This is to certify that the

thesis entitled

TSCAD: A HEAT EXCHANGER ANALYSIS AND DESIGN SOFTWARE PACKAGE

presented by

JON THELEN

has been accepted towards fulfillment
of the requirements for

MASTERS degree in MECHANICAL ENGINEERING

Craig W. Somerton
Major professor

Date February 22, 1991

TSCAD: A HEAT EXCHANGER ANALYSIS AND DESIGN SOFTWARE PACKAGE

By

Jon J. Thelen

A THESIS

Submitted to
Michigan State University
in partial fulfillment of the requirements
for the degree of

MASTER OF SCIENCE

Department of Mechanical Engineering

ABSTRACT

TSCAD: A HEAT EXCHANGER ANALYSIS AND DESIGN SOFTWARE PACKAGE

By

Jon J. Thelen

A software package has been developed for the analysis and design of heat exchangers. A variety of heat exchanger types can be evaluated including double pipe heat exchangers, crossflow heat exchangers, and fifteen different compact heat exchangers. The program can do either a traditional sizing calculation or a rating calculation. It can also handle cases where both mass flow rates are unknown, the mass flow rate of one fluid stream and the exit temperature of the other fluid stream are unknown, and the mass flow rate and exit temperature of the same fluid stream are unknown.

Overall heat transfer coefficients are calculated within the program using state of the art Nusselt number correlations. Property values for air, water, steam, and oil are calculated within the program from curve fit equations.

Several examples are performed to demonstrate the use of the program. A user's manual for the software is included.

Table of Contents

Table of Contents - Continued

LIST OF TABLES

LIST OF FIGURES

LIST OF FIGURES

SYMBOLS AND ABBREVIATIONS

Symbol	Meaning
A	area of heat exchanger
A_f	fin surface area
B	unknown parameter in parameter estimation
C	heat capacity
C_1	constant in compact exchanger data
C_{max}	maximum heat capacity
C_{min}	minimum heat capacity
C_r	C_{min}/C_{max}
c_p	specific heat
di	change in enthalpy
D_i	inside diameter
D_o	outside diameter
dT	change in temperature
E	effectiveness of a heat exchanger
h	convective heat transfer coefficient
i	enthalpy
j_H	Colburn factor
k	thermal conductivity
K	degrees Kelvin
kg	kilograms
kJ	kiloJoules
L	length
M	viscosity

SYMBOLS AND ABBREVIATIONS

Symbol	Meaning
$\dot{m}$	mass flow rate
m	$(2h/kt)^{1/2}$: variable in fin analysis
N	Newtons: unit of work
n	dummy variable: exponent in correlations
n_f	fin efficiency
n_o	fin effectiveness
NTU	Number of Transfer Units
Nu	Nusselt number
P	perimeter of fin
p	density
Pr	Prandtl number
q	heat transfer rate
q_{max}	maximum heat transfer
R''_f	resistance due to fouling
Re	Reynold's number
Re_{cr}	critical Reynold's number ~2300 for pipe flow
R_w	wall conduction resistance
s	percent standard deviation in error analysis
St	Stanton number
T	temperature
t	fin thickness
U	overall heat transfer coefficient
W	weighting factor in transition flow analysis

SYMBOLS AND ABBREVIATIONS

Symbol	Meaning
W	Watts
X	sensitivity coefficient in parameter estimation

Subscript	Meaning
$( )_c$	cold fluid stream
$( )_h$	hot fluid stream
$( )_{in}$	inlet flow condition
$( )_{out}$	outlet flow condition
$( )_{lam}$	laminar flow
$( )_{turb}$	turbulent flow

I. INTRODUCTION

The areas of mechanical design and systems and controls have made considerable use of the capabilities of computers in recent years. The areas of finite element analysis and optimization of mechanical systems are two examples. The area of heat transfer, however, has not made such efficient use of the vast potential ability of computers. There has been more of an effort to close the gap in computer-aided design between the heat transfer field and other areas of engineering lately. This package is a part of the push to increase the computer-aided design capability in heat transfer.

TSCAD is a software package that was written with the intent of providing an expert system which could carry out the calculations involved in heat exchanger analysis and design.

A heat exchanger is a device which exchanges energy, in the form of heat, between two fluids that are separated by a solid wall. Heat exchangers can be classified according to the geometry of the flow. According to this method of classification, three types of heat exchangers are considered in TSCAD: concentric tube, cross-flow, and compact heat exchangers. Concentric tube heat exchangers consist of two fluids flowing in a pair of concentric tubes. Cross-flow consists of transverse flow of one fluid over a tube containing the other fluid. Compact heat exchangers are characterized by a very large surface area to volume ratio, usually achieved by using a bank of finned tubes within a small volume. Typically, one of the fluids in a compact heat exchanger is a gas.

A heat exchanger has six main operating conditions: the inlet temperature, exit temperature, and mass flow rate for each fluid stream. These operating conditions, along with the heat exchange area and the overall heat transfer coefficient, describe the performance of a heat exchanger. TSCAD has the ability to solve for up to three unknown parameters for a given heat exchanger problem. Many of the problems encountered can be solved by the use of effectiveness-NTU (E-NTU) relations. By constructing some simple algorithms and making use of the standard E-NTU relations, all problems involving just the operating conditions can be solved, as well as those that involve just one of the variables U or A.

In the case of both the heat exchange area (A) and the overall heat transfer coefficient (U) unknown, the overall heat transfer coefficient must be calculated using the kinematic and thermal properties of the fluid, the flow velocity, and the geometry of the two flows. Established

Nusselt number correlations were used to calculate the convective coefficient for internal flow in a circular pipe (Dittus-Boelter [1]) and cross-flow over a cylinder (Churchill & Bernstein [2]). Heat transfer data from Kays and London [3] were used for the Colburn factor correlations for compact heat exchangers. No widely accepted correlations are available for flow through an annulus with a convective condition at the inner surface. However, Kays and Leung [4] have assembled experimental heat transfer data for the condition of turbulent flow in an annulus with a constant heat flux at the inner wall. Since these data were similar to that collected in the lab for an inside convective condition, it was used to construct a correlation for heat transfer from an annulus in a concentric tube heat exchanger. The resulting equation, accurate to about 15% in the worst case, is a function of both Reynold's number and Prandtl number and is similar in form to the Dittus-Boelter correlation for internal flow.

The third major portion of this work involves the fluid properties. In order to simplify the software and to save the time spent in opening and closing files, the properties are calculated within the program rather than being stored in a data file. The process of changing from data files to property equations involved plotting the set of available property data and deriving an equation or set of equations that describe the property as a function of temperature. This was done for the density, specific heat, thermal conductivity and kinematic viscosity of each of the four types of fluid, water, steam, air, and oil.

To demonstrate the capability of the system, a number of problems were solved using TSCAD. These are problems taken from selected heat transfer textbooks, as well as problems that make direct use of data obtained in the laboratory.

II. LITERATURE REVIEW

Recently, much work has been done in the area of computer-aided design of thermal systems. A good deal of that work is designed for the purpose of instruction.

One such example is the set of software used by the mechanical engineering department at Mississippi State University [11]. The curriculum at MSU makes use of more than twenty programs for thermal science instruction. Six of these deal specifically with heat exchangers.

Of the heat exchanger software, the program HX is the simplest. HX contains explicit functions of effectiveness as a function of NTU, and NTU as a function of E for some common heat exchanger arrangements. The problems that can be solved are very simple: given E calculate NTU, and given NTU, find E. HX is nearly identical to the E-NTU portion of program TSCAD. TSCAD contains the effectiveness equations in function EFF, and the NTU equations in function TRNS.

A second heat exchanger program used at MSU is CFHX. CFHX deals with cross flow of a fluid over finned tubes containing a second fluid. The main application for this program is in the analysis of compact heat exchangers. In TSCAD, COMPACT solves problems involving compact heat exchangers, while CROSSFLOW solves problems with transverse flow over a single bare tube.

Problems in shell and tube heat exchanger design cannot be solved by TSCAD yet. These types of problems, however, have been the subject of other heat exchanger software work. The curriculum at MSU uses one such program, STAN. STAN can calculate the outlet temperature of both fluid streams of a shell and tube heat exchanger, as well as estimate the pressure drop. TSCAD can calculate both outlet temperatures for concentric tube, cross-flow, and compact heat exchangers, but does not have the ability to calculate pressure drops.

Another example for shell and tube heat exchangers is the expert system created by Wang and Soler [12]. This program uses human expertise to carry out computations involved in heat exchanger analysis, and all decisions depend both on empirical and scientific knowledge. Like TSCAD, it is divided into modules. Among these are modules dealing with heat exchanger type, materials, dimensions, component types, parameter determination and calculation. Wang and Soler's program has the capability to perform mechanical design of a heat exchanger by making use of a data base containing ASME codes. The program uses four data bases, including one containing physical data.

Program MATCODE [13] is another example of the use of data bases. MATCODE stores information dealing only with metals. The properties are stored as functions of temperature, and the program can interpolate between values. The information from the data base is stored in a DOS file which can then be accessed by a Fortran program. This procedure is very similar to the way TSCAD originally accessed fluid properties.

Program HEATER was created for the analysis and design of feedwater heaters through interactive use on personal computers. The Fortran program is driven by screen inputs, from which it predicts off-design point performance. Like TSCAD, established correlations are used to calculate the convective coefficient, h.

TSCAD incorporates many of the assets which are considered the strong points of these other programs. TSCAD uses the same effectiveness and NTU functions used by HX [11]. It can solve those problems of flow over finned tubes which involve available compact heat exchanger data, like CFHX [11]. STAN [11] can compute the outlet fluid temperatures for shell and tube heat exchangers. TSCAD does this for concentric tube, cross-flow, and compact heat exchangers. A full expert system similar to TSCAD, is the program by Wang and Soler [12]. Empirical and scientific knowledge are used by both. Many of the modules in this program are similar to TSCAD as well, including heat exchanger type, materials, dimensions, component types, parameter determination, and calculation. Program HEATER estimates the overall heat transfer coefficient using empirical correlations, like TSCAD.

There are also many facets in these programs which TSCAD does not possess. Some will be incorporated, others will not.

CFHX [11] has the ability to calculate performance for flow over finned tubes. TSCAD is limited in this ability, since data for only fifteen compact heat exchangers is contained in the program. These are all of the circular tube, plate fin type. More compact heat exchanger data, as well as data for flow over banks of tubes, is a good area for future work on TSCAD.

Hydraulic characteristics, such as the pressure drop in STAN, are not calculated in TSCAD. Also, the ability to change units is not present in TSCAD, as it is in MATCODE [13]. The program presently works only in SI units, but this adjustment to the program will require minimal effort.

The ability to analyze shell and tube heat exchangers, present in STAN [11] and Wang and Soler's work [12], will certainly be added to TSCAD.

The ability to utilize graphics in the presentation of output, as well as to assist with input is also an area for future work in TSCAD. Graphics are used by Wang and Soler [12].

The use of data bases, as in MATCODE, will probably not be an area that TSCAD will expand into. It is felt that the computation of properties using equations is a more efficient method of retrieving data.

None of the literature reviewed mentioned any work in the area of concentric tube heat exchangers. The correlation for heat transfer from an annulus is unique work in computer-aided design of heat exchangers. Also not found in the literature is the work done in deriving temperature dependent property functions.

The literature reviewed suggests some areas in which TSCAD could be improved, such as shell and tube heat exchangers and utilization of graphics. These improvements would surely enhance the performance of TSCAD. Many of the facets of the other software are already present in TSCAD, however. This fact, along with the unique components such as the annulus correlation and the property calculation, make TSCAD a valuable tool for the engineer involved in heat exchanger analysis and design.

III. OVERVIEW OF THE PROGRAM

TSCAD was created with the intent of providing an expert system for the analysis and design of heat exchangers. The system can be used as a menu driven program for heat exchanger design, or as a subroutine for a larger thermal system analysis program.

There are four components of TSCAD: the main program, the effectiveness-NTU calculation program, the overall heat transfer coefficient calculation, and the physical property calculation. The main program is the menu-driven interface which collects the known data from the user. This portion of the program decides whether there is enough known information to solve the problem, and if so, calls the proper subroutine to begin the process. The effectiveness-NTU calculation portion includes the effectiveness and NTU functions themselves as well as the subroutines which make use of those functions. The overall heat transfer coefficient set of subroutines is partially menu-driven. It calculates the overall heat transfer coefficient independent of the effectiveness-NTU method,by making use of empirical correlations which relate the fluid dynamics, the fluid properties, and the geometry of the flow to the heat transfer. Finally, the physical property computation consists of a set of equations, derived from curve-fits of published data, which calculate the fluid properties as a function of temperature.

Figure 1 is a flowchart which shows the relationship between the subroutines in the different portions of the program. As shown, the main program receives the input and sorts for the problem type. The properties calculation portion is called by the main program. The properties are calculated by the functions CP, DENSITY, TCOND, and VISCOS. These data then are passed to the subroutines through the main program. The effectiveness-NTU program contains six subroutines: SIZING, RATES, ENERGY, ONE, TWO, and RATING; and two functions: EFF, and TRANS. These subroutines may be called directly by the main program or by another subroutine in the case of more complicated problems involving more than one unknown. The overall heat transfer coefficient calculation contains six subroutines. Subroutine OVERALL is the main program of this section, and all other subroutines are called from it. CONCENTRIC, CROSSFLOW, and COMPACT recieve data and calculate convective coefficients for the various types of heat exchangers. Subroutines RE and NU calculate the Reynold's number and Nusselt number, respectively, for the various correlations.

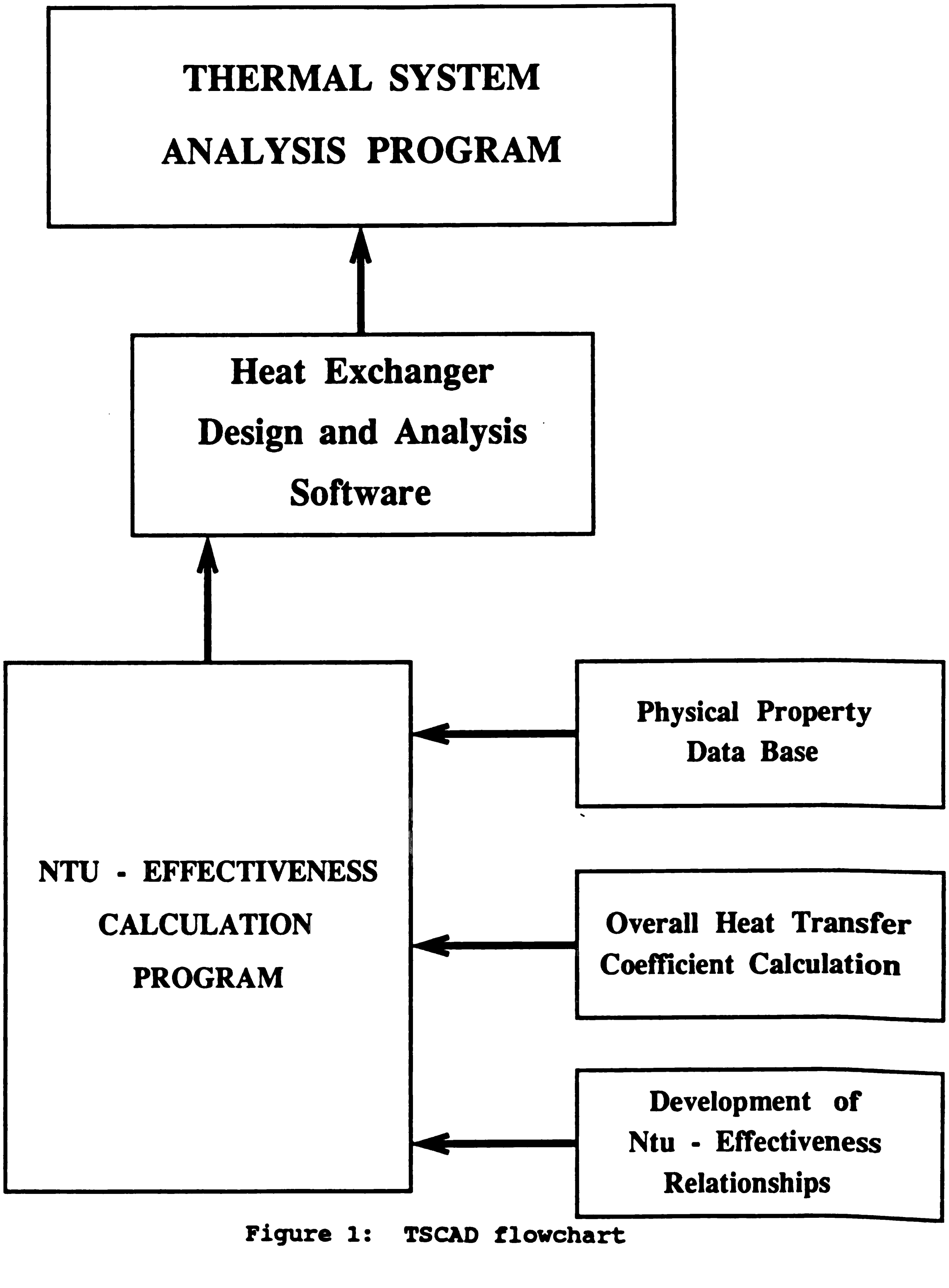


Figure 1: TSCAD flowchart

IV. BACKGROUND

4.1 Effectiveness - NTU Calculations

The most widely used method of heat exchanger analysis is the effectiveness - NTU method. The effectiveness of a heat exchanger is defined as the ratio of the actual heat transfer to the maximum possible heat transfer. The number of transfer units, NTU, is a dimensionless parameter used for heat exchanger analysis. The method is based on the premise that the effectiveness of a heat exchanger is related to the number of transfer units (NTU) by some known function. Using this function, one can solve for any one unknown variable in the relation.

The six operating conditions can be used to calculate the heat transfer rate and the effectiveness of the heat exchanger. If we assume that the heat exchanger is adiabatic, that is, no heat is lost to the environment, we may perform an energy balance on the heat exchanger and equate the enthalpy in to the enthalpy out.

$$\dot{m}_h i_{h,out} + \dot{m}_c i_{c,out} = \dot{m}_h i_{h,in} + \dot{m}_c i_{c,in} \tag{4.1}$$

where $\dot{m}$ = mass flow rate
i = enthalpy
$(\ )_h$ = hot fluid
$(\ )_c$ = cold fluid

This equation can be rearranged in the form of enthalpy change for each fluid.

$$\dot{m}_c(i_{c,in} - i_{c,out}) = \dot{m}_h(i_{h,out} - i_{h,in}) \tag{4.2}$$

For an ideal gas or an incompressible liquid, the enthalpy change can be related to the product of the specific heat and the temperature change, dT.

$$di = c_p dT \tag{4.3}$$

Air can be considered to be an ideal gas, and water and oil as incompressible liquids. The validity of these approximations can be shown by a thermodynamic analysis.

Using equation (4.3) and assuming that the specific heat is constant over a small temperature range, equation (4.2) can be rewritten in terms of the operating conditions.

$$(\dot{m}c_p)_c(T_{c,in} - T_{c,out}) = (\dot{m}c_p)_h(T_{h,out} - T_{h,in}) \tag{4.4}$$

To simplify notation, we may define the heat capacity of a fluid as the product of the mass flow rate and the specific heat.

$$C = \dot{m}c_p \tag{4.5}$$

where $\dot{m}$ = mass flow rate
c_p = specific heat

Equation (4.2) can then finally be rewritten in a compact form to give an expression for the heat transfer rate q, between the two fluids.

$$q = C_c(T_{c,in}-T_{c,out}) = C_h(T_{h,out}-T_{h,in}) \qquad (4.6)$$

Equation (4.6) is a valuable relation, because the energy balance in equation (4.1) has now been rewritten in terms of just operating conditions and fluid properties.

To calculate the effectiveness of a heat exchanger, the maximum heat transfer rate must also be known. To define the maximum heat transfer rate of a heat exchanger first consider a counterflow heat exchanger of infinite length. One of the fluids in this exchanger will achieve a temperature difference of the hot inlet temperature at one end and the cold inlet temperature at the other end. This is recognized as the maximum temperature difference possible. The fluid that will achieve this temperature difference is the one with the lower heat capacity, C_{min}. For a given heat transfer rate q, the fluid with the maximum temperature difference (fluid A), must have a smaller heat capacity than fluid B, since the two are inversely proportional. The maximum heat transfer rate can then be defined as the minimum heat capacity multiplied by the maximum temperature difference, which is the difference between the inlet temperature of the hot fluid and the inlet temperature of the cold fluid.

$$q_{max} = C_{min}(T_{h,in}-T_{c,in}) \tag{4.7}$$

The effectiveness of a heat exchanger is then defined as the ratio of the actual heat transfer rate to the maximum heat transfer rate.

$$E = q/q_{max} \tag{4.8}$$

where E = effectiveness
q = actual heat transfer
q_{max} = maximum heat transfer

Having defined the effectiveness of a heat exchanger, the number of transfer units must now be defined. The number of transfer units (NTU) is a dimensionless parameter, developed by Kays and London, defined as the product of the overall heat transfer coefficient and the heat exchanger area divided by the minimum heat capacity.

$$NTU = UA/C_{min} \tag{4.9}$$

For any heat exchanger, the effectiveness is a known function of the NTU and the ratio of C_{min} and C_{max}.

$$E = f(NTU, C_{min}/C_{max}) \tag{4.10}$$

Thus, heat exchanger performance can be evaluated in terms of effectiveness - NTU functions, with the variables consisting of the six operating conditions, the overall heat transfer coefficient, and the heat exchanger area.

4.2 Overall Heat Transfer Coefficient

The overall heat transfer coefficient accounts for conduction and convection resistances between fluids separated by solid walls. To see the concept of the overall heat transfer coefficient, consider an analogy between the diffusion of heat and electrical charge. Just as there is an electrical resistance associated with the conduction of electricity, there is a thermal resistance associated with the conduction of heat. Electrical resistance is defined as the voltage drop divided by the current. For convection, the thermal resistance is similarly defined as the temperature difference divided by the heat transfer rate, q. Using Newton's Law of Cooling, $R_{t,conv}$ can be written as

$$R_{t,conv} = (T_s - T_f)/q = 1/hA \qquad (4.11)$$

where T_s = surface temperature
T_f = free stream fluid temperature
q = heat transfer rate
h = convective coefficient
A = heat transfer area

The associated thermal resistance for convection in a cylindrical wall is [10]

$$R_{t,cond} = \ln(D_o/D_i)/6.28Lk \qquad (4.12)$$

where D_o = outside diameter
D_i = inside diameter
L = length of the cylindrical wall
k = thermal conductivity

Fouling of the wall surfaces can cause another resistance to the transfer of heat. This process will be described in more detail later. For now, it can be defined as

$$R_{t,foul} = R''_f/A \tag{4.13}$$

where R''_f = fouling factor
A = heat transfer area

Assembling the different thermal resistances into an equivalent thermal circuit gives the total resistance.

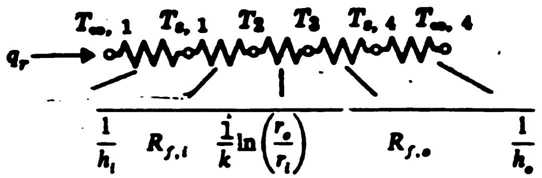


Figure 2: Equivalent thermal circuit

The overall heat transfer coefficient may be defined as the inverse of the total resistance.

$$1/UA = 1/U_cA_c = 1/U_hA_h \tag{4.14}$$

$$= 1/(n_ohA)_c + R''_{f,c}/(n_oA)_c + R_w + R''_{f,h}/(n_oA)_h + 1/(n_ohA)_h$$

where R_w = wall conduction resistance
R''_f = fouling factor
n_o = fin effectiveness
A = area
h = convection coefficient
$(\ )_h$ = hot fluid
$(\ )_c$ = cold fluid

4.2.1 Fouling

The fouling factor, R_f, is a thermal resistance that results from fluid impurities, rust formation, and other reactions between the fluid and the wall material. This fouling, which results from normal heat exchanger operation, can greatly increase the resistance to heat transfer between the fluids. The value of R_f depends on the operating temperature, fluid velocity, and length of service of the heat exchanger.

Table 1: Representative fouling factors [9]

FLUID	R_f'' (m^2*K/W)
Seawater and treated boiler feedwater (below 50°C)	0.0001
Seawater and treated boiler feedwater (above 50°C)	0.0002
River water (below 50°C)	0.0002-0.0001
Fuel oil	0.0009
Refrigerating liquids	0.0002
Steam (nonoil bearing)	0.0001

4.2.2 Finned Surfaces

For the case of a finned surface, the fin effectiveness n_o is defined to be the ratio of the fin heat transfer rate to the heat transfer rate that would exist without the fin. The effectiveness can be calculated as

$$n_o = 1 - (A_f/A)(1 - n_f) \qquad (4.15)$$

where n_f = fin efficiency
A_f = fin area
A = total surface area

For straight fins, such as rectangular, triangular and parabolic fins, as seen in figure 3 [10], the fin efficiency n_f is

$$n_f = \tanh(mL)/mL \qquad (4.16)$$

where $m = (2h/kt)^{1/2}$
L = length of fin
k = thermal conductivity of fin
h = convective coefficient
t = fin thickness

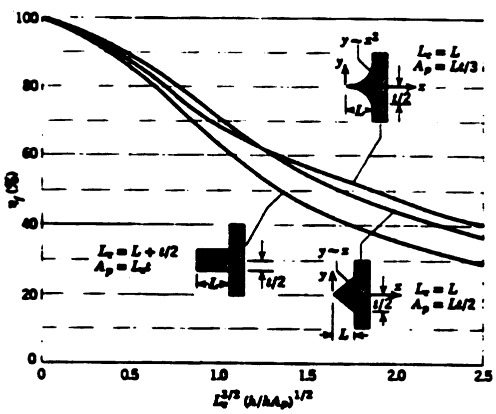


Figure 3: Efficiency of straight fins [10] (rectangular, triangular, and parabolic profiles)

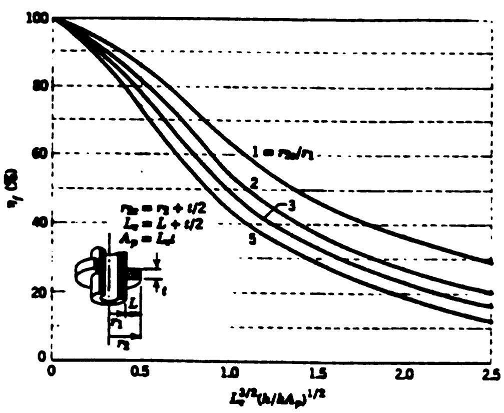


Figure 4: Efficiency of annular fins of rectangular profile

For annular fins, as in figure 4 [10], the efficiency is

$$n_f = (PkA_c/h)^{1/2}(2\ (r_2{}^2-r_1{}^2))^{-1} \qquad (4.17)$$

4.2.3 Unfinned Surfaces

For the case of an unfinned surface, the fin effectiveness is 1 and equation (4.14) reduces to

$$1/UA = 1/(hA)_c + R_w + 1/(hA)_h \qquad (4.18)$$

Fouling effects have been neglected.

For a cylindrical wall, the wall conduction can be calculated to be

$$R_w = \ln(D_o/D_i)/2\ kL \qquad (4.19)$$

where k = thermal conductivity of the fluid
D_o = outside diameter of pipe
D_i = inside diameter of pipe
L = length of pipe

For many heat transfer design problems, the cross-sectional dimensions of a heat exchanger are known, but the length is unknown. For this problem, equation (4.18) can be rewritten as

$$1/U = D[1/(hD)_c + \ln(D_o/D_i)/2k + 1/(hD)_h] \qquad (4.20)$$

Equation (4.20) shows that all that is necessary for calculation of the overall heat transfer coefficient is a knowledge of the geometry (diameters) of the heat exchanger, fluid properties and the convective coefficients of the hot and cold fluids. To estimate the convective coefficient, the correlations require information on the flow geometry, velocity, and fluid properties.

V. METHOD OF SOLUTION

5.1 EFFECTIVENESS - NTU CALCULATIONS

A heat exchanger has six main operating conditions: the inlet temperature, exit temperature, and mass flow rate for each fluid stream. These six operating conditions, along with the overall heat transfer coefficient and the heat exchanger area, make up the variables that are important to the heat exchanger analysis and design program.

Consider the case of any one of the six operating conditions being unknown. This condition can be directly solved for using the energy balance in equation (4.6). This energy balance consists of only the six operating conditions and the fluid property c_p. It is therefore a simple calculation to solve for one unknown condition. Subroutines ENERGY, RATING, and RATES solve for unknown inlet temperature, outlet temperature, and mass flowrate, respectively in TSCAD.

ENERGY solves the simplest problem, that of unknown inlet temperature. The heat transfer rate, q, can be calculated using the data from the known fluid.

$$q = C(T_{out}-T_{in}) \tag{5.1}$$

The heat transfer rate q is then used to solve for the unknown inlet temperature.

$$T_{in} = T_{out} - q/C \tag{5.2}$$

RATING solves a similar problem, that of unknown outlet temperature for one fluid. This routine uses the known heat capacity data, C_c and C_h, along with the overall heat transfer coefficient U, and the area A to calculate the NTU for the problem.

$$NTU = UA/C_{min} \tag{5.3}$$

This information is then used to calculate the effectiveness of the heat exchanger.

$$E = f(NTU, C_{min}/C_{max}) \tag{5.4}$$

The maximum heat transfer rate q_{max} is a function of inlet temperatures and heat capacity, it is therefore known for this problem. The actual heat transfer rate q can be calculated from the effectiveness E and q_{max}.

$$q = Eq_{max} \tag{5.5}$$

The outlet temperature is then calculated from the heat transfer rate, q.

$$T_{out} = T_{in} - q/C \tag{5.6}$$

The reason that RATING is not identical to ENERGY is that, in addition to solving for one exit temperature, it also is capable of solving for the exit temperature of both fluids. RATING was designed to solve this more detailed problem.

