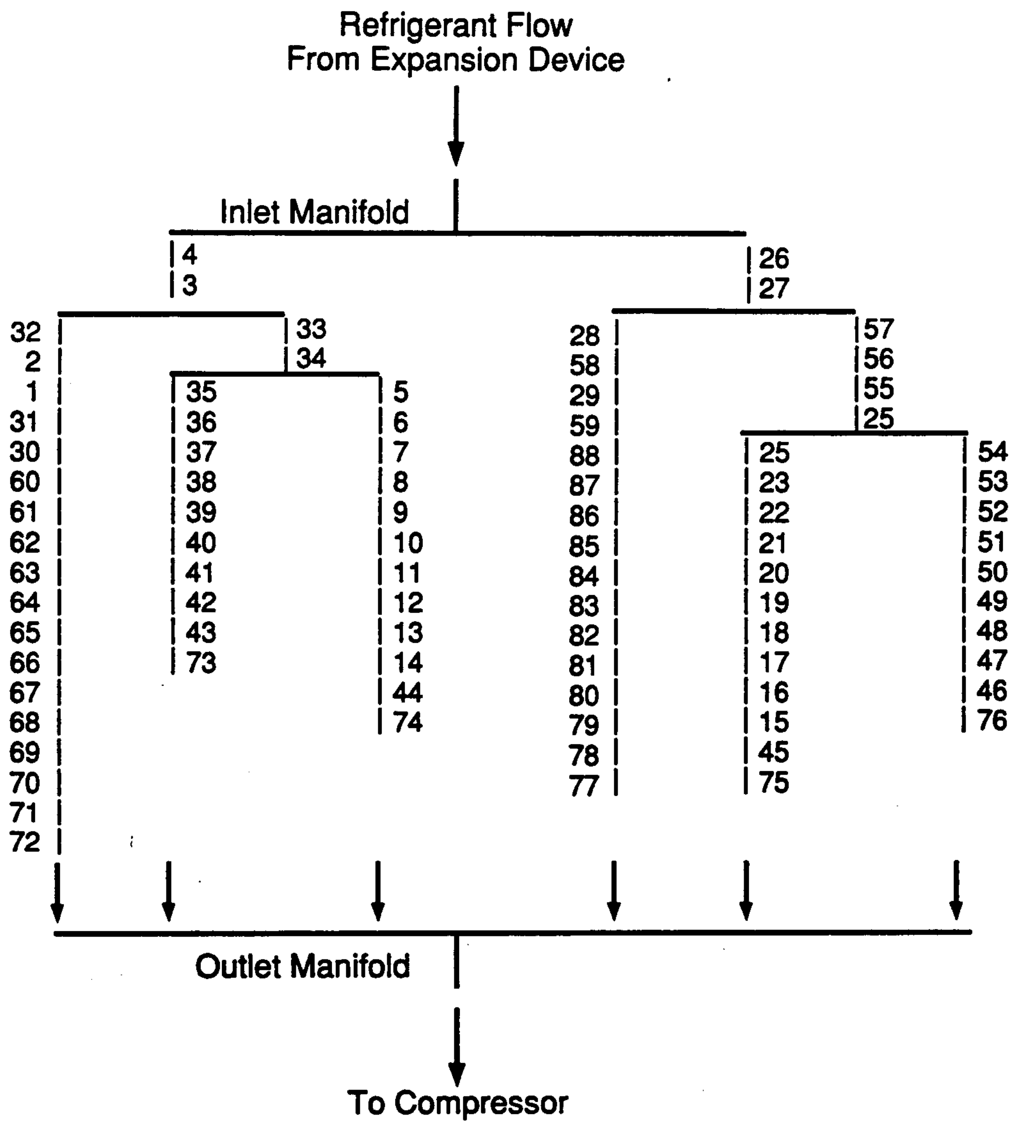
2.3.1 Single-Slab Evaporator

EVSIM can simulate one-slab evaporators employing one expansion device. In such a heat exchanger refrigerant pressure at each inlet tube is the same. Refrigerant mass flow rate through each circuit adjusts itself so pressure at each outlet tube is also the same.

At the outset of simulation the model estimates refrigerant distribution based on the circuitry layout. A uniform resistance to flow at each tube is assumed. This assumption is crude since pressure drop will depend on the refrigerant mass flux and heat transfer rate. In the course of simulation, once pressure drops are calculated for each tube, the refrigerant distribution is updated using the available pressure drop data.

Figure 2 presents an example of refrigerant circuitry on one plane. Each vertical line represents a tube. Numbers in Figure 2 denote location of each tube as explained in Appendix A.

The refrigerant distribution in a coil is determined by sequential analysis of each split point and associated branches. The algorithm for evaluation of refrigerant distribution was derived considering that pressure drop for the evaporative flow, as evaluated by the Pierre's correlation [6], can be



Each vertical line represents a tube. A number next to the line represents location of the tube, as explained in Appendix A.

Figure 2. An Example of Refrigerant Circuitry.

simplistically presented in the following form:

$$\Delta P \propto f \cdot G^2 \qquad (1)$$

where : ΔP = pressure drop
f = friction factor
G = mass flux

Considering that the friction factor in the Pierre's correlation is a function of the Reynolds number to the -0.25 power, we can rearrange equation (1) to obtain the following relations for each branch of circuitry:

$$R_i \cdot G_i^{1.75} = \Delta P_i \qquad (2)$$

where : R_i = resistance to flow offered by a given branch leaving the split point (it accounts for the effects of tube geometry, fluid density and viscosity)
G_i = refrigerant mass flux rate flowing through a given branch
ΔP_i = pressure drop in a given circuitry branch

Equation (2) allows calculation of the flow resistance for each branch using refrigerant pressure drop, ΔP_i, and mass flux, G_i, calculated in the previous iteration loop. Once flow resistances, R_i, are known, equation (2) allows estimation of the ratio of mass fluxes at any two branches, considering the fact that pressure drop through all branches associated with a given split point should be the same.

Since the sum of refrigerant mass flow rates through the branches equals the mass flow rate at the split point, the following equation holds:

$$F_1 + F_2 + \ldots . + F_n = \sum_{i=1}^{n} F_i = 1. \tag{3}$$

where: F_i = fraction of refrigerant mass flow rate at the split point flowing through a given branch
n = number of branches leaving the split point

Equation (3) and mass flux ratios generated by equation (2) constitute a set of n equations with n unknowns. This set can be solved to express the fraction of refrigerant mass flow rate at the split point flowing through any i^{th} of its n branches:

$$F_i = \frac{1.}{\sum_{j=1}^{n} (R_i/R_j)^{0.571}} \tag{4}$$

The above equation is used to update the refrigerant distribution. As result of distribution updates, the model iterates refrigerant pressure to within 0.05 psi at different circuitry outlets.

2.3.2 Two-Slab Evaporator

The model can simulate two types of two-slab evaporators:

1. those equipped with a single expansion device feeding both slabs,
2. those equipped with two expansion devices, one for each slab.

These two cases require somewhat different approaches to simulate refrigerant distribution. In the first case, as for a one-slab coil, refrigerant distribution is governed by the refrigerant pressure drop in the coil circuits. Refrigerant distribution in this case is determined by EVSIM as for a one-slab assembly.

If each coil of a two-slab assembly is fed independently by an individual expansion device, refrigerant distribution is determined in two stages. Firstly, the split of refrigerant between two slabs is determined solely by the relative restrictiveness of the expansion valves used. This is due to the fact that expansion devices used in air-conditioning applications usually choke and exhibit very weak dependence on the evaporator pressure. Secondly, refrigerant flow rate, already determined for each slab, adjusts itself at individual circuits based on their relative restrictiveness.
It should be noted that, unlike for a one expansion device assembly for which refrigerant pressures at all inlet and outlet tubes are respectively equal, a two expansion device assembly will have the same outlet pressure while the inlet pressure may be different between the two slabs involved (this will result in different refrigerant temperature profiles which affects the heat transfer rate). Since the model is set up to arrive at a specified outlet pressure with the inlet pressure unknown, an iterative procedure is employed for the two expansion device coil.

3. MODELING OF HEAT, MASS AND MOMENTUM TRANSFER PROCESSES

3.1 Heat Transfer for a Tube in a Cross-Flow Arrangement

If a single, separate tube of an air-to-refrigerant heat exchanger is considered, the heat transfer problem reduces to one of a pure cross-flow. For this type of heat transfer closed expressions can be derived based on cross-flow heat transfer theory. According to the general heat transfer equation:

$$Q = U \cdot A \cdot \Delta T_m \tag{5}$$

where A = heat transfer surface area
U = overall heat transfer coefficient
ΔT_m = mean temperature difference between the heat exchanging fluids
Q = heat transfer rate

In case of a pure cross-flow arrangements, one of two mean temperature differences applies for ΔT_m, [7]:

$$\Delta T_m = \frac{t_2 - t_1}{\ln \dfrac{T_2 - t_1}{T_1 - t_2}} \tag{6}$$

when temperature of one of the fluids does not change (temperature of that fluid is denoted by T)

$$\Delta T_m = \frac{t_2 - t_1}{\dfrac{\dfrac{T_1 - T_2}{t_2 - t_1}}{\ln \dfrac{\dfrac{T_1 - T_2}{t_2 - t_1}}{\dfrac{T_1 - T_2}{t_2 - t_1} + \ln \dfrac{T_2 - t_1}{T_1 - t_1}}}} \tag{7}$$

when temperature of both fluids change

where T = temperature of one fluid
t = temperature of another fluid
ΔT_m = mean temperature difference
subscripts 1 and 2 refer to tube inlet and outlet conditions

The enthalpy change of the fluid will be according to the equation:

$$Q = m \cdot (i_2 - i_1) \tag{8}$$

or

$$Q = m \cdot C_p \cdot (T_2 - T_1) \tag{9}$$

Equations (5) through (9) allow derivation of formulas for calculation of the heat transfer rate in a tube with two-phase refrigerant or superheated vapor flow. Tubes in which both flow patterns exist can be identified and the fractions of tube length associated with specific flow patterns can be calculated. For refrigerant pressure drop calculations, the model uses different pressure drop correlations for the two-phase portion and the superheated vapor portion of the flow. Heat transfer calculations employ different inside tube heat transfer correlations for annular flow (defined by the flow quality not greater than 0.85), dispersed flow (quality in the range of 0.85 - 1.00), and single-phase, superheated vapor flow (quality greater than 1.00). The derived correlations are presented below and are followed by explanation of the symbols.

<u>Annular Flow</u>

$$Q = m_a \cdot C_{p,a}(t_i - T_i)(1 - \exp(-\frac{U \cdot A_o}{m_a \cdot C_{p,a}})) \tag{10}$$

If the calculated heat transfer rate, Q, results in a refrigerant quality at the tube exit greater than 0.85, the heat transfer rate within the annular flow regime is equal to:

$$Q = m_r(i_{r,85} - i_i) \tag{11}$$

The fraction of the tube with flow quality up to 0.85, ANNUL, can be calculated by the equation:

$$\text{ANNUL} = \frac{m_r(i_{r,85} - i_{r,i})}{m_a \cdot C_{p,a}(t_i - T_i)(1 - \exp(-\frac{U \cdot A_o}{m_a \cdot C_{p,a}}))} \tag{12}$$

Dispersed Flow

$$Q = m_a \cdot C_{p,a}(1 - \text{ANNUL})(t_i - T_i)(1 - \exp(-\frac{U \cdot A_o}{m_a \cdot C_{p,a}})) \tag{13}$$

If the calculated heat transfer rate results in the refrigerant enthalpy exceeding that of the saturated vapor, $i_{r,v}$, the heat transfer rate within the dispersed flow is equal to:

$$Q = m_r \cdot (i_{r,v} - i_{r,i}) \tag{14}$$

The fraction of the tube with dispersed flow, XDRY, can be calculated by the equation:

$$\text{XDRY} = \frac{m_r \cdot (i_{r,v} - i_{r,i})}{m_a \cdot C_{p,a}(1 - \text{ANNUL})(t_i - T_i)(1 - \exp(-\frac{U \cdot A_o}{m_a \cdot C_{p,a}}))} \tag{15}$$

Single-phase Flow (superheated vapor)

$$Q = m_r \cdot C_{p,r}(t_i - T_i)(1-\exp(-\frac{(1-\text{ANNUL}-\text{XDRY})m_a \cdot C_{p,a}}{m_r \cdot C_{p,r}}(1-\exp(-\frac{U \cdot A_o}{m_a \cdot C_{p,a}})))) \tag{16}$$

In equations (10) through (16) the following nomenclature was used:

A_o = total exterior surface area associated with the tube wetted by air
$C_{p,a}$ = air specific heat at constant pressure
$C_{p,r}$ = refrigerant specific heat at constant pressure
$i_{r,i}$ = refrigerant enthalpy at the tube inlet
$i_{r,v}$ = enthalpy of refrigerant saturated vapor
$i_{r,85}$ = refrigerant enthalpy at flow quality equal 0.85
m_a = air mass flow rate associated with the tube
m_r = refrigerant mass flow rate in the tube
t_i = air temperature upstream of the tube
T_i = refrigerant temperature at tube inlet
ANNUL = length fraction of the tube with flow quality up to 0.85
XDRY = length fraction of the tube with flow quality within the range 0.85 - 1.00
U = overall tube heat transfer coefficient

3.2 Overall Heat Transfer Coefficient for a Finned Tube

3.2.1 Dry Tube

The overall heat transfer coefficient, U, for a dry finned tube can be derived by summing up the individual resistances between the refrigerant and the air, [7]:

$$U = \left[\frac{A_o}{A_{p,i}\ h_i} + \frac{A_o\ x_p}{A_{p,m}\ k_p} + \frac{A_o}{A_{p,o}\ h_{tf}} + \frac{1}{h_{c,o}[1 - (A_f/A_o)(1 - \phi)]} \right]^{-1} \tag{17}$$

where
A_f = fin surface area
A_o = total exterior surface area exposed to air
$A_{p,i}$ = pipe inside surface area
$A_{p,m}$ = pipe mean surface area
$A_{p,o}$ = pipe outside surface area
$h_{c,o}$ = convection heat transfer coefficient at the exterior surface
h_i = inside tube heat transfer coefficient
h_{tf} = thermal conductance of the pipe-to-fin contact
k_p = thermal conductivity of pipe material
x_p = thickness of pipe wall

$$\phi = \frac{T_{f,m} - T_a}{T_{f,b} - T_a}, \text{ fin efficiency} \tag{18}$$

T_a = air temperature
$T_{f,b}$ = fin base temperature
$T_{f,m}$ = mean fin temperature

The first term and the fourth term of equation (17) refer to the inside and outside convection resistances, respectively. The second term represents the conductive heat transfer resistance through the tube wall. The third term accounts for the contact resistance between the outside tube surface and the fin collar.

3.2.2 Wet Tube

Wet tube analysis is applicable to an evaporator when its surface temperature is below the dew point of the air. As a result, moisture is removed from the air stream and by condensation on the evaporator external surface. At evaporator temperature above the freezing point (simulation range of EVSIM) the condensate flows down the fins under the influence of gravity.

The heat transfer rate between the air stream and the water surface is described by the following equation:

$$dQ = [h_{c,o}(T_a - T_w) + h_{D,o}(w_a - w_w)i_{fg,w}]dA_o \tag{19}$$

The first term accounts for sensible heat transfer and the second term accounts for latent heat transfer. For air at atmospheric pressure the Lewis number,

$$Le = \frac{h_{c,o}}{h_{D,o} \cdot C_{p,a}}, \tag{20}$$

is close to 1 [7]. Assuming that the fin efficiency approximates the ratio of the moisture content differences,

$$\phi = \frac{w_a - w_{f,m}}{w_a - w_w} , \tag{21}$$

equation (19) assumes the following form for a tube with flat fins:

$$dQ = h_{c,o}(1 + \frac{i_{fg,w}(w_a - w_w)}{C_{p,a}(T_a - T_w)})(1 - \frac{A_f}{A_o}(1 - \phi))(T_a - T_w)dA_o \tag{22}$$

Symbols used in equations (19), (20), (21), and (22) denote:

A_f = fin surface area
A_o = total external area
$C_{p,a}$ = specific heat of air
$h_{c,o}$ = air-side forced convection heat transfer coefficient
$h_{D,o}$ = air-side mass transfer coefficient
$i_{fg,w}$ = latent heat of condensation for water
T_a = temperature of air
T_w = temperature of liquid water at the fin base
Q = heat transfer rate
w_a = humidity ratio of air
w_w = humidity ratio of saturated air at T_w temperature

The one-dimensional heat conduction across the condensate film can be expressed by the equation:

$$dQ = h_w \cdot \Delta T_w \cdot dA_o \tag{23}$$

where $h_w = \frac{k_w}{\delta}$, heat transfer coefficient for the condensate film
k_w = thermal conductivity of water
δ = thickness of condensate film
ΔT_w = temperature difference across the condensate film

Using equations (19) and (23) and referring to equation (17), the following relation for the overall heat transfer coefficient for a wet finned tube can be derived:

$$U = \left[\frac{A_o}{h_i A_{p,i}} + \frac{A_o x_p}{A_{p,m} k_p} + \frac{1}{h_L} + \frac{A_o}{A_{p,o} h_{tf}} + \frac{1}{h_{c,o}\left(1 + \frac{i_{fg,w}(w_a - w_w)}{C_{p,a}(T_a - T_w)}\right)\left(1 - \frac{A_f}{A_o}(1-\phi)\right)} \right]^{-1} \quad (24)$$

where symbols used are defined as in equations (17) and (21).

The presented approach accounts for the impact of moisture condensation on the heat transfer in a few aspects:

(1) the layer of condensate offers additional heat flow resistance (term 3 of equation (24)),

(2) the air-side heat transfer resistance decreases due to effect of condensation (term 4 of equation (24)),

(3) the air-side heat transfer coefficient, $h_{c,o}$, increases since it is sensitive to external surface geometry and the air flow Reynolds number (see equation (42)),

(4) fin efficiency decreases as h increases (see equation (36)).

Equation (24) requires evaluation of thickness of the condensate layer. The analysis presented below allows evaluation of a mean thickness of a condensate film on a flat, vertical fin. Obviously, in a real application, condensation of moisture is not uniform over the fin surface, and condensate thickness varies even for flat fins. Local variations in the condensate thickness are even more severe for wavy and stripped fins. At the lack of a better analytical approach, an average thickness of the water layer is evaluated and used in EVSIM in heat transfer calculations.

In order to evaluate water layer thickness, consider the mass transfer equation:

$$m_{a,d} \cdot dw_a = - h_{D,o}(w_a - w_w)dA_o \tag{25}$$

For the Lewis number equal to 1 equation (25) assumes the following form:

$$m_{a,d} \cdot dw_a = - \frac{h_{c,o}}{C_{p,a}} (w_a - w_w) \cdot dA_o \tag{26}$$

The change in the air humidity ratio can be calculated by integrating equation (26), which yields:

$$w_{a,e} = w_{a,i} - (w_{a,i} - w_w)(1 - \exp \frac{-h_{c,o} \cdot A_o}{C_{p,a} \cdot m_{a,d}}) \tag{27}$$

Condensation of moisture may sometimes occur only on a part of the outside surface associated with a tube. This may happen for example for a tube in which refrigerant is in a superheated vapor state having the inlet and outlet temperature below and above the air dew point. Another probable case is when the refrigerant temperature is slightly below the air dew point. In such a case condensation will occur on the tube surface and on that part of the finned area which is below the dew point. The fin surface, further from the tube, having a temperature above the dew point temperature will not produce condensation.

A rigorous modeling of condensation on a part of the outside surface requires identification of surface areas above and below the dew point. This task is extremely difficult. The effect of many variables affecting the temperature profiles of the surface is unknown. Notably, the effect of the tube staggering pattern and temperature of the neighboring tubes on the fin temperature profile is complex. For the above reasons, this part of simulation is performed using a few simplifying assumptions as it was felt that the phenomena are to complex to model rigorously in a general evaporator simulation program.

Moisture removal from the air stream is calculated separately for a tube and associated fins. Fins associated with a given tube are considered to be circular and of equal area, as shown in Figure 3. A linear temperature profile of a tube surface between inlet and outlet is assumed, and portions of a tube with and without condensation are determined accordingly. Also a linear temperature profile is assumed for a fin.

Mean temperature for a fin surface, $T_{f,m}$, can be expressed by the equation:

$$T_{f,m} = \frac{1}{A_f}\int T \cdot dA_f \tag{28}$$

Applying a linear temperature profile over the fin and integrating we obtain:

$$T_{f,m} = T_o + (T_t - T_o)\left(\frac{D_t^{\,3}}{3} - \frac{D_t^{\,2} \cdot D_o}{2} + \frac{D_o^{\,3}}{6}\right)/(4A_f(D_t - D_o)) \tag{29}$$

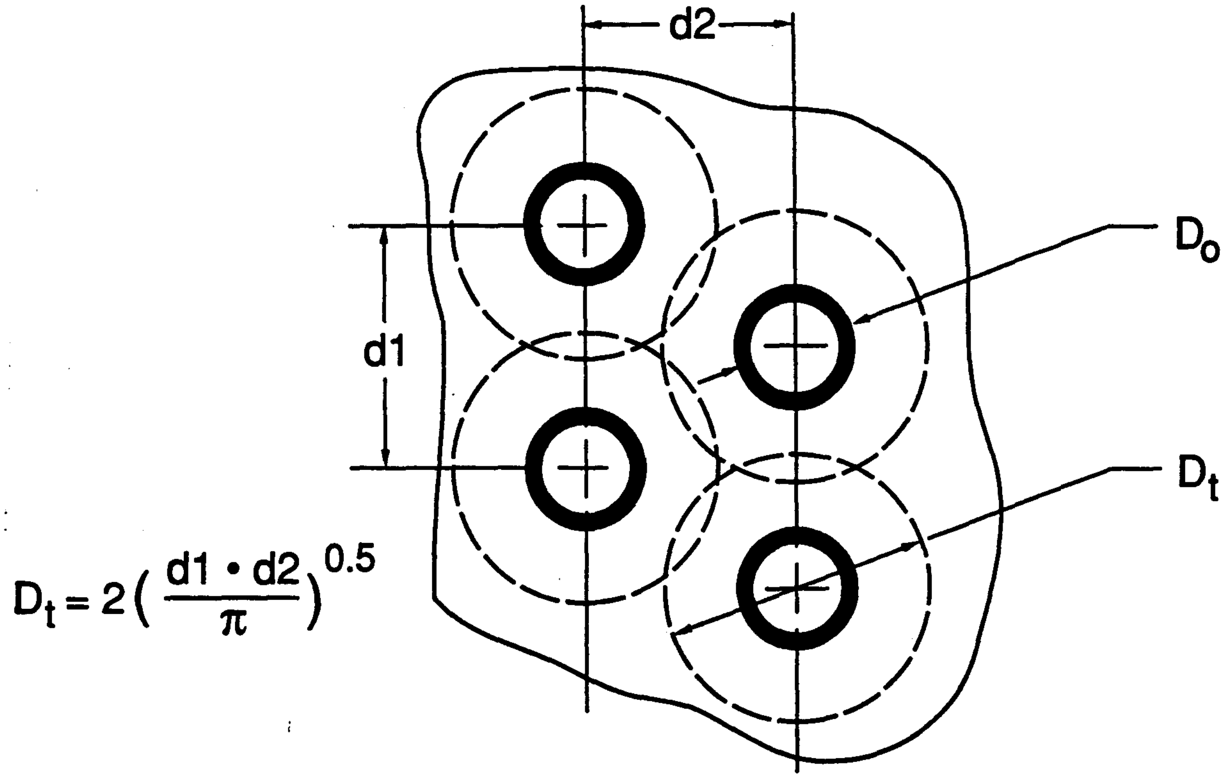


Figure 3. Approximation method for treating a rectangular-plate fin in terms of a circular-plate fin of equal area.

where D_o = tube outside diameter
D_t = fin tip diameter (see Figure 3)
T_o = tube temperature at D_o
T_t = fin temperature at D_t

Since the mean fin temperature can also be related to fin efficiency:

$$T_{f,m} = T_a - \phi(T_a - T_o) \tag{30}$$

the fin tip temperature, T_t, and the fin diameter within which condensation occurs, D_c, can be determined. Assuming that the humidity ratio of saturated air varies linearly with temperature, the humidity ratio of the saturated air corresponding to the mean temperature of the fin surface at temperature below the dew point can be calculated by the following equation:

$$w_w = w_o + (w_c - w_o)\left(\frac{D_c^{\,3}}{3} - \frac{D_c^{\,2} \cdot D_o}{2} + \frac{D_o^{\,3}}{6}\right)/(4A_{f,c}(D_c - D_o)) \tag{31}$$

It should be noted that EVSIM approximates rectangular-plate fins by circular plate fins just for moisture removal calculations only. The condensate flow on fins and the air-side heat transfer calculations recognize the plate finned-tube arrangement.

The rate of moisture removal per unit area, R, can now be calculated:

$$R = m_{a,d}(w_{a,i} - w_{a,e})/A_o \tag{32}$$

where $m_{a,d}$ = mass flow rate of dry air

$w_{a,e}$ = humidity ratio of air at tube row exit

$w_{a,i}$ = humidity ratio of air at tube row inlet

Assuming no air drag on the liquid layer, its local velocity is expressed by the closed solution of the Navier-Stokes equation for a viscous flow on a vertical wall:

$$V_z = \frac{\rho g}{\mu}\,[0.5 \cdot y^2 - y \cdot \delta] \tag{33}$$

where V_z = local liquid layer velocity
ρ = liquid density

g = gravitational acceleration
y = distance from the wall
δ = liquid layer thickness
μ = liquid absolute viscosity

Applying the continuity equation to the liquid film of a unit width:

$$m(z) = \rho \int_0^{\delta} V_z \, dy \tag{34}$$

and assuming uniform condensation rate on the fin (i.e., $m(z) = R' \cdot z/h$, where: $m(z)$ = mass flow rate of condensate at elevation z, R' = water condensation rate by a fin of height h and unit width, z= 0 at the top and z=h at the bottom of the slab), the average condensate layer thickness can be obtained by integrating a local layer thickness over the fin height and dividing the obtained expression by the height. The resulting expression is:

$$\delta_f = 1.082 \left[\frac{\mu_w \cdot R'}{g \cdot \rho_w^{2}}\right]^{1/3} \tag{35}$$

where g = gravitational acceleration
R' = condensation rate per unit width of a fin
μ_w = water dynamic viscosity
ρ_w = water density

3.2.3 Fin Efficiency

The addition of fins to the tubes greatly increases the outer heat transfer area but at the expense of decreasing the mean temperature difference between the surface and the air stream. The parameter called fin efficiency, ϕ, is used to rate the thermal effectiveness of a fin.

Fin efficiency for a circular flat fin on a singular tube was discussed by Gardner in his classic paper [8]. Gardner solved the differential equation for describing the temperature distribution and presented fin efficiency curves.

In the case of residential air-source evaporators, fins are not circular but continuous, rectangular plates serving all tubes in the slab. The shape of fin serving a particular tube depends on the tube pitch and transverse tube spacing in the assembly. The method proposed by Schmidt [9] and described in [10] is sensitive to the pattern of tube staggering and is employed here.

The fin efficiency, ϕ, is calculated in terms of the fin root radius, r_o, and two parameters, M and θ:

$$M = \left(\frac{2 \cdot h}{k_f \cdot t} \right)^{0.5} \tag{36}$$

$$\theta = (R/r_o - 1)(1 + 0.35 \cdot \ln(R/r_o)) \tag{37}$$

$$\phi = \frac{\tanh\ (M \cdot r_o \cdot \theta)}{M \cdot r_o \cdot \theta} \tag{38}$$

$$\text{where} \quad h = \begin{cases} h_{c,o}\left[1 + \dfrac{i_{fg,w}(w_a - w_w)}{C_{p,a}\ (T_a - T_w)}\right] & \text{if } w_a > w_w \\ h_{c,o} & \text{otherwise} \end{cases}$$

R = equivalent fin tip radius
r_o = outside radius of tube
t = fin thickness

The staggered tube configuration results in a hexangular fin as shown in Figure 4. The two dimensions shown in the figure can be defined interchangeably. They must be defined in such a way to have Y always greater than

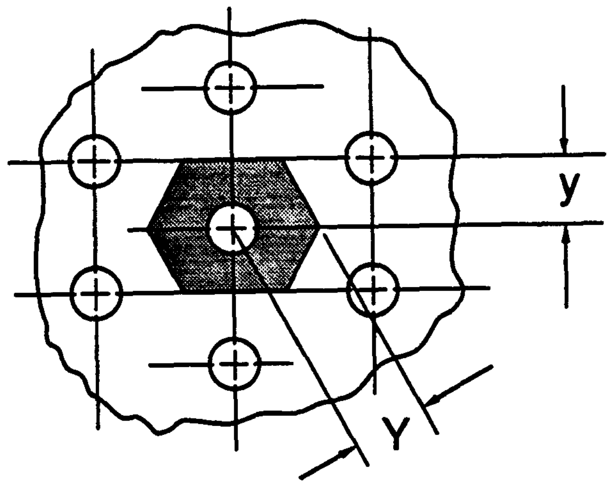


Figure 4. Definition of dimensions for calculating fin efficiency by Schmidt's method.

or equal to y. Following the notation shown in the figure, the ratio of the equivalent fin tip radius and the fin root radius is calculated by the following equations:

$$\psi = y/r_o \tag{39}$$

$$\beta = Y/y \tag{40}$$

$$R/r = 1.27 \cdot \psi(\beta - 0.3)^{0.5} \tag{41}$$

Numerous assumptions are associated with the calculation of the fin efficiency. The fin is assumed to be thin and the air-side heat transfer coefficient to be constant over the fin surface. Several practical aspects of the heat exchanger operation are not taken into account due to inability to model and/or perception that their impact on the fin efficiency value is insignificant. For example, heat transfer between neighboring fins is not taken into account. Similarly, the effect of the discontinuities in the fin material (lanced fins) on the fin efficiency is ignored in the model; it will vary from one lance design to another and is generally unknown. More study is needed to include the above effects.

3.3 Forced Convection Heat Transfer at the Air-Side of a Dry Plate Finned-Tube

EVSIM is concerned with modeling of evaporators having continuous fins on a staggered array of circular tubes. Three consecutive sections describe air-side heat transfer correlations applicable to three different fin designs. Calculation of the air-side heat transfer coefficient is handled by subroutine AIRHT3.

3.3.1 Flat Fins

The correlation of Gray and Webb [11] was selected to calculate the air-side heat transfer coefficient for flat fins. This correlation was developed using laboratory data on 16 heat exchangers applying a multiple regression technique. The rms error was 7.3%.

The correlation provides an average value for the j-factor for a heat exchanger with four or more tube depth rows (no change in the j-factor after 4-rows is assumed). It has the following form:

$$j_4 = 0.14 \cdot Re^{-0.32} \left(\frac{S_t}{S_l}\right)^{-0.502} \left(\frac{s}{D_o}\right)^{0.0312} \tag{42}$$

where $j_4 = \dfrac{h_{c,o} \cdot Pr^{2/3}}{G_c \cdot C_{p,a}}$, j-factor for four or greater number of depth rows (43)

$h_{c,o}$ = outside surface forced convection heat transfer coefficient
Pr = Prandtl number
G_c = air mass flux based in the minimum flow area
$C_{p,a}$ = specific heat of air at constant pressure
$Re = \dfrac{D_o \cdot G_c}{\mu}$, Reynolds number
D_o = outside diameter of tube
μ = dynamic viscosity
S_t = tube spacing normal to air flow
S_l = tube spacing in air flow direction (depth row pitch)
s = spacing between adjacent fins

To calculate an average value for the j-factor for heat exchangers with less than four depth rows, j_N (where N<4), Gray and Webb [11] provided the following equation:

$$j_N = j_4 \cdot 0.991[2.24 \cdot Re^{-0.092}(N/4)^{-0.031}]^{0.607(4-N)} \tag{44}$$

where j_4 = value obtained by equation (42)

Tube-by-tube simulation requires the availability of the air-side heat transfer coefficient for a tube in a given row. Assuming that each row weights equally on the average air-side heat transfer coefficient of the coil, the heat transfer coefficient value for the depth row N, $j_{N,R}$, can be approximated by the formula:

$$j_{N,R} = N \cdot j_N - (N-1) \cdot j_{N-1} \tag{45}$$

where j_N, j_{N-1} = average j-factors for heat exchangers with N and N-1 depth rows, respectively, obtained by equation (43) or (44)

3.3.2 Wavy Fins

Comprehensive data on performance of flat and wavy fins were published by Beecher and Fagan [12]. Webb and Trauger [13] used their results as a base for their correlation which are applied in EVSIM to provide the value of the enhancement of the air-side heat transfer coefficient for a wavy fin over a flat fin. This enhancement value is then used as a multiplier to the heat transfer coefficient calculated for a flat fin by equations presented in Section 3.3.1

Beecher and Fagan reported fins performance in terms of the Nusselt number based on the arithmetic mean temperature difference between the air and refrigerant, Nu_{ar}. Correlations provided by Webb and Trauger express this Nusselt number with the Graetz number, Gz, and non-dimensionalized geometric parameters. For <u>wavy fins</u> the correlations are:

for $Gz \leq 25$

$$Nu_{ar} = 0.5 \cdot Gz^{0.86} \left(\frac{S_t}{D_c}\right)^{0.11} \left(\frac{s}{D_c}\right)^{-0.09} \left(\frac{S_d}{S_1}\right)^{0.12} \left(\frac{2S_p}{S_1}\right)^{-0.34} \tag{46}$$

for $Gz > 25$

$$Nu_{ar} = 0.83 \cdot Gz^{0.76} \left(\frac{S_t}{D_c}\right)^{0.13} \left(\frac{s}{D_c}\right)^{-0.16} \left(\frac{S_d}{S_1}\right)^{0.25} \left(\frac{2S_p}{S_1}\right)^{-0.43} \tag{47}$$

These equations predict 88% of the base data within ±5%, and 99% of the data within 10%.

