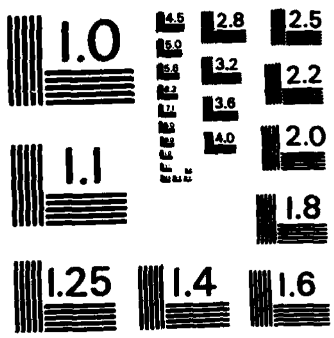
AD-A159 706 THE EFFECTIVENESS OF HEAT EXCHANGERS WITH ONE SHELL PASS AND THREE TUBE PASSES(U) NAVAL POSTGRADUATE SCHOOL MONTEREY CA M S O'HARE JUN 85 1/3

UNCLASSIFIED F/G 20/13 NL



MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

NAVAL POSTGRADUATE SCHOOL
Monterey, California

THESIS

THE EFFECTIVENESS OF HEAT EXCHANGERS
WITH ONE SHELL PASS AND
THREE TUBE PASSES

by

Mark S. O'Hare

June 1985

Thesis Advisor: Allan D. Kraus

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS BEFORE COMPLETING FORM

1. REPORT NUMBER	2. GOVT ACCESSION NO. AD-A159206
3. RECIPIENT'S CATALOG NUMBER	
4. TITLE (and Subtitle) The Effectiveness of Heat Exchangers With One Shell Pass and Three Tube Passes	5. TYPE OF REPORT & PERIOD COVERED Master's Thesis; June 1985
	6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s) Mark S. O'Hare	8. CONTRACT OR GRANT NUMBER(s)
9. PERFORMING ORGANIZATION NAME AND ADDRESS Naval Postgraduate School Monterey, California 93943-5100	10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
11. CONTROLLING OFFICE NAME AND ADDRESS Naval Postgraduate School Monterey, California 93943-5100	12. REPORT DATE June 1985
	13. NUMBER OF PAGES 204
14. MONITORING AGENCY NAME & ADDRESS(if different from Controlling Office) Naval Postgraduate School Monterey, California 93943-5100	15. SECURITY CLASS. (of this report) Unclassified
	15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release; distribution is unlimited

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Heat Exchanger
Effectiveness N_{tu} Method
Three Tube Passes

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

Heat exchangers with one shell pass and n tube passes are often referred to as 1-n exchangers. The heat transfer literature contains many references to studies of 1-n exchangers when n is even but apparently little work has been done with respect to the 1-n exchanger when n is odd. This thesis greatly expands the theoretical study of 1-n exchangers with n being odd. While a completely closed form solution was

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE
S N 0102-LF-014-6601

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

20. (Continued)

found to be unfeasible, a polynomial approximation has been developed that yields the effectiveness (ϵ) of the two possible arrangements of the 1-3 exchanger as a function of the capacity rate ratio (R) and the number of transfer units (N_{tu}). It is also shown that the effectiveness of the arrangement with two counterflow and one parallel flow tube side passes exceeds that of some of the 1-n exchangers with n even.

Approved for public release; distribution is unlimited

The Effectiveness of Heat Exchangers With One Shell Pass and Three Tube Passes.

by

Mark S. O'Hare
Lieutenant Commander, United States Navy
B.S., United States Naval Academy, 1976

Submitted in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE IN MECHANICAL ENGINEERING

from the

NAVAL POSTGRADUATE SCHOOL
June 1985

Author: Mark S. O'Hare

Approved by: Allan D. Kraus, Thesis Advisor

Paul J. Marto, Chairman,
Department of Mechanical Engineering

John N. Dyer, Dean of Science and Engineering

ABSTRACT

Heat exchangers with one shell pass and n tube passes are often referred to as 1-n exchangers. The heat transfer literature contains many references to studies of 1-n exchangers when n is even but apparently little work has been done with respect to the 1-n exchanger when n is odd. This thesis greatly expands the theoretical study of 1-n exchangers with n being odd. While a completely closed form solution was found to be unfeasible, a polynomial approximation has been developed that yields the effectiveness (ϵ) of the two possible arrangements of the 1-3 exchanger as a function of the capacity rate ratio (R) and the number of transfer units (N_{tu}). It is also shown that the effectiveness of the arrangement with two counterflow and one parallel flow tube side passes exceeds that of some of the 1-n exchangers with n even.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

NOMENCLATURE

English Letter Symbols

A = Exchanger heat-transfer surface, sq m

A_m = Coefficient of the mth value dimensionless

A_0 = Coefficient, dimensionless

A_1 = 1st order coefficient to be multiplied by N_{tu}, dimensionless

A_2 = 2nd order coefficient to be multiplied by N_{tu}^2, dimensionless

A_3 = 3rd order coefficient to be multiplied by N_{tu}^3, dimensionless

A_4 = 4th order coefficient to be multiplied by N_{tu}^4, dimensionless

A_5 = 5th order coefficient to be multiplied by N_{tu}^5, dimensionless

a = Exchanger heat-transfer surface, sq m/m

C = Capacity rate, W°/K. Also designates dimensionless arbitrary constant

C_{pc} = Specific heat at constant pressure of cold fluid, J/kg°K

C_{ph} = Specific heat at constant pressure of hot fluid, J/kg°K

D = Empirical value of effectiveness (computer generated), dimensionless

e = Error. Also used as the exponential function

F = Logarithmic mean temperature difference correction factor, dimensionless

L = Exchanger length, m

N = Number of effectiveness empirical data points used to determine a curve for R, dimensionless

n = Number of tube passes, dimensionless. Also number of equations, dimensionless

n_c = A related N_{tu} per unit length hot side, m^{-1}

n_h = A related Nt_u per unit length cold side, m^{-1}

N_{tu} = Number of transfer units, dimensionless

P = Temperature group, dimensionless

q = Total rate of heat transfer, W

q_{max} = Maximum total rate of heat transfer, W

R = Capacity rate ratio, dimensionless

S = Temperature group, dimensionless

S_r = Sum of the squares of the residuals, dimensionless

T = Hot fluid temperature, °C

T_{pi} = Particular integral, dimensionless

T_1 = Hot fluid temperature in, °C

T_2 = Hot fluid temperature out, °C

t_1 = Cold fluid temperature in, °C

t_2 = Cold fluid temperature out, °C

t_a = Cold fluid temperature 1st pass, °C

t_{ab} = Cold fluid temperature between 1st and 2nd passes

t_b = Cold fluid temperature 2nd pass, °C

t_{bc} = Cold fluid temperature between 2nd and 3rd passes

t_c = Cold fluid temperature 3rd pass, °C

U = Overall heat transfer coefficient, W/m^2 - °C

W = Mass flow, kg/sec. Also the product of ω and L, dimensionless

x = length coordinate, m. Also used to represent a constant value in a sequence

y = Sum of mth degree polynomial, defined by eq. (53), dimensionless

Z = A product of z and L, dimensionless

z = A related N_{tu} per unit length, hot side, 1/m

Greek Letter Symbols

α = Root of auxiliary differential equation, 1/m

ε = Exchanger effectiveness, dimensionless

λ = Combination of variables defined by equation (11), dimensionless

ω = A related N_{tu} per unit length, hot side, 1/m

ϕ = A combination of terms defined by eq. (38), dimensionless

Σ = Summation, dimensionless

σ^2 = Variance, dimensionless

θ_m = Mean temperature difference for exchanger, °C

∂ = Indicates partial derivative, dimensionless

Subscripts

c = Cold fluid

h = Hot fluid

i,j,k = Values in a sequence

m = Degree or order, an exponent

1 = inlet

2 = outlet

Special Symbols

[A] = An m x n matrix, symmetric

[K] = An m x n matrix, symmetric

[L] = A lower triangular matrix

[T] = An m x 1 vector

[0] = A null vector

ACKNOWLEDGEMENT

I wish to thank my wife Jayne for her support and never ending faith in me always, especially while here at NPS. Also deep appreciation of thanks to Professor Allan D. Kraus for his guidance, brilliance and friendship in completing this work. A special thanks to those in my Navy career who have encouraged me and given me the chance to prove myself. In closing, I note that I am grateful that the apple did not fall far from the tree.

I. INTRODUCTION

A. BACKGROUND

When analyzing the standard counterflow heat exchanger, it becomes apparent that, from a practical standpoint, it is often difficult to obtain a high velocity for one of the fluids when this fluid is constrained to flow through all of the tubes in a single pass. This leads to a possibility of a low overall heat transfer coefficient which cancels the advantage of the high logarithmic mean temperature difference which is obtainable in true counterflow.

The quest for flow arrangements for increased heat recovery has led to arrangements that yield increased tube-side velocities and higher overall heat transfer coefficients even at the expense of a departure from the ideal true counterflow arrangement. Thus, the design may be modified so that the tube side fluid is carried through fractions of the tubes consecutively.

Heat exchangers of this type with one shell pass and n tube passes are often referred to as 1-n exchangers. These exchangers, such as the one shell pass, two-tube pass (1-2) parallel-counterflow exchanger (see Figure 1.1), are configured such that all of the tube side fluid flows through the two halves of the the tubes successively. A single channel is employed with a partition to permit the

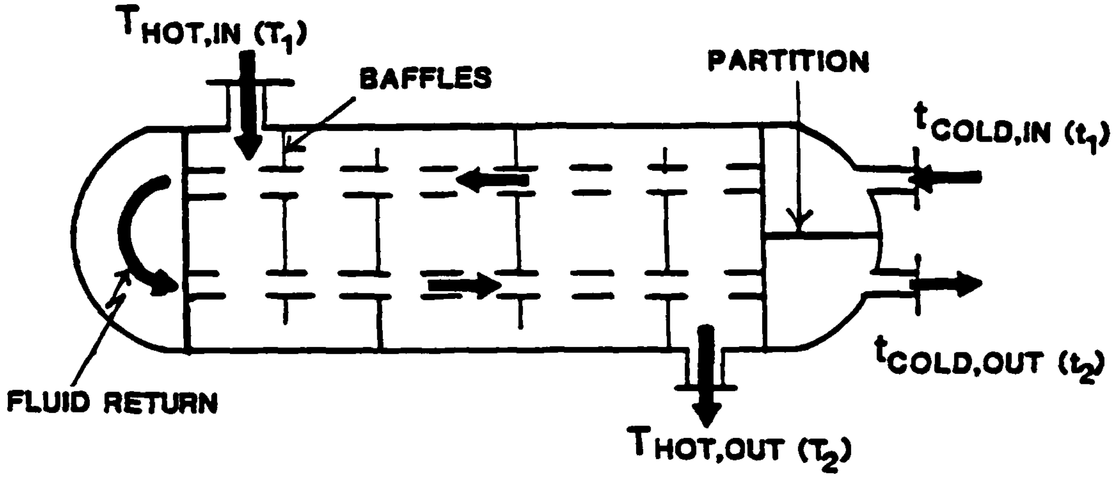


Figure 1.1 1-2 Parallel-Counterflow Exchanger

entry and exit of the tube side fluid from the same channel. Note the baffles used to induce turbulence causing the liquid to flow through the shell at right angles to the axes of the tubes thus helping to create a higher shell side velocity with higher shell side heat-transfer coefficients.

To date, much work has been done on finding the true logarithmic mean temperature difference for heat exchangers with an even number of tube passes. Little work, however, has been done with regard to heat exchangers having an odd number of passes. This is primarily due to the method employed in deriving an analytical solution to measure the overall effectiveness of a heat exchanger.

Exhangers with an even number of tube passes often present a configuration problem especially in a marine application where the inlet and outlet of the cooling fluid must be one the same side of the exchanger header (see Figure 1.1). This particular problem could be alleviated by going to heat exchangers with an odd number of tube passes. Currently, precise mathematical expressions for the effectiveness of the 1-3 exchangers do not exist. Hence, a theoretical examination is reported on here which considers the effectiveness of the 1-3 parallel-counter flow exchanger which is shown in Figure 1.2.

Before doing this it is important to mention the work that lead to the effectiveness method and the development of work on heat exchangers with an odd number of tube passes.