Subroutine RATES can be used to solve for the case of one unknown mass flow rate. For this case, the algorithm begins by calculating the ratio of temperature differences, which is known. It then uses the known temperatures to calculate the specific heat of each fluid. Knowing the temperatures and specific heats, the unknown mass flow rate can be found.

$$m_c = \frac{m_h (c_p)_h (T_o - T_i)_h}{(c_p)_h (T_o - T_i)_c} \qquad (5.7)$$

Each of the subroutines described above can also be used in conjunction with one or more other subroutines to solve for more than one unknown variable. The methods described above are used to solve for only one unknown operating condition.

For the case of two of the operating conditions being unknown, such as both outlet temperatures, an outlet temperature and a mass flowrate, or both mass flowrates, an iterative procedure is used to calculate the unknowns. These situations were solved using RATING, subroutines ONE and TWO, and RATES, respectively.

RATING can calculate outlet temperatures when both are unknown. The procedure is identical to that described above, when just one outlet temperature is unknown. The effectiveness - NTU method is used to find the heat transfer rate q. This is then used to calculate the outlet temperatures for each fluid stream, given the known inlet temperatures and mass flow rates. Since the outlet temperatures are unknown, some iteration is required between the specific heat and the outlet temperature for each fluid.

Subroutine ONE uses an iterative approach to find the mass flow rate and outlet temperature when both are unknown for the same fluid. The heat capacity C and the inlet and outlet temperatures of the known fluid are first used to calculate the heat transfer rate q. As a starting point, the known heat capacity is assumed to be C_{min}. The resulting effectiveness is then the ratio of the known temperature difference to the maximum temperature difference. The known heat capacity C is used to calculate the NTU. A procedure then is begun which iterates on the ratio of heat capacities C_r

$$C_r = C_{min}/C_{max} \qquad (5.8)$$

and alternately calls the E-NTU function TRANS and checks to see if the NTU ever becomes less than that calculated with the known heat capacity.

$$NTU = UA/C_{known} \qquad (5.9)$$

If this happens, it means that the known heat capacity is the minimum. The C_r at this point is then used to calculate the unknown heat capacity.

$$C_{max}\ (\text{unknown}) = C_{min}/C_r \qquad (5.10)$$

If the given problem consists of an unknown mass flow rate for fluid A and unknown outlet temperature for fluid B, subroutine TWO is called. TWO iterates between the two unknown operating conditions until the solution converges.

The process is begun by calculating the known heat capacity, C_B. This is the product of the known mass flow rate of fluid B and the specific heat of B. This is assumed to be C_{min} to begin the iteration. Using U, A and this C_{min}, the NTU is calculated. Using this NTU, the effectiveness is also calculated. The unknown outlet temperature is then computed from the definition of the effectiveness. This outlet temperature is checked for convergence and the iteration is either finished or continued.

The procedure used to solve for two unknown mass flow rates is very similar to that outlined earlier for one unknown flow rate.

Subroutine RATES is again used. An iteration is performed by first calculating NTU, and then using the definition of NTU to find C_{min}. Convergence is checked, and the procedure is either terminated or restarted.

The other possible unknowns, besides the six operating conditions, are the overall heat transfer coefficient U, and the area, A. These are computed by using the operating conditions to calculate the heat transfer and the effectiveness, and then by calling the appropriate effectiveness - NTU function.

$$NTU = f(E, C_{min}/C_{max}) \qquad (5.11)$$

The known NTU is then used to calculate the unknown variable, U or A.

$$\begin{aligned} U &= NTU*C_{min}/A \\ A &= NTU*C_{min}/U \end{aligned} \qquad (5.12)$$

This is done in subroutine SIZING. The E-NTU functions are given in the functions EFF and TRNS.

For the case of both the overall heat transfer coefficient and the area unknown, the effectiveness - NTU method alone is not sufficient to solve the problem. This situation motivates the need for a method of calculating the overall heat transfer coefficient directly, which is described in the next section.

Using this method, TSCAD can solve for three unknown conditions and parameters, provided that two of the unknowns are the overall heat transfer coefficient U and the heat transfer area A. For these cases, subroutine OVERALL is called and the appropriate geometric information is requested. These data are used to calculate U. Subroutine

SIZING then computes the corresponding area, given U. With U and A thus known, the proper E-NTU routine, as described in this section, can be used to find the remaining unknown operating condition.

Each of the routines in TSCAD uses the fluid property information called by PROPS. All properties used are temperature dependent. For this reason, any routine that calculates a new inlet or outlet temperature also contains an iteration loop which computes the fluid properties at the new average temperature, and then recalculates the unknown temperatures.

5.2 Estimating the Overall Heat Transfer Coefficient

Three types of heat exchanger configurations can be analyzed using the program TSCAD: concentric tube, cross-flow, and compact heat exchangers. For these three heat exchanger types four unique kinds of flow exist. All three cases have a common condition of flow inside a cylindrical tube for one of the fluids. The other three types of flow are flow in an annulus for the concentric tube case, transverse flow over a cylinder in the cross-flow case, and flow over a finned tube bank for the fin-side flow in a compact heat exchanger.

The method of solution for the four types was similar. A correlation was found which expressed a dimensionless convective coefficient, such as the Nusselt number or the

Colburn factor, in terms of fluid properties, fluid dynamics and flow geometry.

$$Nu = f(\text{fluid properties}, \text{fluid dynamics}, \text{geometry}) = hD/k \tag{5.13}$$

The convective coefficient (h) was then calculated from the dimensionless coefficient (Nu) and used to calculate the overall heat transfer coefficient, U.

5.2.1 Internal Flow in a Circular Pipe

For the case of flow in a circular pipe, correlations have been found which relate the Nusselt number to the Reynold's number (fluid dynamics), and the Prandtl number (fluid properties). The Dittus-Boelter equation [1] is a version of this correlation which is appropriate for turbulent flows, and is given as

$$Nu = 0.023 Re^{0.8} Pr^{n} \tag{5.14}$$

where $n=0.4$ for heating
$n=0.3$ for cooling

The Dittus-Boelter equation has been confirmed experimentally for the range of conditions

$0.7<Pr<160$
$Re>10,000$
$L/D>10$

For laminar flows, the Nusselt number can be analytically determined for different boundary (wall) conditions.

$$Nu = 4.36 \quad \text{for constant heat flux} \tag{5.15}$$
$$Nu = 3.66 \quad \text{for constant wall temperature}$$

The analytic result for a convective boundary condition is not known exactly, but it is known that the solution for the convective boundary condition must be bounded at the upper end by the constant heat flux solution, and at the lower end by the isothermal solution. The Nusselt number for this condition was therefore hypothesized to be the average of the constant heat flux case and the isothermal wall case.

$$Nu = 4.01 \qquad (5.16)$$

for convective wall condition, laminar flow

Turbulent and laminar correlations thus defined, some relation valid in the transition region between laminar and turbulent flow was needed. The solution for this problem was to use a weighted average of the turbulent solution and the laminar solution. The transition region was assumed to begin at a critical Reynold's number of approximately 2300, and extend to about 10,000.

$$Re_{cr} \sim 2300 \qquad (5.17)$$

Denoting the weighting function W, the solution for the transition region was therefore

$$Nu = WNu_{turb} + (1-W)Nu_{lam} \qquad (5.18)$$

$$\text{where } W = \frac{(Re - 2300)}{(10000-2300)}$$

5.2.2 A Cylinder in Cross-Flow

The method of calculating the convective coefficient, h, for the outside of the crossflow condition is similar to that of the internal flow case. A single empirical correlation is used to calculate a Nusselt number from fluid dynamics, fluid properties, and geometric information. The correlation proposed by Churchill and Bernstein [2], which expresses the Nusselt number as a function of only the Reynold's number and the Prandtl number, covers the entire range of Reynold's numbers for which data is available.

$$Nu = 0.3 + \frac{(0.62 Re^{.5} Pr^{.33})[1+(Re/282{,}000)^{.625}]^{.8}}{[1+(0.4/Pr)^{.67}]^{.25}} \qquad (5.19)$$

5.2.3 Flow Inside an Annulus

The correlations for flow inside a circular tube and external flow over a cylinder are widely used and accepted. Experimental heat transfer data for flow inside an annulus is not as readily available. A theoretical correlation for the Nusselt number at the inner wall of an annulus was derived from plots of available data collected by Kays and Leung.

The data shown in table 2 were collected for an experimental condition of constant wall heat flux on the two sides of an annulus. The specific set of data used had a

Table 2: Heat transfer data for heating from the core tube of an annulus (D_i/D_o=0.50)
Data from Kays and Leung [4]

Pr \ Re ->	10^4	3×10^4	10^5	3×10^5	10^6
	Nusselt number				
0.5	24.6	52.0	130	310	835
0.7	30.9	66.0	166	400	1080
1	38.2	83.5	212	520	1420
3	66.8	152.0	402	1010	2870
10	106.0	260.0	715	1850	5400
30	153.0	386.0	1080	2850	8400
100	220.0	558.0	1600	4250	12600
1000	408.0	1040.0	3000	8000	24000

condition of an adiabatic boundary at the outer surface, similar to the assumption for a concentric tube heat exchanger, and a given heat flux at the inner surface. This data was presented and examined to attempt to find a functional dependence on temperature. It was hypothesized that the dependence should be a power law type of function, similar to the Dittus-Boelter equation.

$$Nu = C_1 Pr^m Re^n \qquad (5.20)$$

Using the Dittus-Boelter correlation as a starting point, the constant C_1 which produced the best fit to the data was found. Since the data was presented for a given Prandtl number and Reynold's number, the functional dependence on either parameter could be easily isolated. The process was begun by plotting Nusselt number versus Reynold's number for each given Prandtl number. The plots were all similar to figure 5, which is for the case of Pr=0.7.

Using the Macintosh software Cricket Graph, a curve fit was performed on these plots. The functional dependence for the curve fit was of the form

$$Nu = CRe^{n} \qquad (5.21)$$

The Prandtl number dependence is absorbed in the constant C in equation 5.21, since each plot was taken for a constant Pr. Table 3 shows the resulting values for C and n for each Prandtl number.

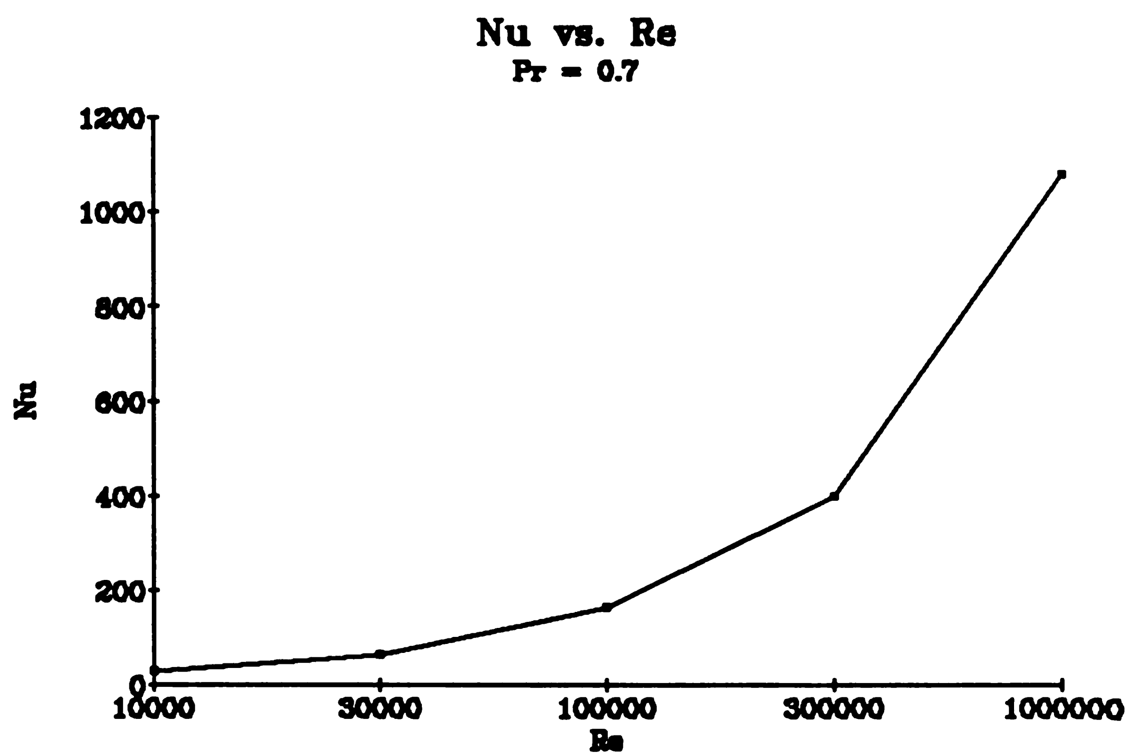


Figure 5: Nu vs. Re for Pr=0.7

Plotting the two variables, C and n, revealed a definite dependence on the Prandtl number for both.

Table 3: Variables for $Nu=CRe^n$

Pr	C	n
0.5	0.0198	0.7676
0.7	0.0234	0.7741
1.0	0.0258	0.7871
3.0	0.0341	0.8179
10.0	0.0398	0.8534
30.0	0.0498	0.8696
100.0	0.0652	0.8796
1000.0	0.1148	0.8851

Examination of the nearly straight line behavior of C between Pr=1.0 and Pr=100 suggests a power law relation between C and Pr, since the scale of the x-axis over this interval is similar to a logarithmic scale. To examine this theory, another curve fit was performed on the plot of C vs. Pr. The same functional behavior was assumed, and this led to the following values of C_1 and m for equation 5.20.

$$C = 0.0248Pr^{0.2174} \tag{5.22}$$

where $C_1 = 0.0248$
$m = 0.2174$

Next, the plot of n vs. Pr was examined more closely. The plot appears to approach a value of about 0.9 asymptotically at large Reynold's numbers. An exponential relationship with the Prandtl number was assumed.

$$n = B_1 + B_2\exp(-Pr) \tag{5.23}$$

Using an ordinary least squares method of parameter estimation, B_1 and B_2 were found. In order to keep the

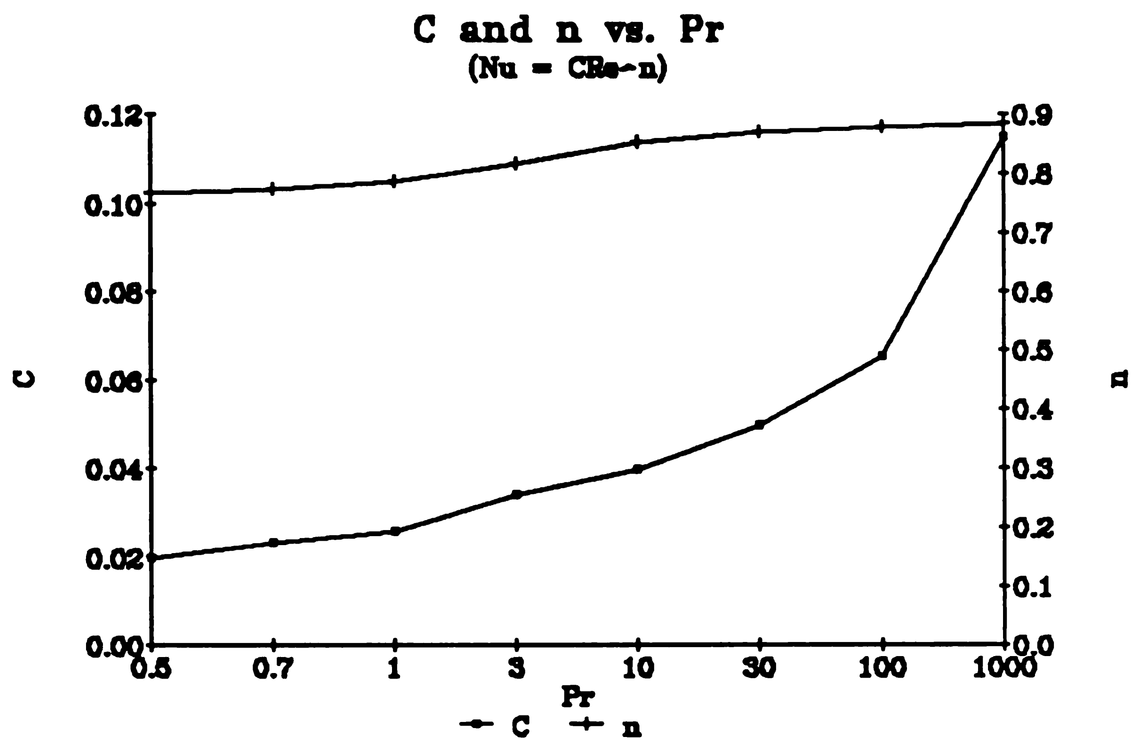


Figure 6: Dependence of C and n on Pr

value of the exponential to a number of significant digits that could be handled by the computer's memory, the Prandtl number was divided by 10, so that the new expression for n was

$$n = B_1 + B_2 \exp(-Pr/10) \qquad (5.24)$$

The estimated parameters were

$$B_1 = 0.8833$$
$$B_2 = -0.1095 \qquad (5.25)$$

The final correlation was then for the Nusselt number at the inner boundary of an annulus is

$$Nu = 0.0248 Pr^{.2174} Re^{n} \qquad (5.26)$$

$$n = 0.8833 - 0.1095 \exp(-Pr/10)$$

Figure 6 shows a comparison of the correlation in equation (5.26) with the data taken from Kays and Leung for Reynold's numbers ranging from 10^4 to 10^6.

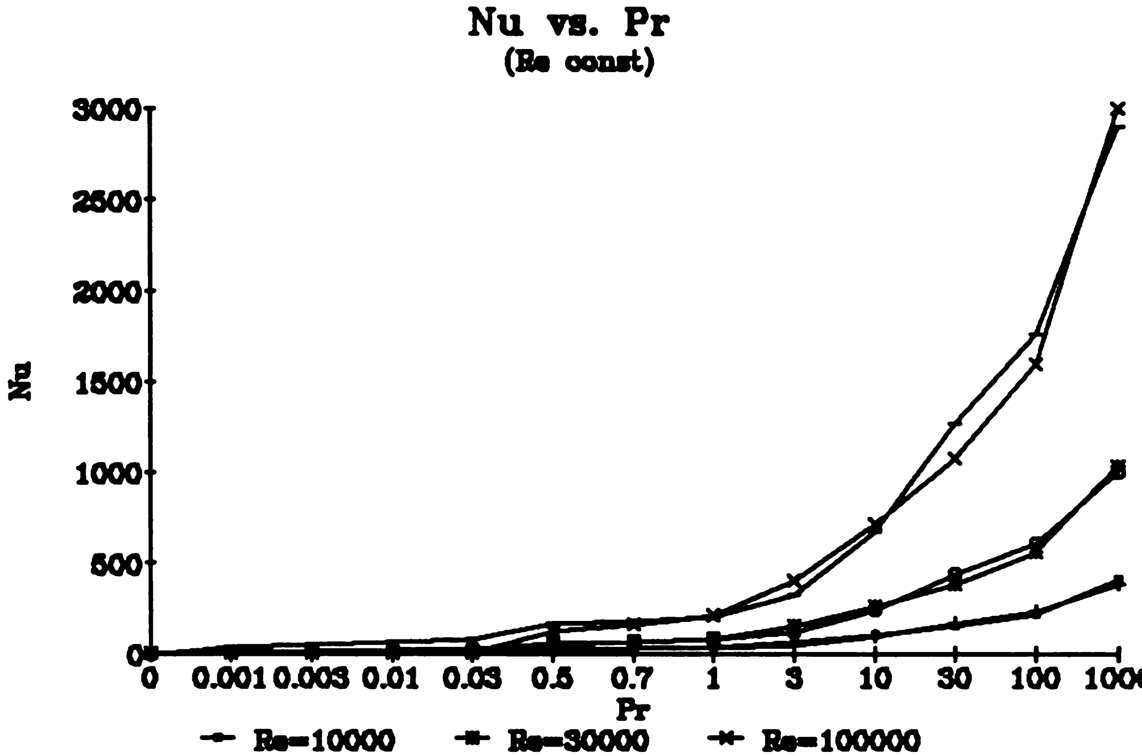
The accuracy of equation 5.26 was calculated by expressing the error between the correlation and the true value (from Kays & Leung data [4]) as a percentage of the true value. Using this method the accuracy of the correlation ranges from 11.9% to 15.3%. As a comparison, the Dittus-Boelter equation, which was used as a starting point, is accurate to between 18.8% and 35.8%.

5.2.4 Fin-Side Heat Transfer in Compact Heat Exchangers

The heat transfer data for compact heat exchangers was taken from the data in Compact Heat Exchangers, 1984, by Kays and London [3]. The Colburn factor

$$j_H = StPr^{.67} \qquad (5.27)$$

is plotted against the fin-side Reynolds number on a log-log plot for each compact heat exchanger considered. In each case it is simply a straight line plot.



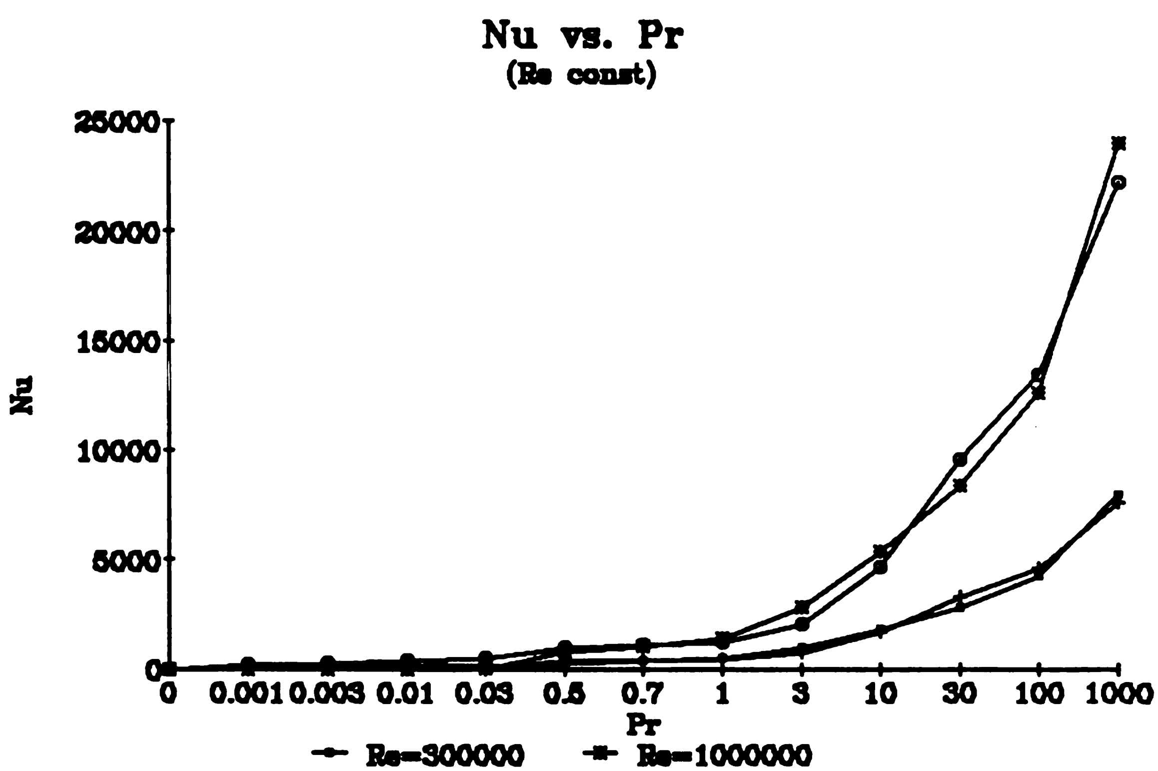
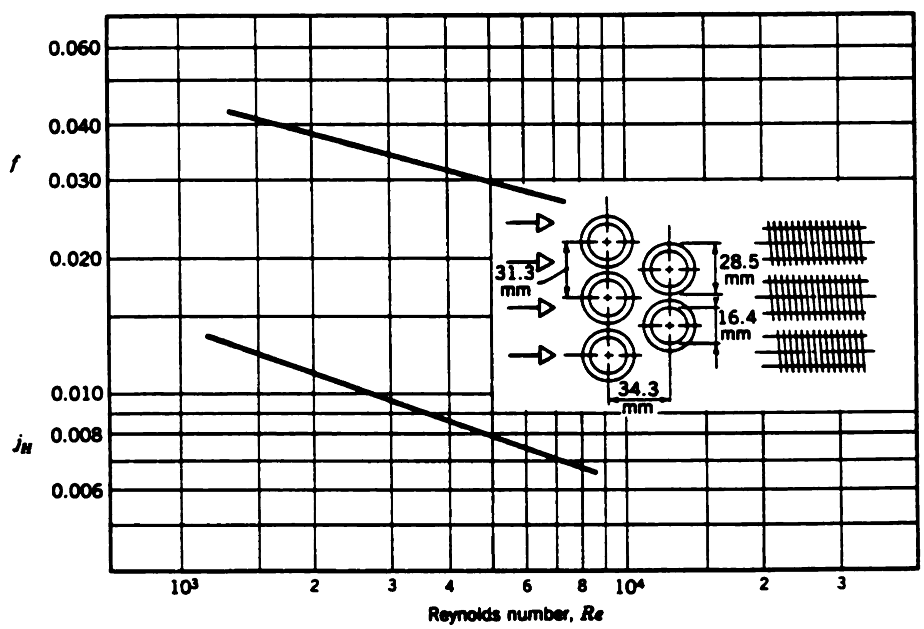


Figure 7: Comparison of Equation 5.26 with Kays and Leung Data [4]



Tube outside diameter, D_o = 16.4 mm
Fin pitch = 275 per meter
Flow passage hydraulic diameter, D_h = 6.68 mm
Fin thickness, t = 0.254 mm
Free-flow area/frontal area, σ = 0.449
Heat transfer area/total volume, α = 269 m^2/m^3
Fin area/total area, A_f/A = 0.830
Note: Minimum free-flow area is in spaces transverse to flow.

Figure 8: Plot of Colburn Factor (j_H) vs. Re
From Kays & London, 1984 [3]

These plots were used to derive analytic correlations for the Colburn factor.

$$j_H = C_1 Re^n \tag{5.28}$$

where C_1 = intercept of plot
n = slope of plot

C_1 and n are stored in a data statement with the rest of the compact heat exchanger information. With these, it is possible to input the calculated fin-side Reynold's number and find the corresponding Colburn factor.

5.3 Property Calculation

Access to the fluid properties of water, oil, steam, and air is necessary for analysis of the heat transfer process in a heat exchanger. The necessary properties include the density, specific heat, thermal conductivity, and viscosity. All of these properties vary with temperature, at least to some degree.

Initially, the program was set up to access a data file which contained the pertinent property information. The process of opening and closing the data file and of interpolating between values within the file cost valuable computer time and also occupied too much memory and disk space. These factors led to the decision to replace the data files with equations which express the properties as functions of temperature.

A process of parameter estimation was used to derive the property equations from the tables of property data versus temperature. The process was begun by plotting the data from the tables. The plots were then examined to try to predict the functional dependence. A least squares method was used to estimate the parameters, given the desired functional form and the tabulated data. The resulting function, using the estimated parameters, was then checked for accuracy. Standard deviation of under one percent was desired for adequate accuracy. If the plot was inaccurate, or accurate for only a portion of the range as was often the case, the process was repeated for the range of temperatures that was unsatisfactory.

The procedure is outlined in more detail as follows. A set of property data, such as that for air in Table 4, is to be replaced by equivalent functions of temperature. Figure 9 shows a plot of the tabulated values of the specific heat of air.

Examination of the plot reveals that the function must be of at least fourth order, because of the three changes in concavity. But, by breaking the data up into two parts, an equivalent function can be found which expresses the behavior in three separate equations, each of which is valid over approximately one third of the domain of temperatures. By solving the simpler set of second order polynomial problems, a more accurate equation can be found.