Correlations for flat fins have the following form:

for $Gz \leq 25$

$$Nu_{ar} = 0.4 \cdot Gz^{0.73} \left(\frac{s}{D_c}\right)^{-0.23} N^{0.23} \tag{48}$$

for $Gz > 25$

$$Nu_{ar} = 0.53 \cdot Gz^{0.62} \left(\frac{s}{D_c}\right)^{-0.23} N^{0.31} \tag{49}$$

Flat fin correlations predict 98% of the source data [12] within 5%.

To convert Nu_{ar} to the Nusselt number based on the logarithmic mean temperature difference, Nu, the following equation is used [12]:

$$Nu = 0.25 \cdot Gz \cdot \ln \frac{1 + 2 \cdot Nu_{ar}/Gz}{1 - 2 \cdot Nu_{ar}/Gz} \tag{50}$$

The symbols used in equations (46) through (50) are defined as follows:

$$Nu = \frac{h \cdot D_H}{k}, \quad \text{Nusselt number} \tag{51}$$

$$Gz = Re \cdot Pr \frac{D_H}{N \cdot S_1}, \quad \text{Graetz number} \tag{52}$$

$$D_H = \frac{2 \cdot s(1 - \beta)}{\gamma \cdot (1 - \beta) + 2 \cdot s \cdot \beta / D_c}, \quad \text{hydraulic diameter} \tag{53}$$

$$Re = \frac{\rho \cdot V_c \cdot D_H}{\mu}, \quad \text{Reynolds number} \tag{54}$$

$$\beta = \frac{\pi \cdot D_c}{4 \cdot S_t \cdot S_l} \tag{55}$$

$$\gamma = [1 + 4(S_d/2 \cdot S_p)^2]^{0.5} \tag{56}$$

$$V_c = V_{max}/(1 - \beta) \tag{57}$$

where
N = number of tube depth rows
V_{max} = velocity based on minimum flow area
S_d = fin pattern depth, tip to valley (see Figure 5)
S_p = fin pattern length, tip to tip
s = spacing between adjacent fins
ρ = density
k = thermal conductivity
μ = dynamic viscosity

The variables characterizing the wave pattern were set within subroutine AIRHT3 as follows: $S_p = 0.5 \cdot S_l$,

$S_d = s$

$D_c = D_o + 2 \cdot t$ where t = fin thickness.

3.3.3 Lanced Fins

Lanced fins are those enhanced fins which have arrays of small strips raised from the base plate. Nakayama and Xu [14] proposed a heat transfer correlation for such fins. Their formula is in the form of a heat transfer correlation for a flat fin and a multiplier which provides correction for heat transfer enhancement due to the raised strips.

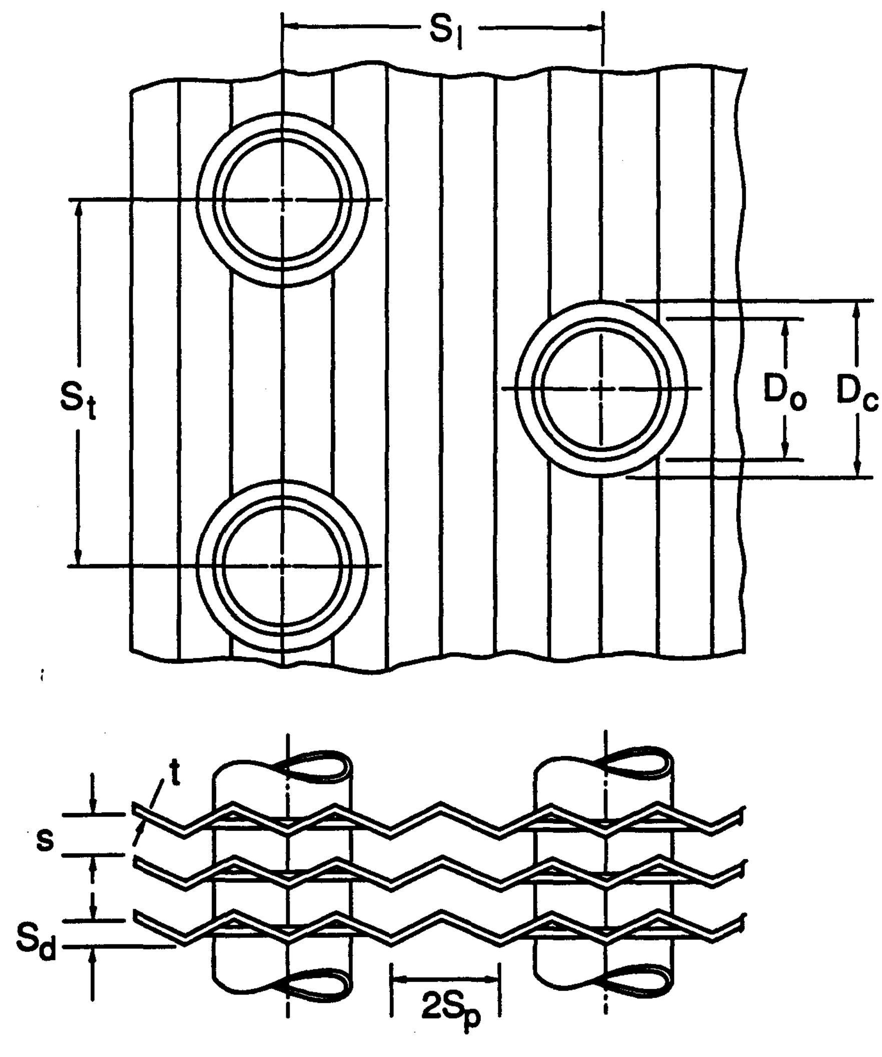


Figure 5. Geometry of a wavy fin.

To obtain a common simulation base for different heat transfer surfaces, EVSIM uses this multiplier in conjunction with the equations presented for a flat fin in Section 3.3.1.

The lanced fin enhancement multiplier is a function of the geometry parameters shown in Figure 6. The proposed correlation has the following form [14]:

$$F_j = 1 + 1093\left(\frac{t}{s}\right)^{1.24} \phi_s^{0.944} Re^{-0.58} + 1.097\left(\frac{t}{s}\right)^{2.09} \phi_s^{2.26} Re^{0.88} \quad (58)$$

where $$\phi_s = \frac{\text{strip area}}{\text{total fin area}} = \frac{(2n_s - 1) l_s \cdot S_s}{St \cdot S_l - 0.25 \cdot \pi \cdot D_o^2} \quad (59)$$

ϕ_s was set to 0.275 in subroutine AIRHT3. The applicable range of ϕ_s is 0.2 - 0.35.

$$Re = \frac{\rho \cdot V_{max} \cdot D_H}{\mu}, \quad \text{Reynolds number} \quad (60)$$

D_H = hydraulic diameter of the minimum free flow area
n_s = number of strips in the enhanced zone
l_s = width of a strip
S_s = length of a strip
t = fin thickness
s = spacing between adjacent fins
V_{max} = velocity in the minimum flow area

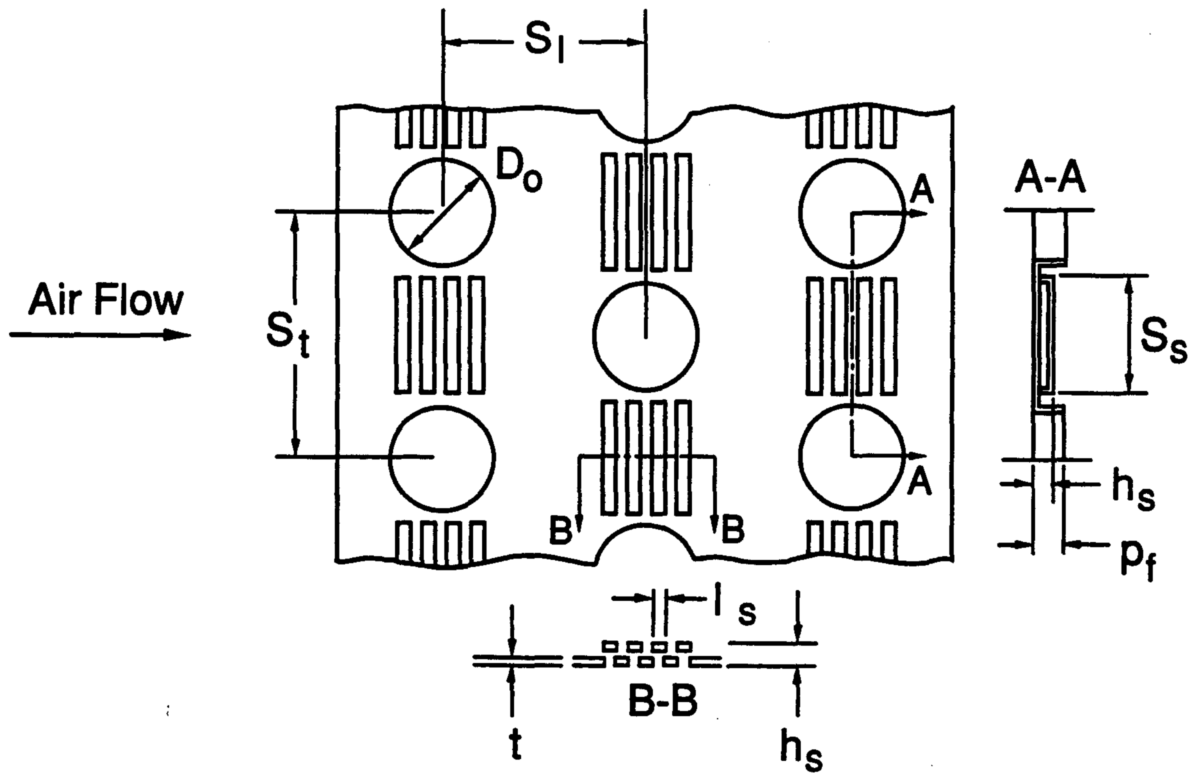


Figure 6. Geometry of a lanced fin.

3.4 Forced Convection Heat Transfer Inside an Externally Heated Tube

Tubes with smooth inner surface are predominantly used in residential heat exchangers, however, tubes with enhanced surfaces are also available. A significant amount of research has been performed over the years on heat transfer for smooth surfaces. Since a smooth surface is clearly defined, laboratory data of different researchers can be pooled together for the correlation development. As a result, a number of correlations for the heat transfer coefficient inside a smooth tube are available.

The situation is different for tubes with enhanced surfaces. Much less research have been done on enhanced tubes than on smooth tubes, and there is an infinite number of ways the inner surface may be enhanced. There is not adequate information in the open literature to allow calculation of the enhanced tube heat transfer coefficient in a generalized way. Consequently, while the internal tube heat transfer coefficient for smooth tubes is calculated in EVSIM by a well founded correlation, the heat transfer coefficient for enhanced tubes is evaluated by multiplying the heat transfer values for smooth tubes by the fixed correction factors. The following discussion on flow patterns refers to smooth tubes and may not necessarily be accurate in all aspects for tubes with enhanced surfaces.

Refrigerant enters the evaporator from an expansion device at a quality of about 20% and forms annular flow almost instantly. The quality increases as the flow proceeds and the liquid layer at the wall gets thinner. At higher qualities, reported by different researchers to be in the range from 65% to 95%, the refrigerant vapor has enough kinetic energy to gradually destroy the liquid layer. Consequently, the annular flow is followed by the mist flow and single-phase superheated vapor flow.

3.4.1 Single-Phase Flow

Smooth tubes. The single-phase forced convection heat transfer coefficient, h_{sp}, for a smooth, heated tube is calculated by a commonly used equation [15]:

$$h_{sp} = 0.023 \cdot Re^{0.8} \cdot Pr^{0.4} \cdot k/D_i \tag{60}$$

where Re = Reynolds number
Pr = Prandtl number
k = thermal conductivity of refrigerant vapor
D_i = tube inner diameter

Enhanced tubes. The heat transfer coefficient for enhanced inner surfaces is calculated within EVSIM by multiplying h_{st} obtained from equations (60) by a correction factor equal to 2.0. The value of the selected correction factor is an average of the heat transfer enhancement reported by Khanpara et al. [16] for micro-fin tubes with R-22.

3.4.2 Two-Phase Flow with Evaporation

Refrigerant flow with evaporation is subdivided in the model in two flow patterns; annular flow and mist flow. The quality value of 0.85 was selected in the model as the border point between these two flow patterns.

Smooth tubes, annular flow. A correlation developed by Gungor and Winterton [17] is used in EVSIM to calculate the evaporative heat transfer coefficient for the annular flow regime in smooth tubes. This correlation was developed with the aid of a large data bank which included 4300 points by 28 authors covering 7 fluids. The form of the correlation is consistent with Chen's [18] approach in that it recognizes two distinct mechanisms for the heat transfer; nucleate boiling and forced convection. The two-phase evaporation heat transfer coefficient, h_{an}, is expressed as a weighted average of the convective single-phase heat transfer coefficient, h_l, and the pool boiling heat transfer coefficient, h_{pool}, responsible in the correlation for the nucleate boiling contribution to the heat transfer:

$$h_{an} = E \cdot h_l + S \cdot h_{pool} \tag{61}$$

$$h_l = 0.023 \cdot Re_l^{0.8} \cdot Pr_l^{0.4} \cdot k_l / D_i \qquad (62)$$

$$h_{pool} = 55 \cdot P_r^{0.12} (-\log_{10} P_r)^{-0.55} \cdot M^{-0.5} \cdot q^{0.67} \quad (W/m^2 \cdot K) \qquad (63)$$

$$E = 1 + 24000 \cdot Bo^{1.16} + 1.37 \cdot X^{-0.86} \qquad (64)$$

$$S = (1 + 1.15 \cdot 10^{-6} \cdot E^2 \cdot Re^{1.17})^{-1} \qquad (65)$$

In the case of a horizontal tube and the Froude number, Fr, smaller than 0.05, E and S should be multiplied by E_2 and S_2, respectively:

$$E_2 = Fr^{(0.1-2 \cdot Fr)} \qquad (66)$$

$$S_2 = Fr^{0.5} \qquad (67)$$

The symbols used in equations (61) through (67) have the following meaning:

$$Re_l = \frac{G(1-x)D_i}{\mu_l}, \quad \text{liquid Reynolds number} \qquad (68)$$

$$Fr = \frac{G^2}{\rho_l^2 \cdot D_i \cdot g}, \quad \text{Froude number} \qquad (69)$$

D_i = tube inside diameter
G = refrigerant mass flux
g = acceleration of gravity
M = molecular weight
P_r = reduced pressure
Pr_l = liquid Prandtl number
q = heat flux (W/m^2)
x = flow quality
μ_l = liquid thermal conductivity
ρ_l = liquid density

$$Bo = \frac{Q}{G \cdot i_{fg}} \text{, boiling number}$$

Smooth tubes, mist flow. The heat transfer coefficient for the mist flow, h_m (flow quality range 0.85 - 1.00), is calculated in EVSIM by weighting by quality the heat transfer coefficient values obtained by equation (60) and equation (61):

$$h_m = (1.0-x)h_{an} + (x-0.85)h_{sp} \tag{70}$$

where x = average fractional flow quality for the mist flow in a tube
h_{an} = heat transfer coefficient obtained by equation (61) for flow quality 0.85
h_{sp} = heat transfer coefficient obtained by equation (60) for saturated vapor flow

Enhanced tubes. The heat transfer coefficient for an enhanced inner surface is calculated by applying a multiplier of 1.45 to the heat transfer coefficient calculated for a smooth tube within the two-phase region. The selected multiplier value is an average of the enhancement range (1.2 to 1.7) reported by Khanpara et al. [16] for a micro-fin tube with R-22.

3.5 Thermal Conductance of the Pipe-to-Fin Contact

The thermal conductance of the pipe-to-fin contact is calculated by the correlation provided by Sheffield et. al. [19].

$$h_{tf} = \exp\left\{6.902 + 2.889 \cdot \left[\frac{I \cdot FPI \cdot d}{D_o}\right]^{0.75}\left[t \cdot FPI\right]^{1.25}\right\} \tag{71}$$

where: h_{tf} = thermal conductance ($Btu/h \cdot ft^2 \cdot F$)
I = tube expansion interference (in)
FPI = number fins per inch (1/in)
d = indentation diameter of Vickers microhardness test, 25g load (μm) (d is set in EVSIM to 44.2 μm)
D_o = tube outside diameter (in)
t = fin thickness (in)

This correlation is restricted to mechanically expanded copper tubes with aluminum fins. Other applicability limits which has to be satisfied are:

$$D_o \le 0.625 \text{ (in)}$$

$$FPI \le 18$$

$$0.003 \le I \le 0.0079 \text{ (in)}$$

3.6 Refrigerant Pressure Drop in a Tube

The total pressure drop for flow in a tube consists of the pressure drops due to friction, momentum change and gravity. The gravitational pressure drop is neglected in EVSIM.

Comments made in the beginning of chapter 3.4 about availability of heat transfer performance data for enhanced surfaces are applicable also for pressure drop. Consequently, pressure drop for enhanced surfaces is calculated applying correlations for smooth tubes and predetermined correction factors.

3.6.1 Single-Phase Flow

Smooth tubes. Frictional pressure drop can be calculated by the Fanning equation with the Fanning friction factor as per equations (72) and (73):

$$\frac{dP}{dL} = \frac{2 \cdot f \cdot G2}{D_i \cdot \rho} \tag{72}$$

$$f = 0.046 \cdot Re^{-0.2} \tag{73}$$

Pressure drop due to momentum change can be calculated by the following equation:

$$\frac{dP}{dL} = - G^2 \frac{dv}{dL} \tag{74}$$

where
- P = pressure
- L = coordinate along the tube axis
- G = refrigerant mass flux
- D_i = tube inner diameter
- v = refrigerant specific volume

Enhanced tubes. Frictional pressure drop is calculated by equations (72) and (73) applying multiplier of 1.5 to the obtained result. This multiplier provides a half of the enhancement provided in EVSIM for the respective inside tube heat transfer coefficient.

3.6.2 Two-Phase Pressure Drop with Evaporation

Smooth tubes. The pressure drop for two-phase flow with evaporation can be calculated by the correlation proposed by Pierre [6], based on his experiments with refrigerants R-12 and R-22. Pierre's correlation combines the frictional and momentum change effects in one equation:

$$\Delta P = \left(f \frac{L}{D_i} + \frac{\Delta x}{x}\right) G^2 \cdot v_m \tag{75}$$

where f = friction factor calculated by equation (76)
x_m = mean quality
Δx = quality change
$v_m = v_L + x_m(v_V - v_L)$, mean specific volume

$$f = 0.0185 \left(\frac{K_f}{Re}\right)^{0.25} \tag{76}$$

where $$K_f = \frac{J \cdot i_{fg} \cdot \Delta x \cdot g_c}{L \cdot g} \tag{77}$$

$$Re = \frac{G \cdot D_i}{\mu_L} \tag{78}$$

J = 778.17 ($lb_f \cdot ft/Btu$), mechanical equivalent of heat

g = 32.2 (ft/s^2), gravitational acceleration

g_c = 32.2 ($ft \cdot lb/(lb_f \cdot s^2)$), dimensional constant

The formula for the friction factor contains the Reynolds number and the term, K_f, referred to by Pierre as a boiling number, making the friction factors sensitive to vapor generation rate at the liquid-vapor interface. Pierre's correlation was verified in [20] providing overall better agreement with experimental data for pressure drop of refrigerants R-12 and R-22 than the other most popular correlation, that of Martinelli and Nelson [21].

Enhanced tubes. Pressure drop for tubes with enhanced surfaced is calculated applying a multiplication factor 1.225 to the pressure drop value obtained from equation (75). The factor 1.225 represents a half of enhancement provided in EVSIM for the two-phase heat transfer coefficient for these surfaces.

4. MODEL VERIFICATION

The verification of the model consisted of comparing the coil capacity prediction with test data at different air distribution. Laboratory test data of an evaporator coil at four different air velocity profiles presented in [1] were used for model verification. The velocity profiles projected on the coil face were used as input to the model along with other operating conditions reported for these tests. Test conditions and laboratory and simulation results are shown in Table 1. The air velocity profiles are presented in Figure 7.

Figure 7 illustrates that the measured air velocity profiles differ from a parabolic profile associated with fully developed flow in a straight duct. The author believes that the odd profiles are due to the duct configuration, short entrance length and the presence of the coil in the duct. Various air velocity profiles were obtained by changing the coil angle with respect to the duct. It should be noted that the velocity profiles at different coil angles were the result not of the coil position alone but rather of the combined effects of change of the distance between coil surface and the duct entrance and the duct bend past the coil.

Total capacity results are also shown in Figure 8. The change of capacity is presented as a function of the coil angle. The figure displays the model sensitivity to the air distribution. Predictions of total capacities are within 8.2 percent of the test results. The model predicted degradation trend for 65° and 45° angles except at 25° where the prediction of capacity is within 1 percent of the test result. A number of reasons could cause the

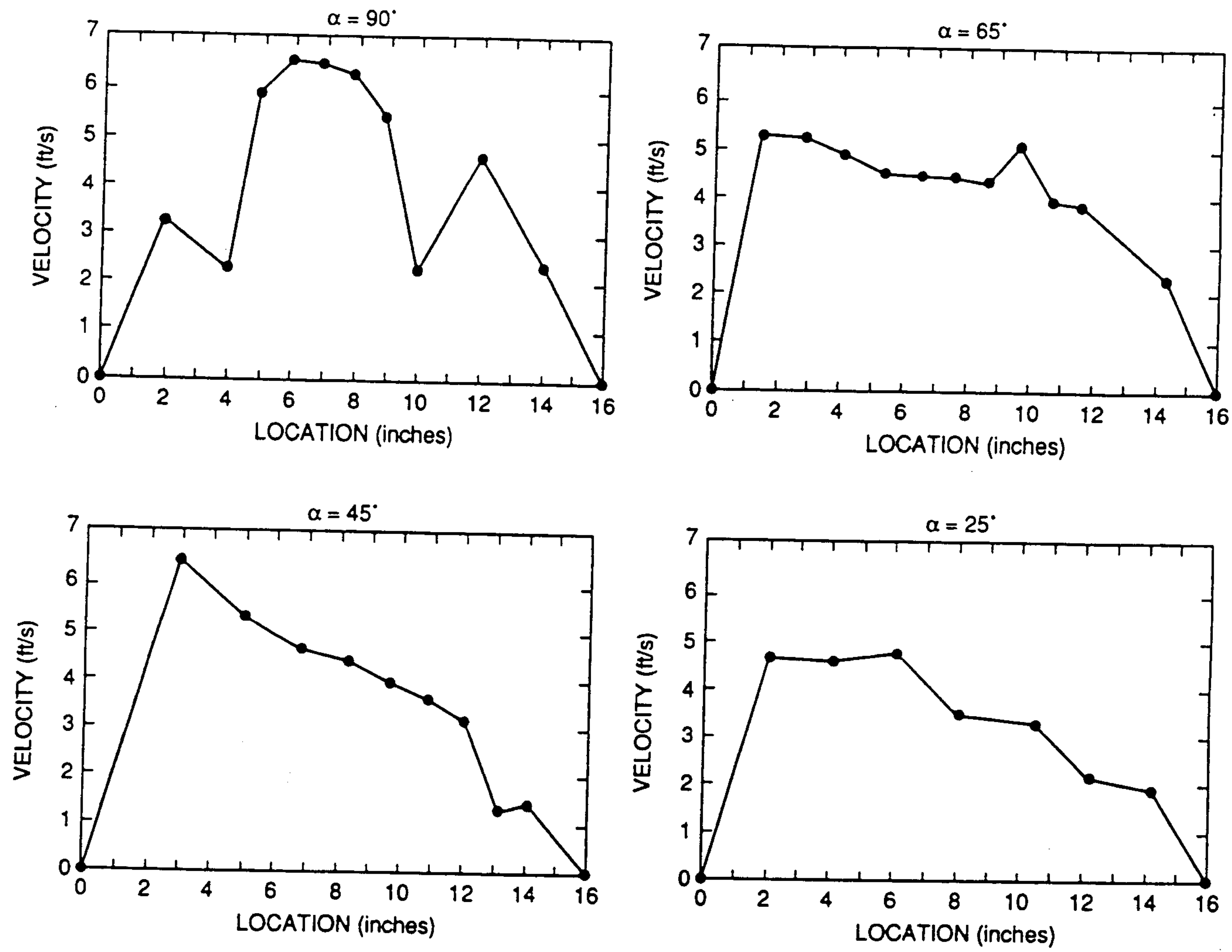
Table 1. Results of Laboratory Tests and Simulation Runs.

Angle	Conditions					Results					
	Air			Refrigerant		Test		Simulation		Discrepancy*	
(deg)	T_{DB} (°F)	T_{WB} (°F)	CFM	T_{SAT} (°F)	T_{SUP} (°F)	Q_T (Btu/h)	Q_L (Btu/h)	Q_T (Btu/h)	Q_L (Btu/h)	Q_T (%)	Q_L (%)
90	80.0	67.0	564	44.8	8.6	18480	5305	17373	5022	-6.0	- 5.3
65	79.8	66.7	566	44.9	7.7	17764	5093	16309	4678	-8.2	- 8.1
45	80.1	66.8	567	44.7	10.5	16580	4774	15417	5692	-7.0	19.2
25	79.3	66.5	559	44.9	5.4	14895	4138	14855	4654	-0.3	12.5

$$^{*}\frac{\text{simulation result - test result}}{\text{test result}} \cdot 100$$

T_{DB} = dry bulb temperature of incoming air
T_{WB} = wet bulb temperature of incoming air
T_{SAT} = refrigerant saturation temperautre at coil outlet
T_{SUP} = superheat of refrigerant at coil outlet

Q_T = total capacity
Q_L = latent capacity



α = coil angle as shown in Figure 8

location = distance from slab edge shown as X coordinate in Figure A3

Figure 7. Air velocity profiles.

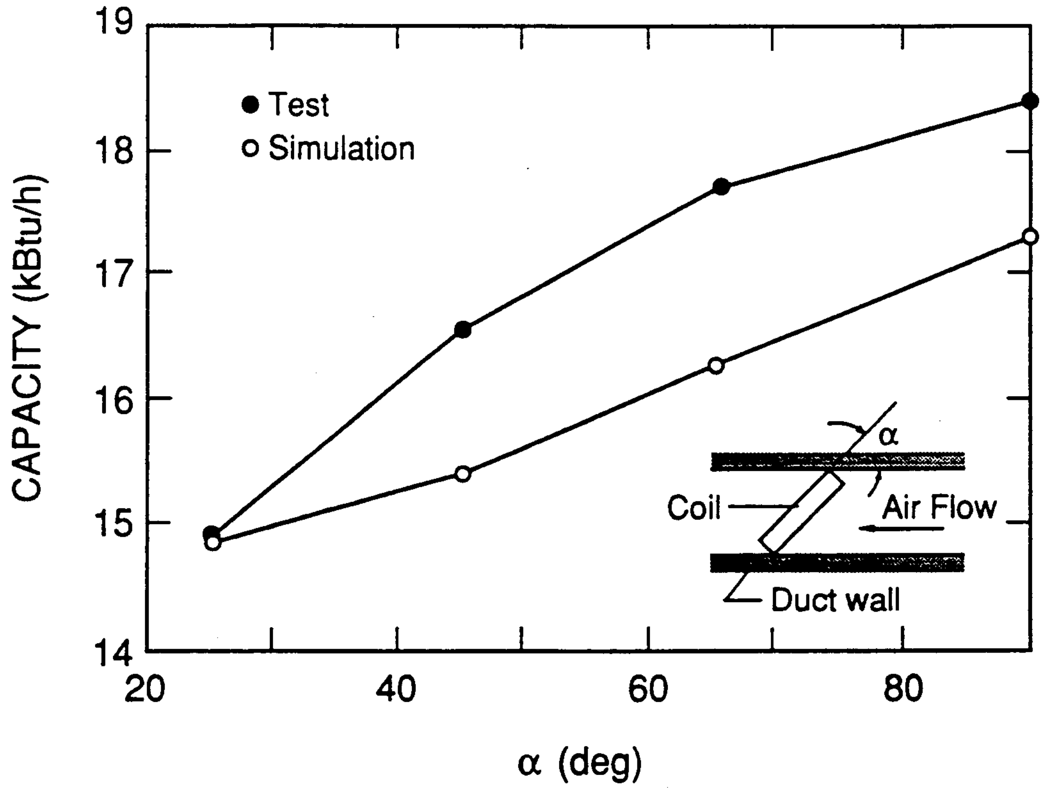


Figure 8. Simulation and test results.

break in the prediction trend between 25° and other angles. One possible reason may be the fact that, due to lack of space needed for the Pitot tube used for velocity measurement, the air velocity profiles for the first three angles were obtained from measurements past the coil while the 25° velocity profile was measured before the coil face. It is also possible that the velocity profiles derived from the air velocity measurements at the center plane represented at different degree the average air flow over the length of each tube at different coil angles.

Prediction of the latent capacity is not as good as of the total capacity. The highest discrepancy between test and simulation results is 19.2%. Lack of a closer agreement is not a complete surprise since the mass transfer and the condensate flow are very difficult to simulate and are represented simplistically in EVSIM. This simulation aspect requires further study to allow for simulation improvement.

5. CONCLUDING REMARKS

The evaporator model presented in this report, EVSIM, possesses two useful features:

- ability to account for air distribution
- ability to simulate refrigerant distribution in complicated circuits

These two capabilities are in a way compatible since refrigerant distribution in a coil depends not only on circuitry specification; change of air distribution affects heat transfer impacting refrigerant quality and pressure drop thus changing distribution of refrigerant. As a result of it's complexity, the model is more likely to provide satisfactory capacity predictions than would simpler models, particulary in cases where air distribution is not uniform over a coil face or refrigerant circuitry is complex. The model should be very helpful as a coil design tool, decreasing/eliminating the need for development tests, since detailed information on performance of each tube and refrigerant superheat at each circuit outlet can be obtained from the model.

Practical aspects of using the model include preparation of a coil data file and computer requirements. The detailed performance information which EVSIM can provide comes at the price of the need for a detailed input data file of the heat exchanger. In addition to general design input like tube diameter, fin spacing and thickness, tube staggering pattern, CFM, etc., specification of refrigerant circuitry and air distribution is needed.

The model is written in FORTRAN 77. Due to detailed algorithms and tube-by-tube performance evaluation scheme, mini and main frame computers are best suited for simulation studies using EVSIM; however the model has been installed on an IBM AT compatible computer converging within 2-6 minutes for a single slab coil, and up to 14 minutes for a two slab coil.

The model is capable of simulating the performance of evaporators with flat, wavy or lanced fins on the air side, and tubes having smooth or enhanced inner surfaces. Capacity predictions for flat fin and smooth tube coils are expected to be most reliable since these basic, plain surfaces are well defined and their respective heat transfer correlations are based on extensive data bases. Capacity prediction for evaporators with enhanced surfaces may not be as accurate since each enhanced surface is different with its own heat transfer characteristics; general correlations may not well represent their performance. This is particulary true for inner tube surfaces for which no general correlations are available and simple enhancement multipliers are used to account for performance change with respect to a smooth tube. Other area in which new algorithms could be applied, if available, to enhance the model are condensation of water vapor and prediction of the coil latent capacity.

6. REFERENCES

1. Chwalowski, M., Didion, D.A., and Domanski, P.A., Verification of Evaporator Computer Models and Analysis of Performance of an Evaporator Coil, ASHRAE Transactions, Vol. 95, Part 1, January, 1989.

2. McQuay, O.E.M. Division, O.E.M. Engineering Manual, Minneapolis, MN.

3. Bohn, Heat Transfer Division, O.E.M. Coil Manual, Danville, Il.

4. Fisher, S.K. and Rice, C.K., A Steady-State Computer Design Model for Air-to-Air Heat Pumps, Oak Ridge National Laboratory, ORNL/CON-8-, Oak Ridge, TN, December 1981.

5. Domanski, P. and Didion, D., Computer Modeling of the Vapor Compression Cycle With Constant Flow Area Expansion Device, NBS Building Science Series 155, Washington, DC, May 1983.

6. Pierre, B., Flow Resistance with Boiling Refrigerants, ASHRAE Journal, September 1964.

7. Threlkeld, J.L., Thermal Environmental Engineering, Prentice-Hall, Inc., New Jersey, 1970.

8. Gardner, K.A., Efficiency of Extended Surface, ASME Transactions, Vol. 67, p. 621, 1945.

9. Schmidt, T.E., La Production Calorifique des Surfaces Munies D'ailettes, Annexe Du Bulletin De L'Institut International Du Froid, Annexe G-5, 1945-46.

10. McQuiston, F.C. and Parker, J.D., Heating, Ventilating, and Air Conditioning, J. Wiley & Sons, 1982.

11. Gray, D.L. and Webb, R.L., Heat Transfer and Friction Correlations for Plate Finned-Tube Heat Exchangers Having Plain Fins, Proc. of Eighth Int. Heat Transfer Conference, San Francisco, 1986.