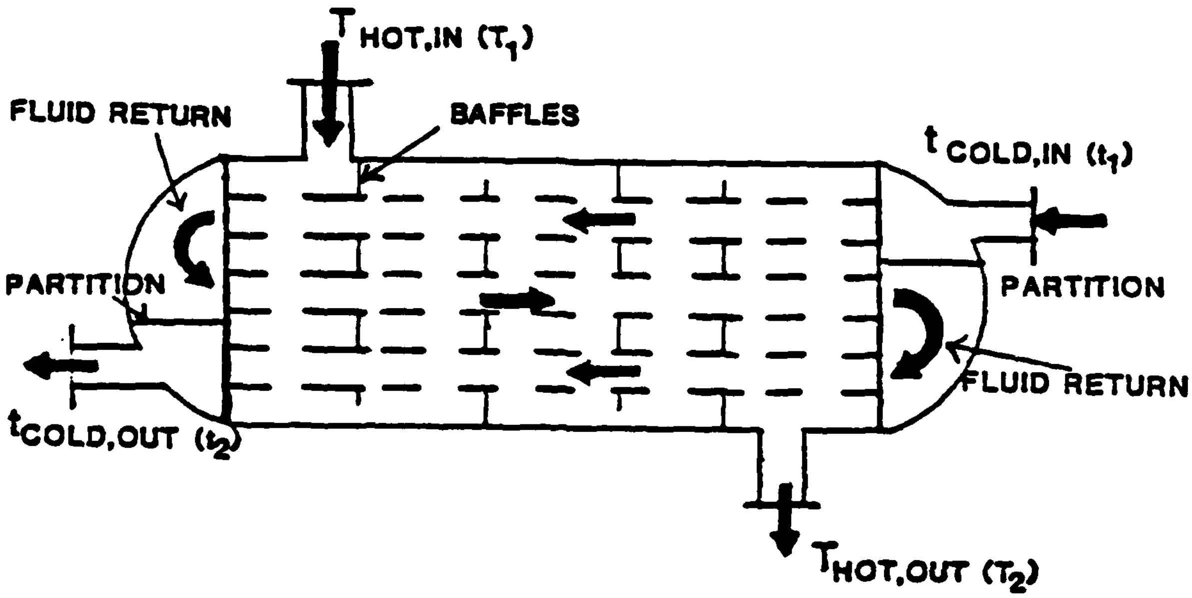


Figure 1.2 1-3 One Shell Pass Three Tube Pass Exchanger

This will be considered in Section II. Kern [Ref. 1: pp. 224-226], makes the interesting point that the optimum exchanger requires an exchanger capable of providing the optimum fluid-flow velocities on the shell as well as the tube sides. This might frequently entail the use of an odd number of tube passes or an odd tube length.

B. WHY EFFECTIVENESS AS A FUNCTION OF N_{tu}

It is also noted that Kays and London [Ref. 2: pp. 24-29] indicated that the effectiveness as a function of N_{tu} (ε - N_{tu}) method is the favored approach for evaluating a heat exchanger's performance because:

1. The effectiveness value stands alone as a dependent variable and should not appear directly in the abscissa and indirectly in the ordinate of a graphical display.

2. The log-mean difference equation misleadingly simplifies the notion of what is involved in heat exchanger design theory, since the implication is that only a rate equation is required.

3. The ε - N_{tu} approach simplifies the algebra involved in predicting the performance of complex flow arrangements.

4. The more meaningful arguments are related to ease of use in design work. Two prime examples of these are:

    a) Given the overal heat transfer coefficient, U, the two fluid capacity rates, C_c and C_h, and the terminal temperatures, determine the required surface area, A.

    b) Given A, U, C_c, C_h and the inlet temperatures of both streams, determine the outlet temperatures.

II. THE DEVELOPMENT OF THE EFFECTIVENESS METHOD

A. LITERATURE SURVEY

It was Nagle [Ref. 3: pp. 604-609], in 1931, who credited Davis [Ref. 4] with a simplified method for computing actual temperature differences between two heat-interchanging streams which depart from true counter or concurrent (parallel) flow. This is now the familiar "F factor" method which expresses the actual mean temperature difference θ_m in $q = UA\theta_m$ as a fraction F of the counterflow logarithmic mean temperature difference, LTMD, θ_{mc} via $\theta_m = F\theta_{mc}$.

The example of initial interest was the 1-2 exchanger with a single shell pass and two continuous tube passes in counter and concurrent flow with it. The method involved derivation of the actual temperature difference for the flow pattern and formed the ratio $F = \theta_m/\theta_{mc}$. This familiar LMTD correction factor was plotted conveniently as functions of the effectiveness, ε, and the capacity rate ratio R with R as a parameter. These mean temperature difference correction charts are available for many flow arrangements [Ref. 1: pp. 829-833 and Ref. 5]. The effectiveness, ε (often called P or S), is always the cold fluid effectiveness and R is always the capacity rate ratio of cold fluid to hot fluid.

Nagle detailed assumptions and derivations for the 1-2, 1-4 and 1-6 exchangers. The F factors were obtained by Nagle through graphical integration and were accompanied by the comment that F factors for the 1-2 exchanger could be applied with negligible error to 1-4 and 1-6 exchangers. Underwood [Ref. 6: pp. 145-148] rederived the equations of Nagle for 1-2 and 1-4 exchangers to eliminate the need for obtaining F factors by graphical integration.

Bowman [Ref. 7: pp. 541-544] pointed out that for a very large or infinite number of tube passes, the F factor approached, as a limit, its value in crossflow with both fluids completely mixed. It was further stated that even at the limit, the F factors were only 1 to 2 percent lower than those for the 1-2 exchanger. A previous paper by Kraus and Kern [Ref. 8] did not confirm the generalization that 1-n exchangers differed only negligibly from the 1-2 exchanger although this lack of confirmation was obtained on an $\varepsilon = f(R, N_{tu})$ basis. Moreover, the Kraus-Kern work does not confirm the generalizations on an $F = f(R, N_{tu}, \varepsilon)$ basis.

From the standpoint of usefulness and good accuracy, it is essential that F factors, if they are to be used in preference to $\varepsilon = f(R, N_{tu}$, flow arrangement), be obtained with precision. Plots of $F = f(R, \varepsilon, = P$ or S) [Ref. 1: pp. 829-833 and Ref. 5] show that the curves for particular values of R approach infinite slope as F decreases. While this can be partially alleviated by restricting $R < 1.0$ (a constraint used in the $\varepsilon = f(R, N_{tu}$, flow arrangement) approach), it is seen that small errors in the interpolation for R or $\varepsilon = P$ or S can result in large fluctuations in the value of F.

In a comprehensive paper, Bowman, Mueller and Nagle [Ref. 9: pp. 283-294] presented graphs of F factors for shells with one through six shell passes and numbers of continuous tube passes respectively double the number of shell passes. In view of the earlier references to Nagle and Bowman, it should be noted that F factors were computed for the 1-2 exchanger in [Ref. 9: pp. 283-294] using the equations of Underwood [Ref. 6: pp. 145-158].

Ten Broeck [Ref. 10: pp. 1041-1042] prepared a graph of the dimensionless groups now known as ε, R and N_{tu} for the 1-2 exchanger. Such a graph had the added versatility of simplifying the calculation of performance in a given exchanger when operating at conditions different for those for which it was designed. Kays and London [Ref. 2: pp. 63-74] prepared similar graphs and tables of $\varepsilon = f(R, N_{tu}$, flow arrangement) for the 1-2 exchanger and for several cases of crossflow and periodic flow.

The foregoing describes the early history of the search for the so-called Logarithmic Mean Temperature Difference Correction Factor, F, with regard to heat exchangers having an even number of tube passes. It is a fact, however, that certain space economies could be realized from exchangers having an odd number of tube passes so that the tube side fluid could enter and leave the exchanger at opposite ends of the exchanger (see Figure 1.2).

B. FISCHER'S WORK

With the foregoing in mind, an extensive search has been conducted to obtain $\varepsilon - N_{tu}$ data for the so called "1-3" and "1-5" exchangers. This search has uncovered a single work, that of Fischer [Ref. 11: pp. 377-383], which summarizes the historical development covered here and contains only a small section on the 1-3 exchanger. This work by Fischer develops an equation for true mean temperature difference of the 1-3 exchanger and casts the results in terms of F rather than ε. Moreover, the work treats only the case where the three tube passes are arranged with two in counterflow and one in parallel flow (1-3:2C) making no mention of the one counterflow and two parallel flow (1-3:2P) case (see Figures 2.1 and 2.2). In addition, the equation developed to yield F must be solved using a trial and error solution.

The present work is aimed at continuing the Fischer investigation for several reasons:

1. A solution is needed for effectiveness, ε, as a function of capacity rate (R) and number of transfer units (N_{tu}).
2. This solution should be in a closed form if at all possible so that it will be computationally efficient and useful in both the design and analysis frameworks.

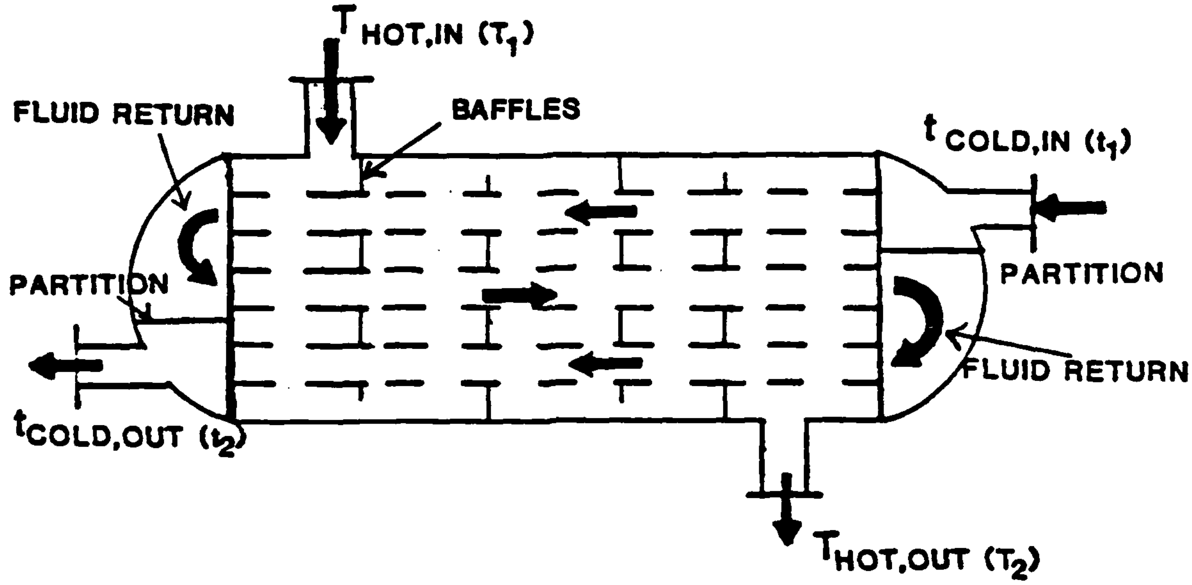


Figure 2.1 1-3:2C Three Tube Passes - Two in Counterflow

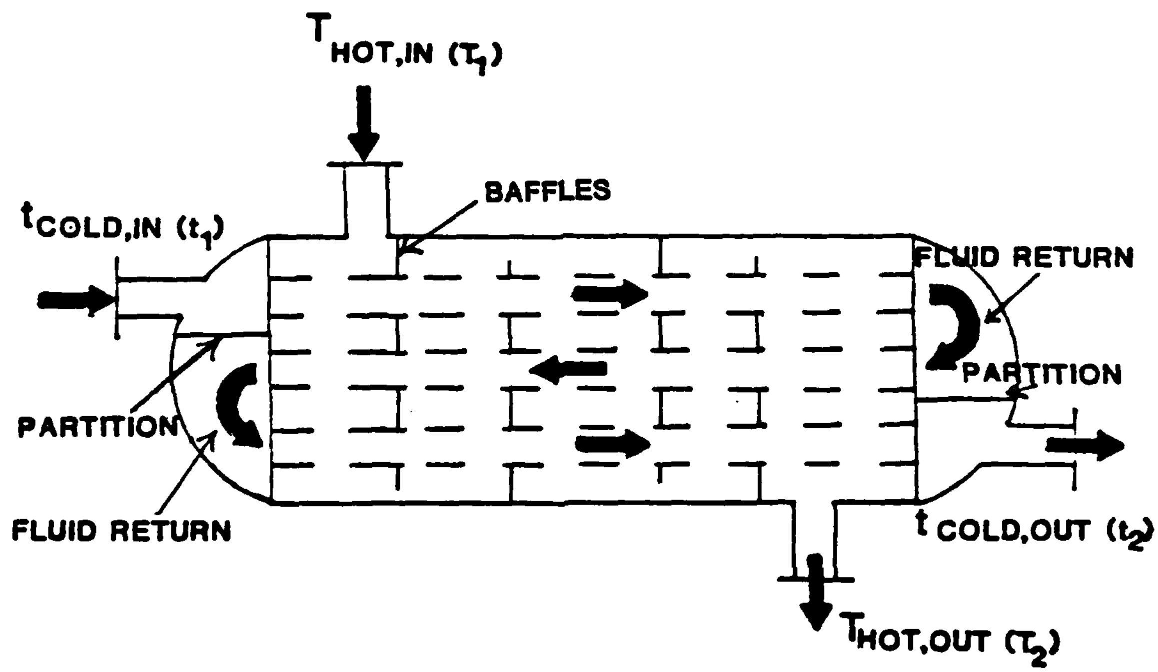


Figure 2.2 1-3:2P Three Tube Passes - Two in Parallel Flow

3. Because valid design data can evolve from a polynomial approximation. The search should not be abandoned just because a closed form solution does not result from an analytical approach.