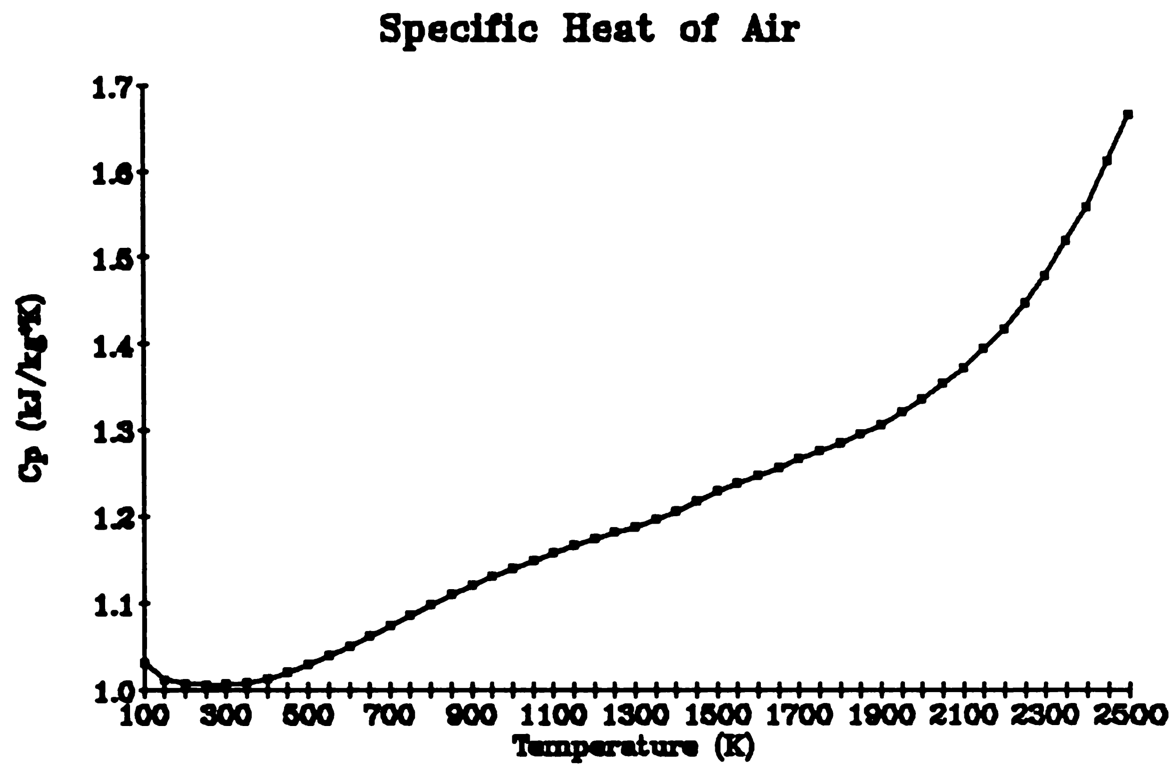


Figure 9: Specific heat of air

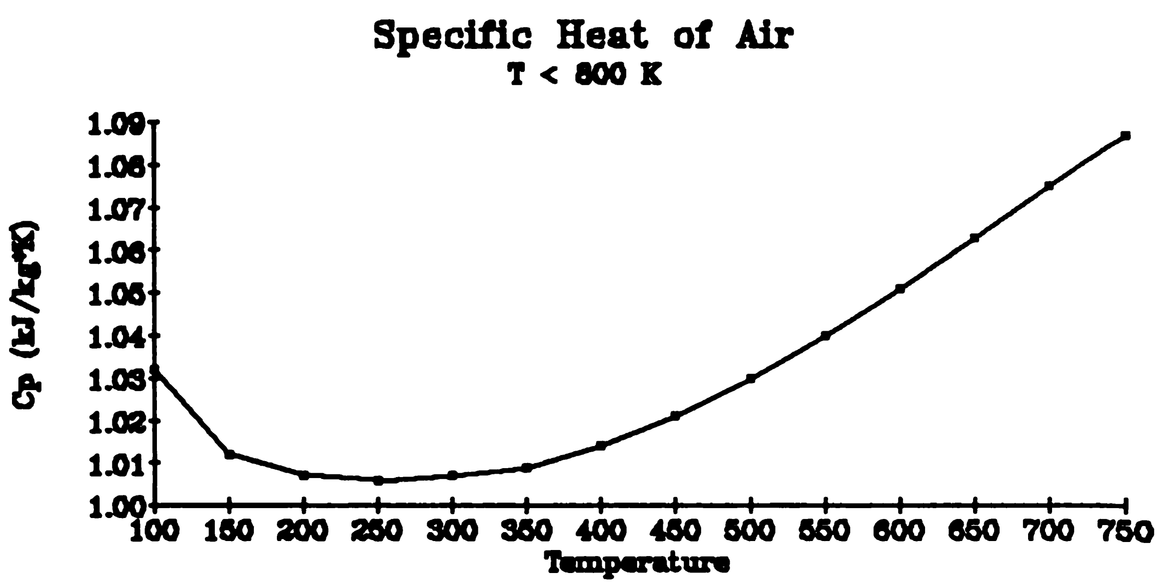


Figure 10: First third of temperature domain

Table 4: Properties of air

Temp [K]	Density [kg/cu.m]	Specific Heat [J/kg*K]	Viscosity [N*s/sq.m]	Thermal Conductivity [W/m*K]
100	3.5562	1032	0.00000711	0.00934
150	2.3364	1012	0.00001034	0.0138
200	1.7458	1007	0.00001325	0.0181
250	1.3947	1006	0.00001596	0.0223
300	1.1614	1007	0.00001846	0.0263
350	0.9950	1009	0.00002082	0.0300
400	0.8711	1014	0.00002301	0.0338
450	0.7740	1021	0.00002507	0.0373
500	0.6964	1030	0.00002701	0.0407
550	0.6329	1040	0.00002884	0.0439
600	0.5804	1051	0.00003058	0.0469
650	0.5356	1063	0.00003225	0.0497
700	0.4975	1075	0.00003388	0.0524
750	0.4643	1087	0.00003546	0.0549
800	0.4354	1099	0.00003698	0.0573
850	0.4097	1110	0.00003843	0.0596
900	0.3868	1121	0.00003981	0.0620
950	0.3666	1131	0.00004113	0.0643
1000	0.3482	1141	0.00004244	0.0667
1100	0.3166	1159	0.00004490	0.0715
1200	0.2902	1175	0.00004730	0.0763
1300	0.2679	1189	0.00004960	0.0820
1400	0.2488	1207	0.00005300	0.0910
1500	0.2322	1230	0.00005570	0.1000
1600	0.2177	1248	0.00005840	0.1060
1700	0.2049	1267	0.00006110	0.1130
1800	0.1935	1286	0.00006370	0.1200
1900	0.1833	1307	0.00006630	0.1280
2000	0.1741	1337	0.00006890	0.1370
2100	0.1658	1372	0.00007150	0.1470
2200	0.1582	1417	0.00007400	0.1600
2300	0.1513	1478	0.00007660	0.1750
2400	0.1448	1558	0.00007920	0.1960
2500	0.1389	1665	0.00008180	0.2220

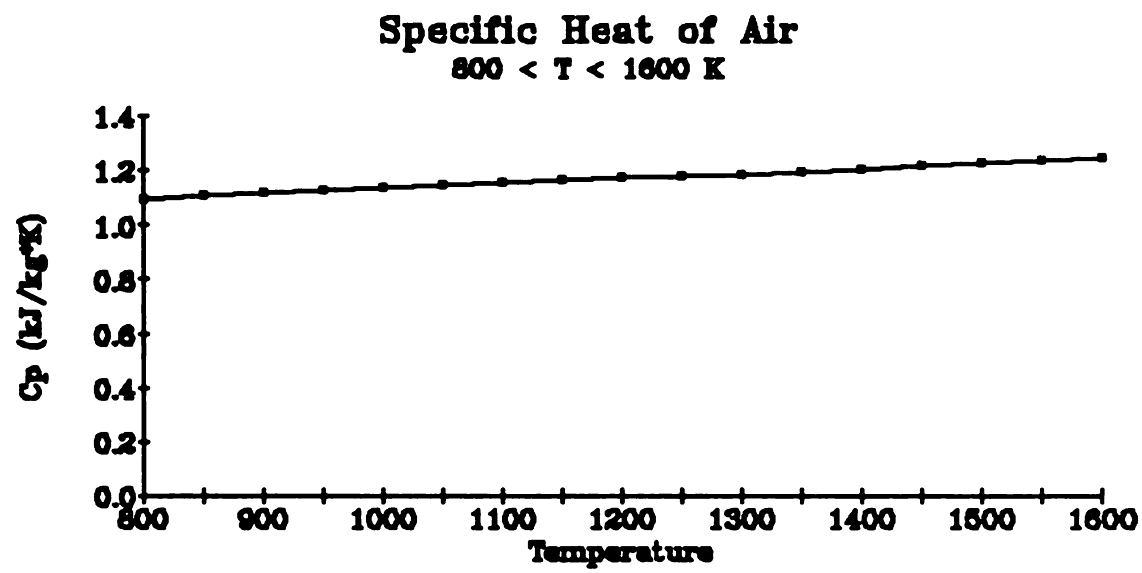


Figure 11: Middle third of temperature domain

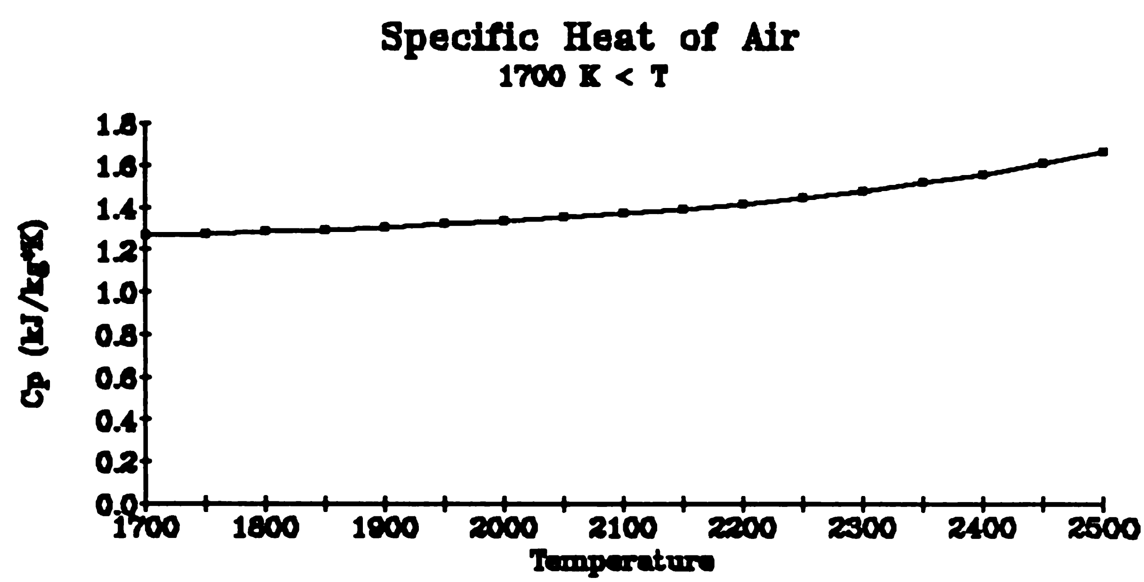


Figure 12: Final third of temperature domain

Once the type of function to use was determined, the problem became one of estimating the parameters in the equation. The method of ordinary least squares was used for the parameter estimation. This method begins by identifying the important parameters. With the function expressed as follows, the parameters B_1, B_2 and B_3 are to be determined.

$$c_p = B_1 + B_2T + B_3T^2 \qquad (5.29)$$

where c_p = specific heat
T = temperature

Having identified the parameters, the sensitivity of each was determined. The sensitivity, $\mathbf{X}$, is defined as the rate of change of the property with change in the parameter.

$$\mathbf{X}_i = dc_p/dB_i \qquad (5.30)$$

The sensitivity matrix has three component vectors, $\mathbf{X}_1$, $\mathbf{X}_2$ and $\mathbf{X}_3$. Expressing the specific heat data as a vector, and the corresponding temperatures as another vector, the property function can be expressed in matrix form.

$$\mathbf{c_p} = B_1 + B_2\mathbf{T} + B_3\mathbf{T}^T\mathbf{T} \qquad (5.31)$$

Or, equivalently, using indicial notation:

$$c_{pi} = B_1 + B_2T_i + B_3T_i^2 \qquad (5.32)$$

The sensitivity of B_1 is then the unity vector, each entry of which is one. The sensitivity of B_2 is the temperature vector, $\mathbf{T}$, and the sensitivity of B_3 is the temperature squared vector, $\mathbf{T}^T\mathbf{T}$.

With each of the sensitivity vectors thus defined, the i^{th} component of the sensitivity matrix is

$$X_i = [1 \; T_i \; T_i^2] \qquad (5.33)$$

The parameter vector B is then defined to be, in terms of the sensitivity matrix,

$$B = [X^T X]^{-1}[X_T Y] \qquad (5.34)$$

The computation of the $X^T X$ matrix and the $X^T Y$ matrix is easily achieved by use of spreadsheet software. The $X^T X$ matrix, is

$$X^T X = \begin{bmatrix} n & \mathrm{sum}(T_i) & \mathrm{sum}(T_i^2) \\ \mathrm{sum}(T_i) & \mathrm{sum}(T_i)\,\mathrm{sum}(T_i) & \mathrm{sum}(T)\,\mathrm{sum}(T_i^2)^2 \\ \mathrm{sum}(T_i^2) & \mathrm{sum}(T_i)\,\mathrm{sum}(T_i^2) & \mathrm{sum}(T_i^2)\,\mathrm{sum}(T_i^2)^2 \end{bmatrix} \qquad (5.35)$$

where n = number of data points

The corresponding $X^T Y$ matrix is

$$X^T Y = \begin{bmatrix} \mathrm{sum}(Y_i) \\ \mathrm{sum}(Y_i)\,\mathrm{sum}(T_i) \\ \mathrm{sum}(Y_i)\,\mathrm{sum}(T_i^2) \end{bmatrix} \qquad (5.36)$$

Each of these summations are carried out, followed by the appropriate multiplications.

The result of the parameter estimation for the first part of the temperature domain is

$$B_1 = 1040.0$$
$$B_2 = -0.217$$
$$B_3 = 3.83E-4$$

For the domain of temperatures

$$T < 800 \text{ K}$$

the corresponding specific heat equation is

$$c_p = 1.0400 - 0.000217*T + 3.83E-7*T^2 \qquad (5.37)$$

The standard deviation, calculated from the tabulated data, is 0.0046. This is approximately 0.44% of the mean value for the thermal conductivity over this range of temperatures.

The equations for the second and third portions of the domain, with the computed parameters included are

$$c_p = 951.0 + 0.1922*T - 4.83E-6*T^2 \qquad (5.38)$$
$$800 < T < 1700 \text{ K}$$

$$c_p = 2988.5 - 2.013*T + 5.916E-4*T^2 \qquad (5.39)$$
$$1700 \text{ K} < T$$

The accompanying standard deviations are 0.0027 and 0.0096, or expressed as percentages 0.23% and 0.68%.
Equations for the rest of the properties of air as well as the properties of water, steam and engine oil, were derived in a similar fashion.

Appendix B has the complete set of property data for the four types of fluids used in TSCAD, along with the corresponding plots of properties as a function of temperature. The equation for each property is given in appendix A, along with the standard deviation expressed as a percentage of the mean property value over the domain, to offer an estimate of the accuracy of each function.

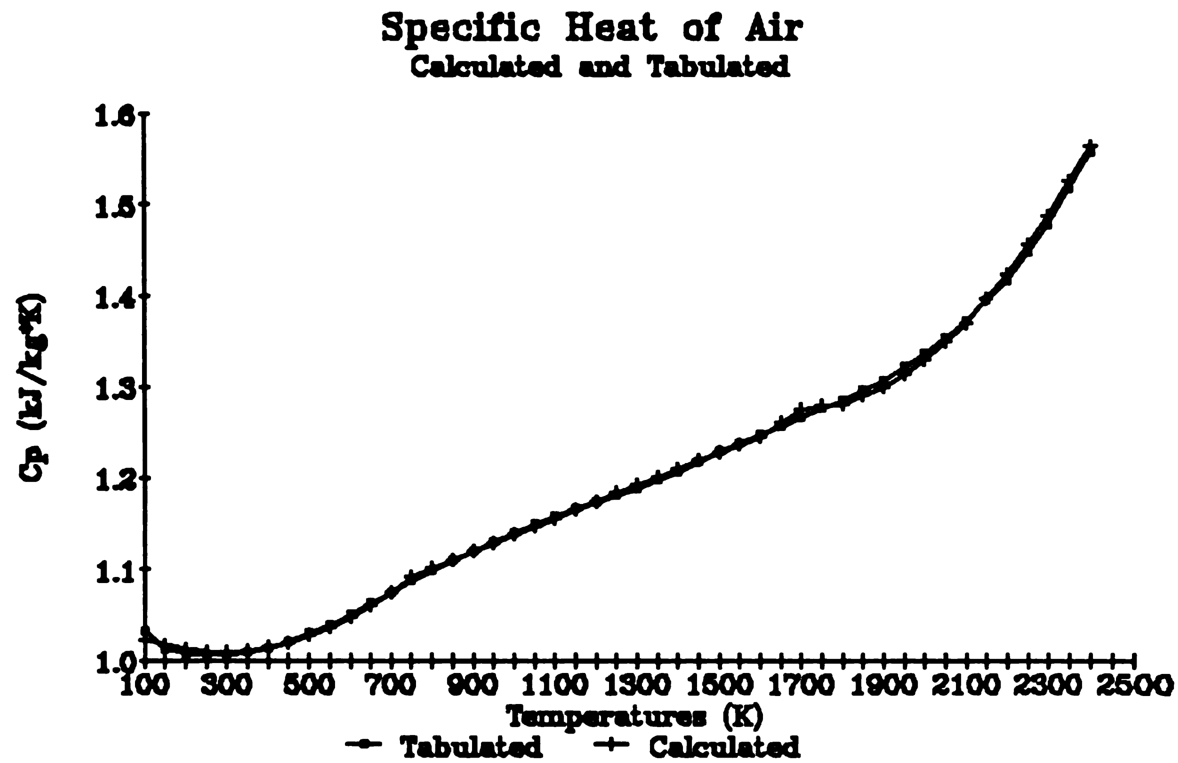


Figure 13: Comparison of calculated vs. tabulated c_p data

VI. EXPERIMENTAL VERIFICATION OF TSCAD PERFORMANCE

TSCAD is a computer program that was designed for use on personal computers. It requires about 43.5 kB of disk space in fortran form, and about 92 kB of disk space in executable form. This is about 3.5% and 7.5%, respectively, of the space available on a 5.25 inch, 1.2 MB high density diskette. The executable form of TSCAD does not require a fortran driver, and can be used on any IBM-compatible PC.

The program was tested on an IBM-compatible PC, running on a 286 processor. With the PC running at 12 MHz, the executable form of TSCAD can produce its results almost instantaneously, its computing time was estimated to be under one second. The fortran file can be used directly by utilizing an interpreter such as WATFOR77. In this form it takes longer to run, since the file has to be interpreted each time the program is used. Once the program is interpreted, the run time is comparable to the executable form of the program. Presently, the data for each test case must be entered by the user via the computer keyboard. This input time far exceeds any computing time required by the program.

This suggests a potentially valuable future improvement: to adapt the program to receive its data from data files, so that changes in one or two variables could be made without retyping the unchanged data.

The experimental verification of the results obtained by TSCAD was an ongoing process occurring simultaneously with the development of the program algorithms. The data set to be presented as evidence of the program's ability was collected on February 20, 1990 and is for a double pipe heat exchanger in both parallel flow and counterflow configuration.

The heat exchanger operating conditions were recorded for each trial. These operating conditions were then used to calculate the overall heat transfer coefficient U three different ways. The log mean temperature difference was used to find the "true" heat transfer coefficient, since this method is exact for a double pipe heat exchanger. Next, TSCAD was used to calculate U by the effectiveness-NTU method and by using the correlations. By inputting the correct area and an unknown overall heat transfer coefficient, TSCAD calculates U using effectiveness-NTU. Finally, the heat transfer correlations within TSCAD were used to calculate the overall heat transfer coefficient. This latter approach is achieved by inputting both an unknown area and an unknown U into TSCAD.

Several factors had to be considered for this particular experimental setup. First, since D/L, the ratio of pipe diameter to pipe length, is relatively large (0.015), entrance effects may have a real impact on the heat transfer. According to Burmeister [15], the average Nusselt number for L/D=70 is increased by 10% over the fully-developed Nusselt number. Secondly, due to the constraints on the water supply, it is difficult to attain a Reynold's number in the annulus which is outside of the transition region ($2300 < Re < 10000$). These two constraints are recognized in the results.

The experimental uncertainty was estimated to be between 5% and 10%, based on unsteadiness in the fluid flow rates and in the inlet fluid temperatures. Four different combinations of mass flow rates were tested in both the parallel flow and counterflow configuration. A total of sixteen sets of data were then available. The operating conditions for each trial are shown in Table 5. The first eight sets of data shown correspond to a counterflow configuration, while the last eight sets are for parallel flow.

The results, shown in Table 6, give the measured heat transfer coefficient U, utilizing the log mean temperature difference method, as well as the results generated by TSCAD. These include results from the effectiveness-NTU method and results from the correlations for heat transfer from an annulus and heat transfer from a circular pipe. Defining an error to be the difference between the TSCAD values and the actual (log mean temperature difference) values, Table 6 shows the raw errors between each TSCAD method and the measured solution for each of the sixteen trials, as well as the average raw error and the average absolute error. The results show that the effectiveness-NTU method is accurate in most cases to within 1% of measured values for the overall heat transfer coefficient.

The correlation results reflect the 10% increase in the Nusselt number due to the entry length effects. Also, it was realized that the linear lever rule hypothesis does not reflect the behavior of the Nusselt number in the transition region accurately. Instead, the turbulent correlation for heat transfer from an annulus was found to perform better over the whole range of Reynold's numbers. Therefore, this correlation was used for all annular flow. With these two points thus compensated for, the correlation predicted the overall heat transfer coefficient and the area with errors generally less than 10%, and averaging 6.3%.

Figure 14 shows a comparison of the measured overall heat transfer coefficient with those calculated by TSCAD, both by the E/NTU method and by use of the correlations for flow through concentric tubes. The measured results are plotted on both the X and the Y axis, these data therefore produce a straight line. The TSCAD results are shown distributed about this line.

The results show that TSCAD's accuracy is well within the experimental uncertainty for the double pipe heat exchanger data. As an extension of the program verification, the case of fully turbulent annular flow and the effect of a thermal entry length on the heat transfer should be examined more thoroughly.

Table 5: Operating Conditions for Verification

T_{ci}	T_{co}	M_c	T_{hi}	T_{ho}	M_h
16.59	23.31	0.1416	58.88	52.57	0.1416
10.25	16.59	0.1416	52.57	45.66	0.1416
16.07	22.02	0.1888	59.27	53.59	0.1888
10.06	16.07	0.1888	53.59	47.08	0.1888
15.49	21.28	0.2360	59.34	54.00	0.2360
10.06	15.49	0.2360	54.00	48.18	0.2360
15.23	20.66	0.2832	59.08	54.13	0.2832
10.06	15.23	0.2832	54.13	48.43	0.2832
11.31	17.23	0.1416	53.74	47.57	0.1416
17.23	21.44	0.1416	47.57	42.77	0.1416
11.04	16.33	0.1888	54.15	48.63	0.1888
16.33	20.28	0.1888	48.63	43.98	0.1888
10.71	16.07	0.2360	56.56	51.25	0.2360
16.07	20.23	0.2360	51.25	46.25	0.2360
10.52	15.70	0.2832	57.27	52.13	0.2832
15.70	20.02	0.2832	52.13	47.27	0.2832

Table 6: TSCAD Program Verification Data

Log Mean Temp Diff	TSCAD E / NTU		TSCAD Correlation		TSCAD Corr.	ACTUAL
U	U	ERROR	U	ERROR	AREA	AREA
2349.4	2363.3	0.6%	2119.5	-9.8%	0.0529	0.0474
2223.6	2206.6	-0.8%	1989.6	-10.5%	0.0526	0.0474
2654.4	2663.3	0.3%	2676.9	0.8%	0.0472	0.0474
2691.9	2674.2	-0.7%	2528.8	-6.1%	0.0501	0.0474
3153.2	3171.5	0.6%	3204.4	1.6%	0.0469	0.0474
2957.3	2942.3	-0.5%	3042.4	2.9%	0.0458	0.0474
3514.3	3535.6	0.6%	3708.5	5.5%	0.0452	0.0474
3351.0	3327.7	-0.7%	3532.3	5.4%	0.0447	0.0474
2055.7	2048.1	-0.4%	2013.0	-2.1%	0.0478	0.0474
2059.3	2033.3	-1.3%	2003.3	-2.7%	0.0477	0.0474
2358.0	2350.4	-0.3%	2549.3	8.1%	0.0434	0.0474
2371.8	2339.5	-1.4%	2535.8	6.9%	0.0434	0.0474
2776.5	2778.8	0.1%	3092.8	11.4%	0.0423	0.0474
2856.0	2813.4	-1.5%	3077.6	7.8%	0.0431	0.0474
3134.8	3136.9	0.1%	3598.7	14.8%	0.0411	0.0474
3418.6	3387.2	-0.9%	3582.6	4.8%	0.0445	0.0474
average error:		-0.4%		2.4%		
average absolute error:		0.7%		6.3%		

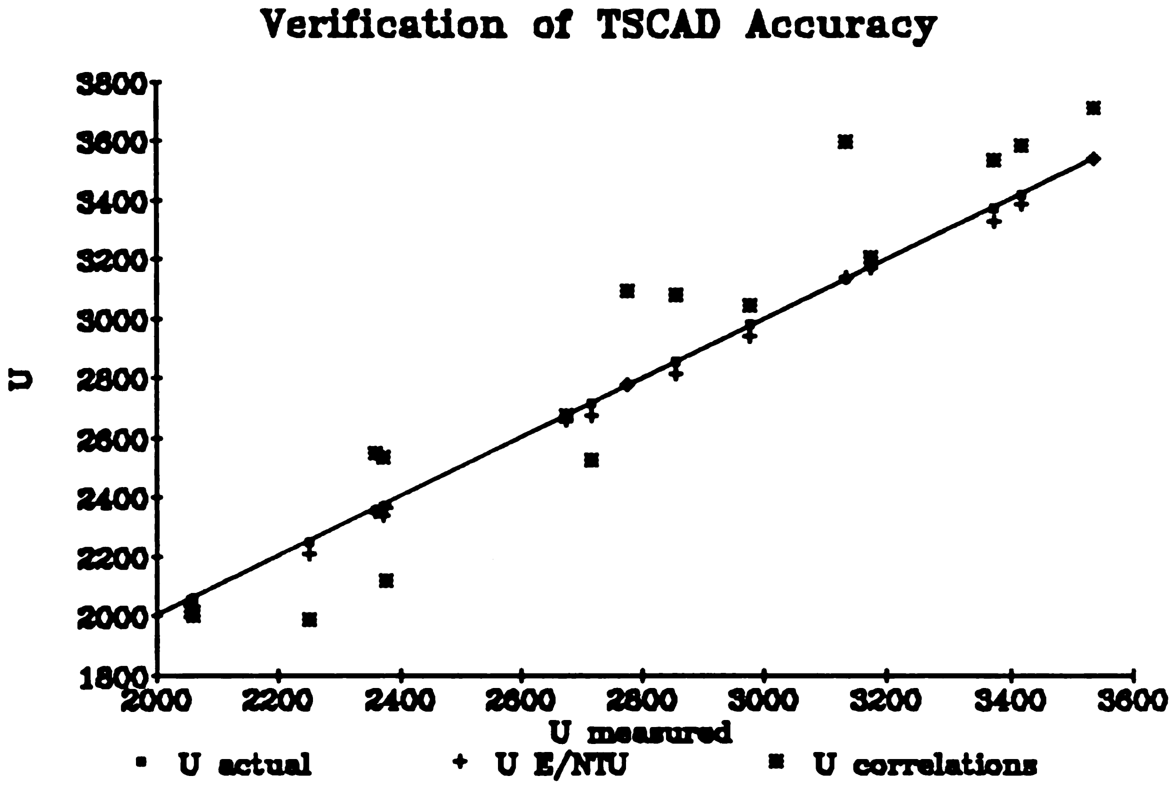


Figure 14: Comparison of calculated U vs. measured U

VII. CLOSURE

The work presented here, which can be used to accurately solve for up to three unknowns in the data set, was intended to offer an easy and fast method of analyzing and designing heat exchangers by making use of the advantages that computers offer.

One of the big advantages of the program is speed. Iterations are performed quickly and accurately. Evaluation of property data was sped up considerably by replacing data files with equations that calculate the property as a function of temperature within the program. This computation time is considerably less than that spent in opening, closing and reading data files.

The program also offers the option of calculating the overall heat transfer coefficient using geometric data and empirical correlations. This option makes use of a correlation that was derived by performing a curve-fit on a set of published experimental data. The work done in deriving a correlation to describe the Nusselt number as a function of the Reynold's number for flow in an annulus is important in that it is not a subject for which a commonly accepted solution is available.

The work has remained very compact, since the full set of required data is contained within the single program. This includes all the required compact heat exchanger data, the thermal conductivity data for different types of pipe material, and the fluid properties mentioned earlier.

The work as presented does leave considerable room for additional improvement, however. One possible improvement would be expanding the capability of the system to include the ability to analyze shell-and-tube and other types of heat exchangers.

Another area that the system could be expanded into is performing flow characteristic analysis. Presently, the program is concerned only with the thermal characteristics of heat exchangers.

The addition of an optimization option would greatly enhance the ability of TSCAD. This would require considerable effort, but the potential benefits are great.

While the addition of the aforementioned improvements could improve TSCAD, the system, as presented, can be a valuable tool for the engineer concerned with heat exchanger design and analysis. TSCAD can calculate the solution of any well-posed heat exchanger problem within its domain quickly and accurately, and it will hopefully be a valuable addition to the area of computer-aided design of thermal systems.

APPENDIX A: Property Functions

The property functions given here were all derived by the methods given in chapter 5.3. The standard deviation, s, for each function is the error between the calculated function and the tabulated values, taken from [5], [6], [7], and [8]. This error was calculated as the standard deviation of the function divided by the average value of the property over the given domain.

The functions for the density, viscosity, specific heat and thermal conductivity are given below. Following the functions are tables which compare the values of the properties over a range of temperatures.