12. Beecher, D.T. and Fagan, T.J., Effects of Fin Pattern of the Air-Side Heat Transfer Coefficient in Plate Finned-Tube Heat Exchangers, ASHRAE Transactions, Vol. 93, Part II, p. 1961, 1987.

13. Trauger, P. and Webb, R.L., private communication on work to be published, August 1988.

14. Nakayama, W. and Xu, L.P., Enhanced Fins for Air-Cooled Heat Exchangers-Heat Transfer and Friction Factor Correlations, Proceedings of ASME-JSME Thermal Engineering Joint Conference, p. 495, March 1983, ASME, New York, 1983.

15. American Society of Heating, Refrigerating and Air Conditioning Engineers, Inc. ASHRAE Handbook, Fundamentals Volume, New York, 1985.

16. Khanpara, J.C., Pate, M.B. and Bergles, A.E., Local Evaporation Heat Transfer in a Smooth Tube and a Micro-fin Tube Using Refrigerants 22 and 113, presented at the Winter Annual Meeting of ASME, Boston, Ma., December 13-18, 1987.

17. Gungor, K.E. and Winterton, R.H.S., A general correlation for flow boiling in tubes and annuli, Int. Journal of Heat and Mass Transfer, Vol. 29, No. 3, pp. 351-358, 1986.

18. Chen, J.C., A Correlation for Boiling Heat Transfer to Saturated Fluids in Convective Flow, Ind. Eng. Chem. Proc. Des. Dev., 5, pp. 322-329, 1966.

19. Sheffield, J.W., Wood, R.A. and Sauer, Jr., H.J., Experimental Investigation of Thermal Conductance of Finned Tube Contacts, Experimental Thermal and Fluid Science, 0894-1777, 1989.

20. Anderson, S.W., Rich, D.G. and Geary, D.F., Evaporation of Refrigerant 22 in a Horizontal 3/4-In. OD Tube, ASHRAE Transactions, Vol. 72, p. 28, 1966.

21. Martinelli, R.C. and Nelson, D.B., Prediction of Pressure Drop During Forced Circulation Boiling of Water, ASME Transactions, Vol. 70, No. 6, 1948.

22. American Society of Heating , Refrigerating, and Air-Conditioning Engineers, ASHRAE Handbook, Equipment Volume, Atlanta, GA, 1988.

APPENDIX A. PROGRAM USER'S GUIDE

The simulation program of an air-source evaporator, EVSIM, predicts performance of a given evaporator at imposed air conditions and CFM, and refrigerant parameters at the heat exchanger inlet and outlet. The model is based on a tube-by-tube approach and account for non-uniform distributions of air and refrigerant.

The program is written in Fortran 77 and makes use of standard Fortran mathematical functions. It consists of a main module, evaporator simulation subroutine, EVPHX2, and 30 subprograms. Capability of the model and relations used for fluid mechanics, heat transfer and mass transfer calculations are described in the main text of this report. Equations used in moist air, water, and refrigerant property routines are described in [5].

A1. Input Data

Input data are read by the program from data files and a terminal (batch file) depending on data category.

A1.1 Data Read from Data Files.

To run the program, a user must establish two data files on the system, DATAREF and DTEV, for refrigerant constants and evaporator data.

Refrigerant property constants. Constants for evaluation of thermodynamic and transport properties for Refrigerant 12 and Refrigerant 22 are shown in Table A1 and Table A2, respectively. Constants of the selected refrigerant have to be in a disk file named DATAREF from which they are read by subroutine

DATAIN. Comments statements inserted into subroutine DATAIN help to identify specific constants.

Evaporator Data. Evaporator data are read by subroutine RDATA3 from a file named DTEV. Coding of evaporator data in DTEV is described in Table A3 which includes Fortran symbols and their short explanation. The convention by which DTEV has to be prepared and some of the variable names are explained in detail below.

EVSIM can simulate one-slab and two-slab evaporator assemblies (coils). The input format shown in Table A3 is complete for a one-slab coil; a data file for a two-slab assembly will have lines 3 through 25 repeated as lines 26 through 48 with input data for slab # 2. An example of a data file for the two-slab assembly of Figure A1 is shown in Table A4.

For two-slab coils, the model can handle coils in which each slab is fed by an individual expansion device, or both slabs are fed by a single, common expansion device. These arrangements are coded in the data file in line 2. In the present version, EVSIM is unable to handle a slab fed by more than one expansion device.

The last input datum at line 7, SFLOW, is the expansion device scaling factor. For capillary tubes, it may be determined with the aid of Fig. 39 of [22]. This scaling factor was included into a data file to allow EVSIM to calculate the refrigerant split between two slabs in two-slab assemblies if separate expansion devices are used for each slab. For evaporator assemblies employing just one expansion device (both single and two-slab coils) the value of SFLOW has no bearing on calculations; any real number may be inputed to satisfy the format of the READ statement. For evaporators with two expansion devices, the ratio of SFLOW(1) and SFLOW(2) has to be equal to the ratio of refrigerant mass flow rates through expansion devices associated with respective slabs. The ratio of flow factors, as presented in Fig. 39 of [22], satisfies this condition; hence flow factors are prescribed as input to the evaporator data file. In the most probable case, the two expansion devices are identical; in such situation two identical real numbers, say 1., satisfy the format of the READ statement and modeling requirements of EVSIM.

Line 3 contains general dimensions of a slab. The symbols used are explained graphically in Figure A2. Note that BSPACE denotes the distance between the edge of the open channel for air flow to the center of tube #1. Numbering of tubes should be done as shown in Figure A1 and Figure A2 for two-slab and one-slab coils, respectively. Numbering should start at the leftmost tube in the first tube row (facing the incoming air). Once the last tube of a given row is given a number, a consecutive number is given to the leftmost tube in the next depth row. It is not important at which coil side tube numbers are assigned so long as it is consistent with the later assignment of the air velocity profile for the coil.

Line 5 contains tube information. ISUR is an identifier for smooth or enhanced inner surface. Only for smooth surface are the heat transfer and pressure drop calculations performed using well establish correlations.

Because of the variety of enhanced surface and lack of confirmed correlations, the heat transfer and pressure drop for enhanced surfaces are evaluated by applying correction factors to smooth tube calculations (refer to section 3.4 and 3.6).

Line 6 groups air side fin data. Heat transfer for plain, wavy and lanced fins is calculated by separate, dedicated correlations. Wave pattern for wavy fins and strip set for lanced fins are assigned within subroutine AIRHT3 (see section 3.3).

Lines 8 through 20 describe refrigerant circuitry. Description of refrigerant circuitry depends on specification for every tube the tube which supplies it with refrigerant. It is done in the numerical order starting from the tube numbered as 1. in line 8, with ten tubes per line. Taking as an example the circuitry shown in Figure A1, since the first field of line 8 is designated for tube #1, tube # 2 shall be placed in this field since tube #2 feeds tube #1. If a given tube receives refrigerant from an inlet manifold, the input shall be zero. EVSIM can handle coils with up to 130 tubes per slab. Enter 999 in the data field if the tube does not exist.

Line 22 through 25 are for air velocity measurement data input. The data consist of location of the measurement and the air velocity at this location. The measurements should be taken in the central plane, perpendicular to coil tubes. Based on these measurements, EVSIM develops the air velocity profile in the central plane and uses it to evaluate the air mass flow rate for individual tubes in the assembly. Figure A3 shows how EVSIM utilizes the air velocity data. A velocity profile is created by straight line interpolation between the discrete data points. Note that the air velocity at the sides of the slab is assigned equal to velocities of respective leftmost and rightmost data points (VX(2,1) and VX(2,6) in Figure A3). The maximum data input is 16 points. The minimum data input is one point. The latter case results in prescription of a uniform velocity profile over the slab face.

It should be noted that air velocity data serve to establish the velocity profile only and are not used as input of air volumetric flow through the coil. The velocity profile is integrated to obtain a CFM value which is then compared by the program with CFMTOT to prorate local velocities so CFM and CFMTOT are equal.

A1.2 Data Read from a Terminal

Data read from a terminal (batch file) include:

- air parameters: dry bulb temperature
  relative humidity
- refrigerant parameters: inlet quality
  outlet saturation temperature
  outlet superheat
- program output controlling parameters.

This input is solicited by the program during program execution in the following format:

1. Request: DATE:
   Response: alphanumeric response up to 16 characters

2. Request:
```
IPR  = 0  FOR MAIN RESULTS OUTPUT ON THE DEFAULT DEVICE
IPR  = 1  FOR MAIN RESULTS OUTPUT TO FILE "RESULT"
IDIA = 0  FOR NO ADDITIONAL, DETAILED RESULTS OUTPUT
IDIA = 1  FOR ADDITIONAL, DETAILED RESULTS OUTPUT
          TOGETHER WITH MAIN RESULTS
IDIA = 2  FOR ADDITIONAL, DETAILED RESULTS OUTPUT TO FILES
          "DIAG" AND "DIAGO"
```

```
               IPR, IDIA =
    Response:  two integer numbers separated by a comma

 3. Request:   TAIR = AIR DRY BULB TEMPERATURE, (F)
               RH   = AIR RELATIVE HUMIDITY, (DECIMAL FRACTION)
               TAIR, RH =
    Response:  two real numbers separated by a comma

 4. Request:   X1    = REFRIGERANT INLET QUALITY, (DECIMAL FRACTION)
               TSAT2= REFRIGERANT SAT. TEMP. AT COIL OUTLET, (F)
               TSUP2= REFRIGERANT SUPERHEAT AT THE COIL OUTLET, (F)
               X1, TSAT2, TSUP2 =
    Response:  three real numbers separated by commas
```

A2. Output Data

EVSIM concludes a run when it converges on the imposed refrigerant saturation temperature and superheat at the evaporator outlet with the following convergence parameters:

saturation temperature: ± 0.05 °F

superheat: ± 2.0 °F.

Once the model converges within 2.0 °F of superheat, the intermediate results from the last two iteration loops are used to interpolate the performance results to the superheat value specified in the input data. If the model is unable to converge, it still performs interpolation and provides a warning message.

A short results output and a detailed results output are available from EVSIM, as indicated in the previous section. The short results version contains a short summary of input data and coil performance information from the last iteration loop which include state information on refrigerant leaving the individual outlet tubes. The last part of the output is the coil performance at the requested superheat at the coil exit. An example of the short results output is given in Appendix E.

A more detailed results output (IDIA=2) contains the intermediate results for individual iteration loops (file DIAG) and counters indicating the number of iteration loops required to converge in a given run (IDIA0). The option of a detailed output may be used if more specific information about the coil performance is sought; this output may be redesigned to suit an individual need.

Table A1. Property Constants for Refrigerant 12 in the Input Format to Program EVSIM

```
REFRIGERANT 12
6.9330000E+02,5.9690000E+02,2.8700000E-02
4.5967000E+02,1.8505300E-01,2.7182818E+00
9.1835883E+01,-7.9131381E+03,-1.2471522E+01
1.0892245E-02,0.,0.
120.,312.
3.4840000E+01,1.8691368E+01,2.1983963E+01
5.3341175E+01,-3.1509939E+00,1.
5.0000000E-01,3.3333333E-01,2.
8.8734000E-02,6.5093890E-03,0.
-3.4097270E+00,1.5943480E-03,-5.6762767E+01
6.0239450E-02,-1.8796180E-05,1.3113990E+00
-5.4873700E-04,0.,0.
0.,3.4688340E-09,-2.5439070E-05
0.,0.,0.
0.,5.4750000E+00
8.0945000E-03,3.3266200E-04,-2.4138960E-07
6.7236300E-11,0.,0.
3.9556551E+01,-1.6537940E-02
0.75800014,-0.44230204E-02,0.22659166E-04
-0.80936502E-07,0.48639626E-09,-0.38992915E-11
0.16672029E+01,-0.37690766E-01,0.48997637E-03
-0.32598077E-05,0.10701433E-07,-0.14068541E-10
0.0262,5.8E-05,0.
0.,0.,0.
-0.10065153E01,0.37970707E-01,-0.53504633E-03
0.36446277E-05,-0.12031578E-07,0.15478212E-10
0.49000002E-01,-0.11950213E-03,0.36320252E-07
0.52080296E-09,-0.31689972E-11,-0.31688932E-13
0.33482605E00,-0.10634881E-01,0.14902322E-03
-0.10213785E-05,0.34027736E-08,-0.44299642E-11
0.0043,1.7E-05,0.
0.,0.,0.
-0.12814194E00,0.48691398E-02,-0.68452384E-04
0.46711199E-06,-0.15483721E-08,0.20037774E-11
0.21699998E00,0.14159610E-03,0.64704511E-06
0.55390004E-08,-0.13778530E-10,-0.17912061E-12
-0.15409625E01,0.64801723E-01,-0.91255037E-03
0.62140834E-05,-0.20422284E-07,0.26255064E-10
```

Table A2. Property Constants For Refrigerant 22 in the Input Format to Program EVSIM

```
REFRIGERANT 22
6.6450000E+02,7.2191000E+02,3.0525000E-02
4.5967000E+02,1.8505300E-01,2.7182818E+00
6.7598246E+01,-8.8538843E+03,-7.8610310E+00
5.0448235E-03,4.4574700E-01,6.8610000E+02
140.,240.
3.2760000E+01,5.4634409E+01,3.6748920E+01
-2.2292560E+01,2.0473289E+01,3.3333333E-01
6.6666667E-01,1.,1.3333333E+00
1.2409800E-01,2.0000000E-03,0.
-4.3535470E+00,2.4072520E-03,-4.4066868E+01
-1.7464000E-02,7.6278900E-05,1.4837630E+00
2.3101420E-03,-3.6057230E-06,0.
-3.7240440E-05,5.3554650E-08,-1.8450510E-04
1.3633870E+08,-1.6726120E+05,0.
5.4820000E+02,4.2000000E+00
2.8128360E-02,2.2554080E-04,-6.5096070E-08
0.,0.,2.5734100E+02
6.2400900E+01,-4.5333500E-02
0.64600E+00,-0.29194E-02,0.12164E-04
-0.74985E-07,0.83951E-09,-0.37512E-11
0.69684E01,-0.24319E00,0.35924E-02
-0.26187E-04,0.93884E-07,-0.13301E-09
0.26600E-01,0.63804E-04,0.10761E-06
-0.32061E-08,0.43463E-10,-0.13175E-12
-0.66330E00,0.25757E-01,-0.37913E-03
0.27734E-05,-0.10053E-07,0.14474E-10
0.63000E-01,-0.15820E-03,-0.12289E-06
0.17453E-08,-0.52685E-11,-0.10857E-13
-0.46705E00,0.19421E-01,-0.28507E-03
0.20429E-05,-0.71963E-08,0.99460E-11
0.48000E-02,0.19881E-04,0.24815E-08
0.28518E-09,-0.62001E-11,0.31001E-13
0.51539E00,-0.18601E-01,0.26762E-03
-0.18936E-05,0.65891E-08,-0.90041E-11
0.27100E+00,0.24054E-03,0.38936E-07
0.23481E-07,-0.97345E-10,0.44953E-12
0.49002E00,-0.83123E-02,0.13105E-03
-0.96884E-06,0.36462E-08,-0.52089E-11
```

Table A3. Input Data Code for an Evaporator Assembly

The input data described below constitute a complete data set for a single slab evaporator. All input data are in Fortran free field input format with data values on the same line separated by commas.

Line 1: COILID
COILID = alphanumeric coil information, maximum 70 characters

Line 2: NSLABS, IEXP, CFMTOT
NSLABS = number of heat exchanger slabs in the coil assembly, possible values: 1 or 2
IEXP = number of expansion devices in the assembly, possible values: 1 if NSLABS = 1
1 or 2 if NSLABS = 2
CFMTOT = total volumetric flow of air through the assembly, (ft^3/min)

Line 3: SLABID
SLABID = alphanumeric slab information, maximum 70 characters

Line 4: BSIDE(1), BSPACE(1), WIDTH(1) (see Figure A2)
BSIDE(1) = height of the coil, (in)
BSPACE(1) = distance between the edge of the coil and location of tube # 1, (in)
WIDTH(1) = width of a coil, equal to the length of tubes exposed to the duct air, (in)

Line 5: TPCH(1), DPCH(1), DI(1), DO(1), TMK(1), ISUR(1) (see Figure A2)
TPCH(1) = tube pitch in each depth row, (in)
DPCH(1) = distance between neighboring tube depth rows, see Fig. A1, (in)
DI(1) = tube inside diameter, for grooved tubes use the minimum diameter, (in)
DO(1) = tube outside diameter, (in)
TKM(1) = thermal conductivity of a tube material, (Btu/(ft•h•F))
ISUR(1) = 1 for a smooth inner surface
= 2 for an enhanced inner surface

Line 6: FPCH(1), FTK(1), FMK(1), IFIN(1)
FPCH(1) = center to center distance between fins, (in)
FTK(1) = fin thickness, (in)
FMK(1) = fin material thermal conductivity, ((Btu/ft•h•F))
IFIN(1) = 1 for flat fins
= 2 for wavy fins
= 3 for lanced fins

```
Line  7: NTUB(1,1), NTUB(1,2), NTUB(1,3), NTUB(1,4), NTUB(1,5), SFLOW(1)
         NTUB(1,1)  = number of tubes in the first depth row (facing the
                      incoming air)
         NTUB(1,2)  = number of tubes in the second depth row
         NTUB(1,3)  = number of tubes in the third depth row
         NTUB(1,4)  = number of tubes in the forth depth row
         NTUB(1,5)  = number of tubes in the fifth depth row
         SFLOW(1)   = expansion device flow factor

Line  8: IFROM(1,I), I = 1,10
         IFROM(1,1) = number of the tube from which tube 1 receives refrig.
         IFROM(1,2) = number of the tube from which tube 2 receives refrig.
         IFROM(1,3)
         .
         .
         IFROM(1,9)
         IFROM(1,10)= number of the tube from which tube 10 receives refrig.

Line  9: IFROM(1,I), I = 11,20
         IFROM(1,11)= number of the tube from which tube 11 receives refrig.
         IFROM(1,12)= number of the tube from which tube 12 receives refrig.
         IFROM(1,13)
         .
         .
         IFROM(1,19)
         IFROM(1,20)= number of the tube from which tube 20 receives refrig.

Line 10: IFROM(1,I), I = 21,30

Line 11: IFROM(1,I), I = 31,40

Line 12: IFROM(1,I), I = 41,50

Line 13: IFROM(1,I), I = 51,60

Line 14: IFROM(1,I), I = 61,70

Line 15: IFROM(1,I), I = 71,80

Line 16: IFROM(1,I), I = 81,90

Line 17: IFROM(1,I), I = 91,100

Line 18: IFROM(1,I), I = 101,110

Line 19: IFROM(1,I), I = 111,120

Line 20: IFROM(1,I), I = 121,130

Line 21: NTEST(1)
         NTEST(1)   = number of air velocity measurement points,
                      possible values: minimum 2, maximum 16
```

Line 22: X(1,N), N=1,8

X(1,1)	=	location of the first air velocity measurement point (distance between the edge of the slab closest to tube #1 and the velocity measuring probe, see Figure A3), (in)
X(1,2)	=	location of the second air velocity measurement point
X(1,3)	=	location of the third air velocity measurement point, if non-existent, input 0.0, (in)
.		
.		
X(1,8)	=	location of the eight air velocity measurement point, if non-existent, input 0.0, (in)

Line 22: X(1,N), N=9,16

X(1,9)	=	location of the ninth air velocity measurement point, if non-existent, input 0.0, (in)
X(1,10)		
.		
.		
X(1,16)	=	location of the sixteenth air velocity measurement point, if non-existent, input 0.0, (in)

Line 23: VX(1,N), N=1,8

VX(1,1)	=	air velocity at the first measurement point, (ft/s)
VX(1,2)	=	air velocity at the second measurement point, (ft/s)
VX(1,3)	=	air velocity at the third measurement point, if non-existent, input 0.0, (ft/s)
.		
.		
VX(1,8)	=	air velocity at the eight measurement point, if non-existent, input 0.0, (ft/s)

Line 25: VX(N), N=9,16

VX(1,9)	=	air velocity at the ninth measurement point, (ft/s)
VX(1,10)		
.		
.		
VX(1,16)	=	air velocity at the eight measurement point, (ft/s) if non-existent, input 0.0, (ft/s)

Line 25 completes data file for a one slab evaporator (slant coil). If two slab evaporator is considered (V-shape, A-shape coil), the data file has to contain additional 23 lines of data in which the second slab is described. In this case, lines 3 through 48 are dedicated to slab # 1, and identical lines 26 through 48 describe slab # 2. Subroutine RDATA3, which reads the evaporator data file, assigns index '2' instead of '1' when reading lines 26 through 48 (e.g., reads VX(2,16) instead VX(1,16)). Table A4 provides an example of a data file for an A-shape coil.

Table A4. Example of a Data File for an A-shape Coil

Below is the coding of the coil shown in Figure A1.

```
***DTEV***    A-SHAPE COIL, 3 DEPTH ROWS, 16 TUBES PER ROW.
2,2,1120.0
***DATA FOR SLAB # 1***
16.0625,0.8125,17.875
1.00,0.875,0.363,0.394,223.,1
0.0789,0.008,128.,2
16,16,16,0,0,1.
2,3,19,5,6,22,23,7,8,25
10,27,12,30,14,15,33,17,18,4
37,21,39,0,9,11,28,45,13,31
32,48,34,35,36,20,38,41,40,24
42,25,26,43,44,29,46,47,999,999
999,999,999,999,999,999,999,999,999,999
999,999,999,999,999,999,999,999,999,999
999,999,999,999,999,999,999,999,999,999
999,999,999,999,999,999,999,999,999,999
999,999,999,999,999,999,999,999,999,999
999,999,999,999,999,999,999,999,999,999
999,999,999,999,999,999,999,999,999,999
999,999,999,999,999,999,999,999,999,999
6
1.0,4.,7.,10.,12.2,14.8,0.0,0.0
0.,0.,0.,0.,0.,0.,0.,0.
3.3,4.8,6.1,4.5,4.2,4.4,0.,0.
0.,0.,0.,0.,0.,0.,0.,0.
***DATA FOR SLAB # 2***
16.0625,0.8125,17.875
1.00,0.875,0.363,0.394,223.,1
0.0789,0.008,128.,2
16,16,16,0,0,1.
2,3,19,5,6,22,23,7,8,25
10,27,12,30,14,15,33,17,18,4
37,21,39,0,9,11,28,45,13,31
32,48,34,35,36,20,38,41,40,24
42,25,26,43,44,29,46,47,999,999
999,999,999,999,999,999,999,999,999,999
999,999,999,999,999,999,999,999,999,999
999,999,999,999,999,999,999,999,999,999
999,999,999,999,999,999,999,999,999,999
999,999,999,999,999,999,999,999,999,999
999,999,999,999,999,999,999,999,999,999
999,999,999,999,999,999,999,999,999,999
999,999,999,999,999,999,999,999,999,999
6
1.0,4.,7.,10.,12.2,14.8,0.0,0.0
0.,0.,0.,0.,0.,0.,0.,0.
4.1,4.9,4.3,2.9,3.9,3.3,0.,0.
0.,0.,0.,0.,0.,0.,0.,0.
```

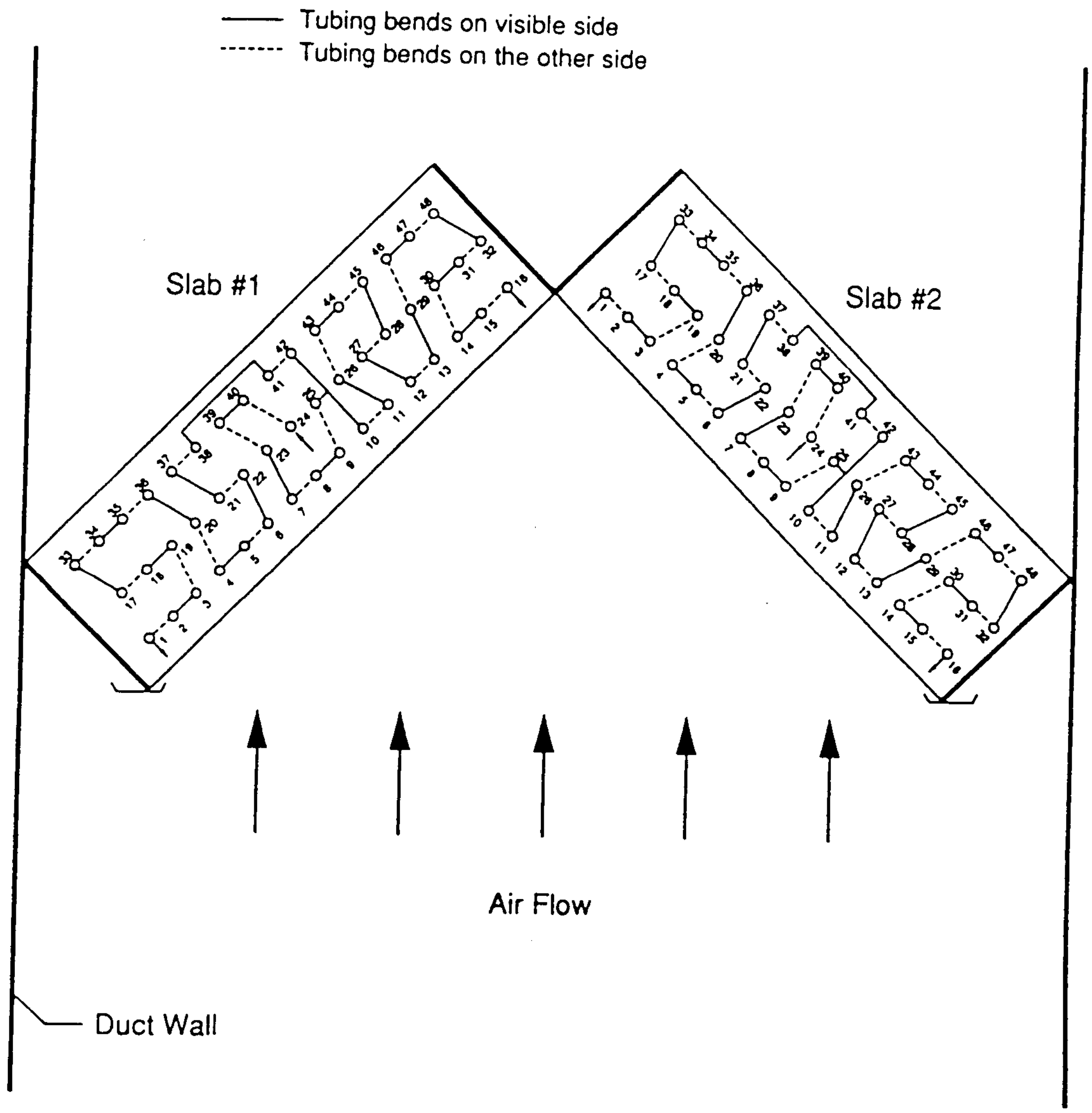


Figure A1. Example of circuitry specification for an A-shape coil.

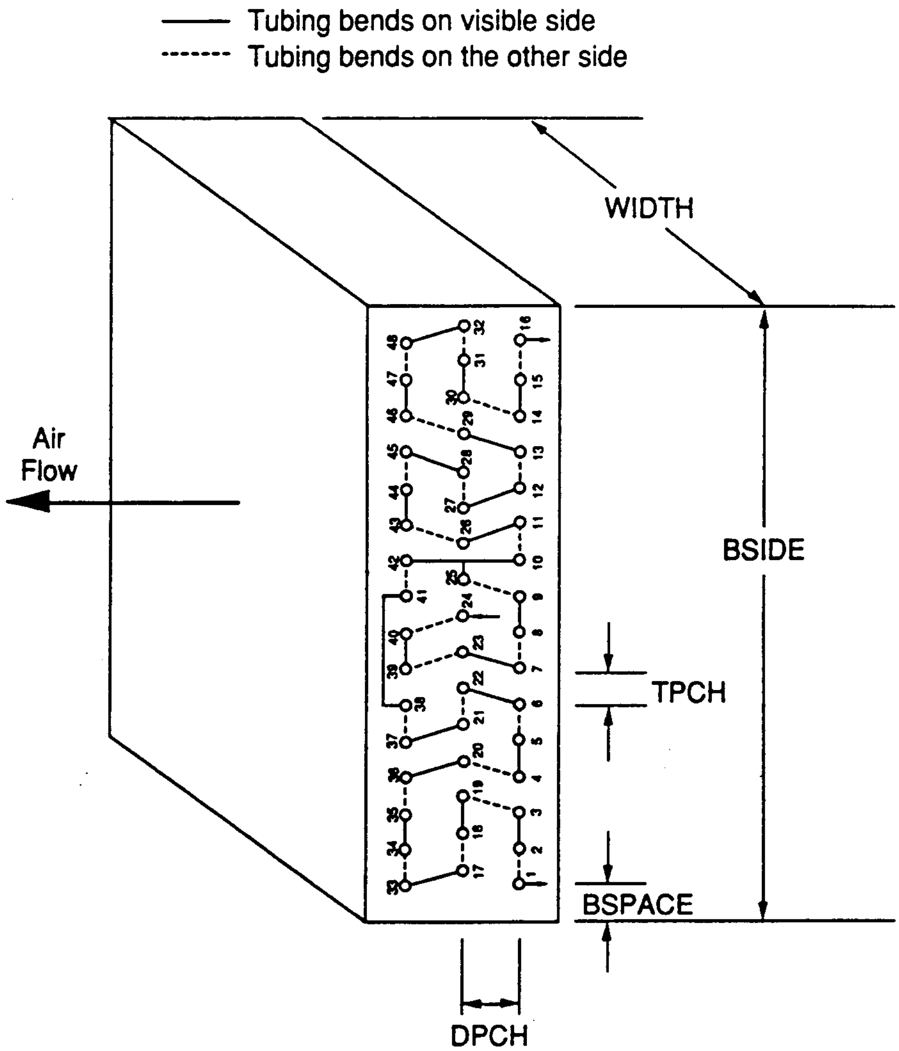


Figure A2. Specification of slab circuitry and dimensions.

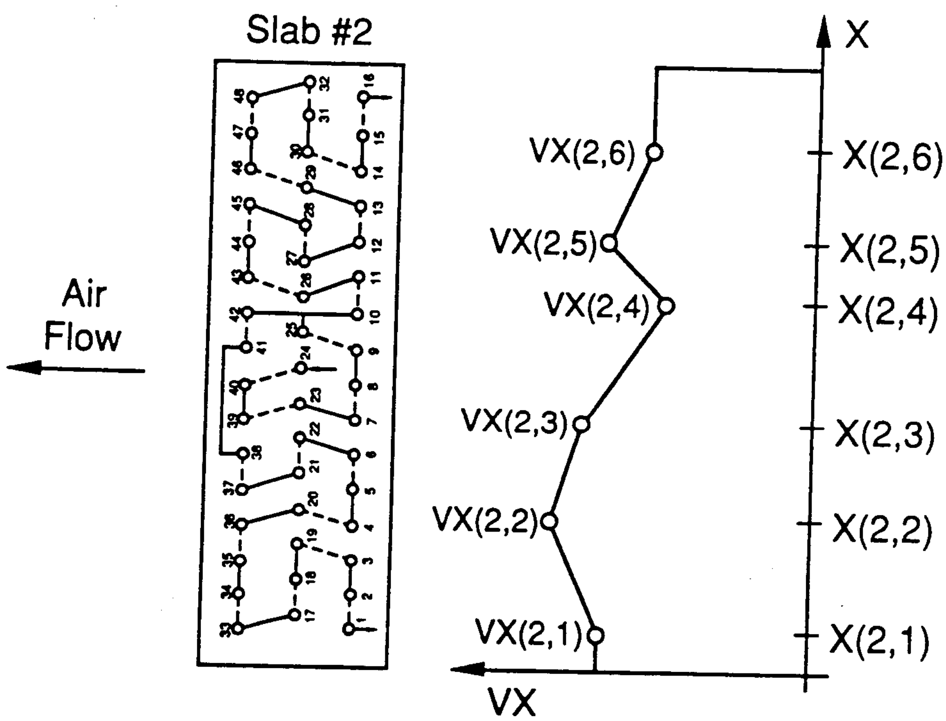


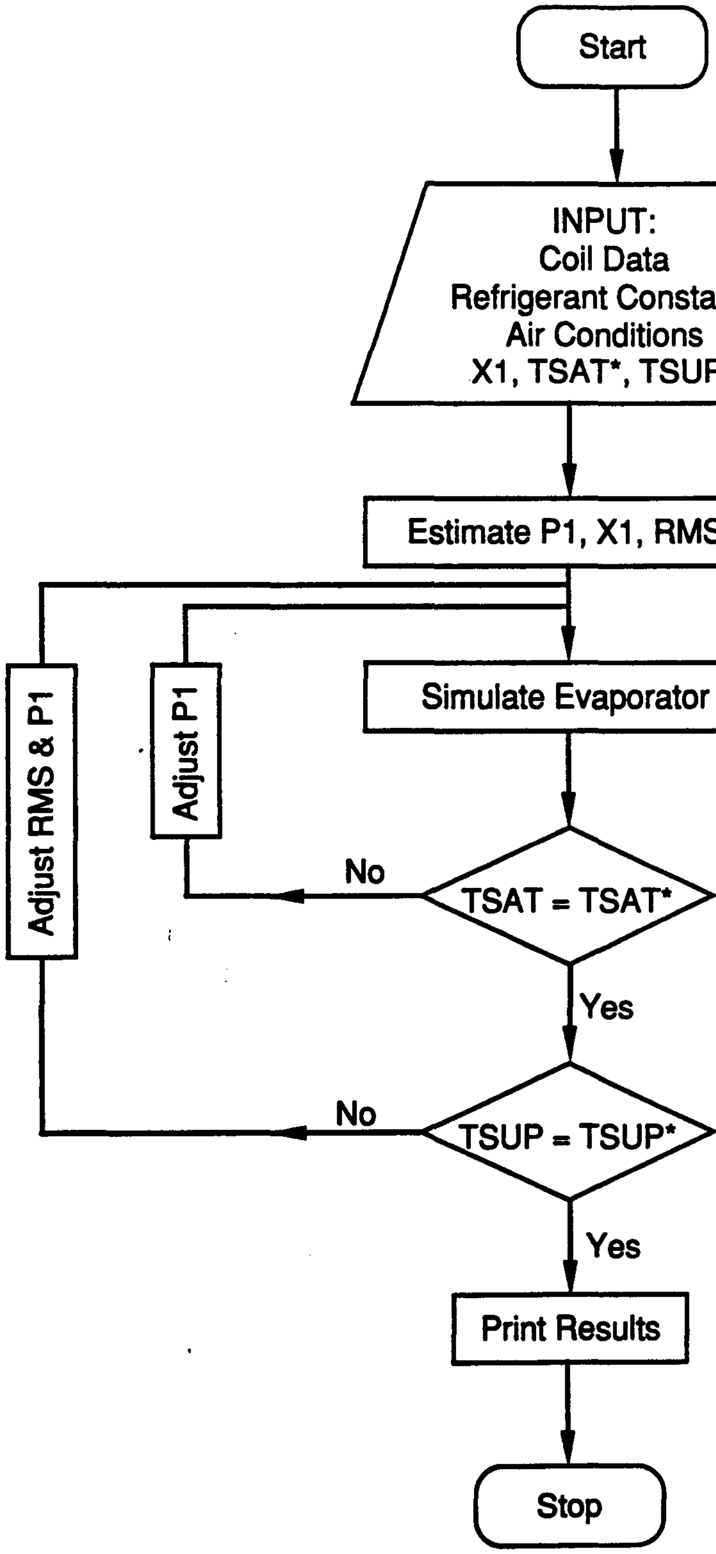
Figure A3. Example of air velocity measurement points and velocity profile.