4. Data is needed for the (1-3:2P) two parallel-one counterflow configuration.

5. A 1-3 exchanger in a marine (shipboard) application may result in a considerable space saving over its 1-n counterpart with n even. This would be evident on the outside of the exchanger where it would be immediately noted that the 1-3 exchanger has tube side inlet and outlet at opposite ends of the exchanger.

The next section confirms Fischer's result and shows that a closed form solution cannot be obtained for the effectiveness of the 1-3 exchanger. Sections IV and V demonstate how, through numerical analysis assisted by a computer, a polynomial solution can be derived that will yield the effectiveness to engineering accuracy.

III. AN ATTEMPT AT A CLOSED FORM SOLUTION

A. EFFECTIVENESS AS A FUNCTION OF CAPACITY RATES AND EXCHANGER SIZE

This section deals with an investigation into the effectiveness, ε, of a one shell pass and three tube pass heat exchanger, whereby ε compares the actual heat transfer rate to the thermodynamically limited, maximum possible heat transfer rate as would be realized only in a counter flow heat exchanger of infinite transfer area. This exchanger heat transfer effectiveness is given by

$$\varepsilon = \frac{q}{q_{max}} = \frac{C_h(T_{hot,in} - T_{hot,out})}{C_{min}(T_{hot,in} - t_{cold,in})} = \frac{C_c(t_{cold,out} - t_{cold,in})}{C_{min}(T_{hot,in} - t_{cold,in})}$$

where C_{min} is the smaller of the C_h and C_c magnitudes. Thus, ε possesses the significance of effectiveness of the heat exchanger from a thermodynamic point of view, with the magnitude of the effectiveness completely defining the heat transfer performance. In general we express $\varepsilon = f(N_{tu}$, R, and flow arrangement) and when the flow arrangement is understood, it is said that $\varepsilon = f(N_{tu}, R)$. [Ref. 2: pp. 14-26].

The number of heat transfer units N_{tu} is a nondimensional expression of the "heat transfer size" of the exchanger. When N_{tu} is small the exchanger effectiveness is low, and when N_{tu} is large, ε approaches the limit imposed by the flow arrangement and thermodynamic conditions asymptotically. From inspection of the definition of N_{tu}

$$N_{tu} = \frac{AU}{C_{min}} = \frac{1}{C_{min}} \int_{0}^{A} UdA$$

it is clear that the overall conductance and transfer area affect the costs of attaining a high value for N_{tu}, ergo high ε. The capacity rate ratio, R, as defined by

$$R = \frac{C_{min}}{C_{max}}$$

is simply the ratio of mass flow rate times specific heat capacity for the two streams. These can be considered as flow stream thermal-capacity rates, i.e., energy storage rate in the stream per unit of temperature change. [Ref. 2: pp. 14-26]

The attempt taken in this thesis to develop a closed form solution has used the basic fundamentals of heat transfer as well as those indicated above. A closed form solution for ε was sought for both 1-3 exchangers with one having two out of three tube passes in parallel flow and the other having two out of three tube passes in counterflow. The analytical approach taken, and demonstrated in this section, is for two out of three tube passes in counterflow.

B. ANALYTICAL DEVELOPMENT

The derivation for the effectiveness, ε, of the 1-3 exchanger as a function of the capacity rate ratio, R, and number of transfer units, N_{tu}, depends on several assumptions.

(1) The overall coefficient of heat transfer, U, does not vary within the exchanger.

(2) The specific heat of both hot side and cold side fluids does not vary.

(3) Each fluid is thoroughly mixed, that is, the temperature of both hot and cold side fluids is uniform over any cross section.

(4) Steady flow conditions are maintained.

(5) Heat losses to or from the environment are negligible.

(6) No change of phase takes place; all heat transferred is sensible heat.

(7) There is equal heat transfer surface in each pass.

The configuration is shown in Figure 3.1 where the three tube passes are designated with subscripts a, b and c. The temperature of the hot (shell side) fluid is indicated by upper case letters. For the cold (tube side) fluid, lower case letters are used. The subscript 1 always refers to the fluid inlet and the subscript 2 always refers to the fluid outlet.

With W_h and C_{ph} designating mass flow (kg/sec) and specific heat (Joules/kg$^{-\circ}$K) of hot fluid entering at T_1 and leaving at T_2 we define a capacity rate for the hot side

$$C_h = W_h C_{ph}$$

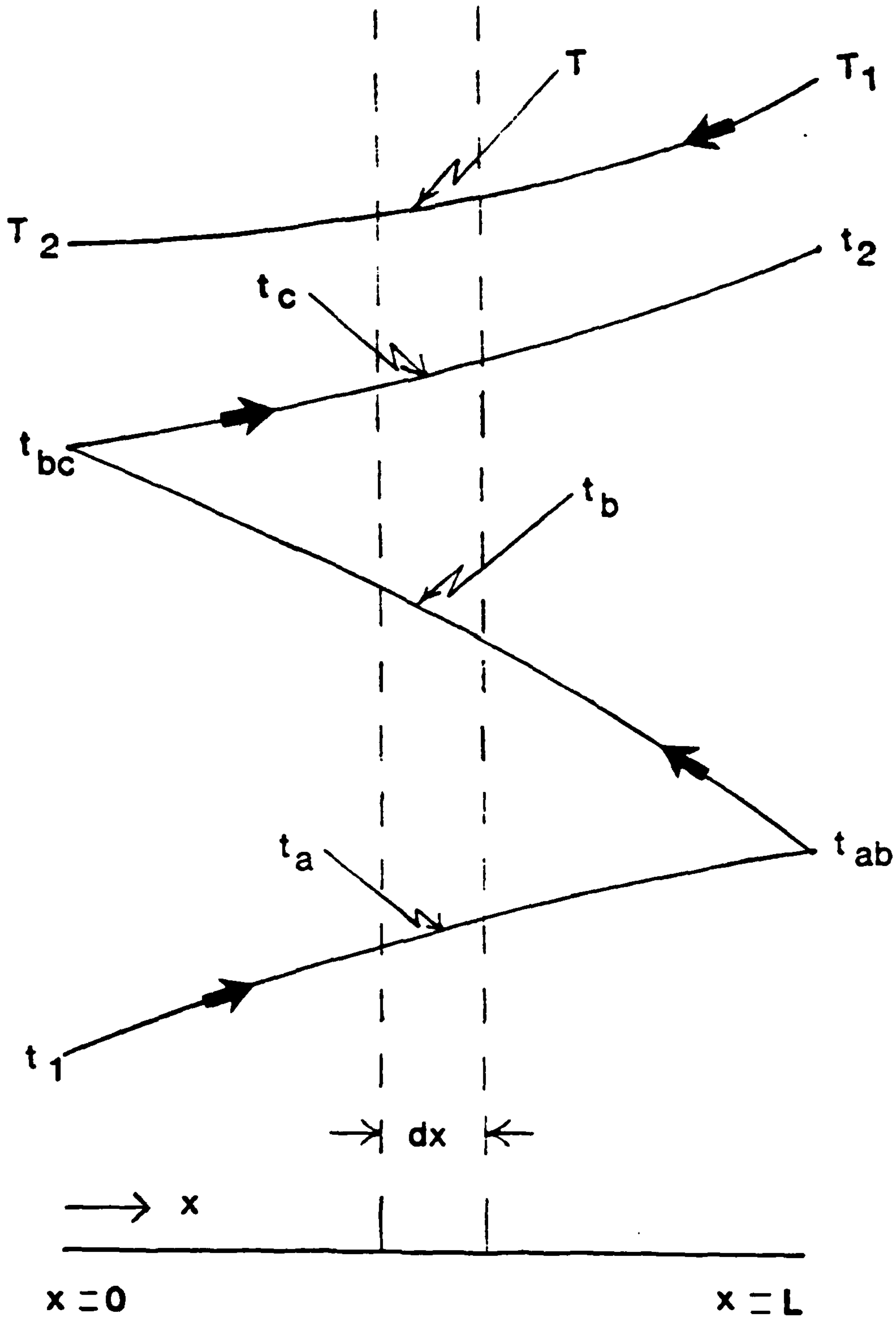


Figure 3.1 Two Counter One Parallel Configuration for Development of Effectiveness Relationship

In similar fashion for the cold side (with W_c and C_{pc}) entering at t_1 and leaving at t_2, we have

$$C_c = W_c C_{pc}$$

We then obtain an energy balance for the entire exchanger

$$C_h(T_1 - T_2) = C_c(t_2 - t_1) \qquad (1)$$

Over the right hand side of the exchanger (Figure 3.1)

$$C_h(T_1 - T) = C_c(t_2 - t_c + t_b - t_a) \qquad (2)$$

and a differentiation gives

$$C_h dT = C_c(dt_c - dt_b + dt_a) \qquad (3)$$

Across dx, with a (m^2/m), the surface per running meter of length of pass so that $A = 3aL$ is the total surface in the exchanger, we may write the heat transferred to the element dx in each cold pass.

$$C_c dt_a = Uadx(T - t_a) \qquad (4a)$$

$$C_c dt_b = -Uadx(T - t_b) \qquad (4b)$$

$$C_c dt_c = Uadx(T - t_c) \qquad (4c)$$

Here it should be observed that due cognizance has been taken of the direction of the flow in each cold fluid pass with respect to the positive sense of the length coordinate, x, and U is the overall heat transfer coefficient (W/m^2-°C).

With eqs. (4) in eq. (3)

$$C_h dT = Ua(3T - t_a - t_b - t_c)dx$$

or

$$\frac{dT}{dx} = n_h(3T - t_a - t_b - t_c) \tag{5}$$

where

$$n_h = \frac{Ua}{C_h}$$

is a sort of N_{tu} per unit length for the hot side.

Now differentiate eq. (5)

$$\frac{d^2T}{dx^2} = n_h\left(3\frac{dT}{dx} - \frac{dt_a}{dx} - \frac{dt_b}{dx} - \frac{dt_c}{dx}\right)$$

and with eqs. (4) substituted

$$\frac{d^2T}{dx^2} = 3n_h \frac{dT}{dx} - n_c n_h(T - t_a + t_b - t_c) \tag{6}$$

where

$$n_c = \frac{Ua}{C_c}$$

where again the resemblence of n_c to N_{tu} can be noted.

From eq. (2) we obtain

$$\frac{C_h}{C_c}(T_1 - T) - t_2 = t_b - t_a - t_c \tag{7}$$

and with eq. (7) put into eq. (6) we obtain

$$\frac{d^2T}{dx^2} - 3n_h \frac{dT}{dx} = - n_c n_h \left[T + \frac{C_h}{C_c}(T_1 - T) - t_2\right]$$

or

$$\frac{d^2T}{dx^2} - 3n_h \frac{dT}{dx} = -n_c n_h \frac{C_h}{C_c}\left[(R_c - 1)T + T_1 - R_c t_2\right] \tag{8}$$

where

$$R_c = C_c/C_h$$

is the capacity rate ratio for the cold side.