Density

water:

T < 500 K

$$p = 0.985 + 0.0215*\exp[(T-275)/100] \qquad s = 0.87\%$$

500 <T< 600 K

$$p = 1.00 + 0.215*\exp[(T-510)/100] \qquad s = 0.30\%$$

600 < T

$$p = 212 - 0.688*T + 0.000564*T^2 \qquad s = 2.04\%$$

air:

$$p = 350/T \qquad s = 1.62\%$$

steam:

$$p = 1.212 - 0.00212*T + 1.18E-6*T^2 \qquad s = 1.66\%$$

oil:

$$p = 1060 - 0.583*T - 0.0000125*T^2 \qquad s = 0.04\%$$

Thermal Conductivity

water:

$$k = -0.3386 + 0.004980*T - 6.03E-6*T^2 \qquad s = 1.18\%$$

air:

T < 1500 K

$$k = 0.003556 + 7.827E-5*T - 1.34E-8*T^2 \qquad s = 1.37\%$$

1500 < T

$$k = 0.2945 - 0.000272*T + 9.662E-8*T^2 \qquad s = 3.46\%$$

steam:

$$k = -0.002674 + 6.635E-5*T + 1.382E-8*T^2 \qquad s = 0.19\%$$

oil:

$$k = 0.1799 - 0.000137*T + 5.66E-8*T^2 \qquad s = 0.70\%$$

<u>Viscosity</u>

water:

T < 330 K

$M = -3344478/T^4 + 9.435E{+}11/T^6$ s = 3.44%

330 < T

$M = 8082231/T^4 - 3.22E{+}11/T^6$ s = 4.46%

air:

T < 800 K

$M = 1.162E{-}6 + 6.499E{-}8{*}T - 2.61E{-}11{*}T^2$ s = 0.74%

800 < T

$M = 1.541E{-}5 + 2.714E{-}8{*}T - 2.21E{-}13{*}T^2$ s = 1.40%

steam:

$M = -9.87E{-}7 + 3.606E{-}8{*}T + 1.382E{-}8{*}T^2$ s = 0.07%

oil:

T< 300 K

M = 1.175E+10*exp(-8T/100) s = 3.80%

300 < T < 340

M = 67594510*exp(-6.25T/100) s = 8.61%

340 < T < 380

M = 9261.90*exp(-3.55T/100) s = 8.78%

380 < T

M = 106.54*exp(-2.35T/100) s = 1.23%

Specific Heat

water:

$T < 430$ K

$$c_p = 5310.79 - 7.141*T + 0.01128*T^2 \qquad s = 0.08\%$$

$430 < T < 580$

$$c_p = 22{,}291 - 79.48*T + 0.08832*T^2 \qquad s = 0.96\%$$

$580 < T < 630$

$$c_p = 1{,}490{,}286 - 5004*T + 4.220*T^2 \qquad s = 3.64\%$$

$630 < T$

$$c_p = 441{,}214{,}463 - 1{,}386{,}078*T + 1{,}088*T^2 \qquad s = 0.12\%$$

air:

$T < 800$ K

$$c_p = 1040.0 - 0.217*T + 3.83E-4*T^2 \qquad s = 0.44\%$$

$800 < T\ 1700$

$$c_p = 951.0 + 0.1922*T - 4.83E-6*T^2 \qquad s = 0.23\%$$

$1700 < T$

$$c_p = 2988.5 - 2.013*T + 5.916E-4*T^2 \qquad s = 0.68\%$$

steam:

$T < 550$ K

$$c_p = 4413 - 10.48*T + 0.01126*T^2 \qquad s = 0.49\%$$

$550 < T$

$$c_p = 1762 + 0.2967*T + 0.0002381*T^2 \qquad s = 0.04\%$$

oil:

$$c_p = 789.56 + 3.314*T + 0.001388*T^2 \qquad s = 0.05\%$$

APPENDIX B: GRAPHS OF PROPERTY FUNCTIONS

The fluid properties density, kinematic viscosity, specific heat, and thermal conductivity, as well as the Prandtl number, are plotted versus temperature. The tabulated values are compared with those values calculated by the equations given in appendix A.

The accuracy of the calculated values is given with each figure. The error is calculated as a percentage of the true value. The average error is then s, the value which is given at the top of each chart.

Properties of Water

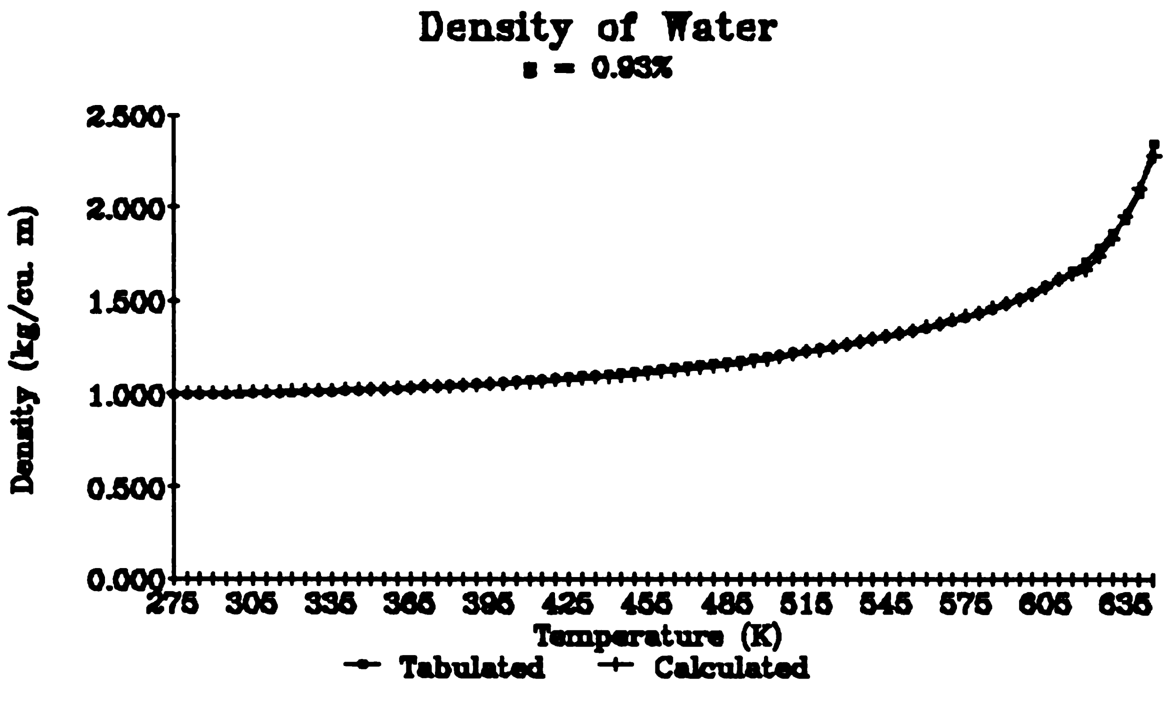


Figure B1: Density of water

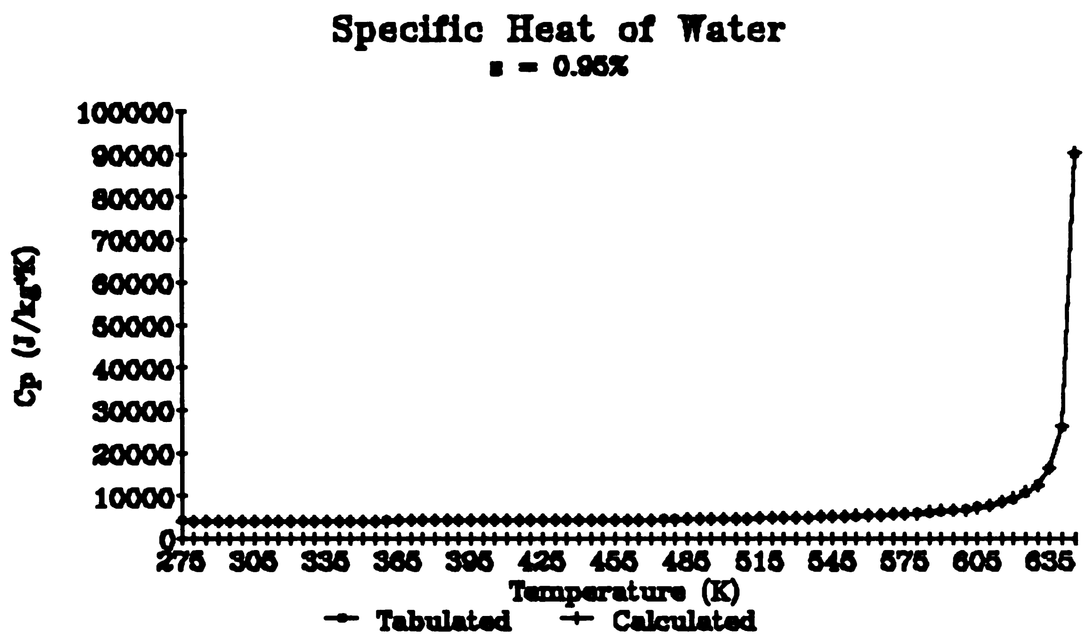


Figure B2: Specific heat of water

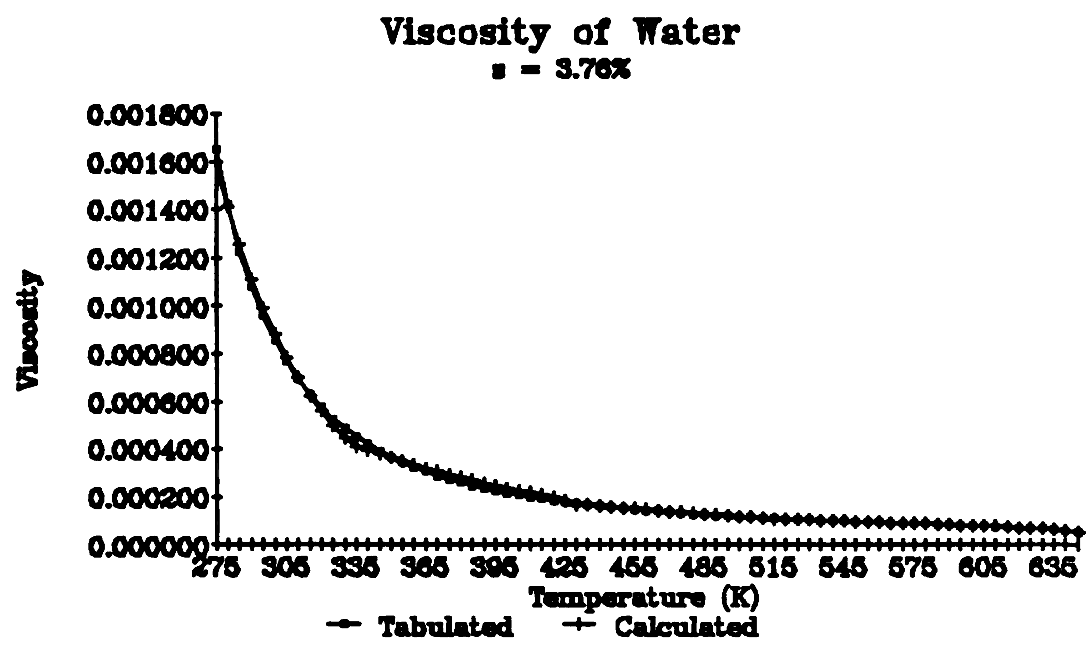


Figure B3: Viscosity of water

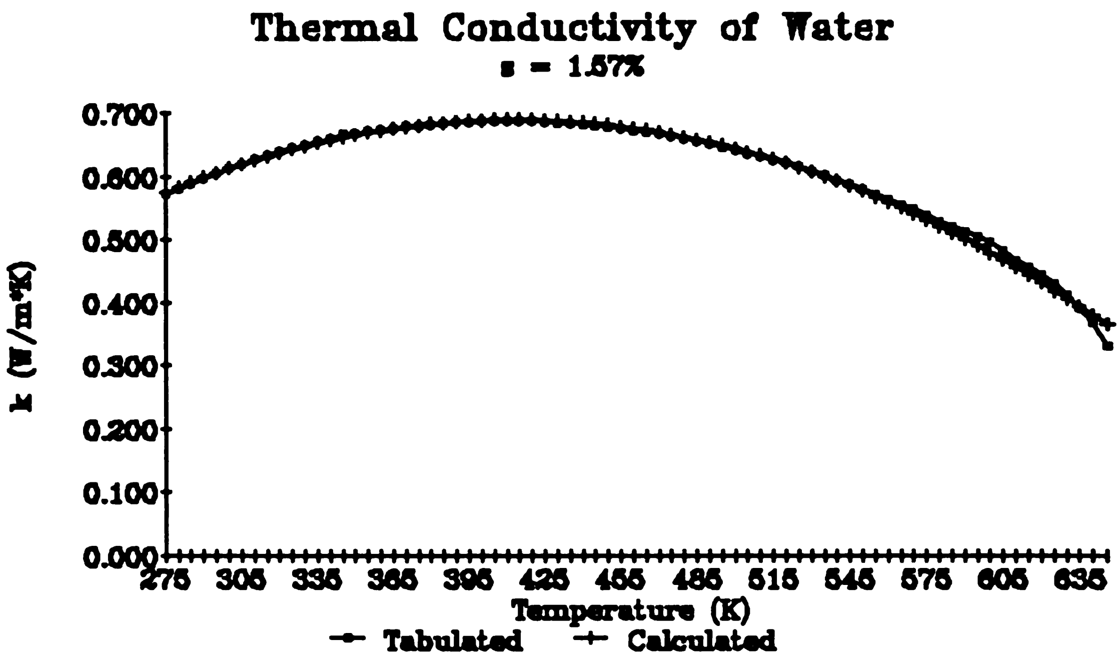


Figure B4: Thermal Conductivity of Water

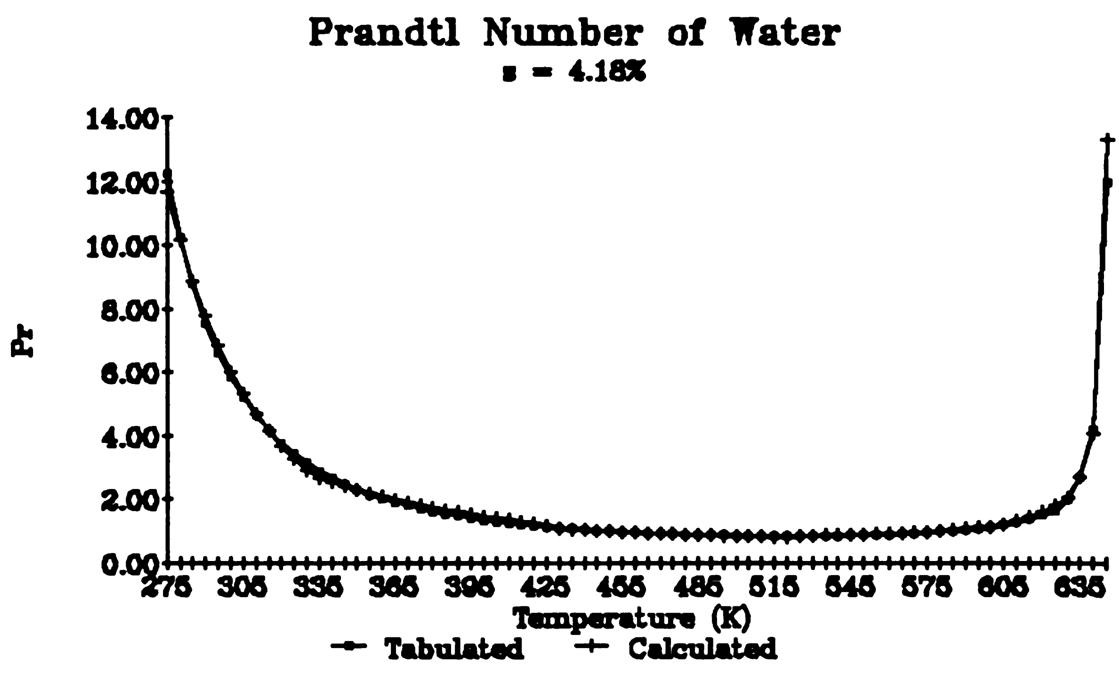


Figure B5: Prandtl Number of Water

Table B1: Properties of Water

Temperature	Density Tab.	Density Eqn.	Specific Heat Tab.	Specific Heat Eqn.
275	1.000	1.007	4211	4200
280	1.000	1.008	4198	4195
285	1.000	1.009	4189	4192
290	1.001	1.010	4184	4188
295	1.002	1.012	4181	4186
300	1.003	1.013	4179	4183
305	1.005	1.014	4178	4182
310	1.007	1.016	4178	4181
315	1.009	1.018	4179	4180
320	1.011	1.019	4180	4181
325	1.013	1.021	4182	4181
330	1.016	1.023	4184	4182
335	1.018	1.025	4186	4184
340	1.021	1.027	4188	4187
345	1.024	1.029	4191	4189
350	1.027	1.031	4195	4193
355	1.030	1.033	4199	4197
360	1.034	1.036	4203	4202
365	1.038	1.038	4209	4207
370	1.041	1.041	4214	4213
375	1.045	1.044	4220	4219
380	1.049	1.047	4226	4226
385	1.053	1.050	4232	4233
390	1.058	1.053	4239	4241
395	1.063	1.057	4248	4250
400	1.067	1.061	4256	4259
405	1.072	1.065	4267	4269
410	1.077	1.069	4278	4279
415	1.083	1.073	4290	4290
420	1.088	1.077	4302	4301
425	1.094	1.082	4317	4313
430	1.099	1.087	4331	4325
435	1.105	1.092	4346	4373
440	1.110	1.098	4360	4420
445	1.117	1.103	4380	4416
450	1.123	1.109	4400	4411
455	1.130	1.116	4420	4416
460	1.137	1.122	4440	4420
465	1.145	1.130	4460	4434
470	1.152	1.137	4480	4447
475	1.160	1.145	4505	4469
480	1.167	1.153	4530	4491
485	1.176	1.162	4560	4522
490	1.184	1.170	4590	4553
495	1.194	1.180	4625	4593
500	1.203	1.190	4660	4633
505	1.213	1.204	4700	4681
510	1.222	1.219	4740	4730

Table B1 - Continued

Temperature	Density Tab.	Density Eqn.	Specific Heat Tab.	Specific Heat Eqn.
515	1.233	1.230	4790	4787
520	1.244	1.242	4840	4845
525	1.256	1.254	4895	4911
530	1.268	1.267	4950	4977
535	1.281	1.280	5015	5052
540	1.294	1.294	5080	5128
545	1.309	1.310	5160	5211
550	1.323	1.325	5240	5295
555	1.339	1.342	5335	5388
560	1.355	1.359	5430	5481
565	1.374	1.377	5555	5583
570	1.392	1.396	5680	5684
575	1.413	1.417	5840	5795
580	1.433	1.437	6000	5905
585	1.458	1.460	6205	6235
590	1.482	1.483	6410	6565
595	1.512	1.508	6705	6650
600	1.541	1.533	7000	6734
605	1.577	1.574	7425	7241
610	1.612	1.614	7850	7748
615	1.659	1.640	8600	8676
620	1.705	1.666	9350	9605
625	1.778	1.734	10600	10850
630	1.856	1.830	12600	12306
635	1.935	1.954	16400	16555
640	2.075	2.107	26000	26162
645	2.351	2.288	90000	90201
	s = 0.93%		s = 0.95%	

Temperature	Viscosity Tab.	Viscosity Eqn.	Thermal Conductivity Tab.	Thermal Conductivity Eqn.
275	0.001652	0.001597	0.574	0.575
280	0.001422	0.001414	0.582	0.583
285	0.001225	0.001254	0.590	0.591
290	0.001080	0.001113	0.598	0.599
295	0.000959	0.000990	0.606	0.606
300	0.000855	0.000881	0.613	0.613
305	0.000769	0.000786	0.620	0.619
310	0.000695	0.000701	0.628	0.626
315	0.000631	0.000626	0.634	0.632
320	0.000577	0.000560	0.640	0.638
325	0.000528	0.000501	0.645	0.643
330	0.000489	0.000449	0.650	0.648

Table B1 - Continued

Temperature	Viscosity Tab.	Viscosity Eqn.	Thermal Conductivity Tab.	Thermal Conductivity Eqn.
335	0.000453	0.000414	0.656	0.653
340	0.000420	0.000396	0.660	0.658
345	0.000389	0.000380	0.668	0.662
350	0.000365	0.000363	0.668	0.666
355	0.000343	0.000348	0.671	0.669
360	0.000324	0.000333	0.674	0.673
365	0.000306	0.000319	0.677	0.676
370	0.000289	0.000306	0.679	0.679
375	0.000274	0.000293	0.681	0.681
380	0.000260	0.000281	0.683	0.683
385	0.000248	0.000269	0.685	0.685
390	0.000237	0.000258	0.686	0.687
395	0.000227	0.000247	0.687	0.688
400	0.000217	0.000237	0.688	0.689
405	0.000209	0.000228	0.688	0.689
410	0.000200	0.000218	0.688	0.690
415	0.000193	0.000210	0.688	0.690
420	0.000185	0.000201	0.688	0.689
425	0.000179	0.000186	0.687	0.689
430	0.000173	0.000170	0.685	0.688
435	0.000168	0.000166	0.684	0.687
440	0.000162	0.000162	0.682	0.685
445	0.000157	0.000158	0.680	0.683
450	0.000152	0.000154	0.678	0.681
455	0.000148	0.000150	0.676	0.679
460	0.000143	0.000146	0.673	0.676
465	0.000140	0.000143	0.670	0.673
470	0.000136	0.000139	0.667	0.670
475	0.000133	0.000135	0.664	0.666
480	0.000129	0.000131	0.660	0.663
485	0.000127	0.000128	0.656	0.658
490	0.000124	0.000124	0.651	0.654
495	0.000121	0.000121	0.647	0.649
500	0.000118	0.000117	0.642	0.644
505	0.000116	0.000114	0.637	0.639
510	0.000113	0.000111	0.631	0.633
515	0.000111	0.000108	0.626	0.627
520	0.000108	0.000105	0.621	0.621
525	0.000106	0.000104	0.615	0.614
530	0.000104	0.000104	0.608	0.607
535	0.000103	0.000102	0.601	0.600
540	0.000101	0.000101	0.594	0.592
545	0.000099	0.000099	0.587	0.584
550	0.000097	0.000098	0.580	0.577
555	0.000096	0.000096	0.572	0.568
560	0.000094	0.000094	0.563	0.559
565	0.000093	0.000093	0.556	0.550
570	0.000091	0.000091	0.548	0.541

Table B1 - Continued

Temperature	Viscosity Tab.	Viscosity Eqn.	Thermal Conductivity Tab.	Thermal Conductivity Eqn.
575	0.000090	0.000089	0.538	0.531
580	0.000088	0.000088	0.528	0.522
585	0.000086	0.000086	0.521	0.511
590	0.000084	0.000084	0.513	0.501
595	0.000083	0.000083	0.505	0.490
600	0.000081	0.000081	0.497	0.479
605	0.000079	0.000079	0.482	0.467
610	0.000077	0.000077	0.467	0.456
615	0.000075	0.000075	0.456	0.443
620	0.000072	0.000072	0.444	0.431
625	0.000070	0.000070	0.430	0.419
630	0.000067	0.000067	0.412	0.406
635	0.000064	0.000064	0.392	0.392
640	0.000059	0.000059	0.367	0.379
645	0.000054	0.000054	0.331	0.365
	s = 3.76%		s = 1.57%	

Prandtl Number Tab.	Prandtl Number Eqn.	Temperature	Prandtl Number Tab.	Prandtl Number Eqn.	Temp
12.22	11.66	275	0.94	0.94	465
10.26	10.17	280	0.92	0.92	470
8.81	8.89	285	0.91	0.91	475
7.56	7.79	290	0.89	0.89	480
6.62	6.84	295	0.88	0.88	485
5.83	6.02	300	0.87	0.86	490
5.20	5.30	305	0.87	0.86	495
4.62	4.68	310	0.86	0.84	500
4.16	4.14	315	0.86	0.84	505
3.77	3.67	320	0.85	0.83	510
3.42	3.26	325	0.85	0.82	515
3.15	2.89	330	0.84	0.82	520
2.88	2.65	335	0.85	0.84	525
2.66	2.52	340	0.85	0.85	530
2.45	2.40	345	0.86	0.86	535
2.29	2.29	350	0.86	0.87	540
2.14	2.18	355	0.87	0.88	545
2.02	2.08	360	0.87	0.90	550
1.91	1.99	365	0.89	0.91	555
1.80	1.90	370	0.90	0.92	560
1.70	1.81	375	0.92	0.94	565
1.61	1.74	380	0.94	0.96	570
1.53	1.66	385	0.97	0.97	575
1.47	1.59	390	0.99	0.99	580
1.41	1.53	395	1.02	1.05	585
1.34	1.47	400	1.05	1.11	590
1.29	1.41	405	1.10	1.12	595