APPENDIX B. Listing of the Program, EVSIM

The list of functions and subroutines used in EVSIM is given in Table B1. Figures B1 and B2 present flow charts of the main program and the evaporator simulation subroutine, respectively. The following pages contain listing of the main program, and subroutines and functions in the alphabetical order.

Table B1. Functions and Subroutines used in EVSIM

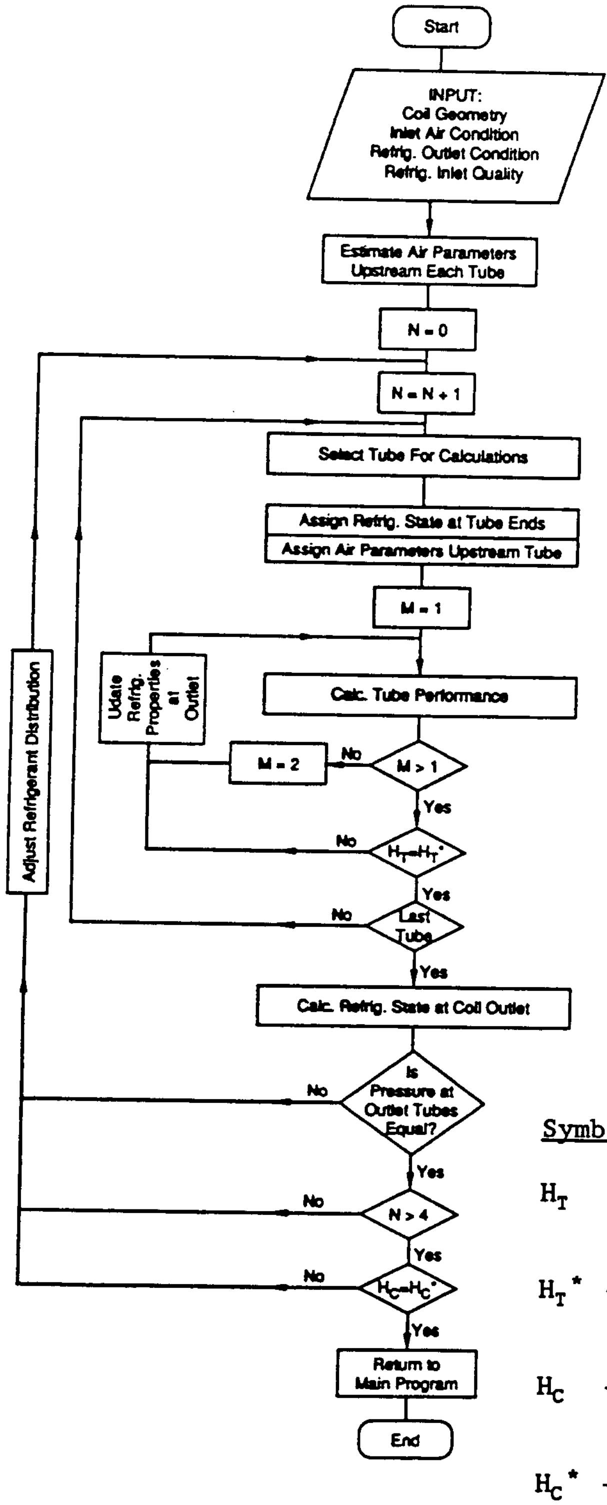
NAME	PURPOSE
AIRPR	Calculate properties of wet air
AIRHT3	Calculate air side heat transfer coefficient for flat, wavy or lanced fin
BALFL1	Adjust refrigerant flow distribution in an evaporator based on pressure drop in different circuits
CPCV	Calculate specific heat at constant volume and at constant pressure, specific heat ratio, and sonic velocity of refrigerant vapor from temperature and pressure
DATAIN	Read file DATAREF with refrigerant constants
DISTR2	Determine air distribution for each tube of the coil
DYNADP	Calculate dynamic pressure drop for a flow in a tube
EVGUNG	Calculate forced-convection evaporative heat transfer coefficient
EVPDP2	Calculate frictional evaporation pressure drop in a tube
EVPHX2	Simulate performance of an evaporator coil
FINCON	Calculate the thermal conductance for a fin-to-fin contact
FINEF2	Calculate fin efficiency
FRACT	Calculate refrigerant distribution in a split point based on pressure drop in the downstream circuits
HYDDIA	Calculate hydraulic diameter for air flow through a slab
ITRPR2	Calculate refrigerant vapor thermodynamic properties from given pressure and specific volume or enthalpy or entropy
MIXAIR	Calculate properties of the air stream resulted from the mixing process of two wet air streams
OVLWET	Calculate overall heat transfer coefficient for a wet finned tube
PHWET2	Calculate refrigerant thermodynamic properties from given pressure and enthalpy
RDATA3	Read file DTEV with evaporator coil data and prepare these data for EVPHX2
SATP	Calculate refrigerant saturation pressure from given temperature
SATPR	Calculate refrigerant dynamic viscosity and thermal conductivity for liquid and vapor, and liquid specific heat at saturation
SATT	Calculate refrigerant saturation temperature from given pressure
SATVF	Calculate specific volume of saturated liquid refrigerant from given temperature
SPHDP	Calculate pressure drop for a single-phase flow in a smooth tube
SPHDP1	Calculate pressure drop for a single-phase flow in a smooth tube
SPHTC	Calculate force-convection single-phase heat transfer coefficient in a smooth tube
TRACE3	Estimate refrigerant distribution in an evaporator coil based on circuitry configuration
VPSV	Calculate specific volume of vapor from given temperature and pressure
VPVHS	Calculate refrigerant vapor thermodynamic properties from given temperature and pressure
WATPR	Calculate water and frost properties



<u>Symbols</u>

P1	-	inlet pressure
RMS	-	refrig. mass flow rate
TSAT	-	calculated saturation temperature at coil outlet
TSAT*	-	requested saturation temperature at coil outlet
TSUP	-	calculated superheat at coil outlet
TSUP*	-	requested superheat at coil outlet
X1	-	inlet quality

Figure B1. Logic of the main program of EVSIM.