Notice that

$$n_c n_h \frac{C_h}{C_c} = \frac{Ua}{C_c} \cdot \frac{Ua}{C_h} \cdot \frac{C_h}{C_c} = \left(\frac{Ua}{C_c}\right)^2 = m$$

and

$$R_h = \frac{1}{R_c} = \frac{C_h}{C_c}$$

a capacity rate ratio for the hot side. Then, algebraic adjustment provides

$$\frac{d^2T}{dx^2} - 3n_h \frac{dT}{dx} + m\left(\frac{1 - R_h}{R_h}\right)T = m\left(\frac{t_2}{R_h} - T_1\right) \tag{9}$$

which is a linear, non-homogeneous, second order differential equation with constant coefficients having a complementary function

$$T_c = C_1 e^{\alpha_1 x} + C_2 e^{\alpha_2 x} \tag{10}$$

where C_1 and C_2 are arbitrary constants and where

$$\alpha_1, \alpha_2 = \frac{3n_h}{2} \pm \frac{1}{2}\left[9n_h^{\;2} - 4m\left(\frac{1 - R_h}{R_h}\right)\right]^{1/2}$$

$$= \frac{3n_h}{2} \pm \frac{n_h}{2}\left[9 - \frac{4m}{n_h^{\;2}}\left(\frac{1 - R_h}{R_h}\right)\right]^{1/2}$$

But

$$\frac{m}{n_h^{\;2}} = \frac{(Ua)^2}{(C_c)^2} \cdot \frac{(C_h)^2}{(Ua)^2} = \left(\frac{C_h}{C_c}\right)^2 = R_h^{\;2} = \frac{1}{R_c^{\;2}}$$

so that

$$\alpha_1, \alpha_2 = \frac{3n_h}{2} \pm \frac{n_h}{2}\left[9 - 4R_h^{\;2}\left(\frac{1 - R_h}{R_h}\right)\right]^{1/2}$$

or

$$\alpha_1, \alpha_2 = \frac{n_h}{2}(3 \pm \lambda) \tag{11}$$

where

$$\lambda = [9 - 4R_h(1 - R_h)]^{1/2} \qquad (12)$$

Designate the particular integral as T_{pi} and by the method of undetermined coefficients let $T_{pi} = P$ so that in eq. (9)

$$m(\frac{1 - R_h}{R_h})\ P = m(\frac{t_2}{R_h} - T_1)$$

This makes

$$T_{pi} = P = [\ \frac{t_2}{R_h} - T_1]\ [\frac{R_h}{1 - R_h}]$$

so that

$$T_{pi} = \frac{t_2 - R_h T_1}{1 - R_h} \qquad (13)$$

The general solution to eq. (9) is the sum of eqs. (10) and (13)

$$T(x) = C_1 e^{\alpha_1 x} + C_2 e^{\alpha_2 x} + \frac{t_2 - R_h T_1}{1 - R_h} \qquad (14)$$

where the arbitary constants, C_1 and C_2 are evaluated from conditions at $x = 0$ and $x = L$. At $x = 0$, $T(x = 0) = T_2$ and at $x = L$, $T(x = L) = T_1$. When these are inserted, in turn, into eq. (14), one obtains a pair of linear algebraic equations in the unknowns C_1 and C_2

$$T_2 = C_1 + C_2 + T_{pi}$$

$$T_1 = C_1 e^{\alpha_1 L} + C_2 e^{\alpha_2 L} + T_{pi}$$

where T_{pi} is given by eq. (13).

It is only a matter of algebra to show that

$$C_1 = \frac{(T_1 - T_{pi}) - (T_2 - T_{pi})e^{\alpha_2 L}}{e^{\alpha_1 L} - e^{\alpha_2 L}} \tag{15a}$$

and

$$C_2 = \frac{(T_2 - T_{pi})e^{\alpha_1 L} - (T_1 - T_{pi})}{e^{\alpha_1 L} - e^{\alpha_2 L}} \tag{15b}$$

It is easy to see from eq. (1) that

$$R_h = \frac{C_h}{C_c} = \frac{(t_2 - t_1)}{(T_1 - T_2)}$$

so that

$$t_2 = t_1 + R_h(T_1 - T_2)$$

Use of this in eq. (13) shows that

$$T_{pi} = \frac{t_1 + R_h(T_1 - T_2) - R_h T_1}{1 - R_h}$$

or

$$T_{pi} = \frac{t_1 - R_h T_2}{1 - R_h} \tag{16}$$

indicating two alternative forms for T_{pi} given by eqs. (13) and (16).

Insertion of eqs. (13) and (16) in eqs. (15) for C_1 and C_2 will yield after some algebra

$$C_1 = \frac{\left(\frac{T_1 - t_2}{1 - R_h}\right) - \left(\frac{T_2 - t_1}{1 - R_h}\right) e^{\alpha_2 L}}{e^{\alpha_1 L} - e^{\alpha_2 L}} \tag{17a}$$

and

$$C_2 = \frac{\left(\frac{T_2 - t_1}{1 - R_h}\right) e^{\alpha_1 L} - \left(\frac{T_1 - t_2}{1 - R_h}\right)}{e^{\alpha_1 L} - e^{\alpha_2 L}} \tag{17b}$$

Equation (14) is an expression for the hot side temperature at any location in the exchanger in terms of the extreme temperatures, t_1, t_2, T_1 and T_2.

Next take eq. (5) and set it equal to the derivative of eq. (14) noting that C_1, C_2 and T_{pi} are all known constants.

$$\frac{dT}{dx} = n_h(3T - t_a - t_b - t_c) = \alpha_1 C_1 e^{\alpha_1 x} + \alpha_2 C_2 e^{\alpha_2 x} \tag{18}$$

At $x = 0$, where $T = T_2$, $t_a = t_1$ and $t_b = t_c = t_{bc}$

$$\frac{dT}{dx} = n_h(3T_2 - t_1 - 2t_{bc}) = \alpha_1 C_1 + \alpha_2 C_2 \qquad (19)$$

and if we subtract eq. (4a) from eq. (4c) we obtain

$$\frac{dt_a - dt_c}{t_a - t_c} = - \frac{Ua}{C_c}\, dx = -n_c dx$$

which can be integrated using C_3 as the constant of integration.

$$t_a - t_c = C_3 e^{-n_c x}$$

and at $x = 0$ where $t_a = t_1$ and $t_c = t_{bc}$

$$t_1 - t_{bc} = C_3$$

or

$$t_{bc} = t_1 - C_3$$

In addition at $x = L$, $t_a = t_{ab}$ and $t_c = t_2$ so that

$$t_{ab} - t_2 = C_3 e^{n_c L}$$

or

$$C_3 = \frac{t_{ab} - t_2}{e^{-n_h L}} = (t_{ab} - t_2) e^{n_c L}$$

This gives a relationship between t_{ab} and t_{bc}

$$t_{bc} = t_1 - (t_{ab} - t_2) e^{N_c} \qquad (20)$$

where $N_c = n_c L$ can be considered as the total number of transfer units for the cold side.

Return now to eq. (18) and look at the conditions at $x = L$ where $t_a = t_b = t_{ab}$, $t_c = t_2$ and $T = T_1$. These conditions in eq. (18) give

$$n_h(3T_1 - 2t_{ab} - t_2) = \alpha_1 C_1 e^{\alpha_1 L} + \alpha_2 C_2 e^{\alpha_2 L}$$

where again we remember that C_1 and C_2 are known constants. Solving for t_{ab}

$$2t_{ab} = -\frac{1}{n_h}(\alpha_1 C_1 e^{\alpha_1 L} + \alpha_2 C_2 e^{\alpha_2 L}) + 3T_1 - t_2$$

and with this in eq. (20)

$$2t_{bc} = e^{N_c}[\frac{1}{n_h}(\alpha_1 C_1 e^{\alpha_1 L} + \alpha_2 C_2 e^{\alpha_2 L}) + 3t_2 - 3T_1] + 2t_1$$

Then with eq. (21) in eq. (19)

$$\alpha_1 C_1 + \alpha_2 C_2 = 3n_h [(T_2 - t_1) + e^{N_c}(T_1 - t_2)]$$
$$-e^{N_c}(\alpha_1 C_1 e^{\alpha_1 L} + \alpha_2 C_2 e^{\alpha_2 L}) \qquad (22)$$

Equation (22) confirms Fischer's result [Ref. 12: pp. 377-383] and at this point in his development he branches off to seek an expression for the Logarithmic Mean

Temperature Difference Correction Factor, F. The attention here is focused on $\varepsilon = f(R, N_{tu})$ and the balance of this section continues in this vein.

Look at the $\alpha_1 C_1 + \alpha_2 C_2$ term in eq. (22). Use of eq. (11) allows the representation of $\alpha_1 C_1 + \alpha_2 C_2$

$$\frac{n_h}{2}(3 + \lambda)C_1 + \frac{n_h}{2}(3 - \lambda)C_2 \tag{23a}$$

or

$$\frac{3n_h}{2}(C_1 + C_2) + \frac{\lambda n_h}{2}(C_1 - C_2) \tag{23b}$$

Then from eqs. (17)

$$C_1 + C_2 = \frac{(T_2 - t_1)[e^{\alpha_1 L} - e^{\alpha_2 L}]}{(1 - R_h)[e^{\alpha_1 L} - e^{\alpha_2 L}]}$$

or

$$C_1 + C_2 = \frac{T_2 - t_1}{1 - R_h} \tag{24}$$

Moreover

$$C_1 - C_2 = \frac{2(T_1 - t_2) - (T_2 - t_1)[e^{\alpha_1 L} + e^{\alpha_2 L}]}{(1 - R_h)[e^{\alpha_1 L} - e^{\alpha_2 L}]} \tag{25}$$

Now let

$$\omega = \frac{3n_h}{2} \tag{26a}$$

and

$$z = \frac{\lambda n_h}{2} \tag{26b}$$

so that

$$e^{\alpha_1 L} - e^{\alpha_2 L} = e^{(\omega+z)L} - e^{(\omega-z)L}$$

$$= e^{\omega L}e^{zL} - e^{\omega L}e^{-zL}$$

$$= e^{\omega L}(e^{zL} - e^{-zL})$$

or

$$e^{\alpha_1 L} - e^{\alpha_2 L} = 2e^{\omega L} \sinh zL \tag{27}$$

Moreover, it is easy to see that

$$e^{\alpha_1 L} + e^{\alpha_2 L} = 2e^{\omega L} \cosh zL \tag{28}$$

If eqs. (24) through (28) are collected and put into the expression of eq. (23b), the result is

$$\alpha_1 C_1 + \alpha_2 C_2 = \frac{3n_h}{2} (C_1 + C_2) + \frac{\lambda n_h}{2} (C_1 - C_2) \tag{23b}$$

$$= \omega \left[\frac{T_2 - t_1}{1 - R_h}\right] + z \left[\frac{2(T_1 - t_2) - (T_2 - t_1)2e^{\omega L} \operatorname{csch} zL}{(1 - R_h)2e^{\omega L} \sinh zL} \right]$$

or

$$\alpha_1 C_1 + \alpha_2 C_2 = \omega\left(\frac{T_2 - t_1}{1 - R_h}\right) +$$

$$z\left[\left(\frac{T_1 - t_2}{1 - R_h}\right)e^{-\omega L}\text{csch } zL - \left(\frac{T_2 - t_1}{1 - R_h}\right)\coth zL\right] \qquad (29a)$$

and this could also be written as

$$\alpha_1 C_1 + \alpha_2 C_2 = \left(\frac{T_2 - t_1}{1 - R_h}\right)[\omega - z \coth zL]$$

$$+ z\left(\frac{T_1 - t_2}{1 - R_h}\right)e^{-\omega L}\text{csch } zL \qquad (29b)$$

The next step is to reduce the right hand side of eq. (22). Use of eq. (11) permits the representation

$$\alpha_1 C_1 e^{\alpha_1 L} + \alpha_2 C_2 e^{\alpha_2 L} = \left(\frac{3n_h}{2} + \frac{\lambda n_h}{2}\right)C_1 e^{\alpha_1 L} + \left(\frac{3n_h}{2} - \frac{\lambda n_h}{2}\right)C_2 e^{\alpha_1 L}$$

or with eqs. (26) inserted

$$\alpha_1 C_1 e^{\alpha_1 L} + \alpha_2 C_2 e^{\alpha_2 L} = \omega(C_1 e^{\alpha_1 L} + C_2 e^{\alpha_2 L}) + z(C_1 e^{\alpha_1 L} - C_2 e^{\alpha_2 L}) \qquad (30)$$