Properties of Steam

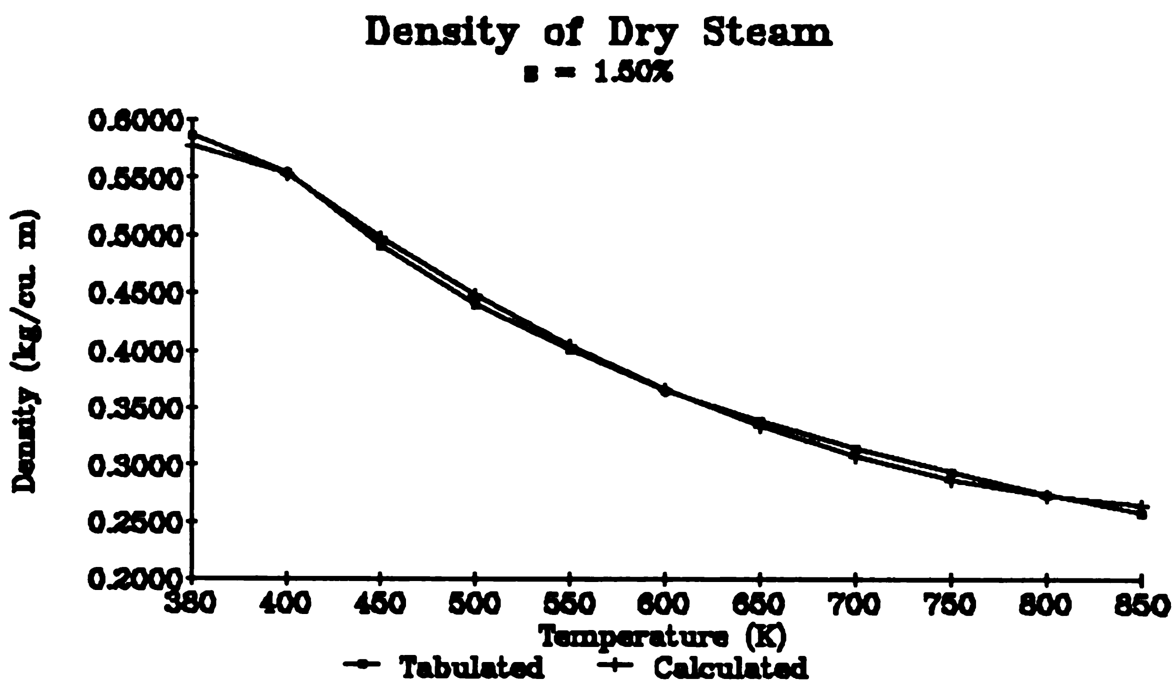


Figure B6: Density of steam

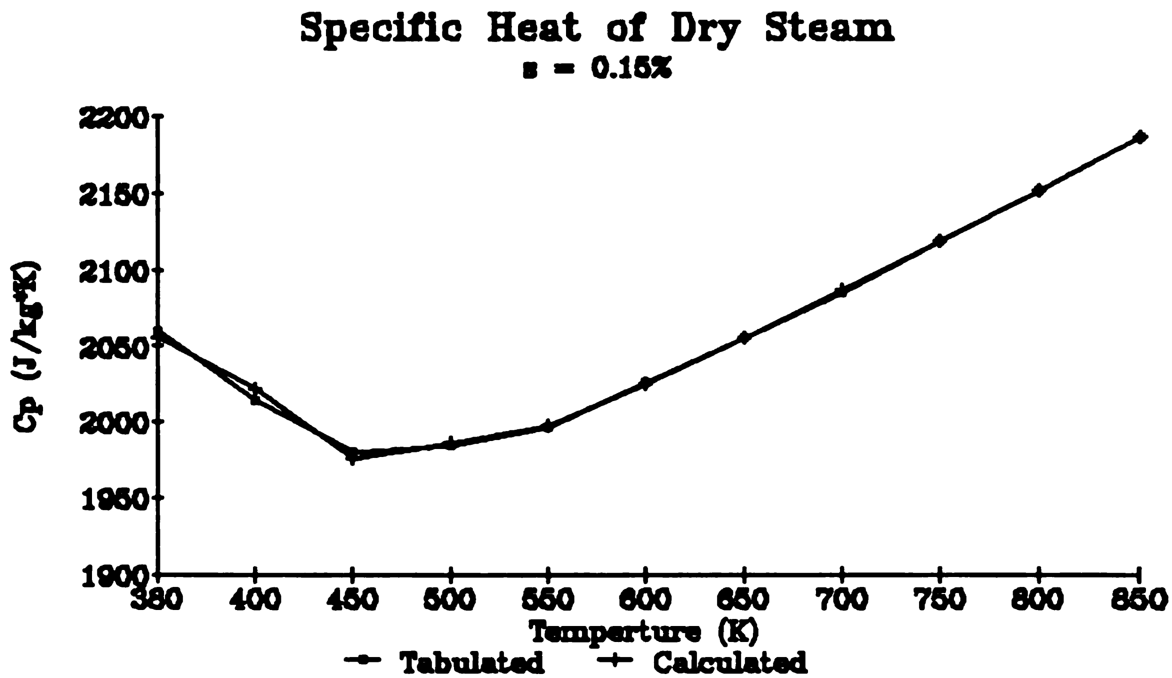


Figure B7: Specific heat of steam

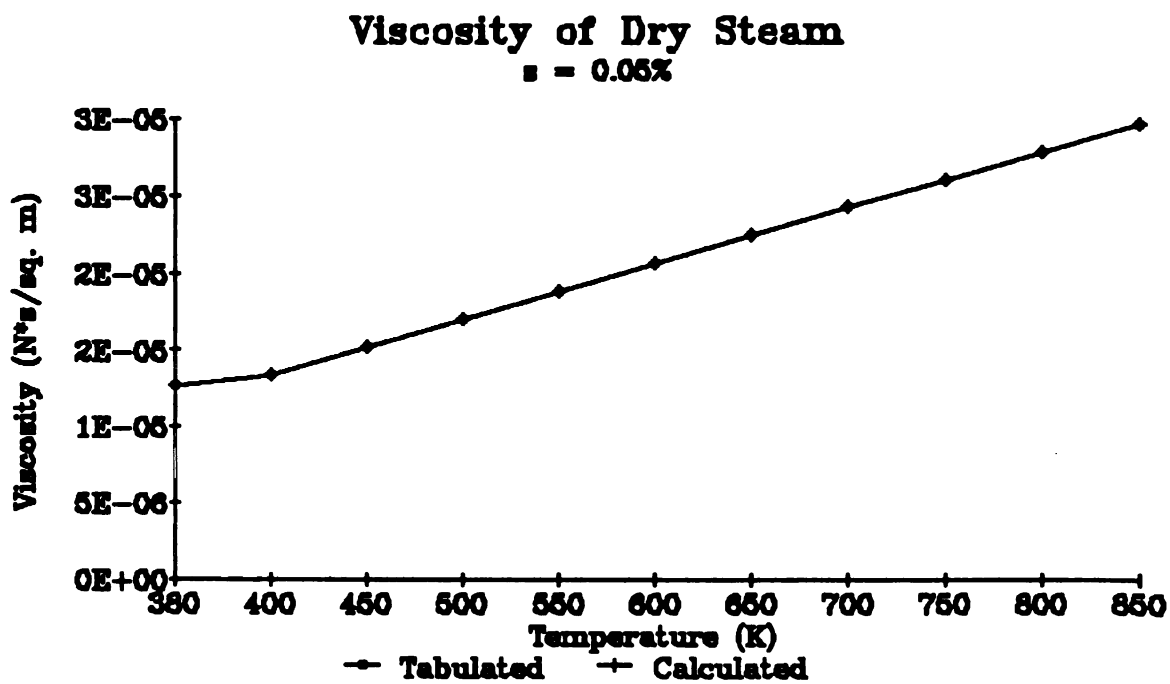


Figure B8: Viscosity of steam

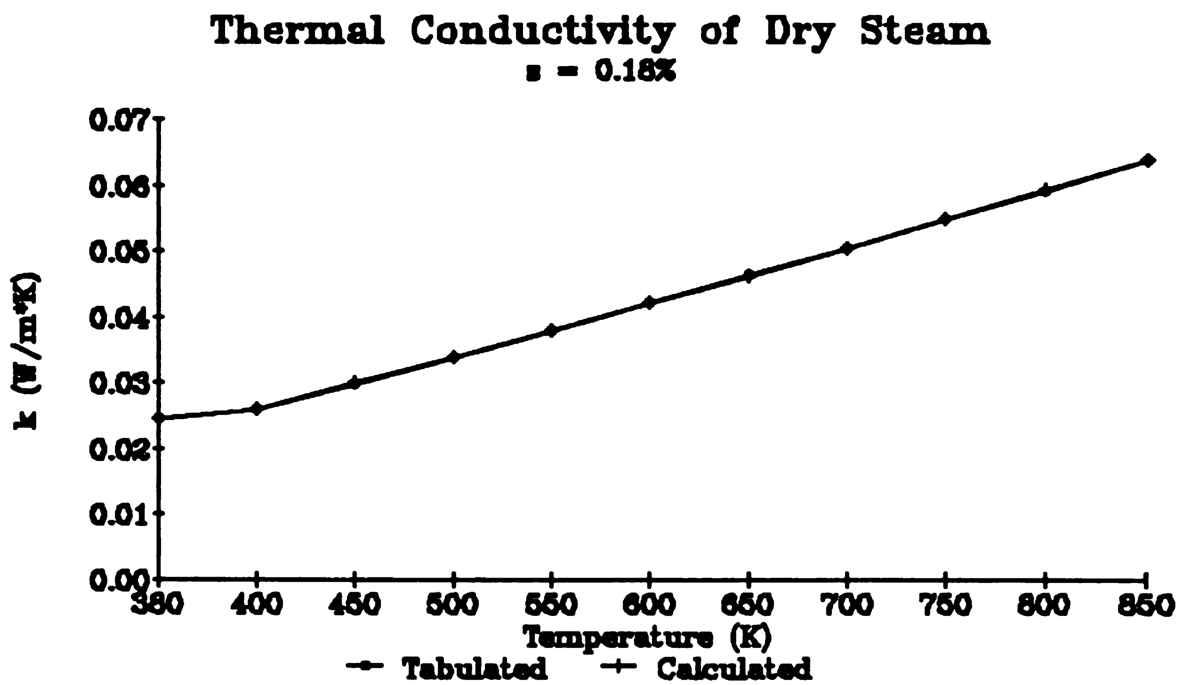


Figure B9: Thermal conductivity of steam

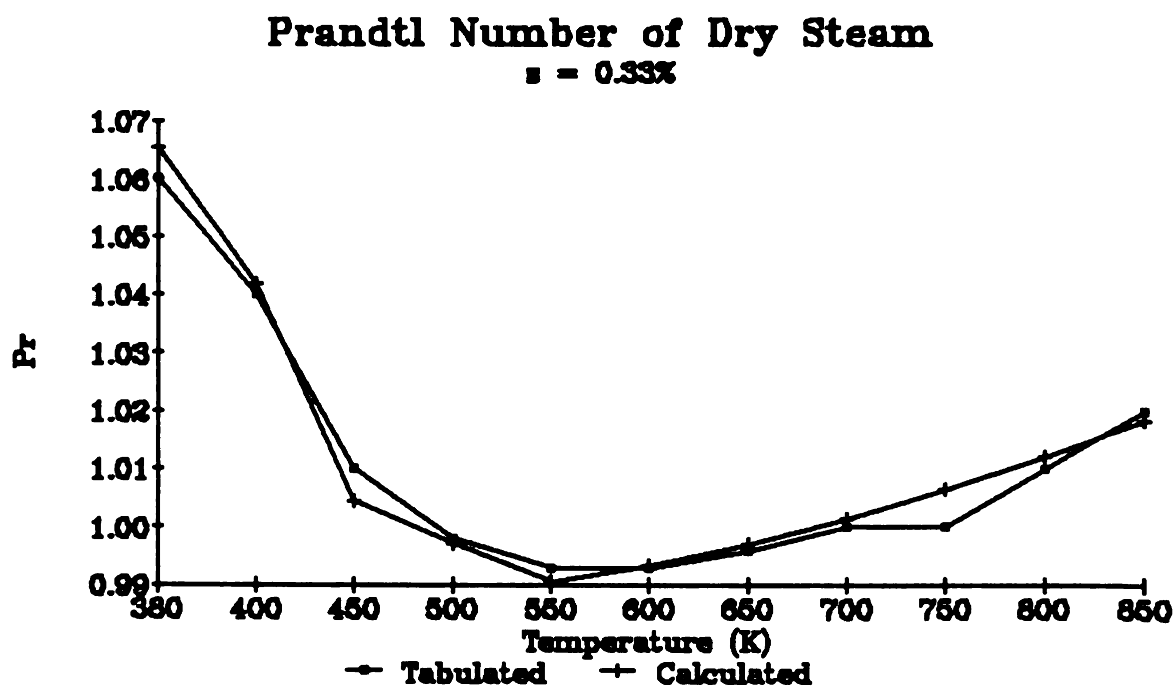


Figure B10: Prandtl number of steam

Table B2: Properties of Dry Steam

Temperature	Density Tab.	Density Eqn.	Specific Heat Tab.	Specific Heat Eqn.
380	0.5863	0.5772	2060	2056
400	0.5542	0.5532	2014	2021
450	0.4902	0.4974	1980	1976
500	0.4405	0.4476	1985	1986
550	0.4005	0.4036	1997	1997
600	0.3652	0.3656	2026	2026
650	0.3380	0.3335	2056	2055
700	0.3140	0.3073	2085	2086
750	0.2931	0.2870	2119	2118
800	0.2739	0.2727	2152	2152
850	0.2579	0.2642	2186	2186
	s = 1.50%		s = 0.15%	

Temperature	Viscosity Tab.	Viscosity Eqn.	Thermal Conductivity Tab.	Thermal Conductivity Eqn.
380	1.271E-05	1.272E-05	0.0246	0.0245
400	1.344E-05	1.344E-05	0.0261	0.0261
450	1.525E-05	1.524E-05	0.0299	0.0300
500	1.704E-05	1.705E-05	0.0339	0.0340
550	1.884E-05	1.885E-05	0.0379	0.0380
600	2.067E-05	2.065E-05	0.0422	0.0421
650	2.247E-05	2.246E-05	0.0464	0.0463
700	2.426E-05	2.426E-05	0.0505	0.0505
750	2.604E-05	2.606E-05	0.0549	0.0549
800	2.786E-05	2.787E-05	0.0592	0.0593
850	2.969E-05	2.967E-05	0.0637	0.0637
	s = 0.05%		s = 0.18%	

Prandtl Number Tab.	Prandtl Number Eqn.	Temperature	
1.06	1.07	380	
1.04	1.04	400	
1.01	1.00	450	
1.00	1.00	500	
0.99	0.99	550	
0.99	0.99	600	
1.00	1.00	650	
1.00	1.00	700	
1.00	1.01	750	
1.01	1.01	800	
1.02	1.02	850	s = 0.33%

Properties of Air

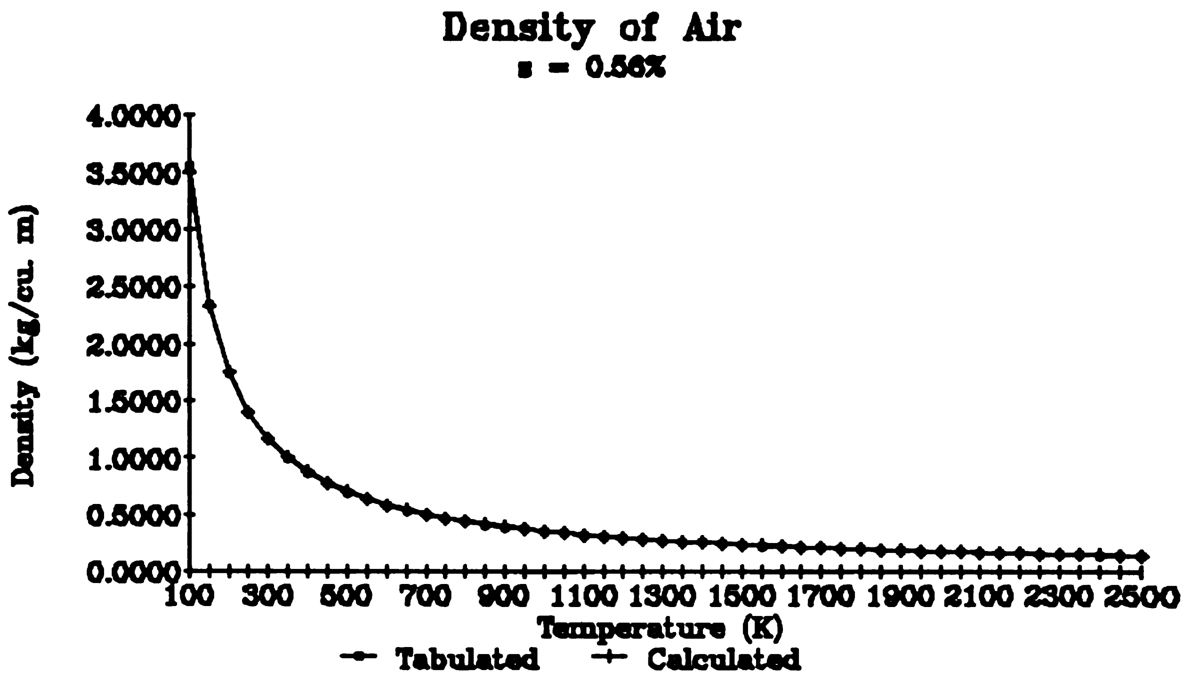


Figure B11: Density of air

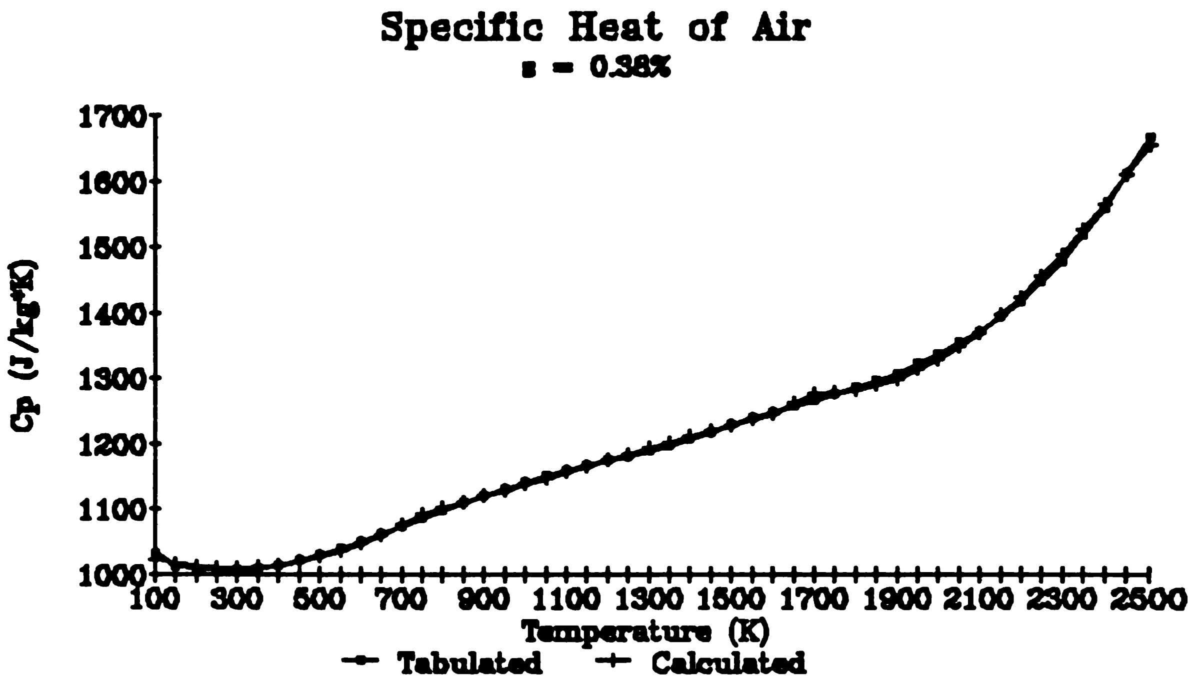


Figure B12: Specific heat of air

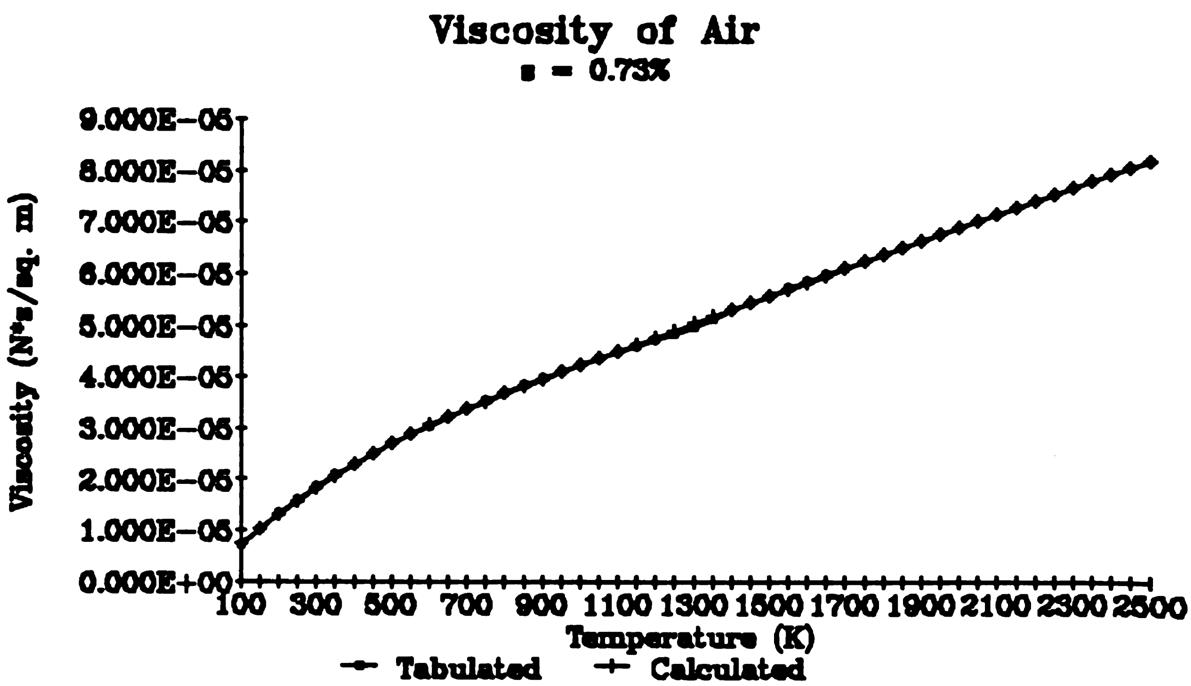


Figure B13: Viscosity of air

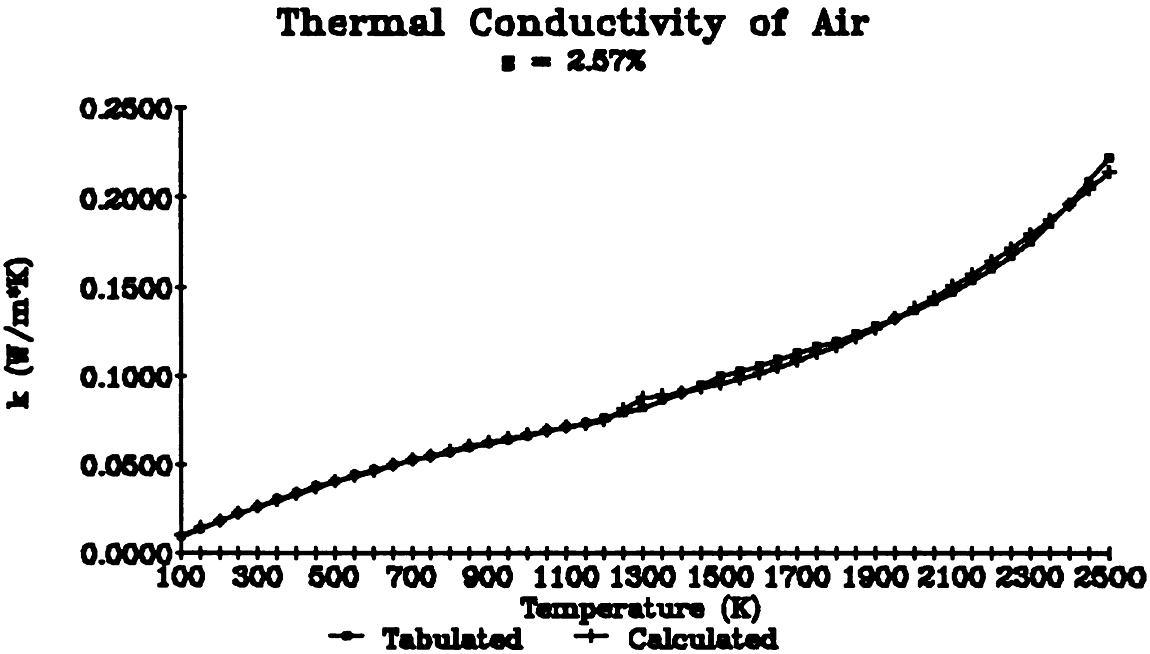


Figure B14: Thermal conductivity of air

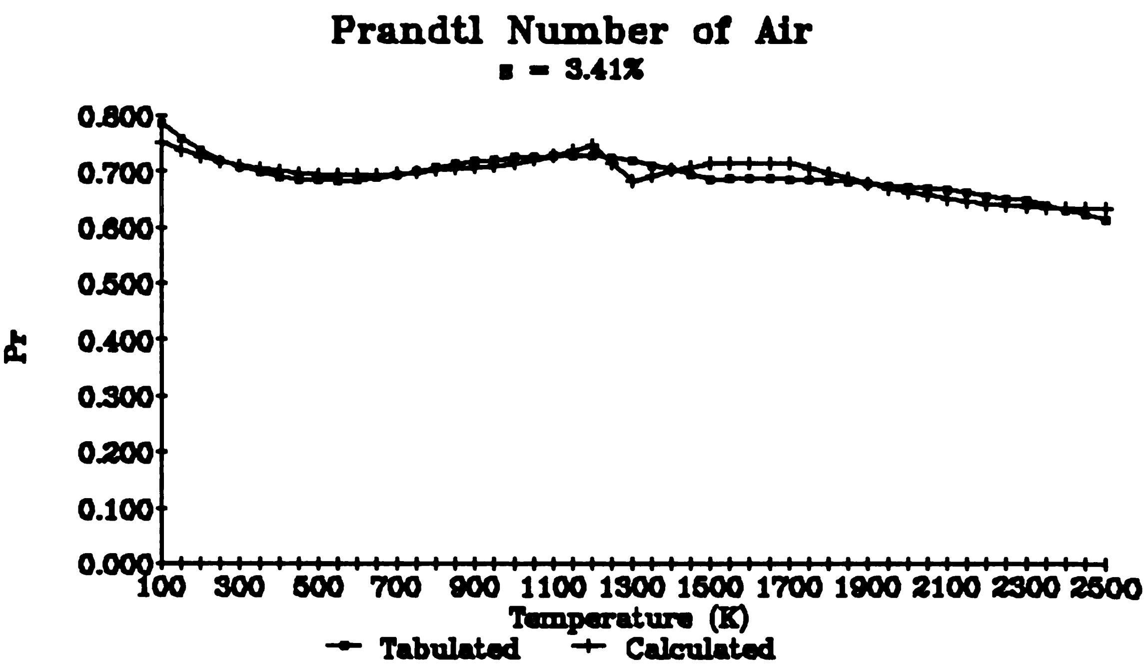


Figure B15: Prandtl number of air

Table B3: Properties of Air

Temperature	Density Tab.	Density Eqn.	Specific Heat Tab.	Specific Heat Eqn.
100	3.5562	3.5000	1032	1022
150	2.3364	2.3333	1012	1016
200	1.7458	1.7500	1007	1012
250	1.3947	1.4000	1006	1010
300	1.1614	1.1667	1007	1009
350	0.9950	1.0000	1009	1011
400	0.8711	0.8750	1014	1015
450	0.7740	0.7778	1021	1020
500	0.6964	0.7000	1030	1027
550	0.6329	0.6364	1040	1037
600	0.5804	0.5833	1051	1048
650	0.5356	0.5385	1063	1061
700	0.4975	0.5000	1075	1076
750	0.4643	0.4667	1087	1093
800	0.4354	0.4375	1099	1102
850	0.4097	0.4118	1110	1111
900	0.3868	0.3889	1121	1120
950	0.3666	0.3684	1131	1129
1000	0.3482	0.3500	1141	1138
1050	0.3324	0.3341	1150	1147
1100	0.3166	0.3182	1159	1157
1150	0.3034	0.3049	1167	1166
1200	0.2902	0.2917	1175	1175
1250	0.2791	0.2804	1182	1184
1300	0.2679	0.2692	1189	1193
1350	0.2584	0.2596	1198	1202
1400	0.2488	0.2500	1207	1211
1450	0.2405	0.2417	1219	1220
1500	0.2322	0.2333	1230	1228
1550	0.2250	0.2260	1239	1237
1600	0.2177	0.2188	1248	1246
1650	0.2113	0.2123	1258	1261
1700	0.2049	0.2059	1267	1276
1750	0.1992	0.2002	1277	1279
1800	0.1935	0.1944	1286	1282
1850	0.1884	0.1893	1297	1291
1900	0.1833	0.1842	1307	1299
1950	0.1787	0.1796	1322	1314
2000	0.1741	0.1750	1337	1329
2050	0.1700	0.1708	1355	1350
2100	0.1658	0.1667	1372	1370
2150	0.1620	0.1629	1395	1397
2200	0.1582	0.1591	1417	1423
2250	0.1548	0.1556	1448	1456
2300	0.1513	0.1522	1478	1488
2350	0.1481	0.1490	1518	1527
2400	0.1448	0.1458	1558	1565

Table B3 - Continued

Temperature	Density Tab.	Density Eqn.	Specific Heat Tab.	Specific Heat Eqn.
2450	0.1419	0.1429	1612	1609
2500	0.1389	0.1400	1665	1654
	s = 0.56%		s = 0.38%	

Temperature	Viscosity Tab.	Viscosity Eqn.	Thermal Conductivity Tab.	Thermal Conductivity Eqn.
100	7.110E-06	7.400E-06	0.0093	0.0101
150	1.034E-05	1.032E-05	0.0138	0.0142
200	1.325E-05	1.312E-05	0.0181	0.0183
250	1.596E-05	1.578E-05	0.0223	0.0222
300	1.846E-05	1.831E-05	0.0263	0.0260
350	2.082E-05	2.071E-05	0.0300	0.0297
400	2.301E-05	2.298E-05	0.0338	0.0333
450	2.507E-05	2.512E-05	0.0373	0.0368
500	2.701E-05	2.713E-05	0.0407	0.0401
550	2.884E-05	2.901E-05	0.0439	0.0434
600	3.058E-05	3.076E-05	0.0469	0.0465
650	3.225E-05	3.238E-05	0.0497	0.0495
700	3.388E-05	3.387E-05	0.0524	0.0524
750	3.546E-05	3.522E-05	0.0549	0.0552
800	3.698E-05	3.698E-05	0.0573	0.0578
850	3.843E-05	3.832E-05	0.0596	0.0604
900	3.981E-05	3.966E-05	0.0620	0.0628
950	4.113E-05	4.099E-05	0.0643	0.0651
1000	4.244E-05	4.233E-05	0.0667	0.0673
1050	4.367E-05	4.366E-05	0.0691	0.0693
1100	4.490E-05	4.500E-05	0.0715	0.0714
1150	4.610E-05	4.633E-05	0.0739	0.0732
1200	4.730E-05	4.766E-05	0.0763	0.0750
1250	4.845E-05	4.899E-05	0.0792	0.0815
1300	4.960E-05	5.032E-05	0.0820	0.0880
1350	5.130E-05	5.165E-05	0.0865	0.0896
1400	5.300E-05	5.297E-05	0.0910	0.0912
1450	5.435E-05	5.430E-05	0.0955	0.0934
1500	5.570E-05	5.562E-05	0.1000	0.0957
1550	5.705E-05	5.695E-05	0.1030	0.0986
1600	5.840E-05	5.827E-05	0.1060	0.1015
1650	5.975E-05	5.959E-05	0.1095	0.1051
1700	6.110E-05	6.091E-05	0.1130	0.1086
1750	6.240E-05	6.223E-05	0.1165	0.1129
1800	6.370E-05	6.355E-05	0.1200	0.1171
1850	6.500E-05	6.486E-05	0.1240	0.1220
1900	6.630E-05	6.618E-05	0.1280	0.1269
1950	6.760E-05	6.749E-05	0.1325	0.1325
2000	6.890E-05	6.881E-05	0.1370	0.1381

Table B3 - Continued

Temperature	Viscosity Tab.	Viscosity Eqn.	Thermal Conductivity Tab.	Thermal Conductivity Eqn.
2050	7.020E-05	7.012E-05	0.1420	0.1443
2100	7.150E-05	7.143E-05	0.1470	0.1506
2150	7.275E-05	7.274E-05	0.1535	0.1575
2200	7.400E-05	7.405E-05	0.1600	0.1644
2250	7.530E-05	7.536E-05	0.1675	0.1719
2300	7.660E-05	7.666E-05	0.1750	0.1795
2350	7.790E-05	7.797E-05	0.1855	0.1877
2400	7.920E-05	7.927E-05	0.1960	0.1960
2450	8.050E-05	8.058E-05	0.2090	0.2049
2500	8.180E-05	8.188E-05	0.2220	0.2138
	s = 0.73%		s = 2.57%	

Prandtl Number Tab.	Prandtl Number Eqn.	Temperature
0.786	0.752	100
0.758	0.737	150
0.737	0.726	200
0.720	0.718	250
0.707	0.710	300
0.700	0.705	350
0.690	0.700	400
0.686	0.697	450
0.684	0.695	500
0.683	0.693	550
0.685	0.693	600
0.690	0.694	650
0.695	0.695	700
0.702	0.698	750
0.709	0.705	800
0.716	0.705	850
0.720	0.707	900
0.723	0.711	950
0.726	0.716	1000
0.727	0.723	1050
0.728	0.729	1100
0.728	0.738	1150
0.728	0.747	1200
0.724	0.714	1250
0.719	0.682	1300
0.711	0.693	1350
0.703	0.704	1400
0.694	0.709	1450
0.685	0.714	1500
0.687	0.715	1550
0.688	0.716	1600