<u>Symbols</u>

H_T - refrig. enthalpy obtained from present iteration loop

H_T^* - refrig. enthalpy obtained from previous iteration loop

H_C - refrig. enthalpy at coil exit obtained from present iteration loop

H_C^* - refrig. enthalpy at coil exit obtained from previous iteration loop

Figure B2. Logic of the evaporator simulation subroutine, EVPHX2.

```
      H1=HF1+X1*(H1-HF1)
C**** CALC. REQUIRED EXIT ENTHALPY, H2REQ
      PSAT=SATP(TSAT2)
      T2REQ=TSAT2+TSUP2
      CALL VPVHS(2,T2REQ,PSAT,VG2,H2REQ,SG2,HFG)
      CALL CPCV(T2REQ,PSAT,CP,CV,GA,SO)
      PSLOPE=1.
C*******************************************************************
C**** START ITERATIONS
      IF(IDIA.NE.0) WRITE(IDIA0,*)'*DATA PREPARATION COMPLETED'
C****
      IF(IDIA.NE.0) WRITE(IDIA0,*)'****ITERATION STARTS****'
      DO 110 N=1,10
      DO 107 IIO=1,IEXP
      DO 105 M=1,5
      IF(IDIA.NE.0) WRITE(IDIA0,*)'N= ',N,'    M=',M,'    SLAB # ',IIO
C*******************************************************************
         CALL EVPHX2(KSLABS,IIO,RMASS,T1,P1,H1,TAIR,14.7,RH,
     &               T2,P2,H2,X2,S2,QL)
C*******************************************************************
      TG2=SATT(P2)
      IF(IDIA.NE.0) WRITE(IDIA,*)'*TG2 =',TG2
      PD=P2-PSAT
      IF(ABS(TG2-TSAT2).LT.0.05)GOTO 106
      IF(M.EQ.1)THEN
         P1N=P1-PD*PSLOPE
      ELSE
         PSLOPE=(P1A-P1)/(PDA-PD)
         P1N=P1-PD*PSLOPE
      END IF
      P1A=P1
      P1=P1N
      T1=SATT(P1)
  105 PDA=PD
      WRITE(IPR,*)' LOOP 105 DID NOT CONVERGE, IIO,PD  =',IIO,PD
  106 IF(IEXP.EQ.2)THEN
         IF(IIO.EQ.1)THEN
            H22=H2
            P22=P2
            QL1=QL
         ELSE
            H2=H22*SFLOW(1)+H2*SFLOW(2)
            P2=P22*SFLOW(1)+P2*SFLOW(2)
            QL=QL+QL1
         END IF
      END IF
  107 CONTINUE
      H2DIF=H2-H2REQ
      IF(IDIA.NE.0) WRITE(IDIA,*)'*********H2DIF=',H2DIF
      RMS(N)=RMASS
      HDS(N)=H2DIF
      QLS(N)=QL
      P2S(N)=P2
      IF(N.GT.2)THEN
         IF(ABS(HDS(N-1)).LT.5.*CP.AND.ABS(H2DIF).LT.2.*CP)GOTO 150
      END IF
      IF(N.EQ.1.)THEN
         RMASSN=RMASS*(H2-H1)/(H2REQ-H1)
      ELSE
         RMASSN=RMASSA-H2DIFA*(RMASS-RMASSA)/(H2DIF-H2DIFA)
      END IF
      IF(RMASSN.GT.RMASS)RMASSN=AMIN1(RMASSN,1.5*RMASS)
      IF(RMASSN.LT.RMASS)RMASSN=AMAX1(RMASSN,0.7*RMASS)
      P1=PSAT+(P1-P2)*(RMASSN/RMASS)**2
      RMASSA=RMASS
      RMASS=RMASSN
  110 H2DIFA=H2DIF
      IF(IDIA.NE.0) WRITE(IDIA,*)'*************************************'
      WRITE(IPR,*)'LOOP 110 DID NOT CONVERGE'
C**** CALC. & PRINT RESULTS FOR THE LAST ITERATION LOOP
  150 QT=RMASS*(H2-H1)
      DTIT=2.
      PP2=P2
```

```
      CALL PHWET2(PP2,H2,DTIT,0.001,T2,V2,S2,X2,TG2)
      TSUP=T2-TG2
      WRITE(IPR,900)
      WRITE(IPR,*)
      WRITE(IPR, *)'                         SIMULATION RESULTS'
      WRITE(IPR,*)
      WRITE(IPR,900)
  900 FORMAT('****************************************************',
     1'*********')
      WRITE(IPR,*)'*RESULTS FROM THE LAST ITERATION LOOP:'
      WRITE(IPR,*)
      WRITE(IPR,905)QT
  905 FORMAT('  TOTAL CAPACITY: ',1F7.0,' BTU/H')
      WRITE(IPR,907)QL
  907 FORMAT('  LATENT CAPACITY:',1F7.0,' BTU/H')
      WRITE(IPR,910)RMASS
  910 FORMAT('  REFRIGERANT MASS FLOW RATE:',1F6.1,' LB/H')
      WRITE(IPR,*)' REFRIGERANT PARAMETERS AT THE COIL OUTLET:'
      WRITE(IPR,915)TG2
  915 FORMAT(9X,'SATURATION TEMPERATURE:',1F6.1,' F')
      WRITE(IPR,920)TSUP
  920 FORMAT(22X,'SUPERHEAT:',1F6.1,' F')
      WRITE(IPR,925)P2
  925 FORMAT(23X,'PRESSURE:',1F6.1,' PSIA')
      WRITE(IPR,930)H2
  930 FORMAT(23X,'ENTHALPY:',1F6.1,' BTU/LB')
C**** PREPARE AND PRINT REFRIGERANT INFO FOR INDIVIDUAL OUTLET TUBES
      WRITE(IPR,*)
      WRITE(IPR,*)' INFORMATION ON REFRIGERANT LEAVING OUTLET TUBES'
      WRITE(IPR,*)
      WRITE(IPR,940)
  940 FORMAT('  SLAB #',3X,'TUBE #',6X,'P',8X,'T',6X,'TSUP',6X,'X',7X,
     1 'RMS',/,'    (-)',6X,'(-)',5X,'(PSIA)',5X,'(F)',6X,'(F)',5X,
     2 '(-)',4X,'(LB/H)')
      DO 160 IIO=1,NSLABS
      DO 160 L=1,NOUT(IIO)
      I=IOUT(IIO,L)
      HOUT=HR(IIO,I)
      POUT=PRM(IIO,2,I)
      CALL PHWET2(POUT,HOUT,DTIT,0.001,T2,V2,S2,X2,TG2)
      RMSI=RMASS*FLOW(IIO,I)
      TSUP=T2-TG2
  160 WRITE(IPR,944)IIO,I,POUT,T2,TSUP,X2,RMSI
  944 FORMAT(I5,1X,I9,1X,1F10.2,1F8.1,1F9.1,1F9.3,1F9.2)
C**** CALC. AND PRINT RESULTS FOR REQUESTED SUPERHEAT
      IF(H2DIF.NE.0.)THEN
         HDH=HDS(N)
         RMSH=RMS(N)
         P2H=P2S(N)
         QLH=QLS(N)
         HDL=HDS(N-1)
         RMSL=RMS(N-1)
         P2L=P2S(N-1)
         QLL=QLS(N-1)
         DO 120 I=1,N-2
         IF(HDH.NE.HDL)GOTO 122
         HDL=HDS(N-1-I)
         RMSL=RMS(N-1-I)
         P2L=P2S(N-1-I)
  120    QLL=QLS(N-1-I)
  122    H2L=H2REQ+HDL
         H2H=H2REQ+HDH
         HSLOPE=(H2REQ-H2L)/(H2H-H2L)
         RMASS=RMSL+(RMSH-RMSL)*HSLOPE
         P2=P2L+(P2H-P2L)*HSLOPE
         QL=QLL+(QLH-QLL)*HSLOPE
         H2=H2REQ
      END IF
      QT=RMASS*(H2-H1)
      DTIT=2.
      PP2=P2
      CALL PHWET2(PP2,H2,DTIT,0.001,T2,V2,S2,X2,TG2)
      TSUP=T2-TG2
```

```
C**** PRINT RESULTS
      WRITE(IPR,*)
      WRITE(IPR,900)
      WRITE(IPR,902)TSUP2
  902 FORMAT(' *RESULTS FOR THE REQUESTED REFRIGERANT SUPERHEAT:',F5.1)
      WRITE(IPR,*)' (INTERPOLATED FROM RESULTS OF LAST TWO ITERATIONS)'
      WRITE(IPR,*)
      WRITE(IPR,905)QT
      WRITE(IPR,907)QL
      WRITE(IPR,910)RMASS
      WRITE(IPR,*)' REFRIGERANT PARAMETERS AT THE COIL OUTLET:'
      WRITE(IPR,915)TG2
      WRITE(IPR,920)TSUP
      WRITE(IPR,925)P2
      WRITE(IPR,930)H2
      WRITE(IPR,900)
      STOP
      END
```

```
      SUBROUTINE AIRPR(I,T,PATM,RH,W,CP,R,AM,AK)
C
C**** PURPOSE:
C         TO CALCULATE AIR PROPERTIES
C         JANUARY 17, 1989
C
C**** INPUT DATA:
C        I     = 1   IF RELATIVE HUMIDITY IS GIVEN    (-)
C              = 2   IF HUMIDITY RATIO IS GIVEN    (-)
C        T     - AIR TEMPERATURE    (-)
C        PATH  - AIR PRESSURE    (PSIA)
C        RH    - RELATIVE HUMIDITY IN FRACTION (IF I=1)    (-)
C        W     - HUMIDITY RATIO (IF I=2)    (LBM H2O/LBM DRY AIR)
C
C**** OUTPUT DATA:
C        AK    - AIR THERMAL CONDUCTIVITY    (BTU/H*F*FT)
C        AM    - AIR DYNAMIC VISCOSITY    (LBM/FT*H)
C        CP    - AIR SPEC. HEAT AT CONSTANT PRESSURE    (BTU/LBM*F)
C        R     - GAS CONSTANT OF AIR    (LBF*FT/LBM*R)
C        RH    - RELATIVE HUMIDITY IN FRACTION (FOR I=2)    (-)
C        W     - HUMIDITY RATIO (FOR I=1)    (LBM H2O/LBM DRY AIR)
C
C
      DOUBLE PRECISION TR,Z,PSAT,P,WD,RHD,PW,CPD
      TR=T+460.
      Z=1000./TR
      IF(TR.GE.492.)GOTO10
      PSAT=DEXP(0.03940*Z**3-0.2755*Z*Z-10.431*Z+19.509)
      GOTO30
   10 IF(TR.GE.672.)GOTO20
      PSAT=DEXP(0.17829*Z*Z*Z-1.6896*Z*Z-5.0988*Z+13.4353)
      GOTO30
   20 PSAT=DEXP(0.71692*Z**4-4.01506*Z**3+7.5568*Z*Z-14.2131*Z+16.8255)
   30 IF(I.EQ.2)THEN
         P=W*PATM/(0.6198D0+W)
         RHD=P/PSAT
         RH=RHD
         WD=W
      ELSE
         RHD=RH
         IF(RH.GE.0.00001)GOTO40
         W=0.
         WD=0.D0
         GOTO50
   40    PW=RHD*PSAT
         WD=0.6198*PW/(PATM-PW)
         W=WD
   50    CONTINUE
      END IF
      CPD=0.2478786-0.4204563E-04*TR+0.5767857E-07*TR**2
     & -0.1493056E-10*TR**3
      CP=(CPD+0.444*WD)/(1.+WD)
      R=(53.34+85.76*WD)/(1.+WD)
      AM=5.5029E-03+8.7157E-05*TR-2.9464E-08*TR**2
     & +6.250E-12*TR**3
      AK=-2.853E-04+3.268E-05*TR-8.253E-09*TR*TR
     & +1.239E-12*TR**3
      RETURN
      END
```

```
      FUNCTION AIRHT3(IROW,DO,TPCH,DPCH,FPCH,FTK,WIDTH,DH,NDEP,IFIN,
     &                 AMASS,AVIS,ACP,AK)
C
C**** PURPOSE: TO CALCULATE DRY AIR-SIDE HEAT TRANSFER COEFFICIENT
C              FOR A TUBE WITH FLAT, WAVY, AND LANCED PLATE FINS.
C              SEPTEMBER 22 ,1988.
C
C**** NOTE: THIS FUNCTION CALCULATES H.T.C FOR A TUBE DEPENDING
C           ON THE DEPTH LOCATION OF THE TUBE IN THE SLAB.
C           GEOMETRIES ARE ASSUMED FOR WAVY & STRIP FINS.
C
C**** INPUT DATA:
C        ACP   - AIR SPEC. HEAT AT CONSTANT PRESSURE   (BTU/LBM*F)
C        AK    - AIR THERMAL CONDUCTIVITY    (BTU/H*FT*F)
C        AMASS - AIR MASS FLOW RATE ASSOCIATED WITH THE TUBE  (LBM/H)
C        AVIS  - AIR DYNAMIC VISCOSITY   (LBM/FT*H)
C        DH    - HYDRAULIC DIAMETER OF THE SLAB  (FT)
C        DO    - TUBE OUTSIDE DIAMETER   (FT)
C        DPCH  - TUBE DEPTH PITCH   (FT)
C        FPCH  - FIN PITCH   (FT)
C        FTK   - FIN THICKNESS   (FT)
C        IFIN  - FIN DESIGN DESCRIPTOR   (-)
C              = 1  FOR A FLAT FIN
C              = 2  FOR A WAVY FIN
C              = 3  FOR AN ENHANCED FIN (LANCED, STRIPPED)
C        IROW  - DEPTH ROW LOCATION OF THE TUBE   (-)
C        NDEP  - NUMBER OF DEPTH ROWS IN THE SLAB   (-)
C        TPCH  - TUBE PITCH IN THE ROW   (FT)
C        WIDTH - HEAT EXCHANGER WIDTH (TUBE LENGHT)  (FT)
C
C**** OUTPUT DATA:
C        AIRHT3 - DRY AIR-SIDE HEAT TRANSFER COEFFICIENT  (BTU/H*F*FT**2)
C
      COMMON/PRINT/IPR
      REAL JM,JN,J4,JTUBE,NUF,NUW
      DATA PI/3.1415927/,IERR/0/
C
C**** FLAT FIN ***************************************************
C**** REFERENCE: GRAY, D.L. AND WEBB, R.L., HEAT TRANSFER AND FRICTION
C                CORRELATIONS FOR PLATE FINNED-TUBE HEAT EXCHANGERS
C                HAVING PLAIN FINS, PROC. OF EIGHTH INT. H.T.
C                CONFERENCE, SAN FRANCISCO, 1986.
      FN=WIDTH/FPCH
      AREAC=(TPCH-DO)*(WIDTH-FN*FTK)
      GC=AMASS/AREAC
      RED=GC*DO/AVIS
      IF(RED.LT.500..OR.RED.GT.24700.)THEN
        WRITE(10,*)'AIRHT3, RED LIMIT 500-24700, RED =',RED
      END IF
      SPACE=FPCH-FTK
      TR=REAL(IROW)
      TR=AMIN1(TR,4.)
      J4=0.14*RED**(-.328)*(TPCH/DPCH)**(-.502)*(SPACE/DO)**0.0312
      JN=J4*0.991*(2.24*RED**(-.092)*(TR/4.)**(-.031))**(.607*(4.-TR))
      IF(TR.EQ.1.)THEN
         JTUBE=JN
      ELSE
         JM=(2.24*RED**(-.092)*((TR-1.)/4.)**(-.031))**(.607*(5.-TR))
         JM=J4*0.991*JM
         JTUBE=TR*JN-(TR-1.)*JM
      END IF
      PR=AVIS*ACP/AK
      AIRHT3=JTUBE*GC*ACP/PR**.66667
      IF(IFIN.EQ.3)THEN
C**** LOUVERED/LANCED FINS *******************************************
C**** REFERENCE: NAKAYAMA, W. AND XU, L.P., ENHANCED FINS FOR AIR-
C                COOLED HEAT EXCHANGERS - HEAT TRANSFER AND FRICTION
C                FACTOR CORRELATIONS, ASME-JSME THERMAL ENGINEERING
C                JOINT CONFERENCE, PROCEEDINGS, PP. 495-510, ASME, NY,
C                MARCH 1983.
         REH=GC*DH/AVIS
C****    NOTE: CORRELATION IS GOOD FOR FIS IN THE RANGE 0.2 - 0.35
```

```
         FIS=0.275
         IF(IERR.EQ.0)THEN
            IERR=1
            GAP=SPACE*3048.
            IF(GAP.LT.0.15.OR.GAP.GT.0.2)WRITE(10,*)'AIRHT3,
     1   GAP LIMIT IS  0.15 - 0.2 MM , GAP = ',GAP
            FP=FPCH*3048.
            IF(FP.LT.1.8.OR.FP.GT.2.5)WRITE(10,*)'AIRHT3,
     1   FIN PITCH LIMIT IS  1.8 - 2.5 MM, FP =',FP
            IF(REH.LT.250..OR.REH.GT.3000.)WRITE(10,*)'AIRHT3,
     1   REH LIMIT IS  250 - 3000, REH =',REH
         END IF
         FJ=1.+1093.*(FTK/SPACE)**1.24*FIS**0.944*REH**(-0.58)+
     1   1.097*(FTK/SPACE)**2.09*FIS**2.26*REH**0.88
         AIRHT3=AIRHT3*FJ
      END IF
      IF(IFIN.EQ.2)THEN
C**** WAVY FINS ******************************************************
C**** REFERENCE: TRAUGER, P. AND WEBB, R.L., A CORRELATION FOR THE
C                AIR SIDE HEAT TRANSFER COEFFICIENT FOR A WAVY FIN,
C                TO BE PUBLISHED, 1988.
C                THIS CORRELATION WAS DEVELOPED FOR 3 DEPTH ROW H-X.
C**** ASSUMPTIONS: 1. TWO FIN WAVES PER DEPTH PITCH
C                  2. WAVE DEPTH EQUALS THE SPACE BETWEEN FINS
         DC=DO+2.*FTK
         SP2=DPCH/2.
         SD=SPACE
         BETA=0.25*PI*DC*DC/(TPCH*DPCH)
         GAMMA=SQRT(1.+4.*(SD/SP2)**2)
         DHW=2.*SPACE*(1.-BETA)/(GAMMA*(1.-BETA)+2.*SPACE*BETA/DC)
         DHF=2.*SPACE*(1.-BETA)/(1.-BETA+2.*SPACE*BETA/DC)
         REDF=AMASS*DHF/(AVIS*AREAC*(1.-BETA))
         REDW=AMASS*DHW/(AVIS*AREAC*(1.-BETA))
         DEP=REAL(NDEP)
         GZW=REDW*PR*DHW/(DPCH*DEP)
         GZF=REDF*PR*DHF/(DPCH*DEP)
         IF(GZW.LT.25.)THEN
            NUW=0.5*GZW**.86*(TPCH/DC)**.11*(SPACE/DC)**(-.09)
            NUW=NUW*(SD/DPCH)**.12*(SP2/DPCH)**(-.34)
            NUF=0.4*GZF**0.73*(SPACE/DC)**(-0.23)*3.**0.23
         ELSE
            NUW=0.83*GZW**0.76*(TPCH/DC)**.13*(SPACE/DC)**(-.16)
            NUW=NUW*(SD/DPCH)**.25*(SP2/DPCH)**(-.43)
            NUF=0.53*GZF**0.62*(SPACE/DC)**(-0.23)*3.**0.31
         END IF
         CF=NUF/GZF
         CW=NUW/GZW
         IF(CF.GE..5.OR.CW.GE..5)THEN
            WRITE(10,*)'AIRHT3, IFIN=2, TOO LOW AIR FLOW RATE'
            WRITE(10,*)'FLAT FIN H.T.C ASSUMED'
         ELSE
            NUF=.25*GZF*ALOG((1.+2.*NUF/GZF)/(1.-2.*NUF/GZF))
            NUW=.25*GZW*ALOG((1.+2.*NUW/GZW)/(1.-2.*NUW/GZW))
            FJ=NUW/NUF
            AIRHT3=AIRHT3*FJ
         END IF
      END IF
      RETURN
      END
```

```
      SUBROUTINE BALFL1(KSLABS,III)
C
C**** PURPOSE:
C        TO ADJUST REFRIGERANT FLOW DISTRIBUTION IN AN EVAPORATOR
C        BASED ON PRESSURE DROP AT DIFFERENT CIRCUITS.
C        THE PROGRAM CAN HANDLE SINGLE SLAB AND TWO SLAB (A & V) COILS.
C        10-14-1988
C
C**** INPUT DATA:
C          IFROM(IIO,J)    - NUMBER OF THE TUBE FROM WHICH TUBE J RECEIVES
C                            REFRIGERANT.  IF THE TUBE IS CONNECTED TO THE
C                            INLET MAINFOLD, IFROM IS SET TO 0.
C          III             = 1  FOR THE FIRST SLAB   (-)
C                          = 2  FOR THE SECOND SLAB   (-)
C          KSLABS          - NUMBER OF SLABS IN THE EVAPORATOR ASSEMBLY
C                            TO BE CONSIDERED (KSLABS=2 OVERRIDES
C                            SPECIFICATION OF III)   (-)
C          IMER(IIO)       - NUMBER OF SPLIT POINTS   (-)
C          IOUT(IIO,L)     - NUMBER OF THE TUBE CONNECTED TO THE OUTLET
C                            MANIFOLD, FOUND AS L SUCH TUBE   (-)
C          KFEED(IIO,J,N) - NUMBER OF THE TUBE RECEIVING REFRIGERANT
C                            FROM TUBE J, FOUND AS N SUCH TUBE.
C                            NOTE THAT TUBE J CAN FEED UP TO 3 TUBES
C                            (N CAN BE 1,2 AND 3). KFEED IS SET TO -1 IF
C                            J TUBE FEEDS THE DISCHARGE MANIFOLD. KFEED
C                            IS SET TO 0 IF A TUBE IS NOT FED.   (-)
C          KSTART(IIO,N)   - NUMBER OF THE TUBE CONNECTED TO THE INLET
C                            MANIFOLD, FOUND AS N SUCH TUBE   (-)
C          KST(IIO)        - NUMBER OF TUBES CONNECTED TO THE INLET
C                            MANIFOLD   (-)
C          MERGE(IIO,K,1) - NUMBER OF THE TUBE WHICH FEEDS A SPLIT POINT,
C                            FOUND AS K SUCH TUBE   (-)
C          MERGE(IIO,K,2) - NUMBER OF TUBES FED BY THE TUBE K   (-)
C          NDEP(IIO)       - NUMBER OF TUBE DEPTH ROWS IN THE SLAB   (-)
C          NOUT(IIO)       - NUMBER OF TUBES CONNECTED TO THE OUTLET
C                            MANIFOLD   (-)
C          NTPS(IIO)       - NUMBER OF TUBES IN THE SLAB   (-)
C          PRM(IIO,1,I)    - REFRIG. PRESSURE AT INLET OF TUBE I (PSIA)
C          PRM(IIO,2,I)    - REFRIG. PRESSURE AT OUTLET OF TUBE I   (PSIA)
C
C**** OUTPUT DATA:
C          FLOW(IIO,J)     - FRACTION OF COIL TOTAL REFRIG. MASS FLOW
C                            PASSING THROUGH TUBE J   (-)
C
C**** SUBPROGRAMS CALLED BY BALFL1:   FRACT
C
      COMMON/HPHX/NSLABS,NDEP(2),NROW(2),DI(2),DO(2),DT(2),RPCH(2),
     & DPCH(2),WIDTH(2),FPCH(2),FTK(2),FMK(2),TMK(2),CFM1,BSIDE(2),
     & NTUB(2,5),IFROM(2,130),NTPS(2),BSPACE(2),
     & ACFM(2),IFIN(2),ISUR(2),SFLOW(2)
      COMMON/MERG/MERGE(2,20,2),IMER(2),IOUT(2,20),NOUT(2),
     & IDEPTH(2,130),FLOW(2,130),KFEED(2,130,3),KSTART(2,130),KST(2)
      COMMON/MASS/PRM(2,2,130)
      DIMENSION LEFT(20),PC(2,130),RN(20),F(20),ITUBE(20),ISEE(20)
C
C**** FIND REFRIGERANT FLOW DISTRIBUTION
C
      II1=III
      DO 120 IDO=1,KSLABS
      IF(KSLABS.EQ.1)THEN
         IIO=II1
      ELSE
         IIO=IDO
      END IF
      DO 65 I=1,IMER(IIO)
   65 LEFT(I)=MERGE(IIO,I,2)
      DO 70 I=1,NTPS(IIO)
   70 PC(IIO,I)=0.
      DO 120 IL=1,NOUT(IIO)
      I=IOUT(IIO,IL)
      POUT=PRM(IIO,2,I)
      DO 100 IT=1,NTPS(IIO)
```

```
      IP=IFROM(IIO,I)
      IF(IP.EQ.0)THEN
        PC(IIO,I)=POUT
        GOTO 120
      END IF
      IF(KFEED(IIO,IP,2).EQ.0)THEN
        I=IP
        GOTO 100
      END IF
      PC(IIO,I)=POUT
      DO 75 IM=1,IMER(IIO)
   75 IF(IP.EQ.MERGE(IIO,IM,1))GOTO 77
   77 LEFT(IM)=LEFT(IM)-1
      IF(LEFT(IM).GT.0)GOTO 120
      POUT=0.
      PUP=PRM(IIO,2,IP)
      NSPLIT=MERGE(IIO,IM,2)
      DO 90 I1=1,NSPLIT
      N=KFEED(IIO,IP,I1)
      POUT=POUT+PC(IIO,N)*FLOW(IIO,N)/FLOW(IIO,IP)
   90 RN(I1)=(PUP-PC(IIO,N))/FLOW(IIO,N)**1.75
      CALL FRACT(NSPLIT,RN,F)
      DO 92 I1=1,NSPLIT
      N=KFEED(IIO,IP,I1)
   92 FLOW(IIO,N)=F(I1)
      I=IP
  100 CONTINUE
  120 CONTINUE
      IF(KSLABS.EQ.1)THEN
         NSTART=KST(IIO)
      ELSE
         NSTART=KST(1)+KST(2)
      END IF
      DO 130 I=1,NSTART
      IF(KSLABS.EQ.1)THEN
         IIO=II1
         N=KSTART(IIO,I)
      ELSE
         IF(I.LE.KST(1))THEN
            IIO=1
            N=KSTART(1,I)
         ELSE
            IIO=2
            N1=KST(1)
            N=KSTART(2,I-N1)
         END IF
      END IF
  130 RN(I)=(PRM(IIO,1,N)-PC(IIO,N))/FLOW(IIO,N)**1.75
      CALL FRACT(NSTART,RN,F)
      DO 132 I=1,NSTART
      IF(KSLABS.EQ.1)THEN
         N=KSTART(IIO,I)
      ELSE
         IF(I.LE.KST(1))THEN
            IIO=1
            N=KSTART(1,I)
         ELSE
            IIO=2
            N1=KST(1)
            N=KSTART(2,I-N1)
         END IF
      END IF
  132 FLOW(IIO,N)=F(I)
C
C**** ASSIGN REFRIGERANT DISTRIBUTION,  FLOW(IIO,I)
      ISTORE=0
      IIO=II1
      DO 170 IDO=1,KSLABS
      IF(KSLABS.EQ.2)IIO=IDO
      DO 136 I=1,IMER(IIO)
      ITUBE(I)=0
  136 ISEE(I)=0
      DO 160 IS=1,KST(IIO)
```

```
      I=KSTART(IIO,IS)
      IL=1
      DO 150 IO=1,NOUT(IIO)
      DO 145 IT=1,NTPS(IIO)
      IN1=KFEED(IIO,I,IL)
      IF(IN1.EQ.-1)THEN
        IF(ISTORE.GT.0)THEN
          I=ITUBE(ISTORE)
          IL=ISEE(ISTORE)
          ISTORE=ISTORE-1
          GOTO 150
        END IF
        GOTO 160
      END IF
      IF(IL.GT.1)GOTO 137
      DO 135 I1=2,3
      IN2=KFEED(IIO,I,I1)
      IF(IN2.EQ.0)GOTO 137
      ISTORE=ISTORE+1
      ITUBE(ISTORE)=I
135   ISEE(ISTORE)=I1
137   IN2=KFEED(IIO,I,2)
      IF(IN2.GT.0)THEN
        FLOW(IIO,IN1)=FLOW(IIO,IN1)*FLOW(IIO,I)
      ELSE
        FLOW(IIO,IN1)=FLOW(IIO,I)
      END IF
      I=IN1
      IL=1
145   CONTINUE
150   CONTINUE
160   CONTINUE
170   CONTINUE
      IF(KSLABS.EQ.1)THEN
         DO 180 I=1,NTPS(IIO)
180      FLOW(IIO,I)=FLOW(IIO,I)*SFLOW(IIO)
      END IF
      RETURN
      END
```

```
      SUBROUTINE CPCV(TS,P,CP,CV,GAMMA,SONIC)
C
C**** PURPOSE:
C        TO CALCULATE FOR REFRIGERANT VAPOR
C        SPECIFIC HEAT AT CONSTANT PRESSURE,
C        SPECIFIC HEAT AT CONSTANT VOLUME,
C        SPECIFIC HEAT RATIO AND SONIC VELOCITY.
C**** JANUARY 9, 1989
C
C**** INPUT DATA:
C        TS     - REFRIG. VAPOR TEMPERATURE    (F)
C        P      - REFRIG. VAPOR PRESSURE    (PSIA)
C        REFRIG. CONSTANTS - REFER TO THE MAIN PROGRAM.
C
C**** OUTPUT DATA:
C        CP     - SPEC. HEAT AT CONSTANT PRESSURE    (BTU/LBM*F)
C        CV     - SPEC. HEAT AT CONSTANT VOLUME    (BTU/LBM*F)
C        GAMMA - SPEC. HEAT RATIO    (-)
C        SONIC - SONIC VELOCITY    (FT/SEC)
C
C**** SUBPROGRAMS CALLED BY CPCV:
C        SATT,VPSV
C
      DOUBLE PRECISION Z,Z2,Z3,V2,V3,V4,V5,V6,AKTTC,AXTTC,FDPDV,
     1 FDPDT,DPDT,FCCV,CVD,CPD,GAMMAD
      COMMON/PRINT/IPR
      COMMON/CONST/TC,PC,VC,TFR,AJ,EEP
      COMMON/STATE/A1,B1,C1,A2,B2,C2,A3,B3,C3,A4,B4,C4,A5,B5,C5,
     &A6,B6,C6,ALPHA,AK
      COMMON/SPHTV/AC,BC,CC,DC,EC,FC,X,Y
      SAVE TSLAST,PLAST,CPLAST,CVLAST,GLAST,SLAST
      DATA TSLAST,PLAST/-1.,-1./
C
      T=TS+TFR
      IF(T.LE.0.)GOTO 999
      IF(P.LT.0.)GOTO 999
      IF(ABS(TS-TSLAST).GT. 1.0E-4)GOTO 5
      IF(ABS(P-PLAST).GT. 1.0E-4)GOTO 5
      CP=CPLAST
      CV=CVLAST
      GAMMA=GLAST
      SONIC=SLAST
         RETURN
    5 TG=SATT(P)
      IF(TS.LT.TG)TS=TG
      V=VPSV(P,TS)
      V1=V-B1
      V2=V1*V1
      V3=V1*V2
      V4=V1*V3
      V5=V1*V4
      V6=V1*V5
      AKTTC=AK*T/TC
      AXTTC=DEXP(-AKTTC)
      Z=ALPHA*V
      Z2=2.*Z
      Z3=3.*Z
      IF(Z.GT.150.D0)Z=150.D0
      IF(Z2.GT.150.D0)Z=150.D0
      IF(Z3.GT.150.D0)Z3=150.D0
      FDPDV=0.
      FDPDT=0.
      IF(ABS(A6).LE.1.E-20)GOTO4
      IF(ABS(C1).GE.1.E-20)GOTO3
      FDPDV=-ALPHA*DEXP(-Z)*(A6+B6*T)
      FDPDT=B6*DEXP(-Z)
      GOTO4
    3 FDPDV=-(ALPHA*(DEXP(-Z3)+2.*C1*DEXP(-Z2))/(DEXP(-Z2)+2.*C1*
     &DEXP(-Z)+C1*C1))*(A6+B6*T+C6*AXTTC)
      FDPDT=(B6-AK*C6*AXTTC/TC)*DEXP(-Z2)/(DEXP(-Z)+C1)
    4 DPDV=-A1*T/V2-2.*(A2+B2*T+C2*AXTTC)/V3-3.*(A3+
     &B3*T+C3*AXTTC)/V4+4.*(A4+B4*T+C4*AXTTC)/V5-5.*(A5+B5
```

```
     &*T+C5*AXTTC)/V6+FDPDV
      DPDT=A1/V1+(B2-AK*C2*AXTTC/TC)/V2+(B3-AK*C3*AXTTC/TC)/V3
     &+(B4-AK*C4*AXTTC/TC)/V4+(B5-AK*C5*AXTTC/TC)/V5+FDPDT
      FCCV=0.
      IF(ABS(C1).GE.1.E-20)FCCV=C6*DEXP(-Z)/ALPHA-(C6*C1/ALPHA)*
     &DLOG(1.D0+DEXP(-Z)/C1)
      CVD=AC+BC*T+CC*T**2+DC*T**3+FC/T**2-(0.185053*AK**2*T*AXTTC/TC**2)
     &*(C2/V1+C3/(2.*V2)+C4/(3.*V3)+C5/(4.*V4)+FCCV)
      CPD=CVD-0.185053*T*DPDT**2/DPDV
      GAMMAD=CPD/CVD
      CP=CPD
      CV=CVD
      GAMMA=GAMMAD
      SONIC=V*DSQRT(857.36091*T*DPDT**2/CVD-4633.056*DPDV)
      CPLAST=CP
      CVLAST=CV
      GLAST=GAMMA
      SLAST=SONIC
      TSLAST=TS
      PLAST=P
         RETURN
  999 WRITE(IPR,100)
  100 FORMAT(5X,'ERROR IN CALLING -CPCV-')
         RETURN
      END
```

```
      SUBROUTINE DATAIN
C
C**** PURPOSE:
C        TO READ REFRIGERANT CONSTANTS
C        1/5/87
C
      COMMON/PRINT/IPR
      COMMON/CONST/TC,PC,VC,TFR,AJ,EEP
      COMMON/TGPG/AG,BG,CG,DG,EG,FG,AA,BB
      COMMON/DENSF/AL,BL,CL,DL,EL,BPL,CPL,DPL,EPL
      COMMON/STATE/A1,B1,C1,A2,B2,C2,A3,B3,C3,A4,B4,C4,A5,B5,C5,
     $ A6,B6,C6,ALPHA,AK
      COMMON/SPHTV/AC,BC,CC,DC,EC,FC,X,Y
      COMMON/COEFPR/A(5,12)
      OPEN (UNIT=7,FILE='DATAREF',STATUS='OLD')
C*******************************************************************
C**** INPUT REFRIGERANT DATA
      READ(7,800)T1,T2,T3,T4,T5
      WRITE(IPR,802)T1,T2,T3,T4,T5
      READ(7,*)TC,PC,VC
      READ(7,*)TFR,AJ,EEP
      READ(7,*)AG,BG,CG
      READ(7,*)DG,EG,FG
      READ(7,*)AA,BB
      READ(7,*)AL,BL,CL
      READ(7,*)DL,EL,BPL
      READ(7,*)CPL,DPL,EPL
      READ(7,*)A1,B1,C1
      READ(7,*)A2,B2,C2
      READ(7,*)A3,B3,C3
      READ(7,*)A4,B4,C4
      READ(7,*)A5,B5,C5
      READ(7,*)A6,B6,C6
      READ(7,*)ALPHA,AK
      READ(7,*)AC,BC,CC
      READ(7,*)DC,EC,FC
      READ(7,*)X,Y
      DO10I=1,5
      READ(7,*)(A(I,J),J=1,3)
      READ(7,*)(A(I,J),J=4,6)
      READ(7,*)(A(I,J),J=7,9)
      READ(7,*)(A(I,J),J=10,12)
   10 CONTINUE
C*******************************************************************
      REWIND 7
      CLOSE (UNIT=7, STATUS='KEEP')
C
  800 FORMAT(5A4)
  802 FORMAT(1X,5A4)
C
      RETURN
      END
```

```
      SUBROUTINE DISTR2
C
C**** PURPOSE:
C       TO DETERMINE AIR FLOW DISTRIBUTION IN A TUBE-FINNED COIL
C       JUNE 15,1988
C
C**** INPUT DATA:
C       II0            = 1  FOR THE FIRST SLAB   (-)
C                      = 2  FOR THE SECOND SLAB   (-)
C       NDEP(II0)      - NUMBER OF TUBE DEPTH ROWS   (-)
C       NTUB(II0,N)    - NUMBER OF TUBES IN ROW N IN THE SLAB   (-)
C       NTPS(II0)      - NUMBER OF TUBES IN THE SLAB   (-)
C
C**** OUTPUT DATA:
C       AMR(II0,I)     - RATIO OF AIR MASS FLOW FOR A GIVEN TUBE TO
C                        THE TOTAL MASS FLOW RATE FOR THE GIVEN SLAB  (-)
C       GET(II0,I,1)   - FRACTION OF THE AIR FLOW OF TUBE IGET(II0,I,1)
C                        PASSING THROUGH TUBE 'I'.
C       GET(II0,I,2)   - FRACTION OF THE AIR FLOW OF TUBE IGET(II0,I,2)
C                        PASSING THROUGH TUBE 'I'. IF IGET(II0,I,2)=0,
C                        GET(II0,I,2) IS SET TO 0.    (-)
C       IGET(II0,I,1) - NUMBER OF A TUBE FROM WHICH TUBE 'I' IS
C                        GETING AIR   (-)
C       IGET(II0,I,2) - NUMBER OF A SECOND TUBE FROM WHICH TUBE 'I'
C                        IS GETING AIR.  IF THE SECOND TUBE
C                        DOES NOT EXIST, IGET(II0,I,2)=0.
C
      COMMON/PRINTD/IDIA,IDIA0,NNN,MMM
      COMMON/PRINT/IPR
      COMMON/HPHX/NSLABS,NDEP(2),NROW(2),DI(2),DO(2),DT(2),TPCH(2),
     & DPCH(2),WIDTH(2),FPCH(2),FTK(2),FMK(2),TMK(2),CFM1,BSIDE(2),
     & NTUB(2,5),IFROM(2,130),NTPS(2),BSPACE(2),
     & ACFM(2),IFIN(2),ISUR(2),SFLOW(2)
      COMMON/ATEST/X(2,18),VX(2,18),NTEST(2)
      COMMON/AIRD/IGET(2,130,2),GET(2,130,2),AMR(2,130)
C
C**** CHECK AGRREMENT BETWEEN LOCAL AND TOTAL CFM MEASUREMENTS
C     APPLY CORRECTION TO THE LOCAL MEASUREMENT VALUES
      CFM2=0.
      DO 10 II0=1,NSLABS
      DO 10 I=2,NTEST(II0)
   10 CFM2=0.5*(VX(II0,I-1)+VX(II0,I))*(X(II0,I)-X(II0,I-1))*WIDTH(II0)
     1    +CFM2
      CFM2=CFM2*60.
      CFMR=CFM2/CFM1
      PER=ABS(CFMR-1.)*100.
      IF(IDIA.NE.0)THEN
         WRITE(IDIA0,*)'CFM1,CFM2=',CFM1,CFM2
         WRITE(IDIA0,*)
         WRITE(IDIA0,*)'LOCAL MEASUREMENTS OF AIR DISTRIBUTION'
         IF(CFMR.GT.1.)THEN
            WRITE(IDIA0,*)'OVERESTIMATE CFM BY ',PER,' %'
         ELSE
            WRITE(IDIA0,*)'UNDERESTIMATE CFM BY ',PER,' %'
         END IF
         WRITE(IDIA0,*)'CORRECTION TO LOCAL VELOCITY VALUES IS APPLIED'
         IF(IDIA.NE.0) WRITE(IDIA0,*)
      END IF
      TBSIDE=BSIDE(1)
      IF(NSLABS.EQ.2)TBSIDE=TBSIDE+BSIDE(2)
      VMEAN=CFM1/(WIDTH(1)*TBSIDE*60.)
      IF(IDIA.NE.0) WRITE(IDIA0,*)'VMEAN=',VMEAN
      DO 16 II0=1,NSLABS
      DO 16 I=1,NTEST(II0)
   16 VX(II0,I)=VX(II0,I)/CFMR
      DO 18 II0=1,NSLABS
      ACFM(II0)=0.
      DO 18 I=2,NTEST(II0)
   18 ACFM(II0)=0.5*(VX(II0,I-1)+VX(II0,I))*(X(II0,I)-X(II0,I-1))*
     1    WIDTH(II0)*60.0+ACFM(II0)
C**** DETERMINE AIR DISTRIBUTION FOR EACH TUBE IN THE FIRST ROW
C     OF EACH SLAB
```

```
      DO 70 IIO=1,NSLABS
      DO 30 ITUBE=1,NTUB(IIO,1)
C     FIND XLL & XRR
      IF(ITUBE.EQ.1)THEN
         XLL=0.
         XRR=BSPACE(IIO)+0.5*TPCH(IIO)
      ELSE
         XLL=BSPACE(IIO)+(REAL(ITUBE)-1.5)*TPCH(IIO)
         XRR=XLL+TPCH(IIO)
         IF(ITUBE.EQ.NTUB(IIO,1))XRR=BSIDE(IIO)
      END IF
      V=0.
      XR=XRR
      DO 20 I=1,NTEST(IIO)
   20 IF(X(IIO,I).GE.XR)GOTO 22
      WRITE(*,*)'BAD INPUT COIL DATA, MESSAGE FROM DISTR2, LOOP 20'
   22 CONTINUE
C**** CALC. AVERAGE VELOCITY FOR A TUBE AND "AMR"
      DO 25 L=1,5
      X1=X(IIO,I)
      X2=X(IIO,I-1)
      V1=VX(IIO,I)
      V2=VX(IIO,I-1)
C     WRITE(*,225)X2,X1,V2,V1
C 225 FORMAT('X2,X1,V2,V1=',4F10.4)
      XL=AMAX1(XLL,X2)
C     WRITE(*,226)XL,XR
C 226 FORMAT('XL,XR=',2F8.4)
      V=V+(V1+(V2-V1)/(X2-X1)*(0.5*(XL+XR)-X1))*(XR-XL)/(XRR-XLL)
      IF(XLL.EQ.XL)GOTO 27
      XR=XL
   25 I=I-1
      WRITE(*,*)'ERROR IN DISTR2, LOOP 25, IIO,ITUBE=',IIO,ITUBE
   27 CFMI=V*WIDTH(IIO)*(XRR-XLL)*60.
      AMR(IIO,ITUBE)=CFMI/ACFM(IIO)
C     WRITE(*,*)'ITUBE,V,AMR=',ITUBE,V,AMR(IIO,ITUBE)
   30 CONTINUE
C
C**** DETERMINE IGET(IIO,130,2) & GET(IIO,130,2) VALUES
C     FOR DEPTH ROWS 2,3,4 AND 5
      NDEPP=NDEP(IIO)
      NTPSS=NTPS(IIO)
C**** FIND THE MAX. NUMBER OF TUBES IN ONE DEPTH ROW
      N1=NTUB(IIO,1)
      NMAX=N1
      DO 32 ICT=2,NDEPP
      N0=NTUB(IIO,ICT)
   32 NMAX=MAX0(N0,N1,NMAX)
      NROW(IIO)=NMAX
      DO 33 I=1,NTPSS
      IGET(IIO,I,2)=0
   33 GET(IIO,I,2)=0.
      N0=N1
      NL=N1
C
      DO 50 ICT=2,NDEPP
      N=NTUB(IIO,ICT)
      NF=NL+1
      NL=NL+N
      IF(N.EQ.N0)THEN
C**** CASE 1,                    N.EQ.N0
        DO 34 I=NF,NL
        IGET(IIO,I,1)=I-N0
   34   GET(IIO,I,1)=1.
      ELSE IF (N.GT.N0) THEN
C**** CASE 2,                    N.GT.N0
        IGET(IIO,NF,1)=NF-N0
        GET(IIO,NF,1)=0.66666
        IGET(IIO,NF+1,1)=NF-N0
        GET(IIO,NF+1,1)=0.33333
        IGET(IIO,NF+1,2)=NF-N0+1
        GET(IIO,NF+1,2)=0.5
        IGET(IIO,NL,1)=NF-1
```

```
      GET(IIO,NL,1)=0.66666
      IGET(IIO,NL-1,1)=NF-1
      GET(IIO,NL-1,1)=0.33333
      IGET(IIO,NL-1,2)=NF-2
      GET(IIO,NL-1,2)=0.5
      DO 38 I=NF+2,NL-2
      IGET(IIO,I,1)=I-N0-1
      IGET(IIO,I,2)=I-N0
      GET(IIO,I,1)=0.5
 38   GET(IIO,I,2)=0.5
     ELSE
C**** CASE 3,                     N.LT.N0
      IGET(IIO,NF,1)=NF-N0
      GET(IIO,NF,1)=1.
      IGET(IIO,NF,2)=NF-N0+1
      GET(IIO,NF,2)=0.5
      IGET(IIO,NL,1)=NF-2
      GET(IIO,NL,1)=0.5
      IGET(IIO,NL,2)=NF-1
      GET(IIO,NL,2)=1.0
      DO 40 I=NF+1,NL-1
      IGET(IIO,I,1)=I-N0
      IGET(IIO,I,2)=I-N0+1
      GET(IIO,I,1)=0.5
 40   GET(IIO,I,2)=0.5
     END IF
     N0=N
 50  CONTINUE
     DO 60 I=N1+1,NTPSS
     J=IGET(IIO,I,1)
     AMR(IIO,I)=AMR(IIO,J)*GET(IIO,I,1)
     J=IGET(IIO,I,2)
 60  IF(J.NE.0)AMR(IIO,I)=AMR(IIO,I)+AMR(IIO,J)*GET(IIO,I,2)
 70  CONTINUE
     RETURN
     END
```

```
      FUNCTION DYNADP(P1,H1,P2,H2,RMS,D)
C
C**** PURPOSE:
C        TO CALCULATE DYNAMIC PRESSURE DROP
C        FOR FLOW IN A TUBE
C
C**** INPUT DATA:
C       D        - TUBE DIAMETER   (FT)
C       H1       - REFRIG. ENTHALPY AT TUBE INLET   (BTU/LBM)
C       H2       - REFRIG. ENTHALPY AT TUBE OUTLET   (BTU/LBM)
C       P1       - REFRIG. PRESSURE AT TUBE INLET   (PSIA)
C       P2       - REFRIG. PRESSURE AT TUBE OUTLET   (PSIA)
C       RMS      - REFRIG. MASS FLOW RATE   (LBM/H)
C
C**** OUTPUT DATA:
C       DYNADP - DYNAMIC PRESSURE DROP   (PSI)
C
C**** SUBPROGRAMS CALLED BY DYNADP:
C       CPCV,ITRPR2,SATPR,SATT,SATVF,VPVHS
C
      DO100I=1,2
      P=P1
      H=H1
      IF(I.EQ.1)GOTO 5
      P=P2
      H=H2
    5 TG=SATT(P)
      CALL VPVHS(1,TG,P,VG,HG,SG,HF)
      IF(H.LT.HF)GOTO 10
      IF(H.GT.HG)GOTO 20
      X=1./(HG-HF)
      X=X*(H-HF)
      VF=SATVF(TG)
      V=VF+X*(VG-VF)
      GOTO 30
   10 CPF=SATPR(5,TG)
      T=TG+(H-HF)/CPF
      V=SATVF(T)
      GOTO 30
   20 TT=TG+10.
      CALL CPCV(TT,P,CP,CV,GA,SO)
      T=TG+(H-HG)/CP
      PR2=H
      CALL ITRPR2(T,2.,P,2,0.001,PR2,V,H,S)
   30 V2=V
      IF(I.EQ.1)V1=V
  100 CONTINUE
      G=0.7853982*D*D
      G=RMS/G
      G=G*G/(32.2*144.*3600.*3600.)
      DYNADP=G*(V2-V1)
      RETURN
      END
```

```
      FUNCTION EVGUNG(RMS,X1,X2,TGI,HFG,VLIQ,VVAP,D,AL)
C
C****PURPOSE:
C        TO COMPUTE EVAP. HEAT TRANSFER COEFF. FOR R22 FLOW IN A SMOOTH
C        TUBE.   JAN 13, 1989
C**** INPUT DATA:
C        AL      - TUBE LENGTH   (FT)
C        D       - TUBE DIAMETER   (FT)
C        HFG     - REFRIG. HEAT OF EVAPORATION   (BTU/LBM)
C        RMS     - REFRIG. MASS FLOW RATE   (LBM/H)
C        TG1     - REFRIG. SATURATION TEMPERATURE   (F)
C        VLIQ    - SPEC. VOLUME OF SAT. LIQUID   (LBM/FT**3)
C        VVAP    - SPEC. VOLUME OF SAT. VAPOR   (LBM/FT**3)
C        X1      - REFRIG. QUALITY AT TUBE INLET   (-)
C        X2      - REFRIG. QUALITY AT TUBE OUTLET   (-)
C**** OUTPUT DATA:
C        EVGUNG - EVAPORATION GUNG TRANSFER COEFFICIENT   (BTU/H*F*FT**2)
C
C**** SUBPROGRAMS CALLED BY EVGUNG: SATP, SATPR
C
C**** REFERENCE:
C        GUNGOR, K.E. AND WINTERTON, R.H.S., A GENERAL CORRELATION FOR
C        FLOW BOILING IN TUBES AND ANNULI, INT.J.HEAT MASS TRANSFER,
C        VOL 29, NO. 3, PP. 352-358, 1986.
C
      DOUBLE PRECISION PRED,G,QFLUX,BO,FRL,XAV,PR,RELIQ,HTCLIQ,XTT,E,
     1 S,HPOOL,EVHTC
      PRED=SATP(TGI)/721.9
      G=RMS/(0.7853982*D*D)
      QFLUX=RMS*HFG*(X2-X1)/(AL*D)
      BO=QFLUX/(G*HFG)
      FRL=(G*VLIQ)**2/(32.2*D)
      XAV=0.5*(X1+X2)
      VISLIQ=SATPR(1,TGI)
      VISVAP=SATPR(2,TGI)
      CONLIQ=SATPR(3,TGI)
      CPLIQ=SATPR(5,TGI)
      PR=VISLIQ*CPLIQ/CONLIQ
      RELIQ=G*(1.-XAV)*D/VISLIQ
      HTCLIQ=0.023D0*RELIQ**0.8*PR**0.4*CONLIQ/D
      XTT=(1.D0/XAV-1.D0)**.9*(VLIQ/VVAP)**.5*(VISLIQ/VISVAP)**.1