But by eqs. (11) and (17)

$$C_1 e^{\alpha_1 L} + C_2 e^{\alpha_2 L} = \frac{(T_1 - t_2)(e^{\alpha_1 L} - e^{\alpha_2 L}) + (T_2 - t_1)[e^{(\alpha_1 + \alpha_2)L} - e^{(\alpha_1 + \alpha_2)L}]}{(1 - R_h)(e^{\alpha_1 L} - e^{\alpha_2 L})}$$

or

$$C_1 e^{\alpha_1 L} + C_2 e^{\alpha_2 L} = \frac{T_1 - t_2}{1 - R_h} \tag{31}$$

$C_1 e^{\alpha_1 L} - C_2 e^{\alpha_2 L}$ may also be simplified. Again using eqs. (11), (17a) and (17b)

$$C_1 e^{\alpha_1 L} - C_2 e^{\alpha_2 L} = \frac{(T_1 - t_2)(e^{\alpha_1 L} + e^{\alpha_2 L}) - (T_2 - t_1)[e^{(\alpha_1 + \alpha_2)L} + e^{(\alpha_1 + \alpha_2)L}]}{(1 - R_h)(e^{\alpha_1 L} - e^{\alpha_2 L})}$$

The exponential term at the far right in the numerator is really quite simple. From eq. (11)

$$\alpha_1 + \alpha_2 = \left(\frac{3n_h}{2} + \frac{\lambda n_h}{2} \right) + \left(\frac{3n_h}{2} - \frac{\lambda n_h}{2} \right) = 3n_h$$

and by eq. (26a) $\alpha_1 + \alpha_2 = 3n_h = 2\omega$. Thus with the combination of exponentials in the numerator and the denominator given by eqs. (27) and (28) we find that

$$C_1 e^{\alpha_1 L} - C_2 e^{\alpha_2 L} = \frac{2(T_1 - t_2)e^{\omega L} \cosh zL - (T_2 - t_1)2e^{2\omega L}}{2(1 - R_h)e^{\omega L} \sinh zL}$$

or

$$C_1 e^{\alpha_1 L} - C_2 e^{\alpha_2 L} = \left(\frac{T_1 - t_2}{1 - R_h}\right) \coth zL - \left(\frac{T_2 - t_1}{1 - R_h}\right) e^{\omega L} \operatorname{csch} zL \tag{32}$$

Now with eqs. (31) and (32) put into eq. (30) we obtain

$$\alpha_1 C_1 e^{\alpha_1 L} + \alpha_2 C_2 e^{\alpha_2 L} = \omega\left(\frac{T_1 - t_2}{1 - R_h}\right)$$

$$+ z\left[\left(\frac{T_1 - t_2}{1 - R_h}\right) \coth zL - \left(\frac{T_2 - t_1}{1 - R_h}\right) e^{\omega L} \operatorname{csch} zL\right]$$

or

$$\alpha_1 C_1 e^{\alpha_1 L} + \alpha_2 C_2 e^{\alpha_2 L} = \left(\frac{T_1 - t_2}{1 - R_h}\right) [\omega + z \coth zL]$$

$$- z\left(\frac{T_2 - t_1}{1 - R_h}\right) e^{\omega L} \operatorname{csch} zL \qquad (33)$$

With eqs. (29b) and (33) inserted into eq. (22) we obtain

$$\left(\frac{T_2 - t_1}{1 - R_h}\right) [\omega - z \coth zL] + z\left(\frac{T_1 - t_2}{1 - R_h}\right) e^{-\omega L} \operatorname{csch} zL =$$

$$3n_h[(T_2 - t_1) + e^{N_c}(T_1 - t_2) -$$

$$e^{-N_c}\left[\left(\frac{T_1 - t_2}{1 - R_h}\right)(\omega + z \coth zL) - z\left(\frac{T_2 - t_1}{1 - R_h}\right) e^{\omega L} \operatorname{csch} zL\right] \qquad (34)$$

We wish to develop an expression for the exchanger effectiveness so we designate the hot side effectiveness as

$$\epsilon_h = \frac{T_1 - T_2}{T_1 - t_1} \tag{35}$$

and we begin by simplifying eq. (34) by dividing throughout by $(T_2 - t_1)/(1 - R_h)$ to obtain

$$(\omega - z \coth Z) + ze^{-W} \operatorname{csch} z\left(\frac{T_1 - t_2}{T_2 - t_1}\right) = R + e^{N_c} R\left(\frac{T_1 - t_2}{T_2 - t_1}\right)$$

$$- e^{-N_c}\left[(\omega + z \coth Z)\left(\frac{T_1 - t_2}{T_2 - t_1}\right) - ze^{W} \operatorname{csch} Z\right] \tag{36}$$

where

$$W = \omega L \tag{37a}$$

$$Z = zL \tag{37b}$$

and

$$R = 3n_h(1 - R_h) \tag{37c}$$

We can then let

$$\phi = \frac{T_1 - t_2}{T_2 - t_1}$$

and get eq. (36) to look like

$$(\omega - z \coth Z) + (ze^{-W} \operatorname{csch} Z)\ \phi =$$

$$R + (e^{N_c}R)\phi - [e^{-N_c}(\omega + \coth Z)]\phi - (ze^{(W-N_c)} \operatorname{csch} Z)$$

It is now a matter of algebra to solve for ϕ

$$\phi = \frac{R + ze^{(W-N_c)} \operatorname{csch} Z - (\omega - z \coth Z)}{ze^{-W} \operatorname{csch} Z - Re^{N_c} + e^{N_c}(\omega + z \coth Z)} \qquad (38)$$

The next step is to represent ϕ as a function of ϵ_h. This is done with some algebraic gymnastics as follows:

$$\phi = \frac{T_1 - t_2}{T_2 - t_1} = \frac{T_1 - t_2}{T_2 - T_1 + T_1 - t_1} = \frac{T_1 - t_2}{(T_1 - t_1)[\frac{T_1 - T_2}{T_1 - t_1}]}$$

or

$$\phi = \frac{T_1 - t_2}{(T_1 - t_1)(1 - \epsilon_h)}$$

Moreover

$$\phi = \frac{T_1 - t_2}{(T_1 - t_1)(1 - \epsilon_h)} = \frac{T_1 - t_1 + t_1 - t_2}{(T_1 - t_1)(1 - \epsilon_h)}$$

$$= \frac{(T_1 - t_1)[1 - (\frac{t_2 - t_1}{T_1 - t_1})\ (\frac{T_1 - T_2}{T_1 - T_2})]}{(T_1 - t_1)(1 - \epsilon_h)}$$

$$= \frac{1 - \left(\frac{T_1 - T_2}{T_1 - t_1}\right)\left(\frac{t_2 - t_1}{T_1 - T_2}\right)}{1 - \epsilon_h}$$

But $R_h = (t_2 - t_1)/(T_1 - T_2)$ so that

$$\phi = \frac{1 - \epsilon_h R_h}{1 - \epsilon_h}$$

or

$$\epsilon_h = \frac{\phi - 1}{\phi - R_h} \tag{39}$$

where ϕ is given by eq. (38)

The neatness of the form of eq. (39) is deceptive because, unfortunately, it cannot be used to determine a unique value for ϵ_h. The reason for this can be found in an inspection of eq. (38) which provides the value of ϕ which is used in eq. (39).

Notice in eq. (38) that Z, W and N_c are all functions of the product aL. On the other hand, w, z and R are functions of a only. Thus, it is impossible to vary a and L independently and still achieve a unique solution.

For example, suppose a = 50 m^2/m and L = 5 m so that aL = 250. A value of ϕ may be obtained from eq. (38) using these values. However, if a = 100 and L = 2.5 so that aL is still equal to 250, an entirely different value of ϕ is obtained because a = 100 rather than 50.

One should resist the temptation to multiply numerator and denominator of eq. (38) by L thereby creating a situation where only Z, W and N_c appear along with a new $R' = RL$. Such a procedure is doomed to failure because in dealing with an equation derived from n equations in n+1 unknowns, one cannot create the $n+1^{th}$ equation by multiplying one of the n equations by a constant. This makes the $n+1^{th}$ equation so obtained linearly dependent on one of the original n equations and the entire set becomes linearly dependent.

This section represents an attempt to obtain $\varepsilon = f(R, N_{tu})$ and the attempt has not been successful. It is now time to turn to the computer and this will be done in Section IV.

IV. NUMERICAL AND COMPUTER ANALYSIS

With a closed form solution for ϵ as a function of R and N_{tu} not attainable as indicated in Section III, it becomes apparent that an alternative method is needed to determine the effectiveness for the 1-3:2C and 1-3:2P heat exchangers. Kern and Kraus [Ref. 12: pp. 306-360] describe a computer code for a thermal analyzer. This code (program) makes use of node equations generated by finite differences and it employs a Cholesky LU decomposition scheme.

The Cholesky decomposition, as explained by Stewart [Ref. 13: pp. 134-144], is best used when decomposition in the presence of positive definite matrices is requested. This is the case at hand where one tries to solve numerically for the temperatures that lead to the effectiveness of the 1-3:2C and 1-3:2P heat exchangers.

A. THERMAL ANALYZER TVSSI

The computer program employed is Program TVSSI (Appendix A) which is an adaptation of the thermal analyzer program called TVSS2 and listed by Kern and Kraus [Ref. 12: pp. 306-360]. The adaptation consisted of changing the program so that it could perform the computations in the SI system of units and be receptive to the use of a specially created input file. Also it should be noted that TVSS2 was written to be used in conjunction with the Honeywell H-1800 computer system. Therefore, it had to be modified to run on the Naval Postgraduate School's IBM 3033 AP system.

The program itself is a non-linear equation solver that determines the temperatures at a prescribed number of node-points or nodes from a set of node equations in almost any framework (i.e., network analysis, field plotting, or fluid flow distribution). It has certain features that make it primarily an equation solver for thermal analysis. These features include:

1. an ability to linearize radiation terms.
2. an ability to allow any of the coefficients in the node equations to vary with temperature.
3. an ability to provide constant heat input and heat input as a function of temperature at any node.
4. an ability to consider other modes of heat transfer that are non-linear such as boiling and natural convection.

As stated earlier, the program utilizes the Cholesky decomposition scheme and, because of the linearization of the radiation terms (a feature of the program that is used even though radiation does not appear in this $\varepsilon - N_{tu}$ study), the program is iterative.

Cholesky's decomposition consists of finding a lower triangular matrix [L] which is capable of reducing the original system of equations.

$$[K][T] = [Q] \tag{40}$$

or

$$[K][T] - [Q] = [0] \tag{41}$$

to the unit triangular form

$$[A][T] - [B] = [0] \tag{42}$$

so that the sought after elements of the column vector [T] can be obtained by backward substitution.