Properties of Oil

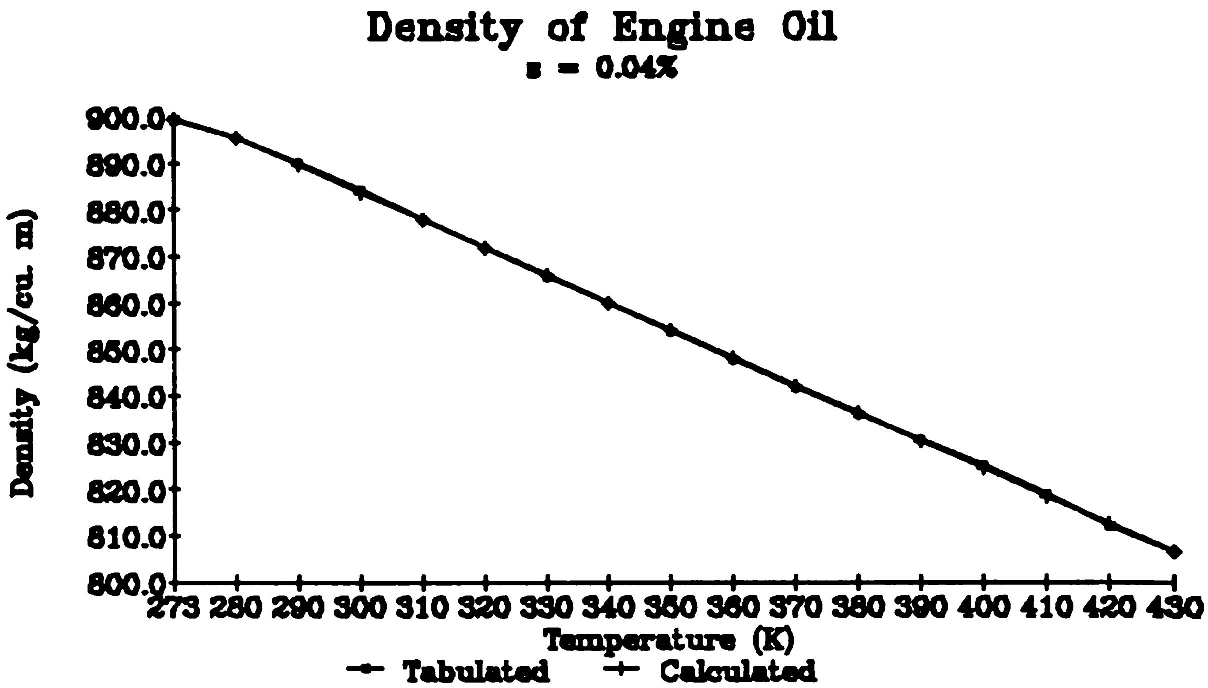


Figure B16: Density of oil

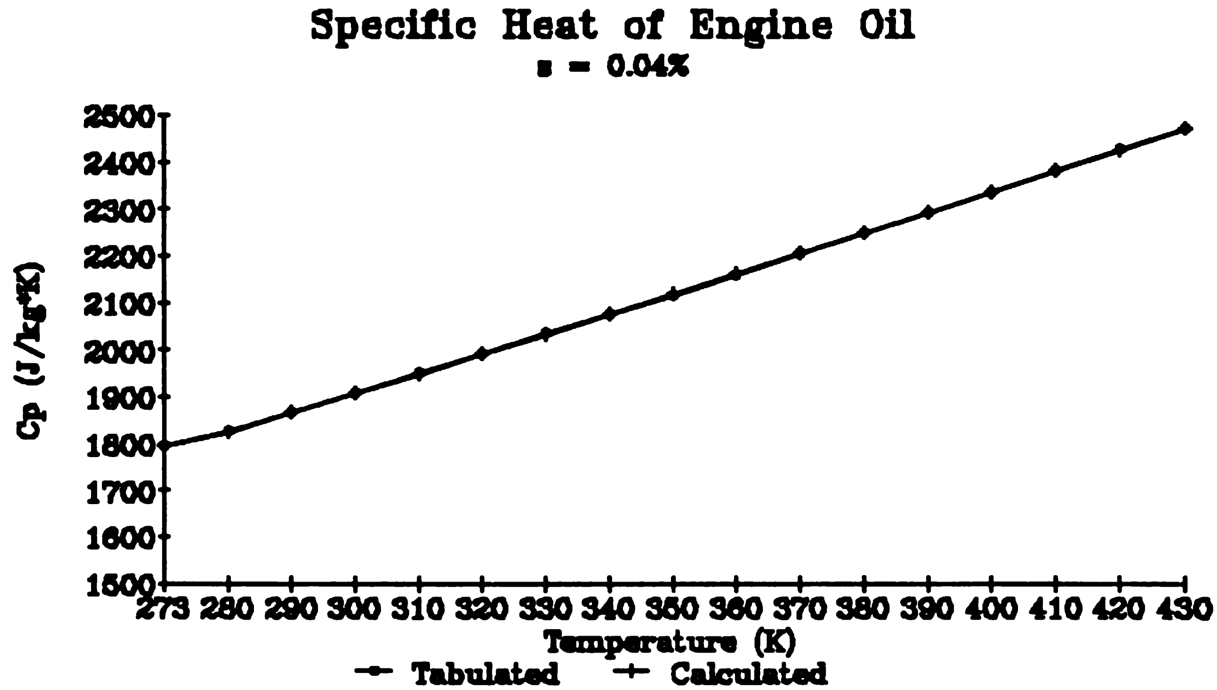


Figure B17: Specific heat of oil

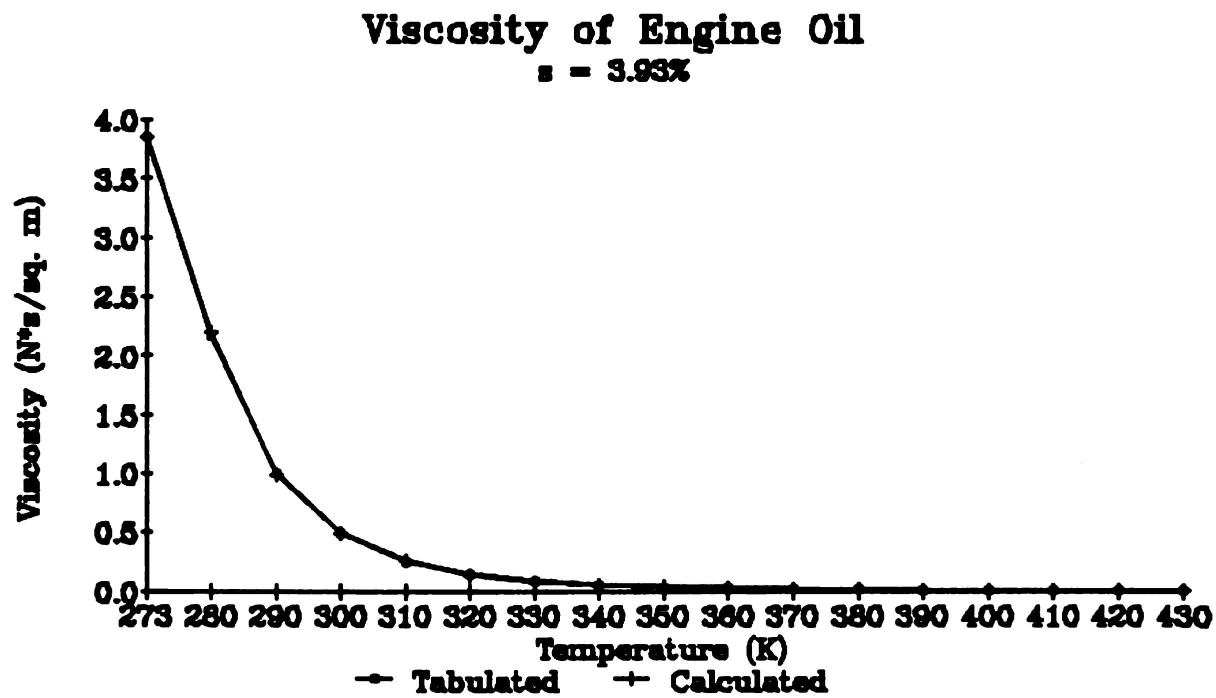


Figure B18: Viscosity of oil

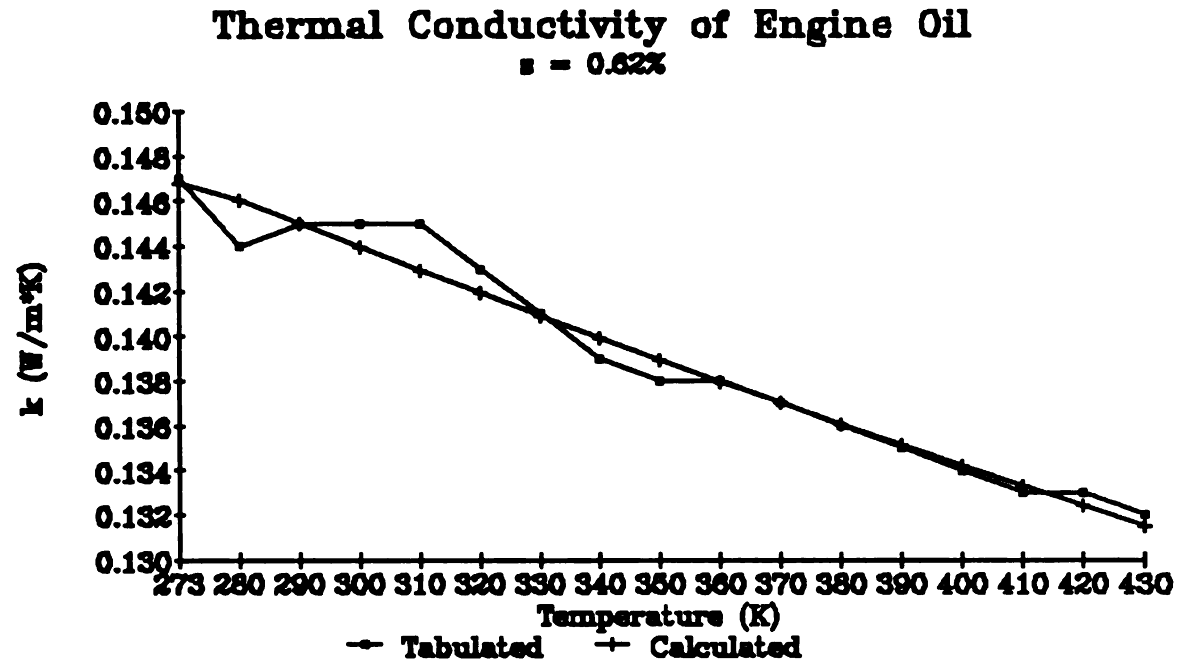


Figure B19: Thermal conductivity of oil

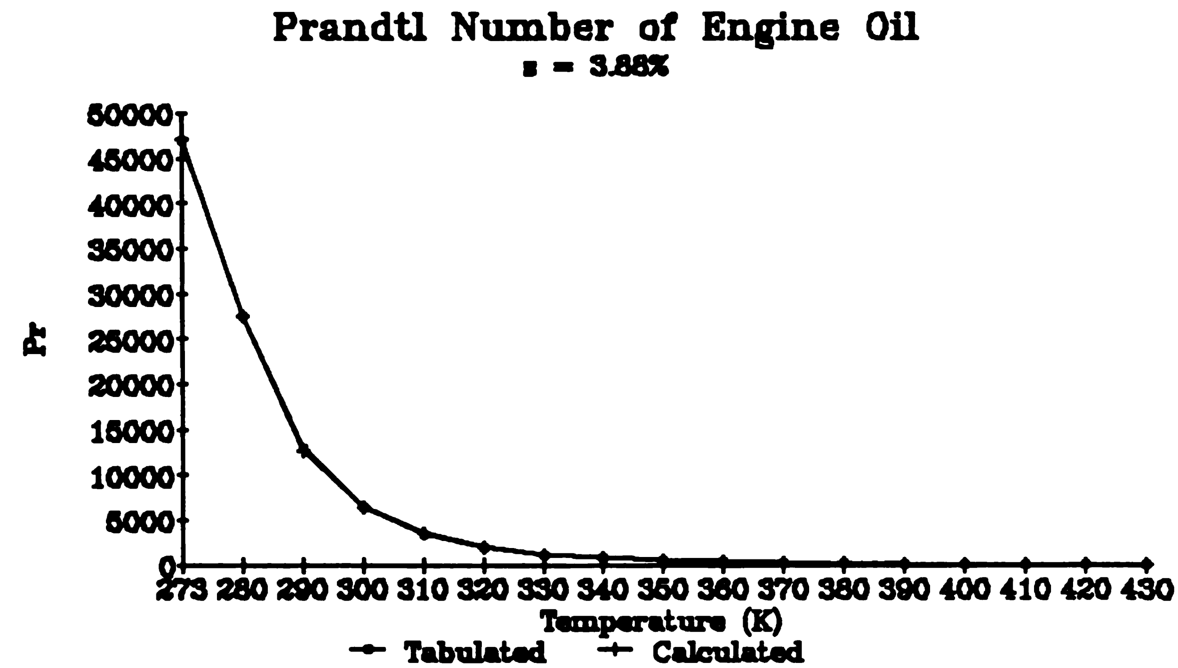


Figure B20: Prandtl number of oil

Table B4: Properties of Engine Oil

Temperature	Density Tab.	Density Eqn.	Specific Heat Tab.	Specific Heat Eqn.
273	899.1	899.6	1796	1798
280	895.3	895.4	1827	1826
290	890.0	889.5	1868	1867
300	884.1	883.6	1909	1909
310	877.9	877.7	1951	1950
320	871.8	871.8	1993	1992
330	865.8	865.9	2035	2034
340	859.9	860.0	2076	2077
350	853.9	854.0	2118	2119
360	847.8	848.1	2161	2162
370	841.8	842.2	2206	2206
380	836.0	836.3	2250	2249
390	830.6	830.3	2294	2293
400	825.1	824.4	2337	2337
410	818.9	818.5	2381	2382
420	812.1	812.5	2427	2426
430	806.5	806.6	2471	2471
	s = 0.04%		s = 0.04%	

Temperature	Viscosity Tab.	Viscosity Eqn.	Thermal Conductivity Tab.	Thermal Conductivity Eqn.
273	3.85	3.85	0.147	0.147
280	2.17	2.20	0.144	0.146
290	0.999	0.987	0.145	0.145
300	0.486	0.486	0.145	0.144
310	0.253	0.260	0.145	0.143
320	0.141	0.139	0.143	0.142
330	0.0836	0.0746	0.141	0.141
340	0.0531	0.0531	0.139	0.140
350	0.0356	0.0372	0.138	0.139
360	0.0252	0.0261	0.138	0.138
370	0.0186	0.0184	0.137	0.137
380	0.0141	0.0141	0.136	0.136
390	0.0110	0.0112	0.135	0.135
400	0.00874	0.00881	0.134	0.134
410	0.00698	0.00697	0.133	0.133
420	0.00564	0.00551	0.133	0.132
430	0.00470	0.00435	0.132	0.132
	s = 3.93%		s = 0.62%	

Table B4: Continued

Prandtl Number Tab.	Eqn.	Temperature
47000	47114	273
27500	27478	280
12900	12715	290
6400	6448	300
3400	3552	310
1965	1956	320
1205	1077	330
793	788	340
546	568	350
395	409	360
300	294	370
233	233	380
187	189	390
152	153	400
125	124	410
103	101	420
88	82	430