      E=1.D0+24000.D0*BO**1.16+1.37D0/XTT**0.86
      IF(FRL.LT.0.05D0)E=E*FRL**(0.1D0-2.D0*FRL)
      S=1.D0/(1.D0+1.15D-6*E*E*RELIQ**1.17)
      IF(FRL.LT.0.05D0)S=S*DSQRT(FRL)
      HPOOL=55.D0*PRED**.12*(-DLOG10(PRED))**(-.55)
      HPOOL=HPOOL*86.46D0**(-.5)*(3.1525D0*QFLUX)**0.67/5.6745D0
      EVHTC=E*HTCLIQ+S*HPOOL
      EVGUNG=EVHTC
      RETURN
      END
```

```
      FUNCTION EVPDP2(RMS,TGI,X1,X2,HFG,VLIQ,VVAP,VISL,VISV,AL,D)
C
C**** PURPOSE:
C       TO COMPUTE FRICTIONAL EVAPORATION PRESSURE DROP
C       FOR FLOW IN A TUBE   , OCT-19-87
C
C**** INPUT DATA:
C       AL    - TUBE LENGTH   (FT)
C       D     - TUBE INSIDE DIAMETER   (FT)
C       HFG   - REFRIG. HEAT OF CONDENSATION    (BTU/LBM)
C       RMS   - REFRIG. MASS FLOW RATE   (LBM/H)
C       TGI   - SAT. TEMPERATURE OF REFRIG.   (F)
C       X1    - REFRIG. QUALITY AT TUBE INLET   (-)
C       X2    - REFRIG. QUALITY AT TUBE OUTLET   (-)
C       VISL  - LIQUID DYNAMIC VISCOSITY   (LBM/H*FT)
C       VISV  - VAPOR DYNAMIC VISCOSITY   (LBM/H*FT)
C       VLIQ  - SPEC. VOLUME OF LIQUID REFRIG.   (FT**3/LBM)
C       VVAP  - SPEC. VOLUME OF VAPOR REFRIG.   (FT**3/LBM)
C
C**** OUTPUT DATA:
C       EVPDP - FRICTIONAL EVAPORATION PRESSURE DROP   (PSI)
C
C**** NOTE: PIERRE CORRELATION USED UNLESS CALCULATED PRESSURE DROP IS
C           SMALLER THAN SINGLE PHASE PRESSURE DROP FOR X*RMS VAPOR.
C
      DATA AC/1.6654E-11/
      G=RMS/(0.78539816*D*D)
      RE=G*D/VISL
      AKF=778.*HFG*(X2-X1)/AL
      RATIO=RE/AKF
      RATIO=AMAX1(1.,RATIO)
      F=0.0185/RATIO**.25
      XM=0.5*(X1+X2)
      V2PH=VLIQ+XM*(VVAP-VLIQ)
      EVPDP2=AC*F*V2PH*G*G*AL/D
      RE=G*XM*D/VISV
      F1=0.046/RE**0.2
      RATE=2.*XM**2*F1*VVAP/(F*V2PH)
      IF(RATE.GT.1.)EVPDP2=EVPDP2*RATE
      RETURN
      END
```

```
      SUBROUTINE EVPHX2(KSLABS,II1,RMASS,T1,P1,H1,ATIN,
     & APIN,ARHIN,T2,P2,H2,X2,S2,QL)
C
C     NOV, 22 1988
C
C**** EVAPORATOR SIMULATION    .
C
C**** THIS PROGRAM COMPUTES IN THE FORWARD TUBE-BY-TUBE SCHEME
C**** PERFORMANCE OF CROSS-FLOW AIR HEATED EVAPORATOR (ONE OR TWO
C**** SLABS ASSEMBLIES) WITH UP TO 130 FINNED TUBES (PER SLAB)
C**** PLACED IN UP TO 5 DEPTH ROWS.
C**** THE MODEL ACCOUNTS FOR AIR AND REFRIGERANT DISTRIBUTION.
C
C**** INPUT DATA:
C         ACFM(IIO)       - AIR MASS FLOW RATE THROUGH SLAB   (LB/H)
C         APIN            - AIR INLET PRESSURE   (PSIA)
C         ARARIO(IIO,I)   - RATIO OF AIR MASS FLOW RATE FOR A GIVEN
C                           TUBE TO THE TOTAL MASS FLOW RATE FOR
C                           A GIVEN SLAB   (-)
C         ARHIN           - AIR INLET RELATIVE HUMIDITY   (-)
C         ATIN            - AIR INLET TEMPERATURE   (F)
C         DI(IIO)         - INNER DIAMETER OF TUBES   (FT)
C         DO(IIO)         - OUTER DIAMETER OF TUBES   (FT)
C         DPCH(IIO)       - TUBE DEPTH PITCH   (FT)
C         DT(IIO)         - FIN TIP DIAMETER   (FT)
C         FLOW(IIO,M)     - FRACTION OF COIL TOTAL REFRIG. MASS FLOW PASSING
C                           THROUGH TUBE M   (-)
C         FMK(IIO)        - FIN MATERIAL THERMAL CONDUCTIVITY   (BTU/FT*H*F)
C         FPCH(IIO)       - FIN PITCH   (FT)
C         FTK(IIO)        - FIN THICKNESS   (FT)
C         GET(IIO,I,1)    - FRACTION OF THE AIR FLOW RATE OF TUBE
C                           IGET(IIO,I,1) PASSING THROUGH TUBE  'I'
C         GET(IIO,I,2)    - FRACTION OF THE AIR FLOW RATE OF TUBE
C                           IGET(IIO,I,2) PASSING THROUGH TUBE  'I'
C                           IF IGET(IIO,I,2)=0, GET(IIO,I,2) IS SET TO 0.
C         H1              - REFRIG. ENTHALPY AT EVAPORATOR INLET   (BTU/LB)
C         IDEPTH(IIO,M)   - DEPTH ROW OF A TUBE M
C         IFROM(IIO,M)    - NUMBER OF TUBE TUBE M RECEIVES REFRIG. FROM
C                           WHEN COIL WORKS AS EVAPORATOR   (-)
C         IGET(IIO,I,1)   - NUMBER OF A TUBE FROM WHICH TUBE  'I'  IS
C                           GETTING AIR   (-)
C         IGET(IIO,I,2)   - NUMBER OF A SECOND TUBE FROM WHICH TUBE 'I'
C                           IS GETTING AIR.  IF THE SECOND TUBE DOES NOT
C                           EXISTS, IGET(IIO,I,2)=0.   (-)
C         IIO             - SLAB INDICATOR FOR THE INPUTED DATA
C                         = 1 OR 2
C         II1             = 1 OR 2, SPECIFIES THE SLAB TO BE CONSIDERED
C                           IF KSLABS=1. IF KSLABS=2, ANY INTEGER VALUE
C                           MAY INPUTED FOR II1  (-)
C         KSLABS          - NUMBER OF EVAPORATOR SLABS TO BE SIMULATED (-)
C         NDEP(IIO)       - NUMBER OF TUBE ROW DEPTHS   (-)
C         NROW(IIO)       - MAX. NUMBER OF TUBES PER DEPTH ROW IN SLAB  (-)
C         NTPS            - NUMBER OF TUBES IN THE SLAB   (-)
C         NTUB(IIO,I)     - NUMBER OF TUBES IN ROW I OF THE SLAB   (-)
C         P1              - REFRIG. PRESSURE AT EVAPORATOR INLET   (PSIA)
C         RMASS           - REFRIG. MASS FLOW RATE THROUGH BOTH SLABS
C                           (IF NSLABS=2) OR THE SLAB (IF NSLABS=1)
C                           (LB/H)
C         TPCH(IIO)       - TUBE ROW PITCH   (FT)
C         TMK(IIO)        - TUBE MATERIAL THERMAL CONDUCTIVITY   (BTU/FT*H*F)
C         T1              - REFRIG. TEMPERATURE AT EVAPORATOR INLET   (F)
C         WIDTH(IIO)      - COIL WIDTH   (FT)
C
C**** OUTPUT DATA:
C         HR(IIO,I)       - REFRIG. ENTHALPY AT THE OUTLET OF TUBE I
C                           (BTU/LBM)
C         H2              - REFRIG. ENTHALPY AT EVAPORATOR OUTLET  (BTU/LB)
C         PRM(IIO,1,I)    - REFRIG. PRESSURE AT I TUBE INLET    (PSIA)
C         PRM(IIO,2,I)    - REFRIG. PRESSURE AT I TUBE OUTLET   (PSIA)
C         P2              - REFRIGERANT PRESSURE AT EVAPORATOR OUTLET  (PSIA)
C         S2              - REFRIG. ENTROPY AT EVAPORATOR OUTLET  (BTU/LB*F)
C         T2              - REFRIG. TEMPERATURE AT EVAPORATOR OUTLET  (F)
C         X2              - REFRIG. QUALITY AT EVAPORATOR OUTLET   (-)
```

```
C                          SAT. TEMP. OF I TUBE OUTLET    (FT**3/LB)
C
C**** SUBPROGRAMS CALLED BY EVPHX2:
C        AIRHT3,AIRPR,CPCV,DYNADP,FINCON,EVGUNG,EVPDP,FINEF2,ITRPR,
C        OVLWET,SATPR,SATT,SPHDP1,SPHTC,UPWAO,VPVHS,VPSV,WATPR
C
      DOUBLE PRECISION DPRES
      COMMON/PRINTD/IDIA,IDIAG0,NNN,MMM
      COMMON/PRINT/IPR
      COMMON/CONST/TC,PC,VC,TFR,AJ,EEP
      COMMON/HPHX/NSLABS,NDEP(2),NROW(2),DI(2),DO(2),DT(2),TPCH(2),
     & DPCH(2),WIDTH(2),FPCH(2),FTK(2),FMK(2),TMK(2),CFM1,BSIDE(2),
     & NTUB(2,5),IFROM(2,130),NTPS(2),BSPACE(2),
     & ACFM(2),IFIN(2),ISUR(2),SFLOW(2)
      COMMON/MERG/MERGE(2,20,2),IMER(2),IOUT(2,20),NOUT(2),
     & IDEPTH(2,130),FLOW(2,130),KFEED(2,130,3),KSTART(2,130),KST(2)
      COMMON/MASS/PRM(2,2,130)
      COMMON/AIRD/IGET(2,130,2),GET(2,130,2),ARATIO(2,130)
      COMMON/OUTPR/HR(2,130)
      DIMENSION TAIR(2,2,6),AIRN(5),H2IAIR(25,2),XRM(2,130),
     & OMEGA(2,2,6),DTHFG(2,130),IMC(20),IEND(10),MY(130),VGM(130),
     &TRM(2,130)
      DIMENSION TPIP(2,5),TWAT(2,5),HICE(2,5),HFGWT(2,5),TKICE(2,5),
     1 KFEED0(10)
      DIMENSION TAIRE(2,130,2),WAIRE(2,130,2),FSTORE(2,130),
     1 HICEX(2,5,25),HFGWTX(2,5,25),TKICEX(2,5,25),FLOW14(2,130),
     2 DPM(25),WAIREX(2,130),TAEX(2,130)
      SAVE TG1P,TAIRE,DTHFG,WAIRE,TAIR,OMEGA,HICE,HFGWT,TKICE,TWAT,
     1 TPIP
      DATA OMEGA/24*0./,PI/3.1415927/
C
      IF(IDIA.NE.0) WRITE(IDIA,900)T1,P1,H1,ATIN,ARHIN,RMASS
C
C**** ESTABLISH PARAMETERS FOR IIO LOOPS
      IF(KSLABS.EQ.1)THEN
         II2=II1
         KSL=II1
      ELSE
         II2=1
         KSL=2
      END IF
C**** CALCULATE INLET PARAMETERS FOR REFRIGERANT AND AIR
      TG1=SATT(P1)
      TG1P=TG1
      CALL VPVHS(1,TG1,P1,VG1,HG1,SG1,HF1)
      X1=(H1-HF1)/(HG1-HF1)
      CALL AIRPR(1,ATIN,APIN,ARHIN,WAIRIN,CPAIR,RAIR,
     &AMAIR,AKAIR)
      VAIR=RAIR*(TFR+ATIN)/144./APIN
C**** PREPARE OTHER VARIABLES
      IDP=0
      HD=5000.
      H2IAIR(1,2)=999.
      DO 1 IIO=II2,KSL
      DO 1 I=1,NTPS(IIO)
    1 FSTORE(IIO,I)=0.
      DO 12 IIO=II2,KSL
      IF(NNN.EQ.1. AND .MMM.EQ.1)THEN
         AAMAS=ACFM(IIO)*60./VAIR
         DTAIR=0.85*RMASS*(HG1-H1)/(CPAIR*AAMAS*NDEP(IIO))
         DO 6 I=1,NTPS(IIO)
         DTHFG(IIO,I)=0.
         ICT=IDEPTH(IIO,I)
         TAIRE(IIO,I,1)=ATIN-FLOAT(ICT)*DTAIR
    6    WAIRE(IIO,I,1)=WAIRIN
         TAIR(IIO,1,1)=ATIN
         OMEGA(IIO,1,1)=WAIRIN
         OMEGA(IIO,2,1)=WAIRIN
         DO 8 I=1,NDEP(IIO)
         J=I+1
         OMEGA(IIO,1,J)=OMEGA(IIO,1,I)
         HICE(IIO,I)=1.E+30
         HFGWT(IIO,I)=0.
```

```
      TKICE(IIO,I)=0.
      TWAT(IIO,I)=0.
      TPIP(IIO,I)=0.
 8    TAIR(IIO,1,J)=TAIR(IIO,1,I)-DTAIR
    ELSE
      DTAIR=TG1-TG1P
      DO 10 I=NTUB(IIO,1)+1,NTPS(IIO)
10    TAIRE(IIO,I,1)=TAIRE(IIO,I,1)+DTAIR
    END IF
 12 CONTINUE
C*****************************************************
C**** START MAIN LOOP
C*****************************************************
    DO 150 IAIR=1,25
    IF(IDIA.NE.0) WRITE(IDIAG0,*)'IAIR= ',IAIR
    DO 110 IIO=II2,KSL
    ILN=0
    IF(ISUR(IIO).EQ.1)THEN
      HTCF1=1.00
      HTCF2=1.00
      DPF1=1.00
      DPF2=1.00
    ELSE
      HTCF1=2.00
      HTCF2=1.45
      DPF1=0.5*HTCF1
      DPF2=0.5*HTCF2
    END IF
    HYDD=HYDDIA(IIO)
    IIFIN=IFIN(IIO)
    NNROW=NROW(IIO)
    DDI=DI(IIO)
    DDO=DO(IIO)
    DDT=DT(IIO)
    TTPCH=TPCH(IIO)
    DDPCH=DPCH(IIO)
    WWIDTH=WIDTH(IIO)
    FFPCH=FPCH(IIO)
    FFTK=FTK(IIO)
    FFMK=FMK(IIO)
    TTMK=TMK(IIO)
    API=PI*DDI*WWIDTH
    APO=PI*DDO*WWIDTH
    APM=0.5*(API+APO)
    APO=APO*(FFPCH-FFTK)/FFPCH
    AF=1.570796*(DDT+DDO)*(DDT-DDO)
    AF=AF*WWIDTH/FFPCH
    AO=APO+AF
    AFLOW=WWIDTH*TTPCH*NNROW
    AFLOW=AFLOW*(TTPCH-DDO)/TTPCH
    AFLOW=AFLOW/FFPCH
    AFLOW=39.75*AFLOW
    WFLW=AO/TTPCH
    HP=2.*TTMK/(DDO-DDI)
    HPF=FINCON(DDO,FFPCH,FFTK)
    AAMAS=ACFM(IIO)*60./VAIR
    AFIN=3.14159*(DDT-DDO)*(DDT+DDO)/4.
    SEGFIN=DDT**3/24.-DDT*DDT*DDO/16.+DDO**3/48.
    SEGFIN=SEGFIN*2./(AFIN*(DDT-DDO))
    DO 4 I=1,NTPS(IIO)
    VGM(I)=0.0
  4 MY(I)=0
    DO 13 I=1,IMER(IIO)
    J=MERGE(IIO,I,1)
 13 MY(J)=1
    DO 15 I=1,5
 15 AIRN(I)=0.
C**** ASSIGN REFRIGERANT AND AIR PARAMETERS AT THE INLET
    DO 110 NUMB=1,KST(IIO)
    I=KSTART(IIO,NUMB)
    VGI=VG1
    TRI=T1
    PRI=P1
```

```
      HRI=H1
      XRI=X1
      GOTO 32
C**** CALCULATE PERFORMANCE OF NEXT TUBE
   18 CONTINUE
      TRI=TRM(IIO,JJ)
      PRI=PRM(IIO,2,JJ)
      HRI=HR(IIO,JJ)
      XRI=XRM(IIO,JJ)
      VGI=VGM(JJ)
   32 PRM(IIO,1,I)=PRI
      HRII=HRI
      XRII=XRI
      TRII=TRI
      TRE=TRI
      PRE=PRI
      HRE=HRI
      XRE=XRI
      RMS=RMASS*FLOW(IIO,I)
      TGI=SATT(PRI)
      CALL VPVHS(1,TGI,PRI,VGI,HGI,SI,HFI)
      VVAP=VGI
      VLIQ=SATVF(TGI)
      HFG=HGI-HFI
      ICT=IDEPTH(IIO,I)
      IF(ICT.EQ.1)THEN
        AMS=ARATIO(IIO,I)*AAMAS
        OMEGI=WAIRIN
        TAI=ATIN
      ELSE
        J1=IGET(IIO,I,1)
        J2=IGET(IIO,I,2)
        IF(J2.EQ.0)THEN
          AMS=AAMAS*ARATIO(IIO,I)
          TAI=TAIRE(IIO,J1,1)
          OMEGI=WAIRE(IIO,J1,1)
        ELSE
          AW1=AAMAS*ARATIO(IIO,J1)*GET(IIO,I,1)
          AW2=AAMAS*ARATIO(IIO,J2)*GET(IIO,I,2)
          TA1=TAIRE(IIO,J1,1)
          TA2=TAIRE(IIO,J2,1)
          W1=WAIRE(IIO,J1,1)
          W2=WAIRE(IIO,J2,1)
          CALL MIXAIR(14.7,AW1,TA1,W1,AW2,TA2,W2,AMS,TAI,OMEGI,WMASS)
        END IF
      END IF
      TAE=TAIRE(IIO,I,1)
      OMEGE=WAIRE(IIO,I,1)
      OMEGE=AMIN1(OMEGE,OMEGI)
      OMEAVE=0.5*(OMEGI+OMEGE)
      TAAV=0.5*(TAI+TAE)
      CALL AIRPR(2,TAAV,APIN,RHA,OMEAVE,CPA,RA,AMA,AKA)
      FFFTK=FFTK+2.*TKICE(IIO,ICT)
      HCOO=AIRHT3(ICT,DDO,TTPCH,DDPCH,FFPCH,FFFTK,WWIDTH,HYDD,NDEP(IIO),
     &            IIFIN,AMS,AMA,CPA,AKA)
      HCO=HCOO*(1.+HFGWT(IIO,ICT))
      FFEE=FINEF2(TTPCH,DDPCH,DDO,FFTK,FFMK,HCO)
      UD1=HCO*(1.-AF*(1.-FFEE)/AO)
      UD2=HICE(IIO,ICT)
      ANNUL=0.
      XDRY=0.
CX     IF(NNN.EQ.2.OR.MMM.EQ.3)THEN
CX    WRITE(IPR,879)I,TRI,TAI,TAE,PRI,XRI,HRI,RMS,AMS,
CX   1 DTHFG(IIO,JJ),Q,CPA,TAAV,HCO
CX879 FORMAT(1I3,4F7.2,1F6.3,2F8.3,1F7.2,2F8.2,1F8.1,3F8.3)
CX    END IF
      DP1=0.
      DP2=0.
      AL1=0.
      IF(TAI.LT.TRI)THEN
         TSATT=SATT(PRI)
         IF(XRI.LT.1.)TAI=TRI+0.05
      END IF
```

```
C       IF(I.EQ.15)THEN
C          WRITE(*,*)'I=',I
C          WRITE(*,*)'PRI,TRI,XRI,HRI,TGI,HGI='
C          WRITE(*,*)PRI,TRI,XRI,HRI,TGI,HGI
C       END IF
      IF(XRI.EQ.1.)GOTO 64
      IF(XRI.GE.0.85)GOTO 54
C**** TUBE SELECTION FOR COMPUTING DONE
C**** PERFORM HEAT TRANSFER & REFRIG. PRESSURE DROP CALCULATIONS
C
C**** CASE 1
C**** INLET QUALITY LESS THAN 0.85
      ANNUL=1.0
      HREE=-999
      DO 50 IHI=1,7
      HI=HTCF2*EVGUNG(RMS,XRI,XRE,TRI,HFG,VLIQ,VVAP,DDI,WWIDTH)
      CALL OVLWET(AO,API,APM,APO,HI,HD,HP,HPF,UD2,UD1,UAO,UPO,UWO)
      DPRES=UAO/(CPA*AMS)
      Q=CPA*AMS*(TAI-TRI)*(1.-DEXP(-DPRES))
      HRE=HRI+Q/RMS
      IF(ABS(HRE-HREE).LT.0.001)GOTO 52
      HREE=HRE
   50 XRE=(HRE-HFI)/HFG
      WRITE(IPR,*)'50 DID NOT CONVERGE. IAIR,I,TAI,AMS=',IAIR,I,TAI,AMS
      WRITE(IPR,*)'RMS,TRI,HRI,XRI=',RMS,TRI,HRI,XRI
      WRITE(IPR,*)'HRE,XRE=',HRE,XRE
   52 IF(XRE.LE.0.85)GOTO 70
C
C**** OUTLET QUALITY ABOVE 0.85 IN A TUBE OF INLET QUALITY BELOW 0.85
C**** FIND FRACTION OF A TUBE WITH QUALITY UP TO 0.85
      X85=0.85
      HI=HTCF2*EVGUNG(RMS,XRI,X85,TRI,HFG,VLIQ,VVAP,DDI,WWIDTH)
      CALL OVLWET(AO,API,APM,APO,HI,HD,HP,HPF,UD2,UD1,UAO,UPO,UWO)
      DPRES=UAO/(CPA*AMS)
      Q=CPA*AMS*(TAI-TRI)*(1.-DEXP(-DPRES))
      HRE=HRI+Q/RMS
      H85=HFI+0.85*HFG
      ANNUL=(H85-HRI)*RMS/Q
      HRI=H85
      XRI=0.85
C
C**** CASE 2
C**** INLET QUALITY AT OR ABOVE 0.85
C**** CALC. INSIDE TUBE H.T.C AT QUALITIES 0.85 AND 1.0
   54 XA=0.85
      XB=0.86
      HCHH=-999.
      DO 58 IHI=1,10
      HIA=HTCF2*EVGUNG(RMS,XA,XB,TGI,HFG,VLIQ,VVAP,DDI,WWIDTH)
      CALL OVLWET(AO,API,APM,APO,HIA,HD,HP,HPF,UD2,UD1,UAO,UPO,UWO)
      DPRES=UAO/(CPA*AMS)
      Q=CPA*AMS*(TAI-TRI)*(1.-DEXP(-DPRES))
      HCH=Q/RMS
      IF(ABS(HCH-HCHH).LT.0.001)GOTO 59
      HCHH=HCH
      XD=HCH/HFG
      XA=0.85-XD/2.
   58 XB=0.85+XD/2.
      WRITE(IPR,*)'58 DID NOT CONVERGE. IAIR,I,XD=',IAIR,I,XD
   59 CONTINUE
      VISV=SATPR(2,TRI)
      CONV=SATPR(4,TRI)
      CALL CPCV(TRI,PRI,CPR,CVR,GAR,SOR)
      HIB=HTCF1*SPHTC(CPR,VISV,CONV,RMS,DDI)
C
C**** CALC. HEAT TRANSFER FOR A TUBE WITH QUALITY RANGE 0.85 - 1.00
      XRAV=XRI
      HREE=-999.0
      DO 60 IHI=1,12
      HI=HIA-(HIA-HIB)*(XRAV-0.85)/0.15
      CALL OVLWET(AO,API,APM,APO,HI,HD,HP,HPF,UD2,UD1,UAO,UPO,UWO)
      DPRES=UAO/(CPA*AMS)
      Q=(1.-ANNUL)*CPA*AMS*(TAI-TRI)*(1.-DEXP(-DPRES))
```

```
      HRE=HRI+Q/RMS
      XRE=(HRE-HFI)/HFG
      IF(XRE.GT.1.0)THEN
         XRE=1.0
         XDRY=(1.-ANNUL)*(HGI-HRI)/(HRE-HRI)
      ELSE
         XDRY=1.-ANNUL
      END IF
      IF(ABS(HRE-HREE).LT.0.001)GOTO 63
      HREE=HRE
   60 XRAV=0.5*(XRI+XRE)
      WRITE(IPR,*)'60 DID NOT CONVERGE, IAIR,I XDRY=',IAIR,I,XDRY
      WRITE(IPR,*)'XRI,XRE,HRI,HRE=',XRI,XRE,HRI,HRE
   63 IF(XRE.LT.1.)GOTO 70
      HRI=HGI
C
C**** CASE 3
C**** INLET QUALITY EQUAL 1.; SATURATED OR SUPERHEATED VAPOR
   64 TGAV=SATT(PRI)
      VISVS=SATPR(2,TGAV)
      CONVS=SATPR(4,TGAV)
      HREE=-999.
      DO 66 IHI=1,7
      TRAV=0.5*(TRI+TRE)
      CALL CPCV(TRAV,PRI,CPR,CVR,GAR,SOR)
      TT=SQRT((TRAV+460.)/(TGAV+460.))
      VISV=TT*VISVS
      CONV=TT*CONVS
      HI=HTCF1*SPHTC(CPR,VISV,CONV,RMS,DDI)
      CALL OVLWET(AO,API,APM,APO,HI,HD,HP,HPF,UD2,UD1,UAO,UPOW,UWOW)
      DPRES=UAO/(CPA*AMS)
      DPRES=(1.-ANNUL-XDRY)*CPA*AMS*(1.-DEXP(-DPRES))/(CPR*RMS)
      Q=CPR*RMS*(TAI-TRI)*(1.-DEXP(-DPRES))
      HRE=HRI+Q/RMS
      TRE=TRI+(HRE-HRI)/CPR
      IF(ABS(HREE-HRE).LT.0.001)GOTO 68
   66 HREE=HRE
      WRITE(IPR,*)'52 DID NOT CONVERGE, IAIR,I,HRE=',IAIR,I,HRE
C
C**** HEAT TRANSFER CALCULATIONS FOR TUBE 'I' COMPLETED
C
C**** CALCULATE PRESSURE DROP
C**** CASE 3,    QUALITY EQUAL 1.
   68 VSP=VPSV(PRI,TRAV)
      AL1=(1.-ANNUL-XDRY)*WWIDTH+12.*DDI
      DP1=DPF1*SPHDP1(RMS,AL1,DDI,VSP,VISV)
      IF(XDRY.EQ.0.)GOTO 72
   70 AL2=WWIDTH-AL1+12.*DDI
      VLIQ=SATVF(TGI)
      VISL=SATPR(1,TGI)
      VISV=SATPR(2,TGI)
      DP2=DPF2*EVPDP2(RMS,TGI,XRII,XRE,HFG,VLIQ,VGI,VISL,VISV,AL2,DDI)
   72 DP3=DYNADP(PRI,HRII,PRE,HRE,RMS,DDI)
      PRE=PRI-DP1-DP2-DP3
C**** ENTHALPY & PRESSURE AT TUBE I INLET ARE KNOWN
C**** FIND TEMP., QUALITY & SPEC. VOLUME
      PRM(IIO,2,I)=PRE
      TGE=SATT(PRE)
      CALL VPVHS(1,TGE,PGE,VGE,HGE,SGE,HFE)
      VGM(I)=VGE
      IF(HRE.LT.HGE)THEN
        XRM(IIO,I)=(HRE-HFE)/(HGE-HFE)
        TRE=TGE
      ELSE
        PROP2=HRE
        CALL ITRPR2(TRE,2.,PRE,2,0.001,PROP2,VRE,HRE,S2)
        XRM(IIO,I)=1.0
      END IF
      TRM(IIO,I)=TRE
      XSUP=(1.-ANNUL-XDRY)
      HR(IIO,I)=HRE
C
      OMECH1=0.
```

```
      OMECH2=0.
C***  FIND AIR STATE PAST TUBE
C
C**** REFRIGERANT IN ANNUAL OR DISPERSED FLOW,   XSUP < 1.
      IF(XSUP.LT.0.9999)THEN
         IF(XSUP.GT.0.0001)THEN
            Q=(HGI-HRII)*RMS
         ELSE
            Q=(HRE-HRII)*RMS
         END IF
         TAE=TAI-Q/(CPA*AMS)
         TAEE=TAE+DTHFG(IIO,I)
         TTAIR=0.5*(TAI+TAEE)
         VELA=AAMAS*(460.+TTAIR)/(AFLOW*(FFPCH-FFTK-2.*TKICE(IIO,ICT)))
         TWATAI=TRI+Q/UWO
         IF(XSUP.GT.0.00001)THEN
            TWATAE=TWATAI
            TPIPAE=TRI+Q/UPO
         ELSE
            TWATAE=TRE+Q/UWO
            TPIPAE=(TRI+TRE)*0.5+Q/UPO
         END IF
         CALL AIRPR(1,TWATAI,APIN,1.,OMEGWI,CPW,RW,AMW,AKW)
         CALL AIRPR(1,TWATAE,APIN,1.,OMEGWE,CPW,RW,AMW,AKW)
         TWATA=0.5*(TWATAI+TWATAE)
         OMEGW=0.5*(OMEGWI+OMEGWE)
         IF(OMEGI.GT.OMEGW)THEN
            DPRES=HCOO*APO/(CPA*AMS)
            OMECHP=(OMEGI-OMEGW)*(1.-DEXP(-DPRES))
            TFM=TTAIR-FFEE*(TTAIR-TWATA)
            TEND=TWATA+(TFM-TWATA)/SEGFIN
            IF(TEND.GT.TAEE)TEND=TAEE
            CALL AIRPR(1,TEND,APIN,1.,OMEGIS,CPW,RW,AMW,AKW)
            DDTS=DDO+(DDT-DDO)*(OMEGI-OMEGW)/(OMEGIS-OMEGW)
            DDTS=AMIN1(DDTS,DDT)
            DDTS=AMAX1(DDTS,DDO)
            IF(DDTS.EQ.DDO)THEN
               OMECHF=0.
            ELSE
               AFIN=3.14159*(DDTS-DDO)*(DDTS+DDO)/4.
               AFS=AFIN*2.*WWIDTH/FFPCH
               SEG=DDTS**3/24.-DDTS*DDTS*DDO/16.+DDO**3/48.
               SEG=SEG*2./(AFIN*(DDTS-DDO))
               OMEGIS=AMIN1(OMEGIS,OMEGI)
               OMEGFM=OMEGW+(OMEGIS-OMEGW)*SEG
               DPRES=HCOO*AFS/(CPA*AMS)
               OMECHF=(OMEGI-OMEGFM)*(1.-DEXP(-DPRES))
            END IF
            OMECH1=(1.-XSUP)*(OMECHP+OMECHF)
         END IF
      END IF
C
C**** REFRIGERANT IS SUPERHEATED IN "XSUP" FRACTION OF A TUBE
      IF( XSUP.GT.0.0001)THEN
         IF(XSUP.LT.0.9999)THEN
            Q=(HRE-HGI)*RMS
         ELSE
            Q=(HRE-HRII)*RMS
         END IF
         TAE=TAI-Q/(CPA*AMS*XSUP)
         TAEE=TAE+DTHFG(IIO,I)
         IF(XSUP.LT.0.9999)THEN
            TWATAI=TGI+Q/UWOW
            TPIPAE=TPIPAE*(1.-XSUP)+((TGI+TRE)*.5+Q/UPOW)*XSUP
         ELSE
            TWATAI=TRI+Q/UWOW
            TPIPAE=(TRII+TRE)*.5+Q/UPOW
         END IF
         TWATAE=TRE+Q/UWOW
         TWATA=0.5*(TWATAI+TWATAE)
         CALL AIRPR(1,TWATAI,APIN,1.,OMEGWI,CPW,RW,AMW,AKW)
         CALL AIRPR(1,TWATAE,APIN,1.,OMEGWE,CPW,RW,AMW,AKW)
         IF(OMEGI.GT.OMEGWI)THEN
```

```
            XCON=(OMEGI-OMEGWI)/(OMEGWE-OMEGWI)
            XCON=AMAX1(XCON,0.)
            XCON=AMIN1(XCON,1.)
            OMECON=AMIN1(OMEGI,OMEGWE)
            OMEGW=0.5*(OMEGWI+OMECON)
            TWATA=TWATAI+0.5*XCON*(TWATAI-TWATAE)
            DPRES=HCOO*APO/(CPA*AMS)
            OMECHP=(OMEGI-OMEGW)*(1.-DEXP(-DPRES))
            TFM=TTAIR-FFEE*(TTAIR-TWATA)
            TEND=TWATA+(TFM-TWATA)/SEGFIN
            IF(TEND.GT.TAEE)TEND=TAEE
            CALL AIRPR(1,TEND,APIN,1.,OMEGIS,CPW,RW,AMW,AKW)
            DDTS=DDO+(DDT-DDO)*(OMEGI-OMEGW)/(OMEGIS-OMEGW)
            DDTS=AMIN1(DDTS,DDT)
            DDTS=AMAX1(DDTS,DDO)
            IF(DDTS.EQ.DDO)THEN
               OMECHF=0.
            ELSE
               AFIN=3.14159*(DDTS-DDO)*(DDTS+DDO)/4.
               AFS=AFIN*2.*WWIDTH/FFPCH
               SEG=DDTS**3/24.-DDTS*DDTS*DDO/16.+DDO**3/48.
               SEG=SEG*2./(AFIN*(DDTS-DDO))
               OMEGIS=AMIN1(OMEGIS,OMEGI)
               OMEGFM=OMEGW+(OMEGIS-OMEGW)*SEG
               DPRES=HCOO*AFS/(CPA*AMS)
               OMECHF=(OMEGI-OMEGFM)*(1.-DEXP(-DPRES))
            END IF
            OMECH2=XCON*XSUP*(OMECHP+OMECHF)
         END IF
      END IF
C
   92 OMEGE=OMEGI-(OMECH1+OMECH2)
      TWATA=TWATAE
      TPIPA=TPIPAE
      CALL WATPR(TWATA,TPIPA,VELA,OMEGI,WATRO,WATK,
     &WATM,WATHFG,WATCP)
      IF(IAIR.LT.6)THEN
         DTHFG(IIO,I)=WATHFG*(OMEGI-OMEGE)/CPA
      ELSE
         DTHFG(IIO,I)=0.5*(DTHFG(IIO,I)+WATHFG*(OMEGI-OMEGE)/CPA)
      END IF
      TAE=TAI-(HRE-HRII)*RMS/(CPA*AMS)
      TAE=TAE+DTHFG(IIO,I)
C     TAE=AMIN1(TAE,TAI-0.1)
      IF(AIRN(ICT).EQ.0)GOTO 96
      AA1=1./(AIRN(ICT)+1.)
      AA2=AA1*AIRN(ICT)
      TAIR(IIO,2,ICT+1)=AA1*TAE+AA2*TAIR(IIO,2,ICT+1)
      OMEGA(IIO,2,ICT+1)=AA1*OMEGE+AA2*OMEGA(IIO,2,ICT+1)
      TWAT(IIO,ICT)=AA1*TWATA+AA2*TWAT(IIO,ICT)
      TPIP(IIO,ICT)=AA1*TPIPA+AA2*TPIP(IIO,ICT)
      AIRN(ICT)=AIRN(ICT)+1.
      GOTO 100
   96 TAIR(IIO,2,ICT+1)=TAE
      OMEGA(IIO,2,ICT+1)=OMEGE
      TWAT(IIO,ICT)=TWATA
      TPIP(IIO,ICT)=TPIPA
      AIRN(ICT)=1.
  100 TAIRE(IIO,I,2)=TAE
      WAIRE(IIO,I,2)=OMEGE
C**** SELECT A NEW TUBE FOR CALCULATION
      IF(MY(I).EQ.1)THEN
        DO 102 N=2,3
        NN=KFEED(IIO,I,N)
        IF(NN.EQ.0)GOTO 104
        ILN=ILN+1
  102   KFEED0(ILN)=NN
      END IF
  104 JJ=I
      I=KFEED(IIO,JJ,1)
      IF(I.NE.-1)GOTO 18
      IF(ILN.NE.0)THEN
        I=KFEED0(ILN)
```

```
          ILN=ILN-1
          JJ=IFROM(IIO,I)
          GOTO 18
        END IF
  110   CONTINUE
C****   ALL TUBES COMPUTED.
C****   CALCULATE REFRIGERANT STATE IN THE OUTLET MANIFOLD  (H2 & P2).
        H2=0.
        P2=0.0
        SUMFLO=0.
        DO 112 IIO=II2,KSL
        DO 112 N=1,NOUT(IIO)
        I=IOUT(IIO,N)
  112   SUMFLO=SUMFLO+FLOW(IIO,I)
        DO 114 IIO=II2,KSL
        DO 114 N=1,NOUT(IIO)
        I=IOUT(IIO,N)
        H2=FLOW(IIO,I)*HR(IIO,I)/SUMFLO+H2
        P2=FLOW(IIO,I)*PRM(IIO,2,I)/SUMFLO+P2
CX      XOUT=XRM(IIO,I)
CX      HOUT=HR(IIO,I)
CX      POUT=PRM(IIO,2,I)
CX      TOUT=TRM(IIO,I)
CX      TSAT=SATT(POUT)
CX      TSUP=TOUT-TSAT
CX      WRITE(IPR,864)I,XOUT,POUT,HOUT,TOUT,TSAT,TSUP
CX864   FORMAT(1I3,1F6.3,5F9.3)
  114   CONTINUE
        H2IAIR(IAIR,1)=H2
        DPMAX=0.0
        DO 118 IIO=II2,KSL
        DO 118 N=1,NOUT(IIO)
        I=IOUT(IIO,N)
        DP=ABS(P2-PRM(IIO,2,I))
        IF(IDIA.NE.0)WRITE(IDIA,786)I,IIO,DP,FLOW(IIO,I)
  786   FORMAT(' I,IIO,DP,FLOW(IIO,I)=',2I4,2F7.3)
  118   DPMAX=AMAX1(DPMAX,DP)
        DPM(IAIR)=DPMAX
C****   CHECK IF CONVERGENCE WAS OBTAINED.
        IF(IAIR.EQ.1)H2S=0.0
        H2PH2=H2S-H2
        IF(IDIA.NE.0) WRITE(IDIA,793)H2,H2PH2,P2,DPMAX
  793   FORMAT(' H2,H2PH2,P2,DPMAX =',2F13.7,2F10.6)
        H2S=H2
        H2IAIR(IAIR,2)=H2PH2
C1
        IF(IAIR.GT.4)THEN
          IF(IAIR.LT.15)THEN
             IF(DPM(IAIR-1).GT.0.1.OR.DPMAX.GT.0.05)GOTO 1182
          END IF
          IF(ABS(H2IAIR(IAIR-1,2)).LT.0.3.AND.ABS(H2PH2).LT.0.15)GOTO 152
        END IF
C
C*********************************************************************
C****   CONVERGENCE WAS NOT OBTAINED.
C****   PREPARE REFRIGERANT DISTRIBUTION FOR THE NEXT LOOP.
 1182   IF(DPMAX.LT.0.05.AND.IAIR.GT.4)THEN
           IDP=IDP+1
           DO 119 IIO=II2,KSL
           DO 119 I=1,NTPS(IIO)
  119      FSTORE(IIO,I)=(FSTORE(IIO,I)*REAL(IDP-1)+FLOW(IIO,I))/REAL(IDP)
        END IF
        IIO=II1
        IF(IAIR.LT.15)CALL BALFL1(KSLABS,IIO)
        DO 1210 IIO=II2,KSL
        IF(IAIR.EQ.14)THEN
           DO 1992 I=1,NTPS(IIO)
 1992      FLOW14(IIO,I)=FLOW(IIO,I)
        END IF
        IF(IAIR.EQ.15)THEN
           IF(IDIA.NE.0)WRITE(IDIA,*)'IDP=',IDP
           IF(IDP.GT.2)THEN
              DO 120 I=1,NTPS(IIO)
```

```
 120        FLOW(IIO,I)=FSTORE(IIO,I)
          ELSE
            DO 1205 I=1,NTPS(IIO)
 1205       FLOW(IIO,I)=0.5*(FLOW(IIO,I)+FLOW14(IIO,I))
          END IF
        END IF
 1210 CONTINUE
C**** PREPARE AIR SIDE DATA FOR NEW LOOP.
C**** PREPARE AIR TEMPERATURE AND HUMIDITY RATIO FOR EACH TUBE.
      DO 1236 IIO=II2,KSL
      IF(IAIR.LT.15)THEN
         DO 121 I=1,NTPS(IIO)
         TAIRE(IIO,I,1)=TAIRE(IIO,I,2)
 121     WAIRE(IIO,I,1)=WAIRE(IIO,I,2)
         IF(IAIR.EQ.12)THEN
            DO 122 I=1,NTPS(IIO)
            TAEX(IIO,I)=0.25*TAIRE(IIO,I,1)
 122        WAIREX(IIO,I)=0.25*WAIRE(IIO,I,1)
         END IF
         IF(IAIR.GE.13)THEN
            DO 1222 I=1,NTPS(IIO)
            TAEX(IIO,I)=TAEX(IIO,I)+0.25*TAIRE(IIO,I,1)
 1222       WAIREX(IIO,I)=WAIREX(IIO,I)+0.25*WAIRE(IIO,I,1)
         END IF
      ELSE
         IF(IAIR.EQ.15)THEN
            DO 123 I=1,NTPS(IIO)
            TAIRE(IIO,I,1)=0.25*TAIRE(IIO,I,1)+TAEX(IIO,I)
 123        WAIRE(IIO,I,1)=0.25*WAIRE(IIO,I,1)+WAIREX(IIO,I)
         ELSE
            DO 1232 I=1,NTPS(IIO)
            TAIRE(IIO,I,1)=0.25*TAIRE(IIO,I,1)+0.75*TAIRE(IIO,I,2)
 1232       WAIRE(IIO,I,1)=0.25*WAIRE(IIO,I,1)+0.75*WAIRE(IIO,I,2)
         END IF
      END IF
 1236 CONTINUE
C**** PREPARE CONDESATE/FROST DATA; AVERAGE FOR EACH DEPTH ROW.
      DO 149 IIO=II2,KSL
      DO 124 I=1,NDEP(IIO)
      J=I+1
      TAIR(IIO,1,J)=TAIR(IIO,2,J)
      OMEGA(IIO,1,J)=OMEGA(IIO,2,J)
      OMEGA(IIO,2,J)=0.
 124  TAIR(IIO,2,J)=0.
      DO 140 I=1,NDEP(IIO)
      TWATER=TWAT(IIO,I)
      TTAIR=TAIR(IIO,1,I)
      WWAIR=OMEGA(IIO,1,I)
      CALL AIRPR(1,TWATER,APIN,1.,WWATER,CCPA,RRA,
     &AAMA,AAKA)
      TPIPE=TPIP(IIO,I)
      VELA=AAMAS*(460.+TTAIR)/AFLOW
      VELA=VELA/(FFPCH-FFTK-2.*TKICE(IIO,I))
      CALL WATPR(TWATER,TPIPE,VELA,WWAIR,WATRO,WATK,
     &WATM,WATHFG,WATCP)
      IF(OMEGA(IIO,1,I).GT.WWATER)GOTO 125
      HICE(IIO,I)=1.E+30
      HFGWT(IIO,I)=0.
      TKICE(IIO,I)=0.
      GOTO 140
 125  IF(OMEGA(IIO,1,I+1).LT.OMEGA(IIO,1,I))GOTO 126
      OMEGA(IIO,1,I+1)=OMEGA(IIO,1,I)
      HICE(IIO,I)=1.E+30
      HFGWT(IIO,I)=0.
      TKICE(IIO,I)=0.
      GOTO 140
 126  WMAS=AAMAS*(OMEGA(IIO,1,I)-OMEGA(IIO,1,I+1))
      HFGWT(IIO,I)=WATHFG*(OMEGA(IIO,1,I)-WWATER)
      CALL AIRPR(2,TTAIR,APIN,RRRH,WWAIR,CCPA,RRA,AAMA,AAKA)
      HFGWT(IIO,I)=HFGWT(IIO,I)/(CCPA*(TAIR(IIO,1,I)-TWATER))
      IF(TWATER.GT.32.)GOTO 128
      TKICE(IIO,I)=0.125*WMAS/(AO*NNROW*WATRO)
      TKMAX=0.5*(FFPCH-FFTK)
```

```
      IF(TKICE(IIO,I).GE.TKMAX)TKICE(IIO,I)=0.9*TKMAX
      GOTO 132
  128 WWW=WMAS/WFLW
      WWW=WATM*WWW/(WATRO*WATRO)
      TKICE(IIO,I)=1.449E-03*WWW**0.333
  132 HICE(IIO,I)=WATK/TKICE(IIO,I)
      IF(HICE(IIO,I).LT.0.)HICE(IIO,I)=0.
      IF(TKICE(IIO,I).LT.0.)TKICE(IIO,I)=0.
      IF(HFGWT(IIO,I).LT.0.)HFGWT(IIO,I)=0.
  140 CONTINUE
      DO 142 ICT=1,NDEP(IIO)
      HICEX(IIO,ICT,IAIR)=HICE(IIO,ICT)
      HFGWTX(IIO,ICT,IAIR)=HFGWT(IIO,ICT)
  142 TKICEX(IIO,ICT,IAIR)=TKICE(IIO,ICT)
      IF(IAIR.GT.9)THEN
        DO 144 ICT=1,NDEP(IIO)
        HICE(IIO,ICT)=0.
        HFGWT(IIO,ICT)=0.
  144   TKICE(IIO,ICT)=0.
        DO 146 ICT=1,NDEP(IIO)
        DO 146 N=1,4
        HICE(IIO,ICT)=HICE(IIO,ICT)+0.25*HICEX(IIO,ICT,IAIR-4+N)
        HFGWT(IIO,ICT)=HFGWT(IIO,ICT)+0.25*HFGWTX(IIO,ICT,IAIR-4+N)
  146   TKICE(IIO,ICT)=TKICE(IIO,ICT)+0.25*TKICEX(IIO,ICT,IAIR-4+N)
      END IF
  149 CONTINUE
  150 CONTINUE
C*****************************************************
C**** END OF MAIN LOOP
C*****************************************************
      IF(IAIR.GT.25)IAIR=25
      DO 529 J=1,IAIR
  529 IF(IDIA.NE.0)WRITE(IDIA,530)H2IAIR(J,1),H2IAIR(J,2)
  530 FORMAT(' H2,H2PH2=',2(1PE11.3))
      I=5
      DO 151 IEV=6,IAIR
  151 IF(ABS(H2IAIR(I,2)).GT.ABS(H2IAIR(IEV,2)))I=IEV
      H2PH2=0.5*H2IAIR(I,2)
      H2=H2IAIR(I,1)+H2PH2
      IF(IDIA.NE.0)WRITE(IDIA,901)H2PH2
  152 CONTINUE
      DTT=4.
      TOL=0.001
      CALL PHWET2(P2,H2,DTT,TOL,T2,V2,S2,X2,TG2)
      TSUP2=T2-TG2
      TSUP2=AMAX1(TSUP2,0.)
      QT=RMASS*(H2-H1)*SUMFLO
      CFMIND=0.
      QL=0.
      DO 160 IIO=II2,KSL
      CFMIND=CFMIND+ACFM(IIO)
  160 QL=ACFM(IIO)*60.*(OMEGA(IIO,1,1)-OMEGA(IIO,1,NDEP(IIO)+1))
     1 /(1.+OMEGA(IIO,1,1))  +   QL
      QL=QL*WATHFG/VAIR
      QS=QT-QL
      IF(IDIA.NE.0)WRITE(IDIA,880)QT,QS,QL
  880 FORMAT('  QT,QS,QL=',3(1PE11.3))
      IF(IDIA.NE.0)WRITE(IDIA,902)T1,P1,H1,X1,T2,P2,H2,X2,TSUP2
      CFMTON=CFMIND/(QT/12000.)
      IF(IDIA.NE.0)WRITE(IDIA,*)'CFMIND,CFM/TON=',CFMIND,CFMTON
CX    DO 310 IIO=II2,KSL
CX    DO 310 N=1,NOUT(IIO)
CX    K=IOUT(IIO,N)
CX    PDIF=P2-PRM(IIO,2,K)
CX310 WRITE(IPR,611)P2,K,PRM(IIO,2,K),PDIF
CX611 FORMAT(F8.3,I5,2X,F8.3,F6.3)
  900 FORMAT(/2X,'INPUT DATA TO EVPHX2:'//2X,'T',
     &10X,'P',10X,'H',10X,'TAIR',7X,'RH',9X,'RMASS'/6(1PE11.3))
  901 FORMAT(' EVPHX2 DOES NOT CONVERGE, MAX. ERROR=',F6.2,' BTU/LB')
  902 FORMAT(/2X,'EVAPORATOR ITERATION:'//2X,
     &'T',10X,'P',10X,'H',10X,'X',10X,'TSUP',/4(1PE11.3)/5(1PE11.3))
 999  RETURN
      END
```

```
      FUNCTION FINCON(DO,FPCH,FTK)
C
C**** PURPOSE: TO CALC. THE THERMAL CONDUCTANCE FOR FIN-TUBE CONTACT
C               MARCH 9, 1989
C
C**** INPUT: DO   - OUTSIDE DIAMETER OF TUBE  (FT)
C            FPCH - FIN PITCH  (FT)
C            FTK  - FIN THICKNESS  (FT)
C
C**** OUTPUT: FINCON - FIN-TUBE CONTACT THERMAL CONDUCTANCE
C                      (BTU/(H*FT}*F)
C
C**** REFERENCE: SHEFFIELD, J.W., WOOD, R.A. AND SAUER, H.J.,
C                EXPERIMENTAL INVESTICATION OF THERMAL CONDUCTANCE
C                OF FINNED TUBE CONTACTS, EXPERIMENTAL THERMAL AND FLUID
C                SCIENCE, 0894-1777, 1988.
C                (SEE ALSO ASHRAE TRANSACTION, PAPER NO. 3103, 1987)
C**** NOTE: THIS CORRELATION IS APPLICABLE TO MECHANICALLY EXPANDED COPPER
C           TUBES WITH WITH ALUMINIUM FINS, DIAMETERS 1/4 TO 5/8 INCH.
C           THE INDENTATION DIAMATER OF THE MICROHARDNESS TEST, D, AND TUBE
C           EXPANSION INTERFERENCE, I, ARE GIVEN TYPICAL VALUES
C           IN CALCULATIONS.
C           NOTE DIMENSIONS USED IN THE CORRELATION:
C                             - D (MICRO METERS)
C                             - ALL OTHER LENGTH DIMENSIONS IN INCHES
C
      REAL I
      DATA D/44.2/,I/0.0065/
C
      DOI=DO*12.
      FPI=1./(12.*FPCH)
      FTKI=12.*FTK
      FINCON=6.902+2.889*(I*FPI*D/DOI)**0.75*(FTKI*FPI)**1.25
      FINCON=EXP(FINCON)
      RETURN
      END
```

```
      FUNCTION FINEF2(TPCH,DPCH,DR,T,AK,H)
C
C**** PURPOSE:
C       CALCULATE FIN EFFICIENCY USING SCHMIDT'S METHOD
C       10-19-88
C
C**** REFERENCE: HEATING, VENTILATING, AND AIR CONDITIONING,
C                MCQUISTON,F.C. AND PARKER, J.D.,
C                J. WILEY & SONS, INC., 2-ND EDITION, P.478, 1982.
C**** INPUT DATA:
C       AK       - FIN MATERIAL THERMAL CONDUCTIVITY   (BTU/FT*H*F)
C       DPCH     - DISTANCE BETWEEN TUBES IN NEIGHBOURING DEPTH ROWS
C                  (FT)
C       DR       - FIN ROOT DIAMETER   (FT)
C       H        - FIN SURFACE HEAT TRNSFER COEFF.   (BTU/H*F*FT**2)
C       T        - FIN THICKNESS   (FT)
C       TPCH     - DISTANCE BETWEEN TUBES IN THE SAME DEPTH ROW  (FT)
C
C**** OUTPUT DATA:  FINEF2  - FIN EFFICIENCY  (-)
C
      REAL L,M
C
C**** CALC. RATIO = REQ/R
      X=0.5*TPCH
      Y=0.5*SQRT(X*X+DPCH*DPCH)
      L=AMAX1(X,Y)
      M=AMIN1(X,Y)
      R=0.5*DR
      PSI=M/R
      BETA=L/M
      RATIO=1.27*PSI*SQRT(BETA-0.3)
C**** CALC. FIN EFFICIENCY
      FI=(RATIO-1.)*(1.+0.35*ALOG(RATIO))
      M=SQRT(2.*H/(AK*T))
      X=M*R*FI
      FINEF2=TANH(X)/X
      RETURN
      END
```

```
      SUBROUTINE FRACT(NSPLIT,RN,F)
C
C**** PURPOSE:
C        TO CALCULATE REFRIGERANT DISTRIBUTION AT A SPLIT POINT