Suppose, for example, that [K] is 3x3 and assume that the system [K][T] = [Q] has been reduced to the form [A][T] - [B] = [O]. In this event a premultiplication by [L] will return the system to its original form, that is

$$[L]([A][T] - [B]) = [K][T] - [Q] = [O]$$

This implies that

$$[L][A] = [K] \tag{43}$$

and

$$[L][B] = [Q] \tag{44}$$

These equations allow the determination of [L], [A], and [B] in a very simple manner and the matrices are uniquely determined because [K] and [Q] are known, or, at least are known after each iteration because the elements of [K] are linearized. For a 3x3 system

$$
\begin{array}{ccc}
[K,Q] & [L] & [A,B] \\
\begin{bmatrix} k_{11} & k_{12} & k_{13} & q_1 \\ k_{21} & k_{22} & k_{23} & q_2 \\ k_{31} & k_{32} & k_{33} & q_3 \end{bmatrix}
= &
\begin{bmatrix} l_{11} & 0 & 0 \\ l_{21} & l_{22} & 0 \\ l_{31} & l_{32} & l_{33} \end{bmatrix}
&
\begin{bmatrix} 1 & a_{12} & a_{13} & b_1 \\ 0 & 1 & a_{23} & b_2 \\ 0 & 0 & 1 & b_3 \end{bmatrix}
\end{array}
$$

one may obtain the following for the elements of [L], [A], and [B].

$$k_{i1} = (1)l_{i1} + (0)l_{i2} + (0)l_{i3} = l_{i1} \qquad (45)$$

which shows that the first column of [L] is identical to the first column of [K].

$$k_{1j} = (l_{11})a_{1j} + (0)a_{2j} + (0)a_{3j} = l_{11}a_{1j} = k_{11}a_{1j} \qquad (46)$$

which shows that the first row of [A] is equal to the first row of [K] divided by $[k_{11}]$ and then

$$k_{22} = l_{21}a_{12} + l_{22}(1) \;, \quad l_{22} = k_{22} - l_{21}a_{12}$$

$$k_{23} = l_{21}a_{13} + l_{22}a_{23} \;, \quad a_{23} = (k_{23} - l_{21}a_{13})/l_{22}$$

$$k_{32} = l_{31}a_{12} + l_{32}(1) \;, \quad l_{32} = k_{32} - l_{31}a_{12}$$

$$q_2 = l_{21}b_1 + l_{22}b_2 \;, \quad b_2 = (q_2 - l_{21}b_1)/l_{22}$$

In the foregoing manner the elements of [L], [A] and [B] are obtained successively in terms of previously determined elements in a progression that goes horizontally from l_{22} on. Thus the general relationship are seen to be

$$l_{ij} = k_{ij} - \sum_{r=1}^{j-1} l_{ir}k_{rj} \qquad (47)$$

with

$$l_{i1} = k_{i1} \qquad (48)$$

and

$$a_{ij} = \frac{1}{l_{ii}} [k_{ij} - \sum_{r=1}^{i-1} l_{ir}a_{rj}] \qquad (49)$$

with

$$a_{ij} = \frac{k_{ij}}{k_{11}} \qquad (50)$$

Moreover, it is observed that if [K] is symmetrical which it must be in our coupled set of equations ($k_{ij} = k_{ji}$) then

$$l_{ij} = a_{ji}l_{ii} \quad (i,j = 1,2,3,\ldots\ n-1;\ i \neq j) \qquad (51)$$

The modification for computation in the SI system was quick and simple. It involved changing a numeral in two places (460 to 273) and some format statements (°F to °C and Btu/hr to Watts).

The conversion of TVSS2 to TVSSI for running on the IBM 3033 AP system (which uses the FORTVS compiler) required a modification to the Fortran program language used in TVSS2 in order to compile the FORTVS (basically international Fortran 77) system used on the IBM 3033.

B. INITIAL MODELING

Initially, the model was designed to find the temperatures that would allow computation of a single effectiveness value after a detailed set of capacity rate, coefficients and surface data were entered into the program. From the scope of the problem it was realized that multiple runs of the thermal analyzer program (TVSSI) would be needed. It therefore became necessary to develop a program that given C_h, C_c, U, A, T_1, and t_1, an input file would be created for use by the modified version of the thermal analyzer (TVSSI).

The first step taken was to develop a program to create an input file for TVSSI that would yield the effectiveness for a 1-4 heat exchanger which could be compared to the existing analytical solution for the effectiveness of a 1-4 exchanger. With this accomplished and confidence established, similar programs for the 1-3:2C and 1-3:2P exchangers could be developed. This program was called NTU14 (See Appendix B) and the following parameters were used for all runs.

1. 250 nodes were used.
2. The initial temperature for the computer to begin the iterative process was set at 200°C.

3. An eventual accuracy of .05 between the final and next to last iterations was used.

4. A radiation coefficient convergence factor of 0.66667 between iterations was used.

5. The maximum number of iterations that the computer was allowed to perform was set at 12.

6. A damping factor of .8 was set as an initial damping based on the number of non-linear terms in all of the node equations.

When the values of C_h, C_c, U, A, T_1 and t_1 are set, N_{tu} and R are then compiled and an input file for TVSSI was generated. In this file all node equations and internode conductance values were determined. This program determined and specified the nodes that interact with each other and the methods by which the interaction takes place such as conduction, forced convection, and fluid flow.

The program, NTU14, makes use of the fact that each term in a node equation shows three things. The first is the node that is coupled for heat flow with the node in question. The second is the method of heat flow between the nodes. In this case forced convection and fluid flow are used. Finally, the node equation shows the magnitude of the internode heat flow. Here all the pieces of information are collected and presented for use by TVSSI as an input file with all items in the proper format.

A comparison of the effectiveness for the 1-4 exchanger developed by the computer to that of using the closed form analytical solution for effectiveness developed by Kraus and Kern [Ref. 8] as shown by equation (52)

$$\varepsilon = \frac{2}{1 + R + \frac{1}{2}[1 + 4R^2]^{1/2}\coth\left(\frac{N_{tu}[1 + 4R^2]}{4}\right) + 4\tanh\frac{N_{tu}}{4}} \quad (52)$$

was then undertaken. The results of this comparison showed that over the entire range of R from .01 to 1.0 for varying values of N_{tu} from 0 to 3.25 less than a 0.5% difference was ever realized. A small sample of these results are provided in Table 1. The conclusion to be drawn here, is that the methodology used to develop the computer program NTU14 for input to TVSSI for finding effectiveness was sound and could then be used in the development of the 1-3:2C and 1-3:2P exchanger methodology.

C. DEVELOPED MODELS FOR 1-3:2C AND 1-3:2P HEAT EXCHANGERS

The same technique used in developing the program NTU14 was used to generate computer programs NTU32C and NTU32P. These are listed in Appendices C and D. The departure for each of these programs from the NTU14 program is in the number of nodes; they are based on 200 node models as shown in Figures 4.1 and 4.2. An example of the output file generated from one of these programs is found at Appendix E. It is these values shown in Appendix E that are used by the thermal analyzer to determine the temperatures T_2 and t_2 for the specific set of given initial parameters C_h, C_c, R, A, T_1, and t_1.

TABLE 1

COMPUTER TO ANALYTICAL COMPARISON

FOR R = .5

N_{tu}	ANALYTICAL RESULTS	COMPUTER RESULTS	% DIFFERENCE
0.05	.0482	.0482	0.00
0.25	.2094	.2090	0.20
0.50	.3569	.3559	0.28
0.75	.4628	.4612	0.35
1.00	.5398	.5377	0.39
1.25	.5963	.5940	0.39
1.50	.6379	.6354	0.40
2.00	.6915	.6886	0.42
2.50	.7206	.7177	0.40
3.00	.7360	.7333	0.37
3.25	.7406	.7381	0.34

D. SCOPE OF COMPUTER ANALYSIS

At this point it is possible to let the computer solve for temperature values that yield a value for effectiveness based on a specific set of initial parameters. However, it must be realized that many computer runs are required to generate enough data to ensure confidence in the results which cover a wide range of capacity rate ratios and N_{tu} values.

To efficiently expedite the computer task, the Multiple Virtual System (MVS) with Job Entry Subsystem and Networking (JES3) was utilized. The MVS coupled with JES3 is more commonly referred to as batch processing. Based on trial and error, it was determined that in order to build a solid data base, eleven different values for effectiveness were needed to best represent a particular value of R. This was required over a range of R from R = 0.1 to 1.0 in increments of .01. In all, 200 curves for both the 1-3:2C and 1-3:2P exchangers were needed. This means that 2,200 unique values of effectiveness needed to be found, plus 100 values of effectiveness to be used for comparison with the 1-4 exchangers.

To complete this task, TVSSI was slightly modified in accordance with the appropriate guidelines of the job control language (JCL) needed to run on the batch processing system. These modifications are few and were needed only at the beginning and end of TVSSI. The modified version of TVSSI has been called TVCOUNT with changes shown in Appendix F. It was TVCOUNT that was then used to activate TVSSI.

It also became necessary to modify the three input file programs NTU14, NTU32P and NTU32C such that they needed to be compiled only once. They were then loaded in a library file to be used when called by another program. New programs utilizing the batch system were written that could easily be loaded with the appropriate input data for a specific R value. These, which are referred to as "sister programs," were used to go from the library file to TVSSI and cause TVSSI to be executed eleven times under TVCOUNT covering the desired range of N_{tu} for a specific R value. The revised input files called NTU14C, NTU32CC, NTU32PC and their associated "sister execution programs," NTU14L, NTU32CL, NTU32PL, are found in Appendices G through L.

The overall system flow chart of how all of the foregoing is accomplished is found in Figures 4.3, 4.4 and 4.5. It is noted from these figures that TVSSI is referred to as TVSSIA through TVSSIV. These are the same programs as TVSSI but for bookkeeping purposes by the computer they are labeled A through V.

E. COMPUTER RESULTS

Upon completion of all data collection from the computer output, plots of effectiveness vs. N_{tu} for the whole range of R were plotted and are shown in Appendices M and N. Thirty-three different plotting programs utilizing the "Display Integrated Software System and Plotting Language (DISSPLA)," were written to graph the data obtained. An example of one of these programs is provided in Appendix O.

Examination of the graphs in Figures 4.6, 4.7, and 4.8 shows that the 1-3:2C exchangers outperform the 1-3:2P, 1-2 and 1-4 exchangers. This is true in all cases, and, because of this, only a sample of the data was chosen to be shown in these figures. Furthermore, it is noted that at higher N_{tu} values, the effectiveness of the 1-3:2C exchanger is better than all of the others considered, while at higher capacity rate ratios, the effectiveness of 1-3:2C exchanger even begins to outperform the others at lower N_{tu} values. This increase in performance is easily understood because it has been proven by Kern [Ref. 1: pp. 139-137] and others that greater temperature differences result when process streams are in counterflow than parallel flow. Thus, when there is a combination of the two phenomena (counterflow and parallel flow) occurring, then it becomes apparent that the extra counterflow pass must increase the exchanger's overall performance.

As shown in the graphs in Appendices M and N, effectiveness increases as both R and N_{tu} increase. These curves can be used to give a graphical approximation of effectiveness if that is all that is required. From the empirical data used to develop these curves, further investigation into the development of equations for these curves which may be used to determine an exact value of effectiveness when R and N_{tu} are known, is undertaken in Section V.