s = 3.88%

APPENDIX C: PROGRAM TSCAD

```
****************************************************************
*
*
*          Program TSCAD is a heat exchanger analysis and
design program.  The user inputs all known data,  signifying
unknown data by entering -1.0 as its value.  If enough
information is available, TSCAD will calculate the unknown
data and display the results.  The total data set includes
inlet and outlet temperatures for each fluid stream, both
mass flow rates, and the parameters U and A.  If U and A are
unknown, the overall heat transfer coefficient U is
calculated.  This requires additional information about the
geometry of the heat exchanger.  In this case, up to
three unknowns can be solved for, two of which must be U and
A.  In cases where U is not calculated independently (either
U or A is known) up to two unknowns can be computed.
Entering "555" at any time during the initial data entry
will restart the process without computing any results.
Entering "999" will cause the program to ask the user
whether he or she wants to stop the program.  If the program
is followed normally, the results will be computed and
displayed.  Property information and, in the case of U and A
both unknown, parameters such as the Nusselt number and the
Reynold's number will also be displayed.  After displaying
results, the program will ask whether or not to
continue.  Entering anything other than "Y" or "y" will
terminate the program.
*
*
****************************************************************
        PROGRAM NTUEFF
C
        DIMENSION C(30),PRIME(9),UNK(8),D(10),F(10),G(10)
        INTEGER CF,HF,UNK
        CHARACTER W*1,TITLE(30)*4
        COMMON C,Q,N,D,F,G
        COMMON /FLUID/ CF,HF
C
      WRITE(*,89)
89    FORMAT(//,//,//,//,//,26X,'TSCAD:',//,18X,'HEAT
EXCHANGER SOLVER')
      WRITE(*,70)
70    FORMAT(//,//,//,//,//,'Press RETURN to begin')
      READ(*,*)
      WRITE(*,80)
```

```
80
FORMAT(//,//,//,//,//,//,//,//,//,//,'**********************
***',
     +'****************************************')
C
C                * Defining the Problem *
C
*
      PRINT *, 'Program TSCAD uses SI units to calculate the
unknown',
     +' parameters.                Enter "-1.0" for any
unknown data.',
     +' If there are too many unknowns,              the',
     +' program will state that the problem is unsolvable
and prompt',
     +'                  for either continuation or
termination of the',
     +' program.'
      PRINT *,'  '
      PRINT *,' '
*
555     PRINT *, 'ENTER PROBLEM PARAMETERS.'
        PRINT *, 'ENTER -1.0 IF NOT KNOWN.'
        PRINT *, 'ENTER 555 TO RESTART.'
        PRINT *, 'ENTER 999 TO QUIT.'
        PRINT *, '  '
C
        PRINT *, '(1) WATER            (3) AIR'
        PRINT *, '(2) STEAM            (4) OIL'
        PRINT *, '  '
C
        PRINT *, 'COLD FLUID'
        READ *,CF
        IF(CF.EQ.555)THEN
          GO TO 555
        ELSEIF(CF.EQ.999)THEN
          GO TO 200
        ENDIF
*
        PRINT *, 'HOT FLUID'
        READ *,HF
        IF(HF.EQ.555)THEN
          GO TO 555
        ELSEIF(HF.EQ.999)THEN
          GO TO 200
        ENDIF
C
        PRINT *, 'COLD INLET TEMP. (C)'
        READ *,C(1)
                IF(C(1).GT.0.0)THEN
                        C(1) = 273 + C(1)
                ENDIF
        IF(C(1).EQ.555)THEN
          GO TO 555
```

```
      ELSEIF(C(1).EQ.999)THEN
        GO TO 200
      ENDIF
*
      PRINT *, 'COLD OUTLET TEMP. (C)'
      READ *,C(3)
              IF(C(3).GT.0.0)THEN
                      C(3) = 273 + C(3)
              ENDIF
      IF(C(3).EQ.555)THEN
        GO TO 555
      ELSEIF(C(3).EQ.999)THEN
        GO TO 200
      ENDIF
*
      PRINT *, 'COLD MASS FLOW RATE (kg/s)'
      READ *,C(5)
      IF(C(5).EQ.555)THEN
        GO TO 555
      ELSEIF(C(5).EQ.999)THEN
        GO TO 200
      ENDIF
C
      PRINT *, 'HOT INLET TEMP. (C)'
      READ *,C(4)
              IF(C(4).GT.0.0)THEN
                      C(4) = 273 + C(4)
              ENDIF
      IF(C(4).EQ.555)THEN
        GO TO 555
      ELSEIF(C(4).EQ.999)THEN
        GO TO 200
      ENDIF
*
      PRINT *, 'HOT OUTLET TEMP. (C)'
      READ *,C(2)
              IF(C(2).GT.0.0)THEN
                      C(2) = 273 + C(2)
              ENDIF
      IF(C(2).EQ.555)THEN
        GO TO 555
      ELSEIF(C(2).EQ.999)THEN
        GO TO 200
      ENDIF
*
      PRINT *, 'HOT MASS FLOW RATE (kg/s)'
      READ *,C(6)
      IF(C(6).EQ.555)THEN
        GO TO 555
      ELSEIF(C(6).EQ.999)THEN
        GO TO 200
      ENDIF
C
      PRINT *, 'OVERALL COEFFICIENT (W K/sq. m)'
```

```
         READ *,C(7)
         IF(C(7).EQ.555)THEN
           GO TO 555
         ELSEIF(C(7).EQ.999)THEN
           GO TO 200
         ENDIF
*
         PRINT *, 'SURFACE AREA (sq. m)'
         READ *,C(8)
         IF(C(8).EQ.555)THEN
           GO TO 555
         ELSEIF(C(8).EQ.999)THEN
           GO TO 200
         ENDIF
C
         PRINT *, '(1) DOUBLE PIPE PARALLEL FLOW'
         PRINT *, '(2) DOUBLE PIPE COUNTERFLOW'
         PRINT *, '(3) CROSSFLOW - COLD UNMIXED/HOT MIXED'
         PRINT *, '(4) CROSSFLOW - COLD MIXED/HOT UNMIXED'
         PRINT *, '(5) COMPACT HEAT EXCHANGER - HOT FLUID
FIN-SIDE'
         PRINT *, '(6) COMPACT HEAT EXCHANGER - COLD FLUID
FIN-SIDE'
         PRINT *, 'CHOOSE A HEAT EXCHANGER TYPE.'
         READ *,N
         IF(N.EQ.555)THEN
           GO TO 555
         ELSEIF(N.EQ.999)THEN
           GO TO 200
         ENDIF
C
C                * Decision Loop for Subprograms *
C
       DO 16 M=1,8
           UNK(M) = 9
16     CONTINUE
*
         PRIME(1)=17.0
         PRIME(2)=3.0
         PRIME(3)=2.0
         PRIME(4)=19.0
         PRIME(5)=5.0
         PRIME(6)=7.0
         PRIME(7)=11.0
         PRIME(8)=13.0
         PRIME(9)=1.0
C
         J = 1
         DO 1 I=1,8
           IF(C(I).EQ.-1.0)THEN
           UNK(J) = I
           J = J + 1
           ENDIF
1        CONTINUE
```

```
         CHK =
PRIME(UNK(1))*PRIME(UNK(2))*PRIME(UNK(3))*PRIME(UNK(4))*
     +
PRIME(UNK(5))*PRIME(UNK(6))*PRIME(UNK(7))*PRIME(UNK(8))
C
C              ****************************
C              * Assign Fluid Properties *
C              ****************************
      CALL PROPS
*
      C(19) = 0.0
      C(20) = 0.0
      C(21) = 0.0
      C(22) = 0.0
         Q  = 0.0
C          *********************************
C          * Call Appropriate Subroutine *
C          *********************************
        IF(CHK.EQ.10.OR.CHK.EQ.21)THEN
           CALL ONE
        ELSE IF(CHK.EQ.14.OR.CHK.EQ.15)THEN
           CALL TWO
        ELSE
IF(CHK.EQ.11.OR.CHK.EQ.13.OR.CHK.EQ.22.OR.CHK.EQ.26.OR.CHK.
     +  EQ.33.OR.CHK.EQ.39)THEN
           CALL SIZING
        ELSE
IF(CHK.EQ.5.OR.CHK.EQ.7.OR.CHK.EQ.35.OR.CHK.EQ.55.OR.CHK.
     +  EQ.65.OR.CHK.EQ.77.OR.CHK.EQ.91)THEN
           CALL RATES
        ELSE IF(CHK.EQ.6.OR.CHK.EQ.2.OR.CHK.EQ.3)THEN
           CALL RATING
        ELSE
IF(CHK.EQ.17.OR.CHK.EQ.19.OR.CHK.EQ.187.OR.CHK.EQ.209.OR.
     +
CHK.EQ.221.OR.CHK.EQ.247.OR.CHK.EQ.2717.OR.CHK.EQ.2431)THEN
           CALL ENERGY
        ELSE IF(CHK.EQ.143)THEN
           CALL OVERALL
           CALL SIZING
        ELSE IF(CHK.EQ.286.OR.CHK.EQ.429)THEN
           CALL OVERALL
           CALL SIZING
           CALL RATING
        ELSE
           PRINT *,'NOT ENOUGH INFORMATION'
        ENDIF
*
C
C               * Output Table *
C
        TITLE(1) = 'TCI'
        TITLE(2) = 'THO'
        TITLE(3) = 'TCO'
```

```
      TITLE(4) = 'THI'
      TITLE(5) = 'MC '
      TITLE(6) = 'MH '
      TITLE(7) = 'U  '
      TITLE(8) = 'A  '
      TITLE(9) = 'CPC'
      TITLE(10)= 'CPH'
      TITLE(23)= 'Q  '
      TITLE(11)= 'VISC'
      TITLE(12)= 'VISH'
      TITLE(13)= 'KC'
      TITLE(14)= 'KH'
      TITLE(15)= 'DENC'
      TITLE(16)= 'DENH'
      TITLE(17)= 'PrC'
      TITLE(18)= 'PrH'
      TITLE(19)= 'ReC'
      TITLE(20)= 'ReH'
      TITLE(21)= 'NuC'
      TITLE(22)= 'NuH'
      TITL(24)='INTERNAL CONVECTION'
      TITL(25)='EXTERNAL CONVECTION'
      TITL(26)='WALL CONDUCTION'
      TITL(27)='INTERNAL FOULING'
      TITL(28)='EXTERNAL FOULING'
C                   ******************
C                   * Print Results *
C                   ******************
      DO 90 M=1,2
      WRITE(*,100)TITLE(M),C(M),TITLE(M+8),C(M+8)
100   FORMAT(' ',10X,A4,' = ',1F11.1,' K',8X,A4,' =
',F7.2,' kJ/kg*K'
     +)
90    CONTINUE

      DO 91 M=3,4
      WRITE(*,101)TITLE(M),C(M),TITLE(M+8),C(M+8)
101   FORMAT(' ',10X,A4,' = ',1F11.1,' K',8X,A4,' =
',F11.6,' N*s/sq m
     +')
91    CONTINUE

      DO 92 M=5,6
      WRITE(*,102)TITLE(M),C(M),TITLE(M+8),C(M+8)
102   FORMAT(' ',10X,A4,' = ',1F13.3,' kg/s',3X,A4,' =
',F8.3,' W/m*K'
     +)
92    CONTINUE

      DO 93 M=7,7
      WRITE(*,103)TITLE(M),C(M),TITLE(M+8),C(M+8)
103   FORMAT(' ',10X,A4,' = ',1F10.0,' W/K*sq m',2X,A4,' =
',F8.3,' kg
     +/cu m')
```

```
93      CONTINUE

        DO 94 M=8,8
        WRITE(*,104)TITLE(M),C(M),TITLE(M+8),C(M+8)
104     FORMAT(' ',10X,A4,' = ',1F14.4,' sq m',2X,A4,' =
',F8.3,' kg/cu
     +m')
94      CONTINUE
C
        WRITE(*,203)TITLE(23),Q
203     FORMAT(' ',10X,A4,' = ',1F11.1,' W')
C
        DO 95 K=17,18
                WRITE(*,105)TITLE(K),C(K)
105             FORMAT(' ',10X,A4,' = ',F13.3)
95      CONTINUE
*

        DO 96 K=19,20
                WRITE(*,106)TITLE(K),C(K)
106             FORMAT(' ',10X,A4,' = ',F10.0)
96      CONTINUE

        DO 97 K=21,22
                WRITE(*,107)TITLE(K),C(K)
107             FORMAT(' ',10X,A4,' = ',F11.1)
97      CONTINUE
*
*
        PRINT *,' '
        PRINT *,' NORMALIZED RESISTANCES: '
*
        DO 12 K=24,28
                WRITE(*,112)TITL(K),C(K)
112             FORMAT(' ',10X,A20,' = ',F4.1,'%')
12      CONTINUE
*
200     PRINT *, 'DO YOU WISH TO CONTINUE?'
        READ(*,300)W
300     FORMAT(A1)
        IF(W.EQ.'Y'.OR.W.EQ.'y')THEN
          GOTO 555
        ENDIF
C
C
        STOP
        END
C
*                             ********************
*                             * Subroutine PROPS *
*************************************************************
SUBROUTINE PROPS
*
        DIMENSION C(30)
```

```
        INTEGER CF,HF
        COMMON C,Q,N
        COMMON /FLUID/ CF,HF
*
      TH = (C(4)+C(2))/2
      TC = (C(3)+C(1))/2
*
      IF(C(1).EQ.-1.0)THEN
          TC = C(3)
      ELSEIF(C(3).EQ.-1.0)THEN
          TC = C(1)
      ENDIF
*
      IF(C(2).EQ.-1.0)THEN
          TH = C(4)
      ELSEIF(C(4).EQ.-1.0)THEN
          TH = C(2)
      ENDIF
C
*
      C(9)  = CP(CF,TC)
      C(10) = CP(HF,TH)
      C(11) = VISCOSITY(CF,TC)
      C(12) = VISCOSITY(HF,TH)
      C(13) = TCOND(CF,TC)
      C(14) = TCOND(HF,TH)
      C(15) = DENSITY(CF,TC)
      C(16) = DENSITY(HF,TH)
      C(17) = C(9)*C(11)/C(13)
      C(18) = C(10)*C(12)/C(14)
*
      RETURN
      END
C                                *****************
C                                * NTU FUNCTIONS *
C**************************************************************
C
        FUNCTION TRNS(N,E,CR,CD)
C

        IF(N.EQ.1)THEN
                CHK1=E*(1.0+CR)
                IF(CHK1.GE.1.0)THEN
                   CALL EXIT
                ENDIF
           TRNS=(-LOG(1.0-E*(1.0+CR)))/(1.0+CR)
        ELSE IF(N.EQ.2)THEN
                IF(CR.EQ.1.0)THEN
                   CR = 0.99
                ENDIF
           TRNS=(1.0/(CR-1.0))*(LOG((E-1.0)/(E*CR-1.0)))
        ELSE
IF(N.EQ.3.AND.CD.LE.1.0.OR.N.EQ.4.AND.CD.LE.1.0)THEN
                CHK2=1.0-E*CR
```

```
              CHK3=(1.0+(1.0/CR)*(LOG(CHK2)))
              IF(CHK2.LE.0.0.OR.CHK3.LE.0.0)THEN
                 CALL EXIT
              ENDIF
           TRNS=-LOG(1.0+(1.0/CR)*(LOG(1.0-E*CR)))
        ELSE
IF(N.EQ.4.AND.CD.GE.1.0.OR.N.EQ.3.AND.CD.GE.1.0)THEN
              CHK4=CR*(LOG(1.0-E)+1.0)
              IF(CHK4.LE.0.0.OR.E.EQ.1.0)THEN
                 CALL EXIT
              ENDIF
           TRNS=(-1.0/CR)*(LOG(CR*LOG(1.0-E)+1.0))
        ELSE
IF(N.EQ.5.AND.CD.LE.1.0.OR.N.EQ.6.AND.CD.LE.1.0)THEN
              CHK2=1.0-E*CR
              CHK3=(1.0+(1.0/CR)*(LOG(CHK2)))
              IF(CHK2.LE.0.0.OR.CHK3.LE.0.0)THEN
                 CALL EXIT
              ENDIF
           TRNS=-LOG(1.0+(1.0/CR)*(LOG(1.0-E*CR)))
        ELSE
IF(N.EQ.6.AND.CD.GE.1.0.OR.N.EQ.5.AND.CD.GE.1.0)THEN
              CHK4=CR*(LOG(1.0-E)+1.0)
              IF(CHK4.LE.0.0.OR.E.EQ.1.0)THEN
                 CALL EXIT
              ENDIF
           TRNS=(-1.0/CR)*(LOG(CR*LOG(1.0-E)+1.0))
        ENDIF
C
        RETURN
        END
C
C                      ****************************
C                      * EFFECTIVENESS FUNCTIONS *
C***********************************************************
C
        FUNCTION EFF(N,NTU,CR,CD)
        REAL NTU
C
        IF(N.EQ.1)THEN
              CHK=EXP(-NTU*(1.0+CR))
              IF(CHK.EQ.1.0)THEN
                 CALL EXIT
              ENDIF
           EFF=(1.0-EXP(-NTU*(1.0+CR)))/(1.0+CR)
        ELSE IF(N.EQ.2)THEN
              IF(CR.EQ.1.0)THEN
                 CR=0.99
              ENDIF
           EFF=(1.0-EXP(-NTU*(1.0-CR)))/(1.0-CR*EXP(-
NTU*(1.0-CR)))
        ELSE
IF(N.EQ.3.AND.CD.LE.1.0.OR.N.EQ.4.AND.CD.LE.1.0)THEN
              CHK5=EXP(-CR*(1.0-EXP(-NTU)))
```

```
                  IF(CHK5.GE.1.0)THEN
                     CALL EXIT
                  ENDIF
              EFF=(1.0/CR)*(1.0-EXP(-CR*(1.0-EXP(-NTU))))
           ELSE
IF(N.EQ.4.AND.CD.GE.1.0.OR.N.EQ.3.AND.CD.GE.1.0)THEN
                  CHK6=EXP((-1.0/CR)*(1.0-EXP(-CR*NTU)))
                  IF(CHK6.GE.1.0)THEN
                     CALL EXIT
                  ENDIF
              EFF=1.0-EXP((-1.0/CR)*(1.0-EXP(-CR*NTU)))
           ELSE
IF(N.EQ.5.AND.CD.LE.1.0.OR.N.EQ.6.AND.CD.LE.1.0)THEN
                  CHK5=EXP(-CR*(1.0-EXP(-NTU)))
                  IF(CHK5.GE.1.0)THEN
                     CALL EXIT
                  ENDIF
              EFF=(1.0/CR)*(1.0-EXP(-CR*(1.0-EXP(-NTU))))
           ELSE
IF(N.EQ.6.AND.CD.GE.1.0.OR.N.EQ.5.AND.CD.GE.1.0)THEN
                  CHK6=EXP((-1.0/CR)*(1.0-EXP(-CR*NTU)))
                  IF(CHK6.GE.1.0)THEN
                     CALL EXIT
                  ENDIF
              EFF=1.0-EXP((-1.0/CR)*(1.0-EXP(-CR*NTU)))
           ENDIF
C
           RETURN
           END
C
C                              *******************
C                              * SIZING PROBLEMS *
C**************************************************************
C
           SUBROUTINE SIZING
           REAL NTU
           DIMENSION C(30),D(10),F(10),G(10)
           COMMON C,Q,N,D,F,G
C
      L = 0
      K = 0
      IF(C(7).LT.0.0.AND.C(8).LT.0.0)THEN
          L = 1
      ENDIF
*
      IF(C(3).LT.0.0)THEN
          K = 1
      ENDIF
C
124   CTD = C(3)-C(1)
      HTD = C(4)-C(2)
*
      IF(L.EQ.1)THEN
          IF(N.EQ.1.OR.N.EQ.2)THEN
```

```
            CALL CONCU
        ELSEIF(N.EQ.3.OR.N.EQ.4)THEN
            CALL CROSSU
        ELSEIF(N.EQ.5.OR.N.EQ.6)THEN
            CALL COMPU
        ENDIF
      ENDIF
*
        CC = C(5)*C(9)
        CH = C(6)*C(10)
        IF(CC.LT.CH)THEN
          CMIN = CC
          CMAX = CH
        ELSE
          CMIN = CH
          CMAX = CC
        ENDIF
C
        CD = CC/CH
        CR = CMIN/CMAX
        QMAX = CMIN * (C(4)-C(1))
        IF(K.EQ.0)THEN
          Q = CC * (C(3)-C(1))
          C(2) = C(4) - Q/CH
        ELSE
          Q = CH * (C(4)-C(2))
          C(3) = C(1) + Q/CC
        ENDIF
*
      CERR = ABS(C(3)-C(1)-CTD)
      HERR = ABS(C(4)-C(2)-HTD)
*
      IF(CERR.GT.1.0.OR.HERR.GT.0.1)THEN
          PRINT *,'       ITERATING...'
          CALL PROPS
          GO TO 124
      ENDIF
*
        E = Q/QMAX
C
        NTU= TRNS(N,E,CR,CD)
C
        C(8) = (NTU*CMIN)/C(7)
*
        IF(C(8).EQ.-1.0)THEN
           C(8) = (NTU*CMIN)/C(7)
        ENDIF
C
        IF(C(7).EQ.-1.0)THEN
           C(7) = (NTU*CMIN)/C(8)
        ENDIF
C
        RETURN
        END
```

```
C
C                             ******************
C                             * RATING PROBLEMS *
C*************************************************************
C
        SUBROUTINE RATING
        REAL NTU
        DIMENSION C(30),D(10),F(10),G(10)
        COMMON C,Q,N,D,F,G
C
      IF(C(7).LT.0.0.AND.C(8).LT.0)THEN
         M = 1
      ELSE
         M = 0
      ENDIF
*
124   CTD = C(3)-C(1)
      HTD = C(4)-C(2)
*
      IF(M.EQ.1)THEN
          IF(N.EQ.1.OR.N.EQ.2)THEN
              CALL CONCU
          ELSEIF(N.EQ.3.OR.N.EQ.4)THEN
              CALL CROSSU
          ELSEIF(N.EQ.5.OR.N.EQ.6)THEN
              CALL COMPU
          ENDIF
          CALL SIZING
      ENDIF
*
        CC = C(5)*C(9)
        CH = C(6)*C(10)
        IF(CC.LT.CH)THEN
          CMIN = CC
          CMAX = CH
        ELSE
          CMIN = CH
          CMAX = CC
        ENDIF
C
        CR = CMIN/CMAX
        NTU = (C(7)*C(8))/CMIN
        CD = CC/CH
C
        E = EFF(N,NTU,CR,CD)
C
        QMAX = CMIN*(C(4)-C(1))
        Q = E*QMAX
        C(2) = C(4) - Q/CH
        C(3) = C(1) + Q/CC
C
      CERR = ABS(C(3)-C(1)-CTD)
      HERR = ABS(C(4)-C(2)-HTD)
*
```

```
      IF(CERR.GT.1.0.OR.HERR.GT.0.1)THEN
          PRINT *,'ITERATING...'
          CALL PROPS
          GO TO 124
      ENDIF
*
        RETURN
        END
C
C                                 ******************
C                                 * ENERGY BALANCE *
C*****************************************************************************
C
        SUBROUTINE ENERGY
        DIMENSION C(30),D(10),F(10),G(10)
        COMMON C,Q,N,D,F,G
C
        IF(C(1).EQ.-1.0)THEN
          W = 1
        ELSE
          W = 0
        ENDIF
*
124   CTD = C(3)-C(1)
      HTD = C(4)-C(2)
*
        IF(W.EQ.1)THEN
          Q = C(10)*C(6)*(C(4)-C(2))
          C(1) = C(3) - Q/(C(5)*C(9))
        ELSE
          Q = C(9)*C(5)*(C(3)-C(1))
          C(4) = C(2) + Q/(C(6)*C(10))
        ENDIF
C
*
      CERR = ABS(C(3)-C(1)-CTD)
      HERR = ABS(C(4)-C(2)-HTD)
*
      IF(CERR.GT.1.0.OR.HERR.GT.0.1)THEN
          PRINT *,'ITERATING...'
          CALL PROPS
          GO TO 124
      ENDIF
*
        IF(C(7).EQ.-1.0.OR.C(8).EQ.-1.0)THEN
          CALL SIZING
        ENDIF
C
        RETURN
        END
C
C                                 ******************
C                                 * RATES PROBLEMS *
```

```
C*****************************************************************
C
        SUBROUTINE RATES
        REAL MR,NTU
        DIMENSION C(30),D(10)
        COMMON C,Q,N,D
C
      IF(C(7).LT.0.0)THEN
         CALL OVERALL
      ENDIF
*
        DT = (C(4)-C(2))/(C(3)-C(1))
        CPR = C(10)/C(9)
        MR = CPR*DT
        CR = MR/CPR
        CD = 1/CPR
C
        IF(C(5).EQ.-1.0.AND.C(6).EQ.-1.0)THEN
           IF(C(7).EQ.-1.0.OR.C(8).EQ.-1.0)THEN
               PRINT *,'NOT ENOUGH INFORMATION'
               RETURN
           ENDIF
C
           IF(CR.GT.1.0)THEN
              CR = 1/CR
              CZ = 1.0
              E = (C(4)-C(2))/(C(4)-C(1))
           ELSE
              CZ = 2
              E = (C(3)-C(1))/(C(4)-C(1))
           ENDIF
C
100        NTU = TRNS(N,E,CR,CD)
           CMIN = C(7)*C(8)/NTU
C
           IF(CZ.EQ.1)THEN
              CH=CMIN
              CC=CH/CR
           ELSE
              CC=CMIN
              CH=CC/CR
           ENDIF
C
           CDO = CD
           CD = CC/CH

IF(CDO.LE.1.0.AND.CD.LE.1.0.OR.CDO.GE.1.0.AND.CD.GE.1.0)THEN
                GOTO 150
           ENDIF
           GOTO 100
C
150        C(5)=CC/C(9)
           C(6)=CH/C(10)
           Q = CH*(C(4)-C(2))
```

```
C
        ELSE IF(C(5).NE.-1.0)THEN
           C(6) = C(5)/MR
           Q = C(6)*C(10)*(C(4)-C(2))
        ELSE IF(C(6).NE.-1.0)THEN
           C(5) = C(6)*MR
           Q = C(5)*C(9)*(C(3)-C(1))
        ENDIF
C
        IF(C(7).EQ.-1.0.OR.C(8).EQ.-1.0)THEN
           CALL SIZING
        ENDIF
C
        RETURN
        END
C
C               ****************************************
C               * SUBPROGRAM ONE - Mi & TiOUT UNKNOWN *
C************************************************************
C
        SUBROUTINE ONE
        DIMENSION C(30)
        REAL NTU, NTU2
        INTEGER W
        COMMON C,Q,N
C
*
        IF(C(5).EQ.-1.0)THEN
          W = 1
        ELSE
          W = 0
        ENDIF
C
124     CTD = C(3)-C(1)
        HTD = C(4)-C(2)
*
        C1 = C(5+W)*C(9+W)
        Q = C1*(C(3+W)-C(1+W))
        CD = C(9)/C(10)
100     E = (C(3+W)-C(1+W))/(C(4)-C(1))
C
C
           CMIN = C1
           NTU = C(7)*C(8)/CMIN
           DIFO = 0.0
           DO 10 K = 2,1001
           CR = 0.001*(K-1.0)
           NTU2 = TRNS(N,E,CR,CD)
           DIF = NTU - NTU2
           DIFC = DIFO/DIF
                IF(DIFC.LT.0.0)THEN
                GOTO 200
                ENDIF
           DIFO = DIF
```

```
10          CONTINUE
C
        CHECKO = 0.0
        DO 15 I=1,1000
           C2 = (1.0-0.001*I)*C1
           CR = C2/C1
           NTU = C(7)*C(8)/C2
           E = EFF(N,NTU,CR,CD)
           CR = (C(3+W)-C(1+W))/(E*(C(4)-C(1)))
           CHECK = C2-CR*C1
           IF(CHECKO/CHECK.LT.0.0)THEN
              GOTO 300
           ENDIF
        CHECKO = CHECK
15      CONTINUE
C
200     C2 = C1/CR
C
300     IF(W.EQ.1)THEN
           CH = C1
           CC = C2
           C(5) = CC/C(9)
           C(3) = C(1) + Q/CC
        ELSE IF(W.EQ.0)THEN
           CC = C1
           CH = C2
           C(6) = CH/C(10)
           C(2) = C(4) - Q/CH
        ENDIF
C
        CDO = CD
        CD = CC/CH

IF(CDO.LE.1.0.AND.CD.LE.1.0.OR.CDO.GE.1.0.AND.CD.GE.1.0)THEN
          GOTO 500
        ELSE
          GOTO 100
        ENDIF
*
      CERR = ABS(C(3)-C(1)-CTD)
      HERR = ABS(C(4)-C(2)-HTD)
*
      IF(CERR.GT.1.0.OR.HERR.GT.0.1)THEN
          PRINT *,'ITERATING'
          CALL PROPS
          GO TO 124
      ENDIF

C
500     RETURN
        END
C
C                 ***************************************
C                 * SUBPROGRAM TWO - Mi & TjOUT UNKNOWN *
```

```
C*************************************************************
C
        SUBROUTINE TWO
        DIMENSION C(30)
        REAL NTU
        INTEGER W
        COMMON C,Q,N
C
*
        IF(C(6).EQ.-1.0)THEN
           W = 1
        ELSE
           W = 0
        ENDIF
C
124   CTD = C(3)-C(1)
      HTD = C(4)-C(2)
*
        TOLD = 0.0
        C1 = C(10-W)*C(6-W)
        CMIN = C1
        CR = 0.5
        CD = C(9)/C(10)
100     NTU = C(7)*C(8)/CMIN
150     E = EFF(N,NTU,CR,CD)
           IF(W.EQ.1)THEN
             E = -1.0*E
           ENDIF
        C(2+W) = C(4-3*W) - (CMIN/C1)*E*(C(4)-C(1))
        C2 = C1*(C(4-W)-C(2-W))/(C(3+W)-C(1+W))
C
        CHECK = ABS(C(2+W) - TOLD)
        IF(CHECK.LT.0.001)THEN
           GOTO 200
        ELSE
           TOLD = C(2+W)
        ENDIF
C
        IF(C2.LT.C1)THEN
           CMIN = C2
           CR = C2/C1
           GOTO 100
        ELSE
           CR = C1/C2
           GOTO 150
        ENDIF
C
200     IF(W.EQ.1)THEN
           CH = C2
           CC = C1
           C(6) = CH/C(10)
        ELSE
           CC = C2
           CH = C1
```

```
          C(5) = CC/C(9)
        ENDIF
C
        CDO = CD
        CD = CC/CH

IF(CDO.LE.1.0.AND.CD.LE.1.0.OR.CDO.GE.1.0.AND.CD.GE.1.0)THEN
            GOTO 300
        ELSE
            GOTO 100
        ENDIF
300     Q = CC*(C(3)-C(1))
*
      CERR = ABS(C(3)-C(1)-CTD)
      HERR = ABS(C(4)-C(2)-HTD)
*
      IF(CERR.GT.1.0.OR.HERR.GT.0.1)THEN
          PRINT *,'ITERATING'
          CALL PROPS
          GO TO 124
      ENDIF
C
        RETURN
        END
C
C                                 ********
C                                 * EXIT *
C************************************************************
C
        SUBROUTINE EXIT
        PRINT *,'PROBLEM OUT OF BOUNDS FOR HEAT EXCHANGE
TYPE'
        RETURN
        END
*************************************************************
*                   *****************************
*                   * OVERALL - Calculation of U *
*************************************************************
SUBROUTINE OVERALL
*
      DIMENSION C(30),D(10),F(10),G(10)
      COMMON C,Q,N,D,F,G
*
      IF(N.EQ.1.OR.N.EQ.2)THEN
          CALL CONCENTRIC
      ELSEIF(N.EQ.3.OR.N.EQ.4)THEN
          CALL CROSSFLOW
      ELSEIF(N.EQ.5.OR.N.EQ.6)THEN
          CALL COMPACT
      ENDIF
*
      RETURN
      END
*                  *************************
```

```
*                        * Subroutine CONCENTRIC *
*******************************************************************
SUBROUTINE CONCENTRIC
*
      DIMENSION C(30),D(10)
      REAL KWALL(7)
      CHARACTER FF*1
      COMMON C,Q,N,D
      DATA KWALL /401,15,999,999,237,999,21.9/
*
      PRINT *, 'WHICH FLUID IS INSIDE'
      PRINT *, '1. COLD FLUID'
      PRINT *, '2. HOT FLUID'
      READ *, D(1)
        IF(D(1).EQ.555)THEN
          GO TO 555
        ELSEIF(D(1).EQ.999)THEN
          CALL EXIT
        ENDIF
*
552     PRINT *, 'INSIDE DIAMETER. (M)'
        READ *,D(2)
        IF(D(2).EQ.555)THEN
          GO TO 555
        ELSEIF(D(2).EQ.999)THEN
          CALL EXIT
        ENDIF
        PRINT *, 'OUTSIDE DIAMETER. (M)'
        READ *,D(3)
*
        IF(D(3).EQ.555)THEN
          GO TO 555
        ELSEIF(D(3).EQ.999)THEN
          CALL EXIT
        ENDIF
*
      IF(D(2).GE.D(3))THEN
          PRINT *,' INSIDE DIAMETER LARGER THAN OUTSIDE'
          GO TO 552
      ENDIF
*
        PRINT *, 'WALL THICKNESS. (M)'
        READ *,D(4)
        IF(D(4).EQ.555)THEN
          GO TO 555
        ELSEIF(D(4).EQ.999)THEN
          CALL EXIT
        ENDIF
*
          DD = D(3)-D(2)
      IF(DD.LE.D(4))THEN
          PRINT *, 'TUBE WALL TOO THICK'
          GO TO 552
      ENDIF
```

```
*
        PRINT *, 'WALL MATERIAL. '
        PRINT *,'(1)  COPPER             (5)  ALUMINUM'
        PRINT *,'(2)  STAINLESS STEEL    (6)  GALVANIZED
STEEL'
        PRINT *,'(3)  HIGH TEMP PVC      (7)  TITANIUM'
        PRINT *,'(4)  LOW TEMP PVC'
        READ *,IWALL
        IF(IWALL.EQ.555)THEN
          GO TO 555
        ELSEIF(IWALL.EQ.999)THEN
          CALL EXIT
        ENDIF
*
          D(7) = KWALL(IWALL)
*
      PRINT *,'DO YOU WANT TO INCLUDE FOULING FACTORS?'
        READ(*,301)FF
301     FORMAT(A1)
*
      FI = 0.0
      FO = 0.0
      IF(FF.EQ.'y'.OR.FF.EQ.'Y')THEN
          PRINT *,'INSIDE FOULING FACTOR'
          READ *,FI
        IF(FI.EQ.555)THEN
          GO TO 555
        ELSEIF(FI.EQ.999)THEN
          CALL EXIT
        ENDIF
          PRINT *,'OUTSIDE FOULING FACTOR'
          READ *,FO
        IF(FO.EQ.555)THEN
          GO TO 555
        ELSEIF(FO.EQ.999)THEN
          CALL EXIT
        ENDIF
      ENDIF
             D(5) = FI
             D(6) = FO
*
      CALL CONCU
*
555   RETURN
      END
*                       ********************
*                       * Subroutine CONCU *
*************************************************************
SUBROUTINE CONCU
*
      DIMENSION C(30),D(10)
      REAL NN,NUI,NUO,NU
      COMMON C,Q,N,D
*
```

```
      IF(D(1).EQ.2)THEN
          J = 1
          K = 0
      ELSEIF(D(1).EQ.1)THEN
          J = 0
          K = 1
      ENDIF
*
      REI = RE(C(5+J),C(11+J),0.0,D(2))
          DOI = D(2) + 2*D(4)
      REO = RE(C(5+K),C(11+K),DOI,D(3))
*

          C1 = 0.023*C(17+J)**.3
          NN = 0.8
      NUI = NU(NN,C1,REI)
*
              C1 = .0248*C(17+K)**.2174
              NN = .8833-.1095*EXP(-C(17+K)/10)
*     IF(REO.LT.2300)THEN
*         NUO = 4.689*(D(2)/D(3))**(0.4133)
*     ELSEIF(REO.GT.10000)THEN
          NUO = NU(NN,C1,REO)
*     ELSE
*         W = (REO-2300)/(10000-2300)
*         NUO = W*NU(NN,C1,REO)+(1-W)*(4.689*(D(2)/D(3))**(-
0.4133))
*     ENDIF
*
      NUI = 1.1 * NUI
      NUO = 1.1 * NUO
*
      HI = NUI*C(13+J)/D(2)
      HO = NUO*C(13+K)/(D(3)-D(2)-(2*D(4)))
*
      R=1/((1/(HI*D(2)))+(D(5)/D(2))+(LOG((D(2)+
     +(2*D(4)))/D(2))/(2*D(7)))+(D(6)/(D(2)+(2*D(4))))+
     +(1/(HO*(D(2)+(2*D(4))))))
      C(7) = R/(D(2)+D(4))
*
        C(24) = R*100/(HI*D(2))
        C(25) = R*100/(HO*(D(2)+(2*D(4))))
        C(26) = R*100*(LOG((D(2)+(2*D(4)))/D(2))/(2*D(7)))
        C(27) = R*100*D(5)/D(2)
        C(28) = R*100*D(6)/(D(2)+(2*D(4)))
*
        IF(D(1).EQ.1)THEN
                C(19) = REI
                C(20) = REO
                C(21) = NUI
                C(22) = NUO
        ELSEIF(D(1).EQ.2)THEN
                C(19) = REO
                C(20) = REI
```

```
                C(21) = NUO
                C(22) = NUI
        ENDIF
*
555   RETURN
      END
*                      **********************
*                      * Subroutine COMPACT *
**************************************************************
SUBROUTINE COMPACT
*
      DIMENSION C(30),F(10)
      COMMON C,Q,N,F
      COMMON /CHEDAT/ E(15,12)
      REAL KWALL(7),KW
      INTEGER CHE
      CHARACTER*1 FF
      DATA KWALL /401,15,999,999,237,999,21.9/
*


*
      PRINT *,' '
      PRINT *,'CONSULT USER MANUAL TO SELECT COMPACT HEAT
EXCHANGER'
*
      PRINT *,' '
      PRINT *,'ENTER DESIRED COMPACT HEAT EXCHANGER TYPE'
*
      PRINT *,' 1.  CF-7.34'
      PRINT *,' 2.  CF-8.72'
      PRINT *,' 3.  CF-8.72(c)'
      PRINT *,' 4.  CF-7.0-5/8J'
      PRINT *,' 5.  CF-8.7-5/8J(a)'
      PRINT *,' 6.  CF-8.7-5/8J(b)'
      PRINT *,' 7.  CF-9.05-3/4J(a)'
      PRINT *,' 8.  CF-9.05-3/4J(b)'
      PRINT *,' 9.  CF-9.05-3/4J(c)'
      PRINT *,'10.  CF-9.05-3/4J(d)'
      PRINT *,'11.  CF-9.05-3/4J(e)'
      PRINT *,'12.  CF-8.8-1.0-J(a)'
      PRINT *,'13.  CF-8.8-1.0-J(b)'
      PRINT *,'14.  8.0-3/8T'
      PRINT *,'15.  7.75-5/8T'
*
      READ *,CHE
        IF(CHE.EQ.555)THEN
          GO TO 555
        ELSEIF(CHE.EQ.999)THEN
          CALL EXIT
        ENDIF
*
*     Dt, Df, a, b, Fin pitch, Dh, t, sigma, alpha, Af/A
*
*
```

```
553     PRINT *, 'INSIDE DIAMETER. (M)'
        READ *,DI
*
        IF(DI.EQ.555)THEN
          GO TO 555
        ELSEIF(DI.EQ.999)THEN
          CALL EXIT
        ENDIF
*
      IF(DI.GE.E(CHE,1))THEN
          PRINT *, 'INSIDE DIAMETER TOO LARGE'
          GO TO 553
      ENDIF
*
        IF(N.EQ.5)THEN
          J = 0
          K = 1
        ELSEIF(N.EQ.6)THEN
          J = 1
          K = 0
        ENDIF
*
        PRINT *, 'CORE MATERIAL. '
        PRINT *,'(1)  COPPER            (5)  ALUMINUM'
        PRINT *,'(2)  STAINLESS STEEL   (6)  GALVANIZED
STEEL'
        PRINT *,'(3)  HIGH TEMP PVC     (7)  TITANIUM'
        PRINT *,'(4)  LOW TEMP PVC'
        READ *,IWALL
        IF(IWALL.EQ.555)THEN
          GO TO 555
        ELSEIF(IWALL.EQ.999)THEN
          CALL EXIT
        ENDIF
*
        KW = KWALL(IWALL)
*
      PRINT *,'ENTER FRONTAL AREA OF COMPACT HEAT EXCHANGER'
      READ *,AFR
        IF(AFR.EQ.555)THEN
          GO TO 555
        ELSEIF(AFR.EQ.999)THEN
          CALL EXIT
        ENDIF
*
      PRINT *,'DO YOU WANT TO INCLUDE FOULING FACTORS?'
        READ(*,301)FF
301     FORMAT(A1)
*
      FI = 0.0
      FO = 0.0
      IF(FF.EQ.'y'.OR.FF.EQ.'Y')THEN
          PRINT *,'INSIDE FOULING FACTOR'
          READ *,FI
```

```
          IF(FI.EQ.555)THEN
            GO TO 555
          ELSEIF(FI.EQ.999)THEN
            CALL EXIT
          ENDIF
            PRINT *,'OUTSIDE FOULING FACTOR'
            READ *,FO
          IF(FO.EQ.555)THEN
            GO TO 555
          ELSEIF(FO.EQ.999)THEN
            CALL EXIT
          ENDIF
        ENDIF
*
        F(1) = CHE
        F(2) = DI
        F(3) = KW
        F(4) = AFR
        F(5) = FI
        F(6) = FO
*
        CALL COMPU
*
555     RETURN
        END
*                       *******************
*                       * Subroutine COMPU *
************************************************************
SUBROUTINE COMPU
*
        DIMENSION C(30),F(10)
        COMMON C,Q,N,F
        COMMON /CHEDAT/ E(15,12)
        INTEGER CHE
        REAL M
*
          IF(N.EQ.5)THEN
            J = 0
            K = 1
          ELSEIF(N.EQ.6)THEN
            J = 1
            K = 0
          ENDIF
*
        FI = F(5)
        FO = F(6)
*
        CHE = F(1)
        G = C(5+K)/(E(CHE,8)*F(4))
*
        DUM1 = G*3.1415927/4.0
        DUM2 = 1/(E(CHE,6))
*
        REO = RE(DUM1,C(11+K),DUM2,0.0)
```

```
*
      JH = E(CHE,11)*REO**E(CHE,12)
*
      HO = JH*G*C(9+K)/(C(17+K)**.667)
      NUO = HO*E(CHE,6)/C(13+K)
*
        LC = (E(CHE,2)-E(CHE,1)+E(CHE,7))/2
        AP = LC*E(CHE,7)
*
        M = SQRT(LC*HO/(F(3)*AP))
*
        EFF = TANH(M*LC)/(M*LC)
        EFFO = 1 - E(CHE,10)*(1-EFF)
*
        ACAH = (F(2)/E(CHE,1))*(1-E(CHE,10))
*
      REI = RE(C(5+J),C(11+J),0.0,F(2))
*
          C1 = 0.023*C(17+J)**.3
          NN = 0.8
      NUI = NU(NN,C1,REI)
*
      HI = NUI*C(13+J)/F(2)
*
        U = 1/((1/(HI*ACAH)) + (1/(EFFO*HO)) + (FO/EFFO) +
(FI/ACAH))
*
        C(7) = U
*
        IF(N.EQ.5)THEN
                C(19) = REI
                C(20) = REO
                C(21) = NUI
                C(22) = NUO
        ELSEIF(N.EQ.6)THEN
                C(19) = REO
                C(20) = REI
                C(21) = NUO
                C(22) = NUI
        ENDIF
*
555   RETURN
      END
*                        *************************
*                        * Subroutine CROSSFLOW *
**************************************************************
      SUBROUTINE CROSSFLOW
*
      DIMENSION C(30),G(10)
      REAL KWALL(7)
      CHARACTER FF*1
      COMMON C,Q,N,G
      DATA KWALL /401,15,999,999,237,999,21.9/
*
```

```
552     PRINT *, 'OUTSIDE DIAMETER. (M)'
        READ *,DO
*
        IF(DO.EQ.555)THEN
          GO TO 555
        ELSEIF(DO.EQ.999)THEN
          CALL EXIT
        ENDIF
*
        PRINT *, 'WALL THICKNESS. (M)'
        READ *,T
        IF(T.EQ.555)THEN
          GO TO 555
        ELSEIF(T.EQ.999)THEN
          CALL EXIT
        ENDIF
*
      IF(DO.LE.T)THEN
          PRINT *, 'TUBE WALL TOO THICK'
          GO TO 552
      ENDIF
*
      PRINT *, 'EXTERNAL FREE FLOW AREA (sq. m)'
      READ *,ARFF
*
        PRINT *, 'WALL MATERIAL. '
        PRINT *,'(1)  COPPER          (5)  ALUMINUM'
        PRINT *,'(2)  STAINLESS STEEL (6)  GALVANIZED
STEEL'
        PRINT *,'(3)  HIGH TEMP PVC   (7)  TITANIUM'
        PRINT *,'(4)  LOW TEMP PVC'
        READ *,IWALL
        IF(IWALL.EQ.555)THEN
          GO TO 555
        ELSEIF(IWALL.EQ.999)THEN
          CALL EXIT
        ENDIF
*
          KW = KWALL(IWALL)
*
      PRINT *,'DO YOU WANT TO INCLUDE FOULING FACTORS?'
        READ(*,301)FF
301     FORMAT(A1)
*
      FI = 0.0
      FO = 0.0
      IF(FF.EQ.'y'.OR.FF.EQ.'Y')THEN
          PRINT *,'INSIDE FOULING FACTOR'
          READ *,FI
        IF(FI.EQ.555)THEN
          GO TO 555
        ELSEIF(FI.EQ.999)THEN
          CALL EXIT
        ENDIF
```

```
          PRINT *,'OUTSIDE FOULING FACTOR'
          READ *,FO
        IF(FO.EQ.555)THEN
          GO TO 555
        ELSEIF(FO.EQ.999)THEN
          CALL EXIT
        ENDIF
      ENDIF
*
      G(1) = DO
      G(2) = T
      G(3) = KW
      G(4) = FI
      G(5) = FO
      G(6) = ARFF
*
      CALL CROSSU
*
555   RETURN
      END
*                       *********************
*                       * Subroutine CROSSU *
**************************************************************
SUBROUTINE CROSSU
*
      DIMENSION C(30),G(10)
      REAL NN,NUI,NUO,NU
      COMMON C,Q,N,G
*
      IF(N.EQ.4)THEN
          J = 1
          K = 0
      ELSE
          J = 0
          K = 1
      ENDIF
*
      DO = G(1)
      T  = G(2)
      KW = G(3)
      FI = G(4)
      FO = G(5)
      ARFF = G(6)
*
          DI = DO - 2*T
      REI = RE(C(5+J),C(11+J),0.0,DI)
*
          DOD = 4*ARFF/(3.1415927*DO)
      REO = RE(C(5+K),C(11+K),0.0,DOD)
*
          C1 = 0.023*C(17+J)**.333
          NN = 0.8
      NUI = NU(NN,C1,REI)
```

```
*
      ARG1 = 0.62*(REO**.5)*(C(17+K)**.333)
      ARG2 = (1. + (0.4/C(17+K))**.666)**.25
      ARG3 = (1. + (REO/282000.)**.625)**.8
*
      NUO = 0.3 + ARG1*ARG3/ARG2
*
      HI = NUI*C(13+J)/DI
      HO = NUO*C(13+K)/DO
*

R=1/((1/(HI*DI))+(FI/DI)+(LOG(DO/DI)/(2*KW))+(FO/DO)+(1/(HO*
DO)))
      C(7) = R/(DI+T)
*
*
        IF(N.EQ.3)THEN
                C(19) = REI
                C(20) = REO
                C(21) = NUI
                C(22) = NUO
        ELSEIF(N.EQ.4)THEN
                C(19) = REO
                C(20) = REI
                C(21) = NUO
                C(22) = NUI
        ENDIF
*
555   RETURN
      END
*************************************************************
*                     ***************************
*                     * Function NU - Nusselt # *
*************************************************************
FUNCTION NU(NN,C1,RE)
      DIMENSION C(30)
      COMMON C,Q,N
*
      REAL NN,NU
*
      IF(RE.LE.2300)THEN
          NU = 4.01
      ELSEIF(RE.GE.10000)THEN
          NU = C1*RE**NN
      ELSE
          WT = (RE-2300)/(10000-2300)
          NU = WT*C1*RE**NN+(1-WT)*4.01
      ENDIF
*
      RETURN
      END
*                     *****************************
*                     * Function RE - Reynold's # *
```

```
*****************************************************************
FUNCTION RE(M,MU,DI,DO)
*
      REAL M,MU
*
      DATA PI /3.1415927/
      RE = 4*M/(PI*MU*(DO+DI))
*
      RETURN
      END
*****************************************************************
*                      ***********************
*                      * Property VISCOSITY *
*****************************************************************
FUNCTION VISCOSITY(FL,T)
*
        INTEGER FL
*
      IF(FL.EQ.1)THEN
           IF(T.LT.335)THEN
               VISCOSITY = -3344478/(T**4)+9.435E11/(T**6)
           ELSEIF(T.LT.430)THEN
               VISCOSITY = 8082231.1/(T**4)-3.22E11/(T**6)
           ELSEIF(T.LT.530)THEN
               VISCOSITY = 11659287/(T**4)-1.08E12/(T**6)
           ELSEIF(T.LT.620)THEN
               VISCOSITY = 0.0002106-8.66E-8*T-2.16E-10*T*T
           ELSE
               VISCOSITY = -0.00605187+2.00636E-5*T-
1.6429795E-8*T*T
           ENDIF
      ELSEIF(FL.EQ.2)THEN
           VISCOSITY = -9.87E-7+3.606E-8*T+1.165E-14*T*T
      ELSEIF(FL.EQ.3)THEN
           IF(T.LT.800)THEN
              VISCOSITY = 1.162E-6 + 6.499E-8*T - 2.61E-
11*T*T
           ELSE
              VISCOSITY = 1.541E-5 + 2.714E-8*T - 2.21E-
13*T*T
           ENDIF
       ELSEIF(FL.EQ.4)THEN
           IF(T.LT.300)THEN
              VISCOSITY = 1.175E10*EXP(-8.*T/100.)
           ELSEIF(T.LT.340)THEN
              VISCOSITY = 67594510*EXP(-6.25*T/100.)
           ELSEIF(T.LT.380)THEN
              VISCOSITY = 9260.8987*EXP(-3.55*T/100.)
           ELSE
              VISCOSITY = 106.53899*EXP(-2.35*T/100.)
           ENDIF
       ENDIF
      RETURN
      END
```

```
*                   ******************
*                   * Property TCOND *
************************************************************
FUNCTION TCOND(FL,T)
*
        INTEGER FL
*
      IF(FL.EQ.1)THEN
          TCOND = -0.338638 + 0.0049804*T - 6.03E-6*T*T
      ELSEIF(FL.EQ.2)THEN
          TCOND = -0.002674 + 6.635E-5*T + 1.382E-8*T*T
      ELSEIF(FL.EQ.3)THEN
          IF(T.LT.1300)THEN
              TCOND = 0.0013906 + 8.902E-5*T - 2.31E-8*T*T
          ELSE
              TCOND = 0.167959 - 0.000148*T + 6.653E-8*T*T
          ENDIF
      ELSEIF(FL.EQ.4)THEN
          TCOND = 0.1799465 - 0.000137*T + 5.66E-8*T*T
      ENDIF
*
      RETURN
      END
*                   ***************
*                   * Property CP *
************************************************************
FUNCTION CP(FL,T)
*
        INTEGER FL
*
      IF(FL.EQ.1)THEN
          IF(T.LT.440)THEN
              CP = 5310.7871-7.14145*T+.0112792*T*T
          ELSEIF(T.LT.590)THEN
              CP = 22291.184-79.47527*T+.0883161*T*T
          ELSEIF(T.LT.635)THEN
              CP = 1490286.2-5004.324*T+4.2195626*T*T
          ELSE
              CP = 441214463-1386078*T+1088.627*T*T
          ENDIF
      ELSEIF(FL.EQ.2)THEN
          IF(T.LT.550)THEN
              CP = 4413.7206 - 10.48318*T + 0.0112565*T*T
          ELSE
              CP = 1762 + 0.2966667*T + 0.0002381*T*T
          ENDIF
      ELSEIF(FL.EQ.3)THEN
          IF(T.LT.800)THEN
              CP = 1040.0505 - 0.217*T + 3.83E-4*T*T
          ELSEIF(T.LT.1700)THEN
              CP = 951.0051 + 0.1922*T - 4.83E-6*T*T
          ELSE
              CP = 2988.5022 - 2.013*T + 5.916E-4*T*T
          ENDIF
```

```
      ELSEIF(FL.EQ.4)THEN
          CP = 789.56439 + 3.3139543*T + 0.001388*T*T
      ENDIF
*
      RETURN
      END
*                               *********************
*                               * Property DENSITY *
*************************************************************
FUNCTION DENSITY(FL,T)
*
        INTEGER FL
*
      IF(FL.EQ.1)THEN
          IF(T.LT.510)THEN
              DENSITY = .9853977+.0215446*EXP((T-275)/100)
          ELSEIF(T.LT.610)THEN
              DENSITY = 1.0035478+.2153583*EXP((T-510)/100)
          ELSE
              DENSITY = 211.70187-.688324*T+.0005638*T*T
          ENDIF
      ELSEIF(FL.EQ.2)THEN
          DENSITY = 1.2123416 - 0.002121*T + 1.183E-6*T*T
      ELSEIF(FL.EQ.3)THEN
          DENSITY = 350.0/T
      ELSEIF(FL.EQ.4)THEN
          DENSITY = 1059.7622 - 0.583379*T - 1.25E-5*T*T
      ENDIF
*
      RETURN
      END
*************************************************************
*                         ********************************
*                         * Compact Heat Exchanger Data *
*************************************************************
      BLOCK DATA
      COMMON /CHEDAT/ E(15,12)
*
        DATA E /

1.00965,.00965,.01067,.01638,.01638,.01638,.01966,.01966,.01
966,
     +.01966,.01966,.026,.026,.01021,.01717,

2.0234,.0234,.0219,.0285,.0285,.0285,.0372,.0372,.0372,.0372
,.0372,
     +.0441,.0441,.0237,.04128,

3.0248,.0248,.0248,.0313,.0313,.0469,.0395,.0503,.0692,.0629
,.0503,
     +.0498,.0782,.0254,.0381,

4.0203,.0203,.0203,.0343,.0343,.0343,.0445,.0445,.0445,.0203
,.0349,
```

```
     +.0524,.0524,.022,.0445,

5289,343,343,276,343,343,356,356,356,356,356,346,346,315,305
,

6.00475,.00393,.00443,.00668,.00548,.01167,.00513,.00818,.01
359,

+.00485,.00643,.00589,.01321,.00363,.00348,.00046,.00046,.00
048,

7.00025,.00025,.00025,.00031,.00031,.00031,.00031,.00031,.00
031,

+.00031,.00033,.00041,.538,.524,.494,.449,.443,.628,.455,.57
2,.688,
     8.537,.572,.439,.642,.534,.481,

9459,535,446,269,324,216,354,279,203,443,354,299,191,587,554
,

A.892,.91,.876,.83,.862,.862,.835,.835,.835,.835,.835,.825,.
825,

B.913,.95,.34,.269,.213,.1615,.096,.133,.092,.0939,.0845,.10
36,
     +.170,.162,.255,.12445,.0527,-.485,-.434,-.394,-.366,-
.294,-.313,
     C-.339,-.323,-.292,-.356,-.387,-.387,-.4056,-.359,-.26/
      END
************************************************************
************************************************************
```