C        WITH ONE TUBE SPLITING INTO UP TO 20 TUBES.
C        10-14-88
C**** INPUT:
C        NSPLIT - NUMBER OF TUBES RECEIVING REFRIG. IN THE SPLIT POINT
C        RN(I)  - FLOW RESISTANCE (OR # OF TUBES) IN A GIVEN CIRCUIT
C**** OUTPUT:
C        F(I)   - FRACTION OF REFRIGERANT MASS FLOW RATE THROUGH A SPLIT
C                 FLOWING IN TUBE I
C
      DIMENSION RN(20),F(20),FL1(20)
C
      IF(NSPLIT.GT.20)THEN
         WRITE(*,*)' ERROR IN CALLING FRACT, NSPLIT = ',NSPLIT
         RETURN
      END IF
C
      DO 10 I=1,20
   10 F(I)=0.0
      R1=RN(1)
      DO 20 I=2,NSPLIT
      A=R1/RN(I)
      IF(A.EQ.1.)THEN
        FL1(I)=0.5
      ELSE
        FL1(I)=1./(1.+A**0.571)
      END IF
   20 F(1)=F(1)+(1.-FL1(I))/FL1(I)
      F(1)=1./(1.+F(1))
      DO 30 I=2,NSPLIT
   30 F(I)=F(1)*(1.-FL1(I))/FL1(I)
      RETURN
      END
```

```
      FUNCTION HYDDIA(IIO)
C
C**** PURPOSE: TO CALC. HYDRAULIC DIAMETER FOR SLAB # IIO
C               JULY 15, 1988
C
C**** DEFINITION:
C     HYDRAQULIC DIAMETER = 4*L*(CRIT. FLOW AREA)/(TOTAL WETTED AREA)
C     WHERE:   L = NDEP(IIO)*DPCH(IIO)
C
      COMMON/HPHX/NSLABS,NDEP(2),NROW(2),DI(2),DO(2),DT(2),TPCH(2),
     & DPCH(2),WIDTH(2),FPCH(2),FTK(2),FMK(2),TMK(2),CFM1,BSIDE(2),
     & NTUB(2,5),IFROM(2,130),NTPS(2),BSPACE(2),
     & ACFM(2),IFIN(2),ISUR(2),SFLOW(2)
      DATA PI/3.141592654/
C
      NDEPX=NDEP(IIO)
      DOX=DO(IIO)
      TPCHX=TPCH(IIO)
      DPCHX=DPCH(IIO)
      W=WIDTH(IIO)
      FTKX=FTK(IIO)
      NTPSX=NTPS(IIO)
C     CALC. MAX. AND MIN. # OF TUBES IN DEPTH ROWS
      NMAX=NTUB(IIO,1)
      NMIN=NMAX
      DO 10 I=2,5
      N=NTUB(IIO,I)
      NMAX=MAX0(NMAX,N)
   10 NMIN=MIN0(NMIN,N)
C     CALC. HEIGHT OF THE COIL
      HGT=REAL(NMAX-1)*TPCHX+0.75*TPCHX
      IF(NMAX.EQ.NMIN)HGT=HGT+0.5*TPCHX
C     FLOW AREA BLOCKED BY FINS
      FN=W/FPCH(IIO)
      ABFINS=FN*FTKX*HGT
C     FLOW AREA BLOCKED BY TUBES (ADJUSTED FOR AREA BLOCKED BY FINS)
      ABTUBS=DOX*(W-FN*FTKX)*REAL(NMAX)
C     MIN. FLOW AREA
      AC=W*HGT-ABFINS-ABTUBS
C     TOTAL HEAT TRANSFER AREA ON THE AIR SIDE, ATOTAL
      AFINS=REAL(NDEPX)*DPCHX*HGT-0.25*PI*DOX*DOX*REAL(NTPSX)
      AFINS=2.*AFINS*FN
      ATUBES=PI*DOX*(W-FN*FTKX)*REAL(NTPSX)
      ATOTAL=AFINS+ATUBES
C     HYDRAULIC DIAMETER
      HYDDIA=4*AC*NDEP(IIO)*DPCH(IIO)/ATOTAL
      RETURN
      END
```

```
      SUBROUTINE ITRPR2(T,DT,P,I,AER,PROP2,V,H,S)
C
C**** PURPOSE:
C       TO ITERATE REFRIGERANT VAPOR PROPERTIES FROM GIVEN PRESSURE
C       AND ONE OF THE FOLLOWING PROPERTIES:
C       SPECIFIC VOLUME, ENTHALPY, ENTROPY
C
C**** INPUT DATA:
C       AER    - CONVERGENCE PARAMETER NOT TO BE EXEEDED
C                (IN UNITS OF RESPECTIVE PROPERTY)
C       DT     - TEMPERATURE INITIAL ITERATION STEP    (F)
C       I      = 1    IF SPECIFIC VOLUME IS PROP2    (-)
C              = 2    IF ENTHALPY IS PROP2    (-)
C              = 3    IF ENTROPY IS PROP2    (-)
C       P      - REFRIG. VAPOR PRESSURE    (PSIA)
C       PROP2  - VALUE OF OTHER THEN PRESSURE KNOWN PROPERTY:
C                SPECIFIC VOLUME (I=1)    (FT**3/LBM)
C                ENTHALPY (I=2)    (BTU/LBM)
C                ENTROPY (I=3)    (BTU/LBM*F)
C       T      - APPROXIMATE REFRIG. VAPOR TEMPERATURE    (F)
C
C**** OUTPUT DATA:
C       H      - VAPOR ENTHALPY    (BTU/LBM)
C       T      - VAPOR TEMPERATURE    (F)
C       S      - VAPOR ENTROPY    (BTU/LBM*F)
C       V      - VAPOR SPEC. VOLUME    (FT**3/LBM)
C
C**** SUBPROGRAMS CALLED BY ITRPR2:
C       SATT,VPVHS
C
      COMMON/PRINT/IPR
      TG=SATT(P)
      DO 100 IT=1,40
      IF(T.GT.TG)THEN
         CALL VPVHS(2,T,P,V1,H1,S1,HF)
      ELSE
         T=TG
         CALL VPVHS(1,T,P,V1,H1,S1,HF)
      END IF
      IF(I.EQ.1)PROP22=V1
      IF(I.EQ.2)PROP22=H1
      IF(I.EQ.3)PROP22=S1
      DIFF=PROP22-PROP2
      IF(ABS(DIFF).LE.AER)GOTO 110
      IF(T.EQ.TG.AND.DIFF.GT.0.)THEN
         WRITE(IPR,600)DIFF
         GOTO 110
      END IF
      IF(IT.NE.1)GOTO 10
      T1=T
      DTT=SIGN(DT,DIFF)
      T=T-DTT
      DIFF1=DIFF
      GOTO 100
   10 DDT=(T-T1)/(DIFF-DIFF1)
      IF(ABS(DIFF1).LE.ABS(DIFF))GOTO 20
      DIFF1=DIFF
      T1=T
   20 T=T1-DDT*DIFF1
  100 CONTINUE
      WRITE(IPR,*)'ITRPR2 DID NOT CONVERGE, I,P,PROP2= ',I,P,PROP2
  110 V=V1
      H=H1
      S=S1
C
  600 FORMAT(' ERROR IN CALLING ITRPR2, DIFF=',1PE11.3)
C
      RETURN
      END
```

```
      SUBROUTINE MIXAIR(P,AW1,T1,W1,AW2,T2,W2,AW3,T3,W3,WMASS)
C
C**** PURPOSE:
C       TO CALC. PROPERTIES OF THE AIR STREAM RESULTED FROM
C       THE MIXING PROCESS OF TWO WET AIR STREAMS
C       MAR/4/1987
C
C**** INPUT:
C       AW1    - MASS FLOW RATE OF THE STREAM #1    (LB OF WET AIR/H)
C       AW2    - MASS FLOW RATE OF THE STREAM #2    (LB OF WET AIR/H)
C       P      - AIR TOTAL PRESSURE    (PSI)
C       T1     - TEMPERATURE OF THE STREAM #1    (F)
C       T2     - TEMPERATURE OF THE STREAM #2    (F)
C       W1     - HUMIDITY RATIO OF THE STREAM #1    (LB H20/LB DRY AIR)
C       W2     - HUMIDITY RATIO OF THE STREAM #2    (LB H20/LB DRY AIR)
C
C**** OUTPUT:
C       AW3    - MASS FLOW RATE OF THE MIXED STREAM    (LB OF WET AIR/H)
C       T3     - TEMPERATURE OF THE MIXED AIR STREAM    (F)
C       W3     - HUMIDITY RATIO OF THE MIXED STREAM    (LB H20/LB DRY AIR)
C       WMASS - CONDENSATION RATE AT MIXING    (LB H20/H)
C
C**** NOTE: APPLICATIONS RANGE OF THIS SUBROUTINE IS  32<T(F)<80.
C
      H(TF,W)=0.240*TF+W*(1061.0+0.444*TF)
      PSAT(Z)=EXP(0.17829*Z**3-1.6896*Z**2-5.0988*Z+13.4353)
C
      WMASS=0.0
      AD1=AW1/(1.+W1)
      AD2=AW2/(1.+W2)
      AD3=AD1+AD2
      H1=H(T1,W1)
      H2=H(T2,W2)
      HTOTAL=AD1*H1+AD2*H2
      H3=HTOTAL/AD3
      W3=(W1*AD1+W2*AD2)/AD3
      T3=(H3-1061.0*W3)/(0.240+W3*0.444)
      Z=1000.0/(T3+459.67)
      PSAT3=PSAT(Z)
      WSAT3=0.62198*PSAT3/(P-PSAT3)
      IF(W3.LT.(WSAT3+0.0001))     GOTO 60
C
C**** MIXED STREAM IS SATURATED, CONDENSATION DURING MIXING OCCURS
      T3=T3+1.0
      DO 50 N=1,10
      Z3=1000.0/(T3+459.67)
      PSAT3=PSAT(Z3)
      W3=0.62198*PSAT3/(P-PSAT3)
      H3=H(T3,W3)
      WMASS=AD1*W1+AD2*W2-AD3*W3
      HWATER=-32.01+1.002*T3
      DH=HTOTAL-AD3*H3-WMASS*HWATER
      FR=DH/HTOTAL
      IF(ABS(DH/HTOTAL).LT.0.00001)     GOTO 60
      IF(N.EQ.1)THEN
        DT=SIGN(0.2,DH)
      ELSE
        DT=-DH*(T31-T3)/(DH1-DH)
      END IF
      DH1=DH
      T31=T3
   50 T3=T3+DT
      WRITE(*,*)'MIXAIR, LOOP 50 DID NOT CONVERGE'
   60 AW3=AD3*(1.+W3)
      RETURN
      END
```

```
      SUBROUTINE OVLWET(AO,API,APM,APO,HI,HD,HP,HPF,HL,HO,
     1                  UAO,UPO,UWO)
C
C**** PURPOSE:
C        TO COMPUTE OVERALL HEAT TRANSFER COEFFICIENT
C        FOR A WET FINNED TUBE       (MAR 9, 1989)
C
C**** INPUT DATA:
C        AO     - TOTAL OUTSIDE SURFACE AREA    (FT**2)
C        API    - TUBE INSIDE SURFACE AREA    (FT**2)
C        APM    - SURFACE AREA BASED ON TUBE MEAN DIAMETER    (FT**2)
C        APO    - TUBE OUTSIDE AREA    (FT**2)
C        HD     - TUBE INSIDE SURFACE DEPOSIT HEAT TRANSFER COEFF.
C                 (BTU/H*F*FT**2)
C        HI     - INSIDE TUBE HEAT TRANSFER COEFFICIENT
C                 (BTU/H*F*FT**2)
C        HP     - TUBE WALL HEAT TRANSFER COEFFICIENT    (BTU/H*F*FT**2)
C        HPF    - THERMAL CONDUCTANCE OF THE PIPE-FIN CONTACT
C                 (BTU/H*F*FT**2)
C        HO     - AIR-SIDE HEAT TRANSFER COEFF. FOR WET FINNED TUBE
C                 (BTU/H*F*FT**2)
C
C**** OUTPUT DATA:
C        UAO    - OVERALL HEAT TRANSFER COEFF. FOR WET FINNED TUBE
C                 (BTU/H*F*FT**2)
C        UPO    - HEAT CONDUCTANCE FROM REFRIGERANT TO TUBE SURFACE
C                 (BTU/H*F)
C        UWO    - HEAT CONDUCTANCE FROM REFRIGERANT TO WATER (FROST)
C                 SURFACE    (BTU/H*F)
C
      U=AO/(API*HI)+AO/(API*HD)+AO/(APM*HP)+AO/(APO*HPF)
      UPO=AO/U
      U=U+1./HL
      UWO=AO/U
      U=U+1./HO
      UAO=AO/U
      RETURN
      END
```

```
      SUBROUTINE PHWET2(P,H,DT,TOL,T,V,S,X,TG)
C
C**** PURPOSE:
C       TO FIND REFRIGERANT PARAMETERS
C       FROM KNOWN PRESSURE & ENTHALPY.
C       11/5/86
C
C**** INPUT:
C       DT  - ITERATION STEP OF TEMPERATURE   (F)
C       H   - REFRIG. ENTHALPY    (BTU/LBM)
C       P   - REFRIG. PRESSURE    (PSIA)
C       TOL - CONVERGENCE TOLERANCE OF ENTHALPY   (BTU/LBM)
C
C**** OUTPUT:
C       H   - REFRIG.ENTHALPY    (BTU/LBM)
C       S   - REFRIG. ENTROPY    (BTU/LBM*F)
C       T   - REFRIG. TEMPERATURE    (F)
C       TG  - REFRIG. SAT. TEMPERATURE    (F)
C       V   - REFRIG. SPEC VOLUME    (FT**3/LBM)
C       X   - REFRIG. QUALITY    (-)
C
C**** SUBPROGRAMS CALLED BY PHWET2:
C       CPCV,ITRPR2,SATT,SATVF,VPVHS
C
      TG=SATT(P)
      CALL VPVHS(1,TG,P,VG,HG,SG,HF)
      IF(H.LT.HF)THEN
        TT=TG
        DO 10 K=1,5
        TAV=0.5*(TG+TT)
        T=TG-(HF-H)/SATPR(5,TAV)
        IF(ABS(T-TT).LT.0.05)GOTO 11
        TT=T
   10   CONTINUE
   11   V=SATVF(T)
        S=SG-(HF-H)/(0.5*(TG+T)+459.67)-(HG-HF)/(459.67+TG)
        X=0.0
      ELSE
        IF(H.LE.HG)THEN
            T=TG
            X=(H-HF)/(HG-HF)
            V=X*VG+(1.-X)*SATVF(TG)
            S=SG-(HG-H)/(460.+TG)
        ELSE
            CALL CPCV(TG,P,CP,CV,GA,SO)
            T=TG+(H-HG)/CP
            PROP=H
            CALL ITRPR2(T,DT,P,2,TOL,PROP,V,H,S)
            X=1.0001
        END IF
      END IF
      RETURN
      END
```

```
      SUBROUTINE RDATA3
C
C**** PURPOSE:
C        TO READ FROM FILE 8 AND PREPARE THE EVAPORATOR DATA.
C        JUNE 15, 1988
C**** NOTE:
C        AT THE EXIT OF RDATA3, ALL COMMON STATEMENT VALUES ARE DEFINED
C        WITH EXCEPTION TO /HPHX/ACFM(2) AND /PRINT/IPR
C
      COMMON/PRINT/IPR
      COMMON/RESTR/IEXP
      COMMON/HPHX/NSLABS,NDEP(2),NROW(2),DI(2),DO(2),DT(2),TPCH(2),
     & DPCH(2),WIDTH(2),FPCH(2),FTK(2),FMK(2),TMK(2),CFMTOT,BSIDE(2),
     & NTUB(2,5),IFROM(2,130),NTPS(2),BSPACE(2),
     & ACFM(2),IFIN(2),ISUR(2),SFLOW(2)
      COMMON/ATEST/X(2,18),VX(2,18),NTEST(2)
      DIMENSION ATITLE(20)
      OPEN (UNIT=8, FILE='DTEV', STATUS='OLD')
C
C**** INPUT EVAPORATOR COIL DATA
      READ(8,800)ATITLE
      WRITE(IPR,800)ATITLE
  800 FORMAT( 20A4)
      READ(8,*)NSLABS,IEXP,CFMTOT
C
C**** READ & REDUCE DATA FOR EACH SLAB SEPERATELY
      DO 100 N=1,NSLABS
      READ(8,800)ATITLE
      READ(8,*)BSIDE(N),BSPACE(N),WIDTH(N)
      READ(8,*)TPCH(N),DPCH(N),DI(N),DO(N),TMK(N),ISUR(N)
      READ(8,*)FPCH(N),FTK(N),FMK(N),IFIN(N)
      READ(8,*)(NTUB(N,I),I=1,5),SFLOW(N)
      DO 12 I=1,13
      K=10*I
      M=K-9
   12 READ(8,*)(IFROM(N,J),J=M,K)
C**** READ AIR DISTRIBUTION DATA
      READ(8,*)NTEST(N)
      READ(8,*)(X(N,I),I=1,8)
      READ(8,*)(X(N,I),I=9,16)
      READ(8,*)(VX(N,I),I=1,8)
      READ(8,*)(VX(N,I),I=9,16)
C**** ASSIGN AIR VELOCITY AT THE DUCT WALL (IF NOT SPECIFIED BY DTEV)
      IF(X(N,1).NE.0.)THEN
         DO 14 IT=1,NTEST(N)
         I=NTEST(N)+1-IT
         X(N,I+1)=X(N,I)
   14    VX(N,I+1)=VX(N,I)
         X(N,1)=0.0
         VX(N,1)=VX(N,2)
         NTEST(N)=NTEST(N)+1
      END IF
      IT=NTEST(N)
      IF(X(N,IT).LT.BSIDE(N))THEN
         NTEST(N)=NTEST(N)+1
         I=NTEST(N)
         X(N,I)=BSIDE(N)
         VX(N,I)=VX(N,I-1)
      END IF
C**** PREPARE DATA FOR CALCULATIONS
      BSIDE(N)=BSIDE(N)/12.
      BSPACE(N)=BSPACE(N)/12.
      WIDTH(N)=WIDTH(N)/12.
      TPCH(N)=TPCH(N)/12.
      DPCH(N)=DPCH(N)/12.
      DI(N)=DI(N)/12.
      DO(N)=DO(N)/12.
      FPCH(N)=FPCH(N)/12.
      FTK(N)=FTK(N)/12.
      DT(N)=SQRT(4.*TPCH(N)*DPCH(N)/3.14159)
      DO 20 I=1,NTEST(N)
   20 X(N,I)=X(N,I)/12.
```

```
C**** FIND # OF TUBES IN THE SLAB, NTPS(N)
C     FIND # OF DEPTH ROWS, NDEP(N)
      NTPS(N)=0
      DO 50 I=1,5
      IF(NTUB(N,I).NE.0)NDEP(N)=I
   50 NTPS(N)=NTPS(N)+NTUB(N,I)
C**** FIND MAX. NUMBER OF TUBES IN A DEPTH ROW, NROW(N)
      NMAX=0
      DO 60 ICT=1,NDEP(N)
      NX=NTUB(N,ICT)
   60 NMAX=MAX0(NX,NMAX)
      NROW(N)=NMAX
  100 CONTINUE
      REWIND 8
      CLOSE (UNIT=8, STATUS='KEEP')
      RETURN
      END
```

```
      FUNCTION SATP(TG)
C
C**** PURPOSE:
C       TO COMPUTE REFRIGERANT SATURATION PRESSURE
C       FOR GIVEN TEMPERATURE
C
C**** INPUT DATA:
C       TG   - REFRIG. TEMPERATURE   (F)
C       REFRIG. CONSTANTS - REFER TO THE MAIN PROGRAM.
C
C**** OUTPUT DATA:
C       SATP - REFRIG. SATURATION PRESSURE   (PSIA)
C
      COMMON/PRINT/IPR
      COMMON/TGPG/AG,BG,CG,DG,EG,FG,AA,BB
      COMMON/CONST/TC,PC,VC,TFR,AJ,EEP
      SAVE TGLAST,STPLST
      DATA TGLAST/-111./
C
      T=TG+TFR
      IF(T.LE.0.)GOTO 999
      IF(T.GT.TC)GOTO 999
      IF(ABS(TG-TGLAST).GT. 1.0E-4)GOTO 5
      SATP=STPLST
          RETURN
    5 C=ALOG(ABS(FG-T))
      SATP=EEP**(AG+BG/T+CG*ALOG(T)+DG*T+EG*C*((FG-T)/T))
      STPLST=SATP
      TGLAST=TG
          RETURN
  999 WRITE(IPR,100)
  100 FORMAT(5X,'ERROR IN CALLING -SATP-')
          RETURN
      END
```

```
      FUNCTION SATPR(I,TG)
C
C**** PURPOSE:
C        TO CALCULATE REFRIGERANT PROPERTIES AT SATURATION
C        JANUARY 13, 1989
C
C**** INPUT DATA:
C        I      = 1,2,3,4 OR 5
C        TG     - REFRIG. TEMPERATURE   (F)
C        REFRIG. CONSTANTS - REFER TO THE MAIN PROGRAM.
C
C**** OUTPUT DATA:
C        SATPR(1,TG) - SAT. LIQUID DYNAMIC VISCOSITY   (LBM/H*FT)
C        SATPR(2,TG) - SAT. VAPOR DYNAMIC VISCOSITY   (LBM/H*FT)
C        SATPR(3,TG) - SAT. LIQUID THERMAL CONDUCTIVITY   (BTU/H*FT*F)
C        SATPR(4,TG) - SAT. VAPOR THERMAL CONDUCTIVITY   (BTU/H*FT*F)
C        SATPR(5,TG) - SPEC. HEAT OF SAT. LIQUID   (BTU/LBM*F)
C
      DOUBLE PRECISION DPRES,T,AA
      COMMON/COEFPR/A(5,12)
      DIMENSION PRLAST(5),TGLAST(5)
      DATA TGLAST/5*-1111./
C
      IF(ABS(TG-TGLAST(I)).GT.1.0E-5)GOTO 5
      SATPR=PRLAST(I)
         RETURN
    5 T=TG
      DPRES=0.D0
      K=-1
      M=1
      IF(TG.GT.100.)M=M+6
      N=M+5
      DO10J=M,N
      K=K+1
      AA=A(I,J)
   10 DPRES=DPRES+AA*T**K
      SATPR=DPRES
      PRLAST(I)=SATPR
      TGLAST(I)=TG
         RETURN
      END
```

```
      FUNCTION SATT(PG)
C
C**** PURPOSE:
C        TO COMPUTE SATURATION TEMPERATURE FOR GIVEN PRESSURE
C
C**** INPUT DATA:
C        PG   - REFRIG. PRESSURE   (PSIA)
C        REFRIG. CONSTANTS - REFER TO THE MAIN PROGRAM.
C
C**** OUTPUT DATA:
C        SATT - REFRIG. SATURATION TEMPERATURE   (F)
C
      COMMON/PRINT/IPR
      COMMON/TGPG/AG,BG,CG,DG,EG,FG,AA,BB
      COMMON/CONST/TC,PC,VC,TFR,AJ,EEP
      SAVE PGLAST,SATLST
      DATA PGLAST/0./
C
      IF(PG.LE.0.)GOTO 999
      IF(ABS(PG-PGLAST).GT.1.0E-3)GOTO 5
      SATT=SATLST
      RETURN
    5 PLOG=ALOG(PG)
      TR=0.43429448*AA*PLOG+BB
      DO10ITR=1,30
      TRO=TR
      C=ALOG(ABS(FG-TRO))
      F=AG+BG/TRO+CG*ALOG(TRO)+DG*TRO+EG*((FG-TRO)*C/TRO)-PLOG
      FT=-BG/TRO**2+CG/TRO+DG
      IF(ABS(EG).GE.1.E-20)FT=FT-EG*(1./TRO+FG*C/TRO**2)
      TR=TRO-F/FT
      IF(ABS(TR-TRO).LE.0.05)GOTO 20
   10 CONTINUE
  999 WRITE(IPR,100)PG
  100 FORMAT(5X,'ERROR IN CALLING SATT, PG=',1PE11.3)
      RETURN
   20 SATT=TR-TFR
      SATLST=SATT
      PGLAST=PG
      RETURN
      END
```

```
      FUNCTION SATVF(TF)
C
C**** PURPOSE:
C       TO COMPUTE SPECIFIC VOLUME OF SATURATED LIQUID REFRIGERANT
C
C**** INPUT DATA:
C       TF    - SATURATED LIQUID TEMPERATURE   (F)
C       REFRIG. CONSTANTS - REFER TO THE MAIN PROGRAM.
C
C**** OUTPUT DATA:
C       SATVF - SPECIFIC VOLUME OF SAT. LIQUID REFRIGERANT   (FT**3/LBM)
C
      COMMON/PRINT/IPR
      COMMON/DENSF/AL,BL,CL,DL,EL,BPL,CPL,DPL,EPL
      COMMON/CONST/TC,PC,VC,TFR,AJ,EEP
      SAVE TFLAST,VFLAST
      DATA TFLAST/-1111./
C
      T=TF+TFR
      IF(T.GT.TC)GOTO 999
      IF(T.LE.0.)GOTO 999
      IF(ABS(TF-TFLAST).GT. 1.0E-3)GOTO 5
      SATVF=VLAST
         RETURN
    5 T=1.-T/TC
      SATVF=1./(AL+BL*T**BPL+CL*T**CPL+DL*T**DPL+EL*T**EPL)
      TFLAST=TF
      VLAST=SATVF
         RETURN
  999 WRITE(IPR,100)TF
  100 FORMAT(5X,'ERROR IN CALLING SATVF, TF = ',1PE10.3)
         RETURN
      END
```

```
      FUNCTION SPHDP(I,T,P,RMAS,AL,D)
C
C**** PURPOSE:
C       TO COMPUTE FRICTIONAL SINGLE PHASE PRESSURE DROP IN A TUBE
C
C**** INPUT DATA:
C       AL    - TUBE LENGTH   (FT)
C       D     - TUBE INNER DIAMETER   (FT)
C       I     = 1   FOR SATURATED LIQUID   (-)
C             = 2   FOR SATURATED VAPOR   (-)
C             = 3   FOR SUPERHEATED VAPOR   (-)
C       P     - REFRIG. PRESSURE AT INLET   (PSIA)
C       RMAS  - REFRIG. MASS FLOW RATE   (LBM/H)
C       T     - REFRIG. TEMPERATURE AT INLET   (F)
C
C**** OUTPUT DATA:
C       SPHDP - PRESSURE DROP OVER TUBE LENGTH   (PSIA)
C
C**** SUBPROGRAMS CALLED BY SPHDP:
C       SATP,SATPR,SATT,SATVF,VPSV
C
      IF(I.EQ.2)GOTO 1
      IF(I.EQ.3)GOTO 2
      VSP=SATVF(T)
      AMU=SATPR(1,T)
      GOTO 3
    1 P=SATP(T)
    2 VSP=VPSV(P,T)
      TG=SATT(P)
      AA=((T+460.)/(TG+460.))**0.5
      AMU=AA*SATPR(2,TG)
    3 AC=2./(32.174*144.*3600.**2)
      G=0.7853981*D*D
      G=RMAS/G
      RE=G*D/AMU
      F=0.046/RE**0.2
      IF(RE.LT.2000.)F=16./RE
      SPHDP=AC*F*VSP*AL*G*G/D
      RETURN
      END
```

```
      FUNCTION SPHDP1(AM,AL,D,VSP,AMU)
C
C**** PURPOSE:
C        TO COMPUTE FRICTIONAL PRESSURE DROP
C        FOR SINGLE PHASE FLOW IN A TUBE
C
C**** INPUT DATA:
C        AL      - TUBE LENGTH   (FT)
C        AM      - FLUID MASS FLOW RATE   (LBM/H)
C        AMU     - FLUID DYNAMIC VISCOSITY   (LBM/H*FT)
C        D       - TUBE DIAMETER   (FT)
C        VSP     - FLUID SPECIFIC VOLUME   (FT**3/LBM)
C
      ACC=3.3309E-11
      G=0.78539816*D*D
      G=AM/G
      RE=G*D/AMU
      F=0.046/RE**0.2
      SPHDP1=ACC*F*VSP*AL*G*G/D
      RETURN
      END
```

```
      FUNCTION SPHTC(CP,AM,AK,RMASS,D)
C
C**** PURPOSE:
C       TO COMPUTE SINGLE PHASE HEAT TRANSFER COEFFICIENT
C       FOR FLOW INSIDE TUBE
C
C**** INPUT DATA:
C       AM     - FLUID DYNAMIC VISCOSITY   (LBM/FT*H)
C       AK     - FLUID THERMAL CONDUCTIVITY   (BTU/H*F*FT)
C       CP     - FLUID SPECIFIC HEAT AT CONST. PRESSURE   (BTU/LBM*F)
C       D      - TUBE DIAMETER   (FT)
C       RMASS  - FLUID MASS FLOW RATE   (LBM/H)
C
C**** OUTPUT DATA:
C       SPHTC - SINGLE PHASE HEAT TRANSFER COEFF.   (BTU/H*F*FT**2)
C
      G=RMASS/(0.7853982*D*D)
      RE=D*G/AM
      IF(RE.GE.2000.)GOTO10
      SPHTC=4.36*AK/D
      GOTO20
   10 PR=(AM*CP/AK)**0.4
      RE=RE**0.8
      SPHTC=0.023*AK*PR*RE/D
   20 RETURN
      END
```

```
      SUBROUTINE TRACE3
C
C**** THIS PROGRAM DETERMINES REFRIGERANT DISTRIBUTION AMONG TUBES
C     IN A CROSS FLOW EVAPORATOR BASED ON CIRCUITRY CONFIGURATION.
C**** THE EVAPORATOR ASSEMBLY MAY CONSIST OF ONE OR TWO COILS (SLABS).
C**** 11-18-1988
C
C**** INPUT DATA:
C        IFROM(IIO,J)    - NUMBER OF THE TUBE FROM WHICH TUBE J RECEIVES
C                          REFRIGERANT.  IF THE TUBE IS CONNECTED TO THE
C                          INLET MAINFOLD, IFROM IS SET TO 0.
C        IIO             = 1  FOR THE FIRST SLAB   (-)
C                        = 2  FOR THE SECOND SLAB   (-)
C        NSLABS          - NUMBER OF SLABS IN THE EVAPORATOR ASSEMBLY   (-)
C        NTUB(IIO,N)     - NUMBER OF TUBES IN THE ROW N   (-)
C
C**** OUTPUT DATA:
C        FLOW(IIO,J)     - FRACTION OF COIL TOTAL REFRIG. MASS FLOW
C                          PASSING THROUGH TUBE J   (-)
C        IDEPTH(IIO,J)   - DEPTH ROW OF THE TUBE J   (-)
C        IMER(IIO)       - NUMBER OF SPLIT POINTS   (-)
C        IOUT(IIO,L)     - NUMBER OF THE TUBE CONNECTED TO THE OUTLET
C                          MANIFOLD, FOUND AS L SUCH TUBE   (-)
C        KFEED(IIO,J,N)  - NUMBER OF THE TUBE RECEIVING REFRIGERANT
C                          FROM TUBE J, FOUND AS N SUCH A TUBE
C                          NOTE THAT TUBE J CAN FEED UP TO 3 TUBES
C                          (N CAN BE 1,2 AND 3). KFEED IS SET TO -1 IF
C                          J TUBE FEEDS THE DISCHARGE MANIFOLD. KFEED
C                          IS SET TO 0 IF A TUBE IS NOT FED.   (-)
C        KSTART(IIO,N)   - NUMBER OF THE TUBE CONNECTED TO THE INLET
C                          MANIFOLD, FOUND AS N SUCH TUBE   (-)
C        KST(IIO)        - NUMBER OF TUBES CONNECTED TO THE INLET
C                          MANIFOLD   (-)
C        MERGE(IIO,K,1)  - NUMBER OF THE TUBE WHICH FEEDS A SPLIT POINT,
C                          FOUND AS K SUCH TUBE   (-)
C        MERGE(IIO,K,2)  - NUMBER OF TUBES FED BY THE TUBE K   (-)
C        NDEP(IIO)       - NUMBER OF TUBE DEPTH ROWS IN THE SLAB   (-)
C        NOUT(IIO)       - NUMBER OF TUBES CONNECTED TO THE OUTLET
C                          MANIFOLD   (-)
C        NTPS(IIO)       - NUMBER OF TUBES IN THE SLAB   (-)
C        SFLOW(IIO)      - FRACTION OF TOTAL REFRIGERANT MASS FLOW RATE
C                          FOR THE COIL FLOWING THROUGH SLAB IIO
C
C**** SUBPROGRAMS CALLED BY TRACE3: FRACT
C
      COMMON/HPHX/NSLABS,NDEP(2),NROW(2),DI(2),DO(2),DT(2),TPCH(2),
     & DPCH(2),WIDTH(2),FPCH(2),FTK(2),FMK(2),TMK(2),CFM1,BSIDE(2),
     & NTUB(2,5),IFROM(2,130),NTPS(2),BSPACE(2),
     & ACFM(2),IFIN(2),ISUR(2),SFLOW(2)
      COMMON/MERG/MERGE(2,20,2),IMER(2),IOUT(2,20),NOUT(2),
     & IDEPTH(2,130),FLOW(2,130),KFEED(2,130,3),KSTART(2,130),KST(2)
      DIMENSION IDID(130),LEFT(20),TC(130),RN(20),F(20),ITUBE(20),
     & ISEE(20)
C
      DO 200 IIO=1,NSLABS
C**** FIND NUMBER TUBES IN THE SLAB
      NTPS(IIO)=0
      DO 1 I=1,5
      IF(NTUB(IIO,I).NE.0)NDEP(IIO)=I
    1 NTPS(IIO)=NTPS(IIO)+NTUB(IIO,I)
C
      DO 3 I=1,NTPS(IIO)
      FLOW(IIO,I)=0.
      DO 3 J=1,3
    3 KFEED(IIO,I,J)=0
      DO 4 I=1,20
    4 MERGE(IIO,I,1)=0
C
C**** FIND TUBES CONNECTED TO THE OUTLET MANIFOLD
C**** FIND TUBES WHICH FEED SPLIT POINTS
      IS=0
      IM=0
```

```
      DO 10 J=1,NTPS(IIO)
      NM=0
      DO 6 I=1,NTPS(IIO)
      IF(IFROM(IIO,I).NE.J)GOTO 6
      NM=NM+1
    6 CONTINUE
      IF(NM.EQ.0)GOTO 8
      IF(NM.EQ.1)GOTO 10
      IM=IM+1
      MERGE(IIO,IM,1)=J
      MERGE(IIO,IM,2)=NM
      GOTO 10
    8 IS=IS+1
      IOUT(IIO,IS)=J
   10 CONTINUE
      NOUT(IIO)=IS
      IMER(IIO)=IM
C
C**** FIND DEPTH ROW FOR EACH TUBE
      NNDEP=NDEP(IIO)
      ILAST=0
      DO 22 J=1,NNDEP
      IFIRST=ILAST+1
      ILAST=ILAST+NTUB(IIO,J)
      DO 22 I=IFIRST,ILAST
      IDEPTH(IIO,I)=J
   22 CONTINUE
C
C**** FIND REFRIG. FLOW PATH FROM THE INLET TO THE OUTLET  (KFEED ARRAY)
      DO 50 I=1,NTPS(IIO)
      DO 48 IK=1,IMER(IIO)
      IF(I.NE.MERGE(IIO,IK,1))GOTO 48
      IDID(I)=MERGE(IIO,IK,2)
      GOTO 50
   48 CONTINUE
      IDID(I)=1
   50 CONTINUE
C
      DO 60 IS=1,NOUT(IIO)
      I=IOUT(IIO,IS)
      KFEED(IIO,I,1)=-1
   54 J=I
      I=IFROM(IIO,J)
      IF(I.EQ.0)GOTO 60
      N=IDID(I)
      KFEED(IIO,I,N)=J
      IDID(I)=IDID(I)-1
      IF(IDID(I).EQ.0)GOTO 54
   60 CONTINUE
C
C**** FIND THE INLET EVAPORATOR TUBES, KSTART(IIO,N)
      KS=0
      DO 63 I=1,NTPS(IIO)
      IO=IFROM(IIO,I)
      IF(IO.EQ.0)THEN
         KS=KS+1
         KSTART(IIO,KS)=I
      END IF
   63 CONTINUE
      KST(IIO)=KS
C
C**** FIND REFRIGERANT FLOW DISTRIBUTION
C**** FIND FLOW RESISTANCE FOR EACH TUBE
      DO 65 I=1,IMER(IIO)
   65 LEFT(I)=MERGE(IIO,I,2)
      DO 70 I=1,NTPS(IIO)
   70 TC(I)=0.
      DO 120 IL=1,NOUT(IIO)
      I=IOUT(IIO,IL)
      RC=1.
      DO 100 IT=1,NTPS(IIO)
      IP=IFROM(IIO,I)
      IF(IP.EQ.0)THEN
```

```
          TC(I)=RC
          GOTO 120
        END IF
        IF(KFEED(IIO,IP,2).EQ.0)THEN
          RC=RC+1
          I=IP
          GOTO 100
        END IF
        TC(I)=RC
        DO 75 IM=1,IMER(IIO)
   75   IF(IP.EQ.MERGE(IIO,IM,1))GOTO 77
   77   LEFT(IM)=LEFT(IM)-1
        IF(LEFT(IM).GT.0)GOTO 120
        NSPLIT=MERGE(IIO,IM,2)
        DO 90 I1=1,NSPLIT
        N=KFEED(IIO,IP,I1)
   90   RN(I1)=TC(N)
        CALL FRACT(NSPLIT,RN,F)
        DO 92 I1=1,NSPLIT
        N=KFEED(IIO,IP,I1)
   92   FLOW(IIO,N)=F(I1)
        RC=TC(N)*FLOW(IIO,N)**1.75
        I=IP
  100   CONTINUE
  120   CONTINUE
        NSTART=KST(IIO)
        DO 130 I=1,NSTART
        N=KSTART(IIO,I)
  130   RN(I)=TC(N)
        CALL FRACT(NSTART,RN,F)
        DO 132 I=1,NSTART
        N=KSTART(IIO,I)
  132   FLOW(IIO,N)=F(I)
C
C**** ASSIGN REFRIGERANT DISTRIBUTION,  FLOW(IIO,I)
        ISTORE=0
        DO 136 I=1,IMER(IIO)
        ITUBE(I)=0
  136   ISEE(I)=0
        DO 160 IS=1,NSTART
        I=KSTART(IIO,IS)
        IL=1
        DO 150 IO=1,NOUT(IIO)
        DO 145 IT=1,NTPS(IIO)
        IN1=KFEED(IIO,I,IL)
        IF(IN1.EQ.-1)THEN
          IF(ISTORE.GT.0)THEN
            I=ITUBE(ISTORE)
            IL=ISEE(ISTORE)
            ISTORE=ISTORE-1
            GOTO 150
          END IF
          GOTO 160
        END IF
        IF(IL.GT.1)GOTO 137
        DO 135 I1=2,3
        IN2=KFEED(IIO,I,I1)
        IF(IN2.EQ.0)GOTO 137
        ISTORE=ISTORE+1
        ITUBE(ISTORE)=I
  135   ISEE(ISTORE)=I1
  137   IN2=KFEED(IIO,I,2)
        IF(IN2.GT.0)THEN
          FLOW(IIO,IN1)=FLOW(IIO,IN1)*FLOW(IIO,I)
        ELSE
          FLOW(IIO,IN1)=FLOW(IIO,I)
        END IF
        I=IN1
        IL=1
  145   CONTINUE
  150   CONTINUE
  160   CONTINUE
        DO 170 I=1,NTPS(IIO)
```

```
170 FLOW(IIO,I)=FLOW(IIO,I)*SFLOW(IIO)
200 CONTINUE
    RETURN
    END
```

```
      FUNCTION VPSV(P,TG)
C
C**** PURPOSE:
C        TO COMPUTE REFRIGERANT VAPOR SPECIFIC VOLUME
C**** JANUARY 11, 1989
C
C**** INPUT DATA:
C        P    - REFRIG. VAPOR PRESSURE    (PSIA)
C        TG   - REFRIG. VAPOR TEMPERATURE    (F)
C        REFRIG. CONSTANTS - REFER TO THE MAIN PROGRAM.
C
C**** OUTPUT DATA:
C        VPSV - SPECIFIC VOLUME OF REFRIG. VAPOR    (FT**3/LBM)
C
C**** SUBPROGRAMS CALLED BY VPSV:
C        SATT
C
      DOUBLE PRECISION F,FV,V,VN,Z,EMAV,T,AKTTC,ES0,ES1,ES2,ES3,
     1 ES4,ES5,ES6,ES7,ES32,ES43,ES54,ES65,V2,V3,V4,V5,V6
      COMMON/PRINT/IPR
      COMMON/STATE/A1,B1,C1,A2,B2,C2,A3,B3,C3,A4,B4,C4,A5,B5,C5,
     &A6,B6,C6,ALPHA,AK
      COMMON/CONST/TC,PC,VC,TFR,AJ,EEP
      SAVE PLAST,TGLAST,VLAST
      DATA PLAST,TGLAST/-1.,-111./
C
      T=TG+TFR
      IF(T.LT.0.)GOTO999
      IF(P.LE.0.)GOTO999
      IF(ABS(P-PLAST).GT. 1.0E-4)GOTO 5
      IF(ABS(TG-TGLAST).GT. 1.0E-4)GOTO 5
      VPSV=VLAST
          RETURN
    5 TSAT=SATT(P)
      IF(TG.LT.(TSAT-0.05))GOTO999
      AKTTC=AK*T/TC
      ES0=DEXP(-AKTTC)
      ES1=P
      ES2=A1*T
      ES3=A2+B2*T+C2*ES0
      ES4=A3+B3*T+C3*ES0
      ES5=A4+B4*T+C4*ES0
      ES6=A5+B5*T+C5*ES0
      ES7=A6+B6*T+C6*ES0
      ES32=2.*ES3
      ES43=3.*ES4
      ES54=4.*ES5
      ES65=5.*ES6
      VN=A1*T/P
      DO 10 ITR=1,30
      V=VN
      V2=V*V
      V3=V*V2
      V4=V*V3
      V5=V*V4
      V6=V*V5
      Z=ALPHA*(V+B1)
      IF(Z.GT.150.D0)Z=150.D0
      EMAV=DEXP(-Z)
      F=ES1-ES2/V-ES3/V2-ES4/V3-ES5/V4-ES6/V5-ES7*EMAV
      FV=ES2/V2+ES32/V3+ES43/V4+ES54/V5+ES65/V6+ES7*ALPHA*EMAV
      VN=V-F/FV
      IF(DABS((VN-V)/V).LE.1.D-06)GOTO20
   10 CONTINUE
      VPSV=VN+B1
      WRITE(IPR,50)
   50 FORMAT(5X,'VPSV DOES NOT CONVERGE')
          RETURN
  999 VPSV=0.
      WRITE(IPR,100)TG,TSAT,P
  100 FORMAT(5X,'ERROR IN CALLING -VPSV-',
     &  /' TG,TSAT,P=',3(1PE11.3))
```

```
      RETURN
20 VPSV=VN+B1
   VLAST=VPSV
   PLAST=P
   TGLAST=TG
      RETURN
   END
```

```
      SUBROUTINE VPVHS(I,TSG,P,V,H,S,HF)
C
C**** PURPOSE:
C       TO COMPUTE REFRIGERANT PARAMETERS
C       AT AND ABOVE SATURATION
C**** JANUARY 11, 1989
C
C**** INPUT DATA:
C       I   = 1  FOR SATURATED REFRIGERANT   (-)
C           = 2  FOR SUPERHEATED REFRIG. VAPOR   (-)
C       P   - REFRIG. PRESSURE (REQUIRED FOR I=2 ONLY)   (PSIA)
C       TSG - REFRIG. TEMPERATURE   (F)
C       REFRIG. CONSTANTS - REFER TO THE MAIN PROGRAM.
C
C**** OUTPUT DATA:
C       H   - ENTHALPY OF VAPOR   (BTU/LBM)
C       HF  - ENTHALPY OF SAT. LIQUID (FOR I=1 ONLY)   (BTU/LBM)
C       P   - SATURATION PRESSURE (FOR I=1 ONLY)   (PSIA)
C       S   - ENTROPY OF VAPOR   (BTU/LBM*F)
C       V   - SPECIFIC VOLUME OF VAPOR   (FT**3/LBM)
C
C**** SUBPROGRAMS CALLED BY VPVHS:
C       SATP,SATT,SATVF,VPSV
C
      DOUBLE PRECISION Z,T,C,T2,T3,T4,VR,VR2,VR3,VR4,AKE,AKEXP,
     1 H1,H2,H3,H4,HFGD,HD,CD,EMAV,S1,S2,S3,S4,HO
      COMMON/PRINT/IPR
      COMMON/TGPG/AG,BG,CG,DG,EG,FG,AA,BB
      COMMON/STATE/A1,B1,C1,A2,B2,C2,A3,B3,C3,A4,B4,C4,A5,B5,C5,
     &A6,B6,C6,ALPHA,AK
      COMMON/SPHTV/AC,BC,CC,DC,EC,FC,X,Y
      COMMON/CONST/TC,PC,VC,TFR,AJ,EEP
      SAVE TSGLST,PLAST,VLAST,HLAST,SLAST,HFLAST
      DATA TSGLST,PLAST/2*-111./
C
      T=TSG+TFR
      IF(T.LE.0.D0)GOTO 999
      IF(I.EQ.2)THEN
         IF(P.LT.0.)GOTO 999
      END IF
      IF(ABS(TSG-TSGLST).GT. 1.0E-4)GOTO 5
      IF(I.EQ.2)THEN
         IF(ABS(P-PLAST).GT.  1.0E-4)GOTO 5
      END IF
      IF(I.EQ.1)P=PLAST
      V=VLAST
      H=HLAST
      S=SLAST
      HF=HFLAST
         RETURN
    5 IF(I.EQ.1)GOTO10
      TSAT=SATT(P)
      IF(TSG.LT.TSAT)GOTO 999
      V=VPSV(P,TSG)
      GOTO20
   10 P=SATP(TSG)
      V=VPSV(P,TSG)
      C=DLOG(DABS(FG-T))/T
      VF=SATVF(TSG)
      HFGD=(V-VF)*P*AJ*(-BG/T+CG+DG*T-EG*(1.+FG*C))
   20 T2=T*T
      T3=T*T2
      T4=T*T3
      VR=V-B1
      VR2=2.D0*VR*VR
      VR3=3.D0*VR*VR2/2.D0
      VR4=4.D0*VR*VR3/3.D0
      AKE=AK*T/TC
      AKEXP =DEXP(-AKE)
      Z=ALPHA*V
      IF(Z.GT.150.D0)Z=150.D0
      EMAV=DEXP(-Z)
```

```
      H1=AC*T+BC*(T2/2)+CC*(T3/3.)+DC*(T4/4.)-FC/T
      H2=AJ*P*V
      H3=A2/VR+A3/VR2+A4/VR3+A5/VR4
      H4=C2/VR+C3/VR2+C4/VR3+C5/VR4
      S1=AC*DLOG(T)+BC*T+CC*T2/2.+DC*T3/3.-FC/(2.*T2)
      S2=AJ*A1*DLOG(VR)
      S3=B2/VR+B3/VR2+B4/VR3+B5/VR4
      S4=H4
      IF(ABS(A6).LE.1.E-20)GOTO30
      H0=EMAV
      IF(ABS(C1).GT.1.E-20)H0=H0-C1*DLOG(1.D0+EMAV/C1)
      H0=H0/ALPHA
      H3=H3+A6*H0
      H4=H4-C6*H0
      S3=S3+B6*H0
      S4=S4-C6*H0
   30 HD=H1+H2+AJ*H3+AJ*AKEXP*(1.+AKE)*H4+X
      S=S1+S2-AJ*S3+AJ*AKEXP*AK/TC*S4+Y
      HFG=HFGD
      H=HD
      IF(I.EQ.1)HF=H-HFG
      TSGLST=TSG
      PLAST=P
      VLAST=V
      HLAST=H
      SLAST=S
      HFLAST=HF
         RETURN
  999 WRITE(IPR,100)TSG,P
  100 FORMAT(5X,'ERROR IN CALLING -VPVHS-',2(1PE11.3))
         RETURN
      END
```

```
      SUBROUTINE WATPR(TW,TP,VA,WA,WATRO,WATK,WATM,WATHFG,WATCP)
C
C****PURPOSE:
C        TO COMPUTE WATER AND FROST PROPERTIES
C
C**** INPUT DATA:
C        TW      - WATER (FROST) TEMPERATURE    (F)
C        TP      - TUBE TEMPERATURE    (F)
C        WA      - AIR HUMIDITY RATIO    (LBM H2O/LBM DRY AIR)
C        VA      - AIR VELOCITY    (FT/SEC)
C
C**** OUTPUT DATA:
C        WATRO   - DENSITY OF WATER (FROST)    (LBM/FT**3)
C        WATK    - THERMAL CONDUCTIVITY OF WATER (FROST)    (BTU/H*F*FT)
C        WATM    - DYNAMIC VISCOSITY OF WATER (FROST)    (LBM/H*FT)
C        WATHFG  - WATER HEAT OF CONDENSATION    OR
C                  FROST HEAT OF SUBLIMATION    (BTU/LBM)
C        WATCP   - SPECIFIC HEAT OF WATER (FROST)    (BTU/LBM*F)
C
      DIMENSION ARO(5),AK(5),AM(5),AHFG(5)
C
      ARO(1)=0.11647E03
      ARO(2)=-0.40054E00
      ARO(3)=0.10815E-02
      ARO(4)=-0.12387E-05
      ARO(5)=0.49002E-09
      AK(1)=-0.27694E00
      AK(2)=0.45215E-03
      AK(3)=0.49008E-05
      AK(4)=-0.88613E-08
      AK(5)=0.41387E-11
      AM(1)=0.79424E03
      AM(2)=-0.47589E01
      AM(3)=0.10622E-01
      AM(4)=-0.10416E-04
      AM(5)=0.37690E-08
      AHFG(1)=0.31514E04
      AHFG(2)=-0.13714E02
      AHFG(3)=0.35945E-01
      AHFG(4)=-0.43525E-04
      AHFG(5)=0.19695E-07
      TWR=TW+460.
      TPR=TP+460.
      WATRO=0.
      WATK=0.
      WATM=0.
      WATHFG=0.
      IF(TW.LE.32.)GOTO 100
      DO10I=1,5
      J=I-1
      WATRO=WATRO+ARO(I)*TWR**J
      WATK=WATK+AK(I)*TWR**J
      WATM=WATM+AM(I)*TWR**J
   10 WATHFG=WATHFG+AHFG(I)*TWR**J
      WATCP=1.
      GOTO 200
  100 B1=-11.9521+0.02422*TPR+35.5498*WA
     &-9.1742E-07*VA+3.1138E-09*VA*TPR-0.03838
      B2=(13.1606-0.02133*TPR-81.955*WA)/(32.018-TP)
      WATRO=10.*EXP(B1+B2*(TW-TP))
      WATK=0.012138+3.8909E-03*WATRO+5.1409E-06*WATRO**3
      WATM=1.E25
      WATHFG=1219.
      WATCP=0.46
  200 RETURN
      END
```