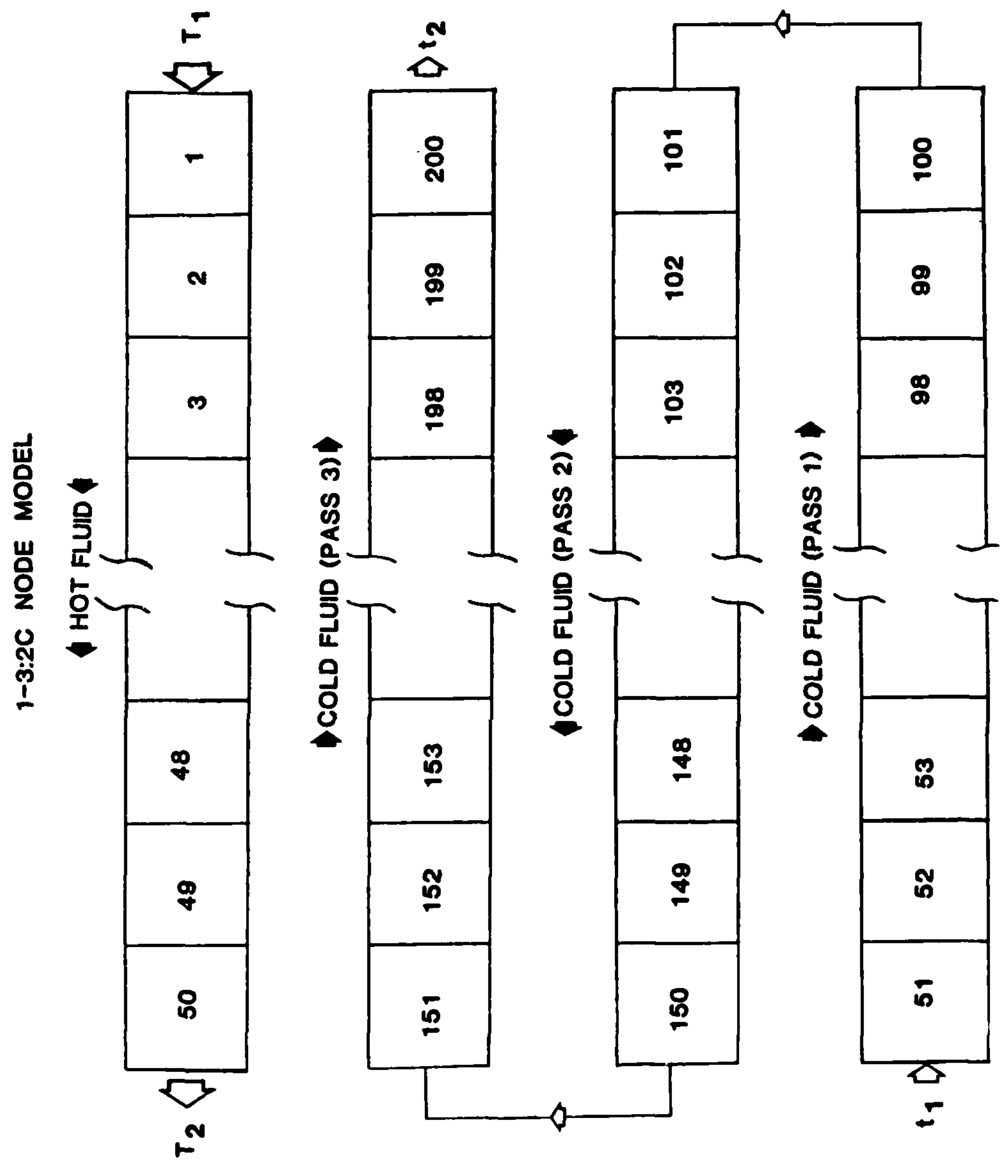


Figure 4.1 Nodal Arrangement for the 1-3:2C Exchanger

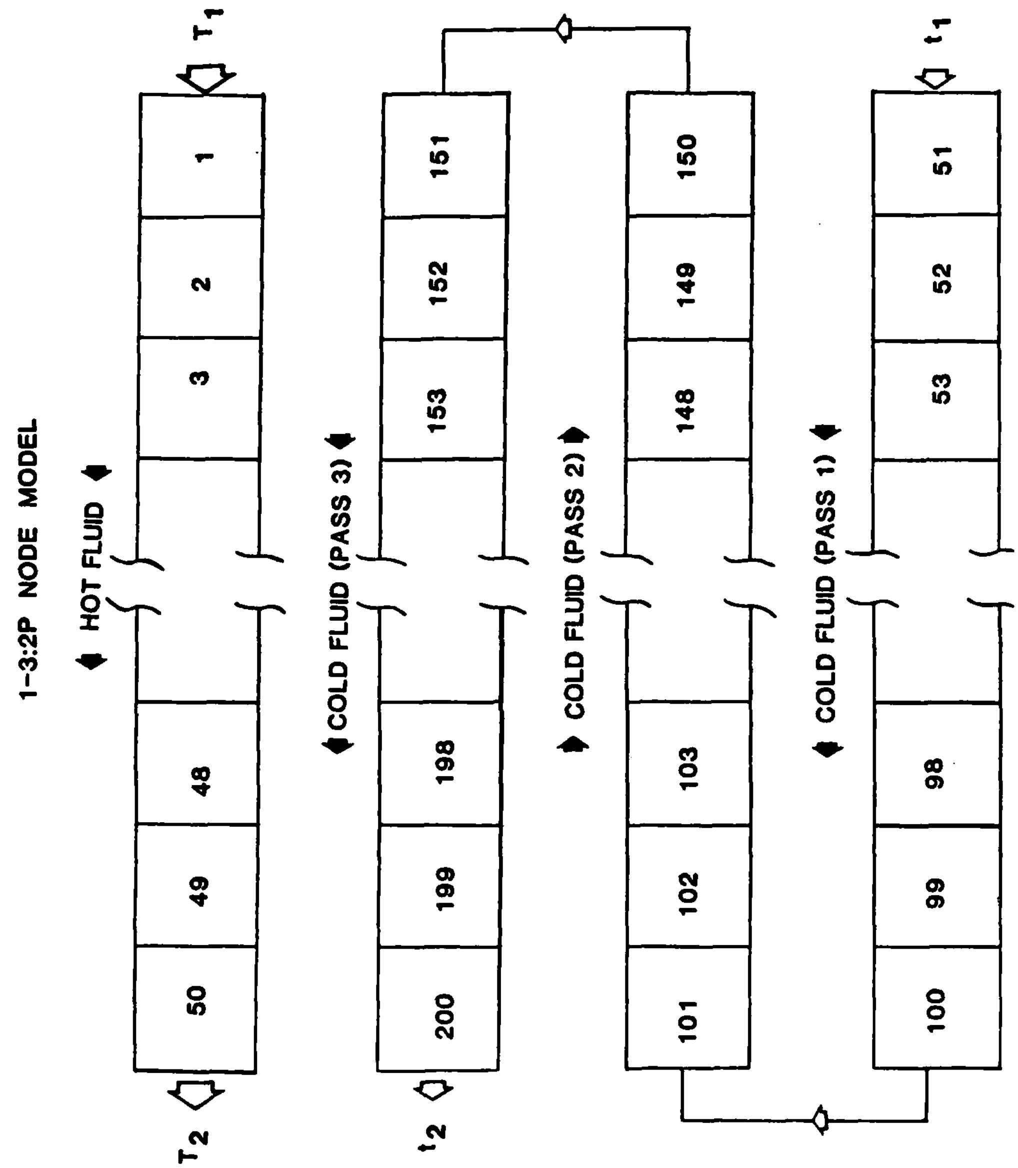


Figure 4.2 Nodal Arrangement for the 1-3:2P Exchanger

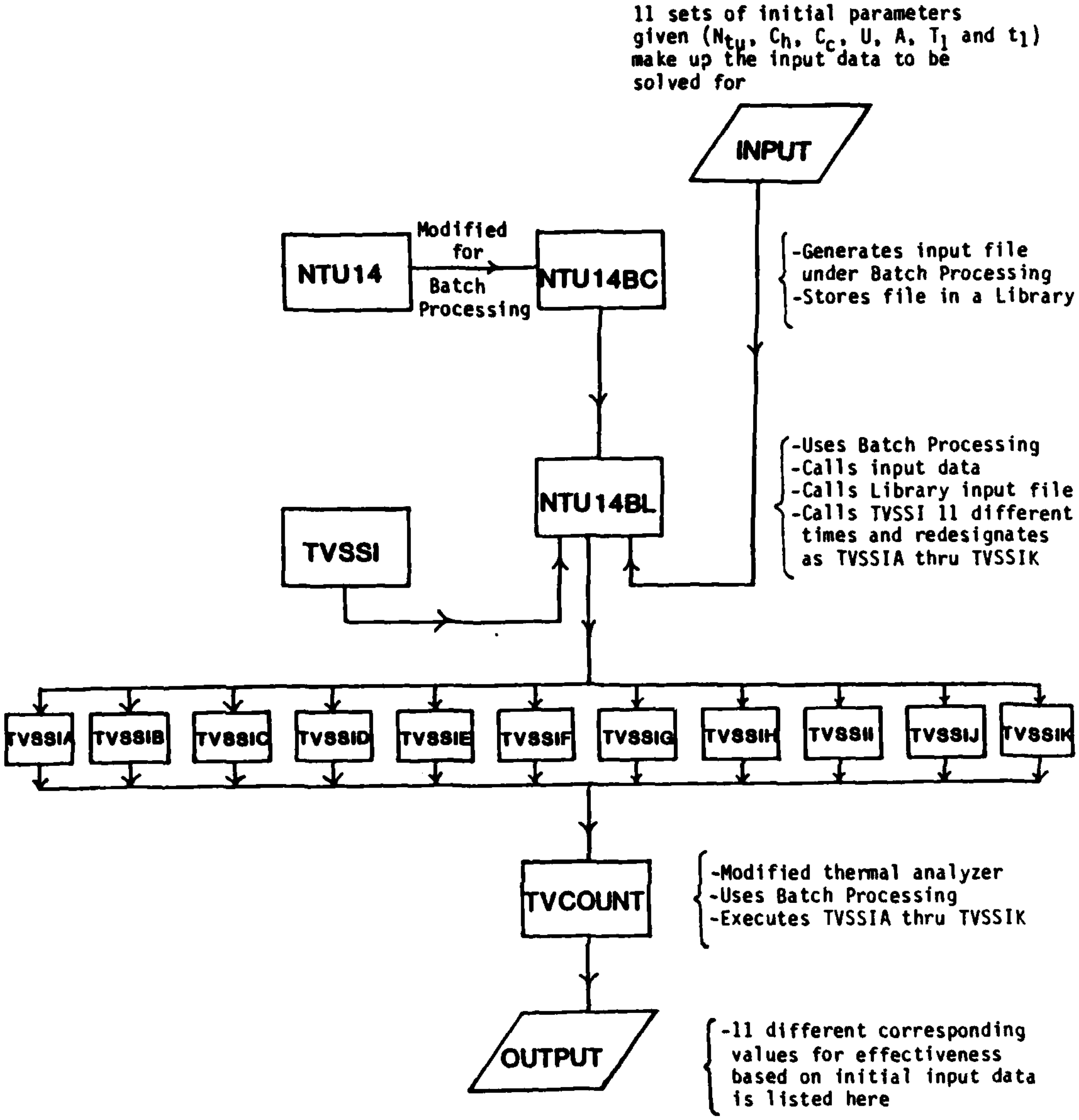


Figure 4.3 Computer Systems Flow Chart for 1-4 Exchanger Model

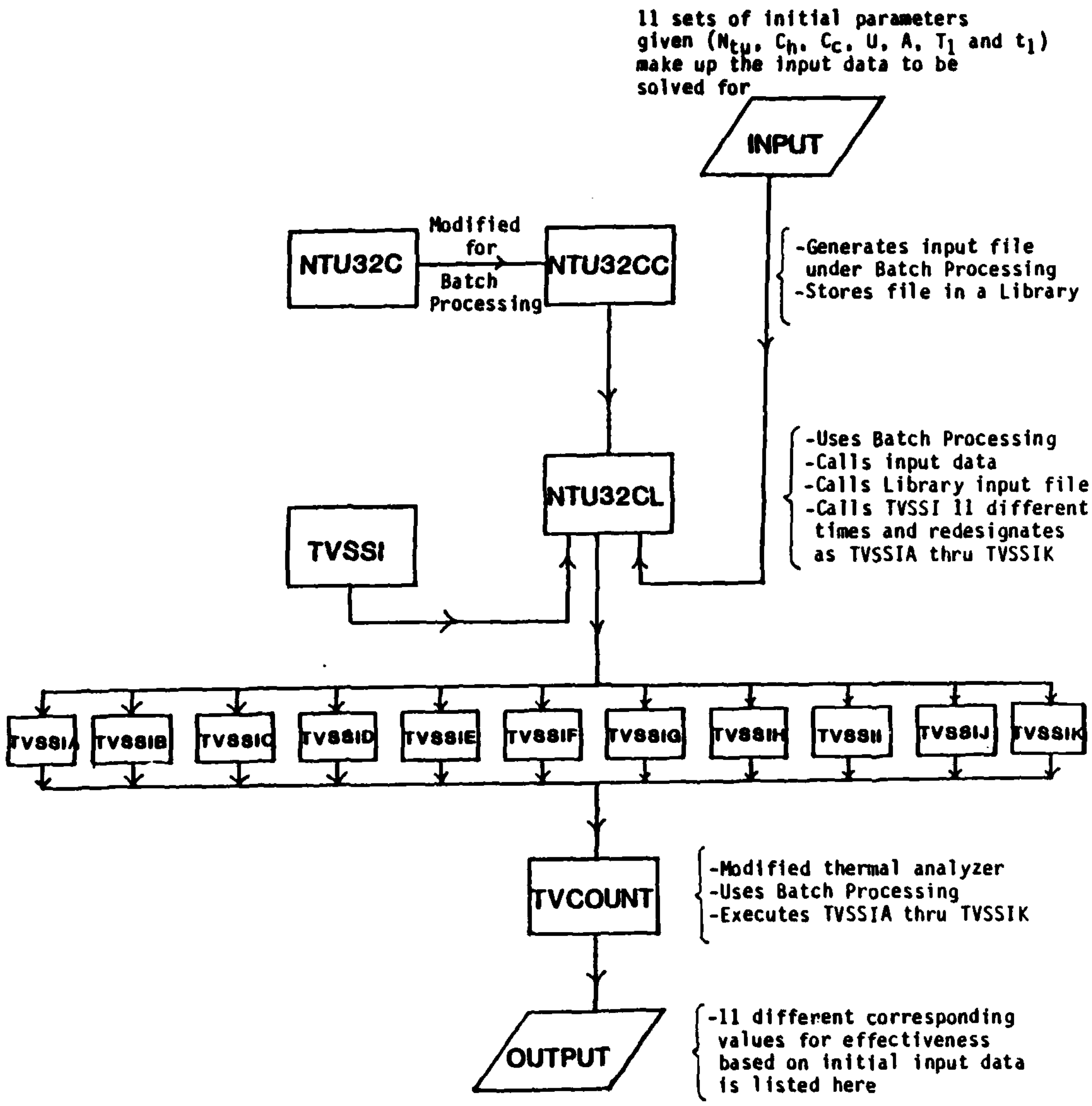


Figure 4.4 Computer Systems Flow Chart for 1-3:2C Exchanger Model