APPENDIX D: USER'S GUIDE

General Comments

Program TSCAD is a heat exchanger analysis and design program. The user inputs all known data, signifying unknown data by entering "-1.0" as its value. If enough information is available, TSCAD will calculate the unknown data and display the results.

The total data set includes inlet and outlet temperatures for each fluid stream, both mass flow rates, and the parameters U and A. If U and A are unknown, the overall heat transfer coefficient U is calculated. This requires additional information about the geometry of the heat exchanger. In this case, up to three unknowns can be solved for, two of which must be U and A. In cases where U is not calculated independently (either U or A is known) up to two unknowns can be computed.

Entering "555" at any time during the initial data entry will restart the process without computing any results. Entering "999" will cause the program to ask the user whether he or she wants to stop the program. If the program is followed normally, the results will be computed and displayed.

Property information and, in the case of U and A both unknown, parameters such as the Nusselt number and the Reynold's number will also be displayed. After displaying results, the program will ask whether or not to continue. Entering anything other than "Y" or "y" will terminate the program.

Limitations of the Program

* Only concentric tube, cross-flow over a single pipe, and compact heat exchangers can be analyzed.
* Only thermal analysis of heat exchanger is performed. No consideration is given to hydraulic design.

How to Use the Program

Input of the six operating conditions and the two important parameters is first requested. These are the four temperatures, the two mass flow rates and the area and overall heat transfer coefficient. Known data should be given in SI units. Unknown data should be signified by entering "-1.0".

An example of the input for the classic rating problem of both outlet temperatures unknown is given below.

Input screens:

```
**********************************************************************
Program TSCAD uses SI units to calculate the unknown parameters.
Enter "-1.0" for any unknown data. If there are too many unknowns,
the program will state that the problem is unsolvable and prompt
for either continuation or termination of the program.


ENTER PROBLEM PARAMETERS.
ENTER -1.0 IF NOT KNOWN.
ENTER 555 TO RESTART.
ENTER 999 TO QUIT.

(1) WATER              (3) AIR
(2) STEAM              (4) OIL

COLD FLUID
1
HOT FLUID
1
COLD INLET TEMP. (C)
35
COLD OUTLET TEMP. (C)
-1.0

COLD MASS FLOW RATE (kg/s)
.02333
HOT INLET TEMP. (C)
200
HOT OUTLET TEMP. (C)
-1.0
HOT MASS FLOW RATE (kg/s)
.01167
OVERALL COEFFICIENT (W K/sq. m)
180
SURFACE AREA (sq. m)
0.33
(1) DOUBLE PIPE PARALLEL FLOW
(2) DOUBLE PIPE COUNTERFLOW
(3) CROSSFLOW - COLD UNMIXED/HOT MIXED
(4) CROSSFLOW - COLD MIXED/HOT UNMIXED
(5) COMPACT HEAT EXCHANGER - HOT FLUID FIN-SIDE
(6) COMPACT HEAT EXCHANGER - COLD FLUID FIN-SIDE
CHOOSE A HEAT EXCHANGER TYPE.




2
```

Output screen:

```
ITERATING...
          TCI  =     308.0 K          CPC  = 4183.85 kJ/kg*K
          THO  =     371.6 K          CPH  = 4306.45 kJ/kg*K
          TCO  =     360.2 K          VISC =    0.000410 N*s/sq m
          THI  =     473.0 K          VISH =    0.000197 N*s/sq m
          MC   =       0.023 kg/s     KC   =    0.652 W/m*K
          MH   =       0.012 kg/s     KH   =    0.689 W/m*K
          U    =     180. W/K*sq m    DENC =    1.024 kg/cu m
          A    =       0.3300 sq m    DENH =    1.079 kg/cu m
          Q    =    5098.1 W
          PrC  =       2.629
          PrH  =       1.233
          ReC  =       0.
          ReH  =       0.
          NuC  =       0.0
          NuH  =       0.0

 NORMALIZED RESISTANCES:
          INTERNAL CONVECTION  = ????%
          EXTERNAL CONVECTION  = ????%
          WALL CONDUCTION      = ????%
          INTERNAL FOULING     = ????%
          EXTERNAL FOULING     = ????%
DO YOU WISH TO CONTINUE?
```

The example shown above is problem 11.26(b) from Incropera and DeWitt, Introduction to Heat Transfer, 1st ed. The solution is $T_{c,o}$=87.2^{o}C and $T_{h,o}$=98.6^{o}C compared to 87.2^{o}C and 98.4^{o}C in solution manual. Notice that all inputted data is presented with the calculated results. Also presented is the total heat transfer rate, Q, and the property data which has been calculated as a function of the average temperature of each fluid stream. Finally, notice that the Reynold's numbers and Nusselt numbers for this problem are given as zero. This will be the case for all problems in which the overall heat transfer coefficient is not calculated independently (either U or A is known).

Compact Heat Exchanger Data:

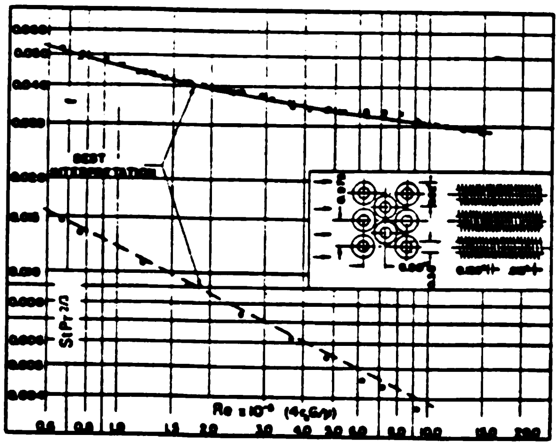
The data which follows corresponds to the set of compact heat exchangers that can be analyzed using TSCAD. All of these compact heat exchangers are finned circular tube types. Problems requiring other compact heat exchangers will require additional data to be added to the program. Data taken from Compact Heat Exchangers, by Kays and London.

Compact Heat Exchanger Types:

1. CF-7.34
2. CF-8.72
3. CF-8.72(c)
4. CF-7.0-5/8J
5. CF-8.7-5/8J(a)
6. CF-8.7-5/8J(b)
7. CF-9.05-3/4J(a)
8. CF-9.05-3/4J(b)
9. CF-9.05-3/4J(c)
10. CF-9.05-3/4J(d)
11. CF-9.05-3/4J(e)
12. CF-8.8-1.0-J(a)
13. CF-8.8-1.0-J(b)
14. 8.0-3/8T
15. 7.75-5/8T

CF-7.34

Finned circular tubes, surface CF-7.34.

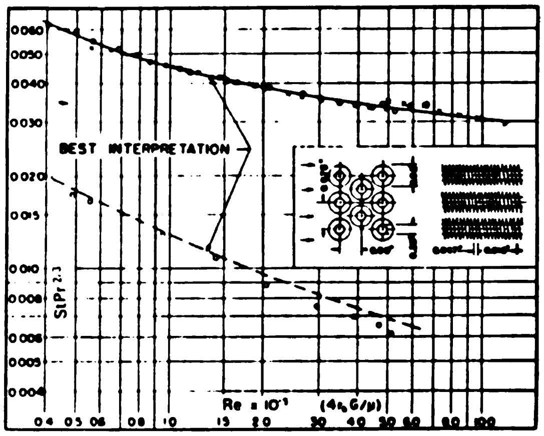


Tube outside diameter = 0.38 in = 9.65 × 10^{-3} m
Fin pitch = 7.34 per in = 289 per m
Flow passage hydraulic diameter, $4r_h$ = 0.0154 ft = 4.75 × 10^{-3} m
Fin thickness (average)† = 0.018 in, aluminum = 0.46 × 10^{-3} m
Free-flow area/frontal area, σ = 0.538
Heat transfer area/total volume, α = 140 ft^2/ft^3 = 459 m^2/m^3
Fin area/total area = 0.882
Note: Experimental uncertainty for heat transfer results possibly somewhat greater than the nominal ±5% quoted for the other surfaces because of the necessity of estimating a contact resistance in the bimetal tubes.

†Fins slightly tapered.

CF-8.72

Finned circular tubes, surface CF-8.72.



Tube outside diameter = 0.38 in = 9.65×10^{-3} m

Fin pitch = 8.72 per in = 343 per m

Flow passage hydraulic diameter, $4r_h$ = 0.01288 ft = 3.929×10^{-3} m

Fin thickness (average)† = 0.018 in, aluminum = 0.46×10^{-3} m

Free flow area/frontal area, σ = 0.524

Heat transfer area/total volume, α = 163 ft^2/ft^3 = 535 m^2/m^3

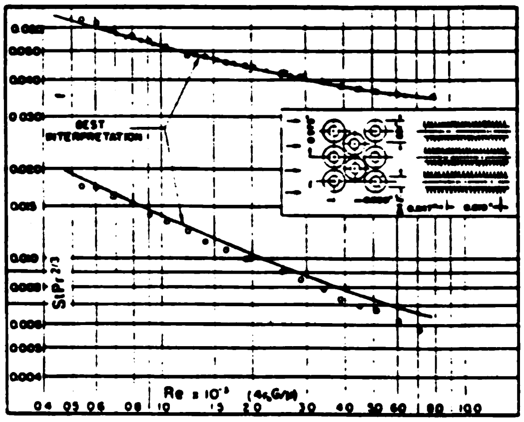
Fin area/total area = 0.910

Note: Experimental uncertainty for heat transfer results possibly somewhat greater than the nominal ±5% quoted for the other surfaces because of the necessity of estimating a contact resistance in the bimetal tubes.

†Fins slightly tapered.

CF-8.72(c)

Finned circular tubes, surface CF-8.72(c).



Tube outside diameter = 0.42 in = 10.67×10^{-3} m

Fin pitch = 8.72 per in = 343 per m

Flow passage hydraulic diameter, $4r_h$ = 0.01452 ft = 4.425×10^{-3} m

Fin thickness (average) † = 0.019 in, copper = 0.48×10^{-3} m

Free-flow area/frontal area, σ = 0.494

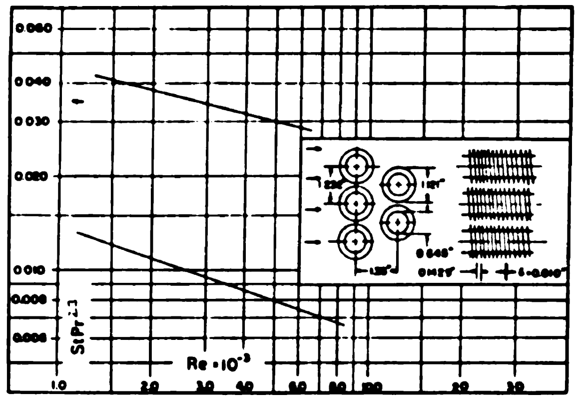
Heat transfer area/total volume, α = 136 ft^2/ft^3 = 446 m^2/m^3

Fin area/total area = 0.876

†Fins slightly tapered.

CF-7.0-5/8J

Finned circular tubes, surface CF-7.0-5/8J. *(Data of Jameson.)*



Tube outside diameter = 0.645 in = 16.38×10^{-3} m

Fin pitch = 7.0 per in = 276 per m

Flow passage hydraulic diameter, $4r_h$ = 0.0219 ft = 6.68×10^{-3} m

Fin thickness = 0.010 in = 0.25×10^{-3} m

Free-flow area/frontal area, σ = 0.449

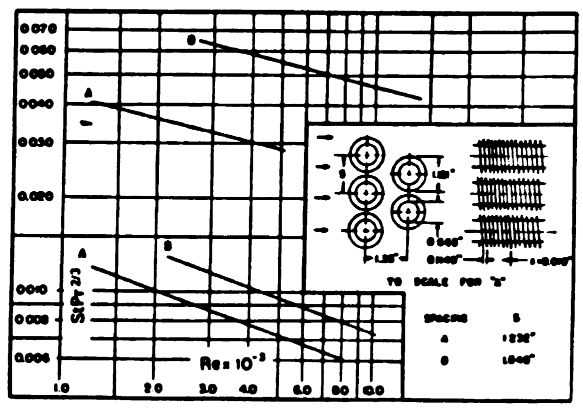
Heat transfer area/total volume, α = 82 ft^2/ft^3 = 269×10^{-3} m

Fin area/total area = 0.830

Note: Minimum free-flow area is in spaces transverse to flow.

CF-8.7-5/8J

Finned circular tubes, surface CF-8.7-5/8J. *(Data of Jameson.)*



Tube outside diameter = 0.645 in = 16.38×10^{-3} m

Fin pitch = 8.7 per in = 343 per m

Fin thickness = 0.010 in = 0.25×10^{-3} m

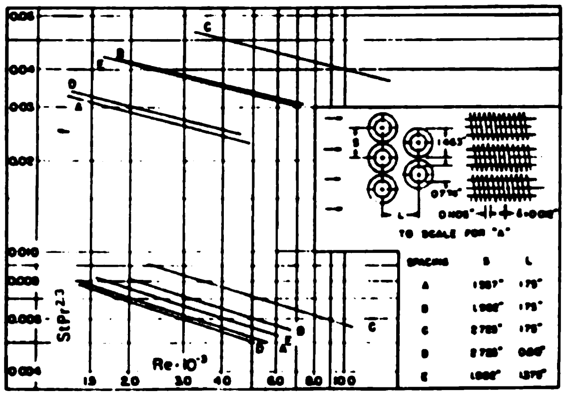
Fin area/total area = 0.862

	A	B	A	B	
Flow passage hydraulic diameter, $4r_h$	= 0.01797	0.0383 ft	5.48×10^{-3}	11.67×10^{-3}	m
Free-flow area/frontal area, σ	= 0.443	0.628			
Heat transfer area/total volume, α	= 98.7	65.7 ft^2/ft^3	324	216	m^2/m^3

Note: Minimum free-flow area is in spaces transverse to flow.

CF-9.05-3/4J

Finned circular tubes, surface CF-9.05-3/4J. *(Data of Jameson.)*



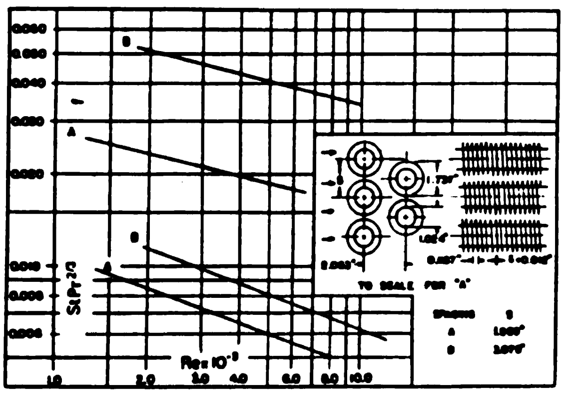
Tube outside diameter = 0.774 in = 19.66 x 10^{-3} m
Fin pitch = 9.05 per in = 356 per m
Fin thickness = 0.012 in = 0.305 x 10^{-3} m
Fin area/total area = 0.835

	A	*B*	*C*	*D*	*E*
Flow passage hydraulic diameter, $4r_h$ =	0.01681	0.02685	0.0445	0.01587	0.02108 ft
=	5.131 x 10^{-3}	8.179 x 10^{-3}	13.59 x 10^{-3}	4.846 x 10^{-3}	6.426 x 10^{-3} m
Free-flow area/frontal area, σ =	0.455	0.572	0.688	0.537	0.572
Heat transfer area/total volume, α =	108	85.1	61.9	135	108 ft^2/ft^3
=	354	279	203	443	354 m^2/m^3

Note: Minimum free-flow area in all cases occurs in the spaces transverse to the flow, except for *D*, in which the minimum area is in the diagonals.

CF-8.8-1.0-J

Finned circular tubes, surface CF-8.8-1.0J. *(Data of Jameson.)*



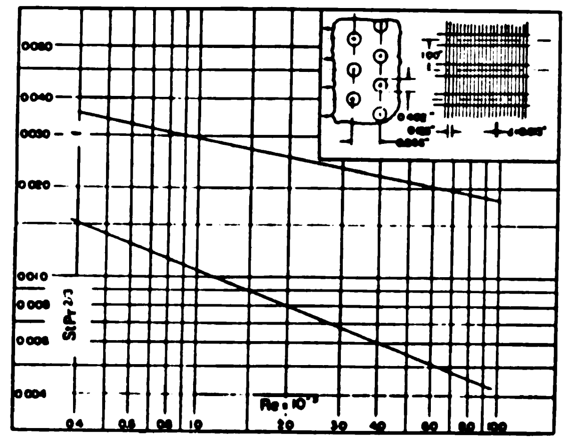
Tube outside diameter = 1.024 in = 26.01 x 10^{-3} m
Fin pitch = 8.8 per in = 346 per m
Fin thickness = 0.012 in = 0.305 x 10^{-3} m
Fin area/total area = 0.825

Flow passage hydraulic diameter, $4r_h$	= 0.01927	0.0443 ft
	= 5.893 x 10^{-3}	13.21 x 10^{-3} m
Free-flow/frontal area, σ	= 0.439	0.642
Heat transfer area/total volume, α	= 91.2	58.1 ft^2/ft^3
	= 299	191 m^2/m^3

Note: minimum free-flow is in spaces transverse to flow.

8.0-3/8T

Finned circular tubes, surface 8.0-3/8T.



Tube outside diameter = 0.402 in = 10.2×10^{-3} m

Fin pitch = 8.0 per in = 315 per m

Flow passage hydraulic diameter, $4r_h$ = 0.01192 ft = 3.632×10^{-3} m

Fin thickness = 0.013 in = 0.33×10^{-3} m

Free flow area/frontal area, σ = 0.534

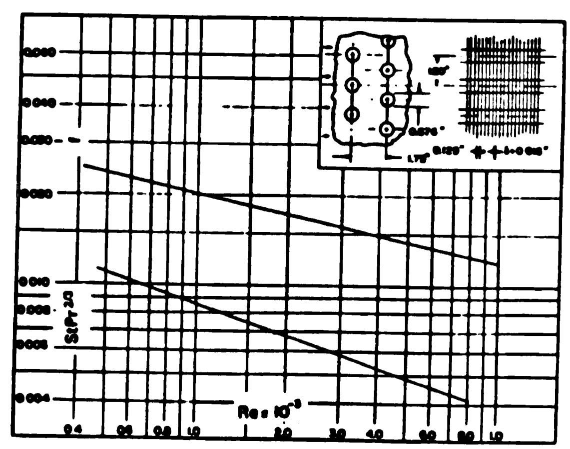
Heat transfer area/total volume, α = 179 ft^2/ft^3 = 587 m^2/m^3

Fin area/total area = 0.913

Note: Minimum free-flow area in spaces transverse to flow.

7.75-5/8T

Finned circular tubes, surface 7.75-5/8T.



Tube outside diameter = 0.676 in = 17.17×10^{-3} m

Fin pitch = 7.75 per in = 305 per m

Flow passage hydraulic diameter, $4r_h$ = 0.0114 ft = 3.48×10^{-3} m

Fin-flow area/frontal area, σ = 0.481

Heat transfer area/total volume, α = 169 ft^2/ft^3 = 554 m^2/m^3

Fin area/total area = 0.950

Note: Minimum free-flow area in spaces transverse to flow.

LIST OF REFERENCES

1. Dittus, F.W., and L.M.K. Boelter, University of California, Berkeley, Publications on Engineering, vol. 2, 1930, p.443.

2. Churchill, S.W., and M. Bernstein, Journal of Heat Transfer, 99, 300, 1977.

3. Kays, W.M., and A.L. London, Compact Heat Exchangers, 3rd ed., McGraw-Hill, New York, 1984.

4. Kays, W.M., and E.Y.Leung, International Journal of Heat and Mass Transfer, vol. 6, p.537, 1963.

5. Liley, P.E., Steam Tables in SI Units Private Communication, School of Mechanical Engineering, Purdue University, West Lafayette, Ind., Dec. 1978.

6. American Society of Heating, Refrigerating and Air Conditioning Engineers, ASHRAE Handbook of Fundamentals, ASHRAE, New York, 1981.

7. Vargaftik, N.B., Tables of Thermophysical Properties of Liquids and Gases, 2nd ed., Hemisphere Publishing Corporation, New York, 1975.

8. Eckert, E.R.G., and R.M. Drake, Analysis of Heat and Mass Transfer, McGraw-Hill, New York, 1972.

9. Standards of the Tubular Exchange Manufacturers Association, 6th ed., Tubular Exchanger Manufacturer's Association, New York, 1978.

10. Incropera, F.P. and D.P. DeWitt, Introduction to Heat Transfer, John Wiley and Sons, Inc., 1985

11. Hodge, B.K., A Suite of Personal Computer Programs For Undergraduate Education in the Thermal Sciences, Numerical Heat Transfer With Personal Computers and Supercomputing, American Society of Mechanical Engineers, 1989.

12. Wang, Y., and Soler, A.I., An Expert System for the Mechanical Design of Shell and Tube Heat Exchangers, Heat Transfer Equipment Fundamentals, Design, Applications, and Operating Problems, American Society of Mechanical Engineers, 1989.

13. Knoll, K., Computer Program MATCODE - A Material Properties Data Base, Heat Transfer Equipment Fundamentals, Design, Applications, and Operating Problems, American Society of Mechanical Engineers, 1989.

14. Singh, K.P. and Maciunas, V., Feedwater Heater Performance Prediction Code "Heater", Heat Transfer Equipment Fundamentals, Design, Applications, and Operating Problems, American Society of Mechanical Engineers, 1989.

15. Burmeister, Louis, Convective Heat Transfer, John Wiley and Sons, Inc., 1983