APPENDIX C. Example of Run of the Program, EVSIM

Below is a printout from execution of EVSIM. The simulated coil is shown in Figure A1. The coil data file is presented in Table A4. The working fluid is Refrigerant 22.

```
**************************************************************
     EVSIM (VER. 1.1): SIMULATION OF AN EVAPORATOR COIL
**************************************************************
 March 14, 1989
 COIL INFORMATION:
***DTEV***   A-SHAPE COIL, 3 DEPTH ROWS, 16 TUBES PER ROW.

      NUMBERS OF SLABS IN THE ASSEMBLY:    2
      NUMBER OF EXPANSION DEVICES:         2
      TOTAL CFM:                        1120
 AIR CONDITION:
      AIR DRY BULB TEMPERATURE:           80.00 F
      AIR RELATIVE HUMIDITY:               0.51
 REFRIGERANT 22
 REFRIGERANT CONDITIONS:
      REFRIG. QUALITY AT INLET:            0.20
      REFRIG. SAT. TEMPERATURE AT EXIT:   45.00 F
```

NBS-114A (REV. 2-80)

U.S. DEPT. OF COMM. BIBLIOGRAPHIC DATA SHEET *(See instructions)*	1. PUBLICATION OR REPORT NO.	2. Performing Organ. Report No.	3. Publication Date
	NISTIR 89-4133		AUGUST 1989

4. TITLE AND SUBTITLE

EVSIM - An Evaporator Simulation Model Accounting for Refrigerant and One Dimensional Air Distribution

5. AUTHOR(S)

Piotr A. Domanski

6. PERFORMING ORGANIZATION *(If joint or other than NBS, see instructions)*

NATIONAL BUREAU OF STANDARDS
DEPARTMENT OF COMMERCE
WASHINGTON, D.C. 20234

7. Contract/Grant No.

8. Type of Report & Period Covered

9. SPONSORING ORGANIZATION NAME AND COMPLETE ADDRESS *(Street, City, State, ZIP)*

U.S. Department of Energy
1000 Independence Ave., SW
Washington, DC 20585

10. SUPPLEMENTARY NOTES

☐ Document describes a computer program; SF-185, FIPS Software Summary, is attached.

11. ABSTRACT *(A 200-word or less factual summary of most significant information. If document includes a significant bibliography or literature survey, mention it here)*

The report describes a computer model, EVSIM, of a refrigerant-to-air heat exchanger of the type used in residential air conditioning as an evaporator. The model provides performance predictions of a one-slab or two-slab evaporator for a given refrigerant enthalpy at the coil inlet, saturation temperature and superheat at the coil outlet, and at imposed one dimensional air mass flow distribution over the coil face.

The model accounts for air distribution and for complex refrigerant circuitry designs by simulating refrigerant distribution. Performance of the coil is calculated employing a tube-by-tube scheme. Performance of each tube is evaluated individually based on individual air and refrigerant mass flow rates and their respective thermodynamic states assigned for each tube. The modelling effort emphasis was on the local thermodynamic phenomena which were described by fundamental heat transfer equations and equations of state relationships among material properties.

This report includes a User's Guide and a listing written in FORTRAN 77. Due to the detailed algorithms and tube-by-tube performance evaluation scheme, mini and main frame computers are best suited for simulation studies using EVSIM. Nevertheless, the model converges on an IBM AT compatible machine within 2-6 minutes when simulating a single slab evaporator.

12. KEY WORDS *(Six to twelve entries; alphabetical order; capitalize only proper names; and separate key words by semicolons)*