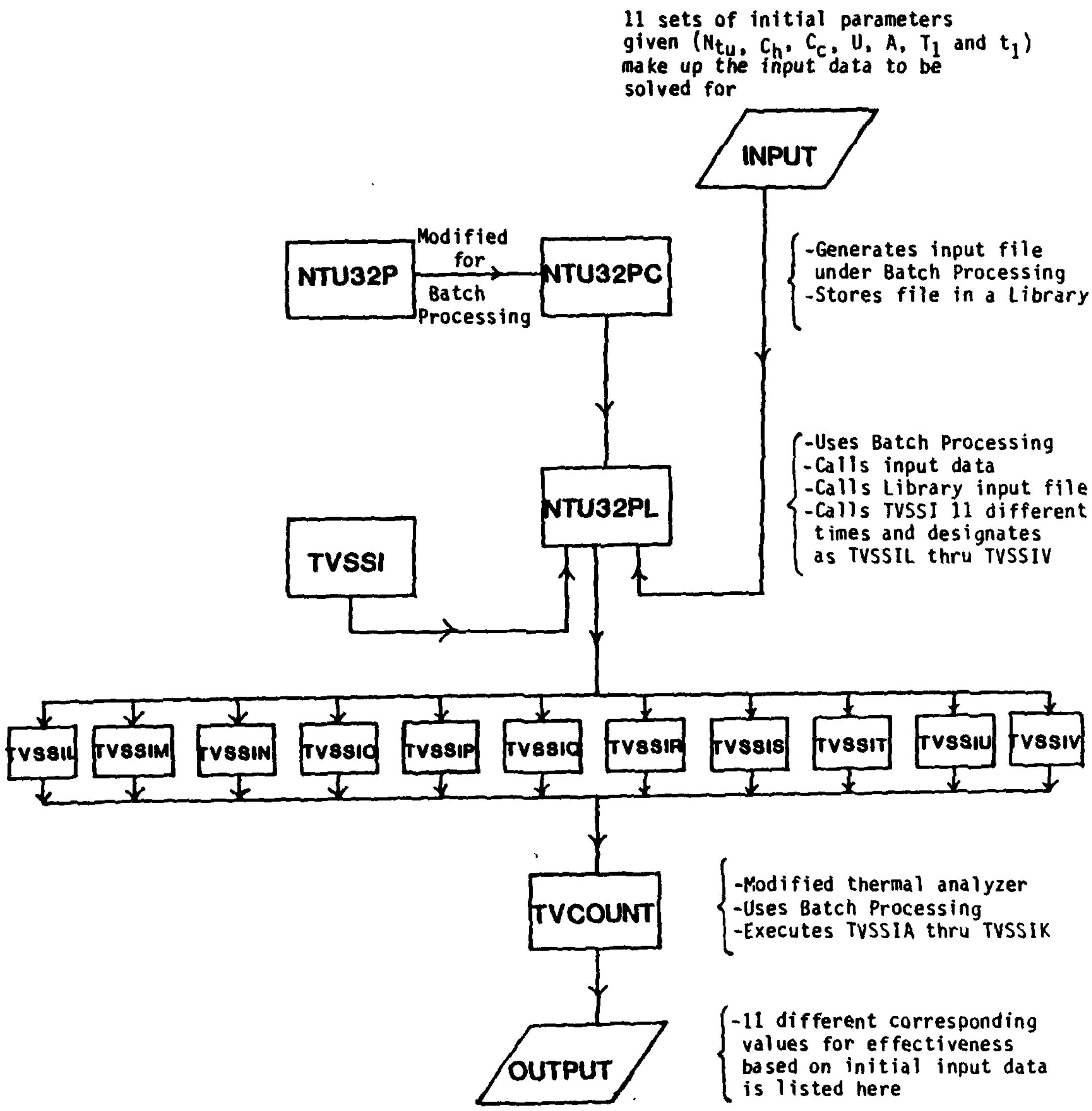


Figure 4.5 Computer Systems Flow Chart for 1-3:2P Exchanger Model

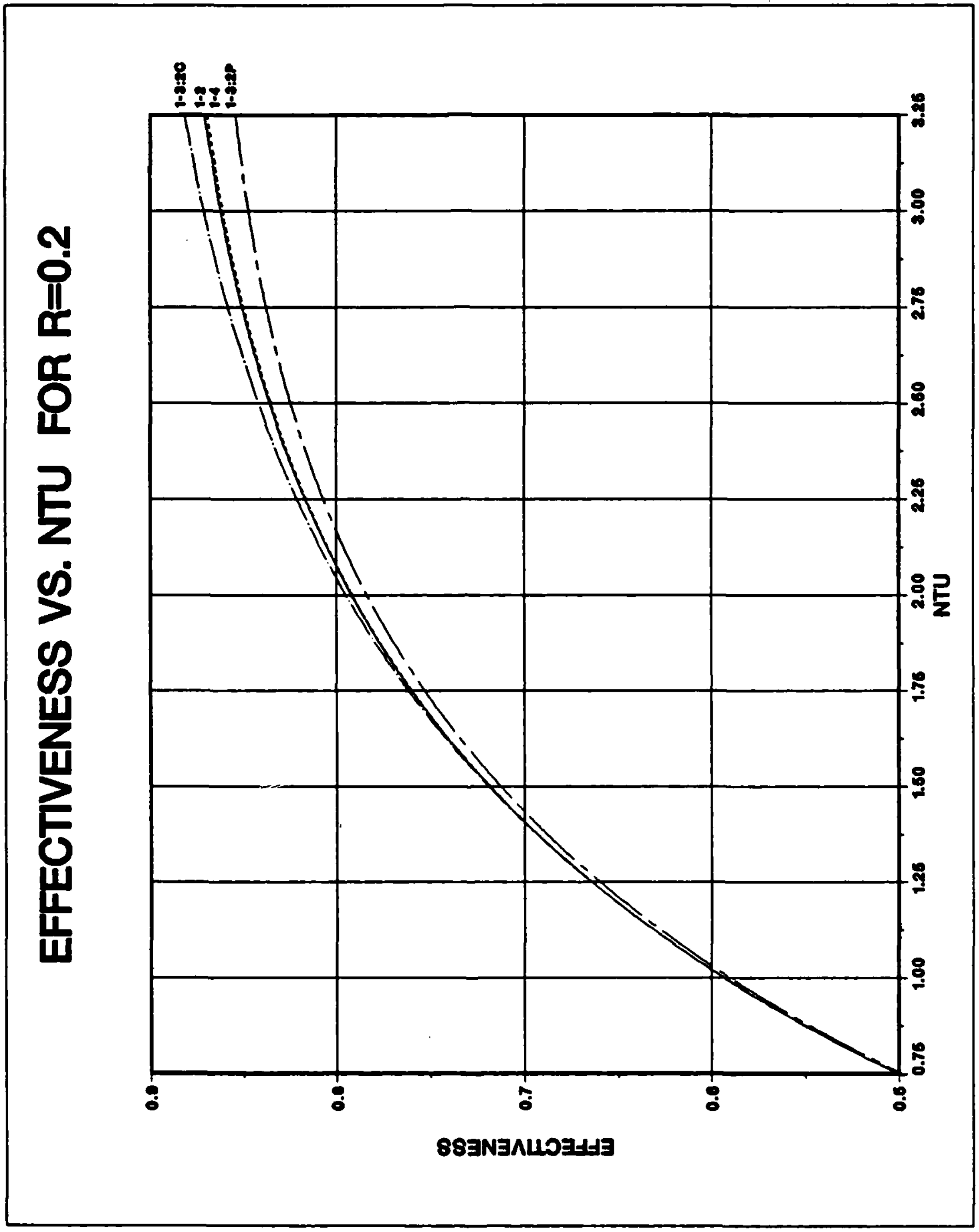


Figure 4.6 Comparison of Analytical (1-2 and 1-4) to Computer Results (1-3:2C and 1-3:2P) at R = .2

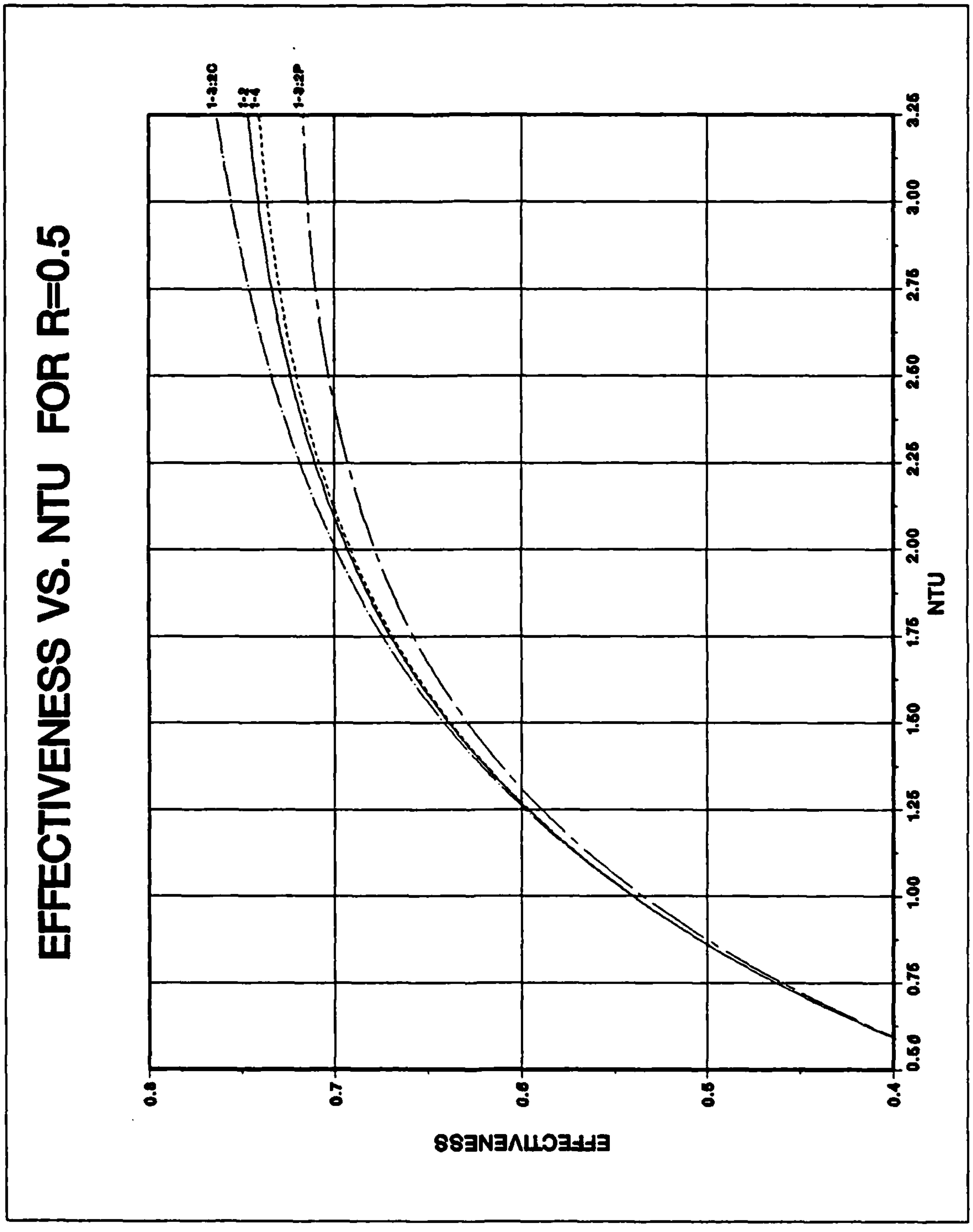


Figure 4.7 Comparison of Analytical (1-2 and 1-4) to Computer Results (1-3:2C and 1-3:2P) at R = .5

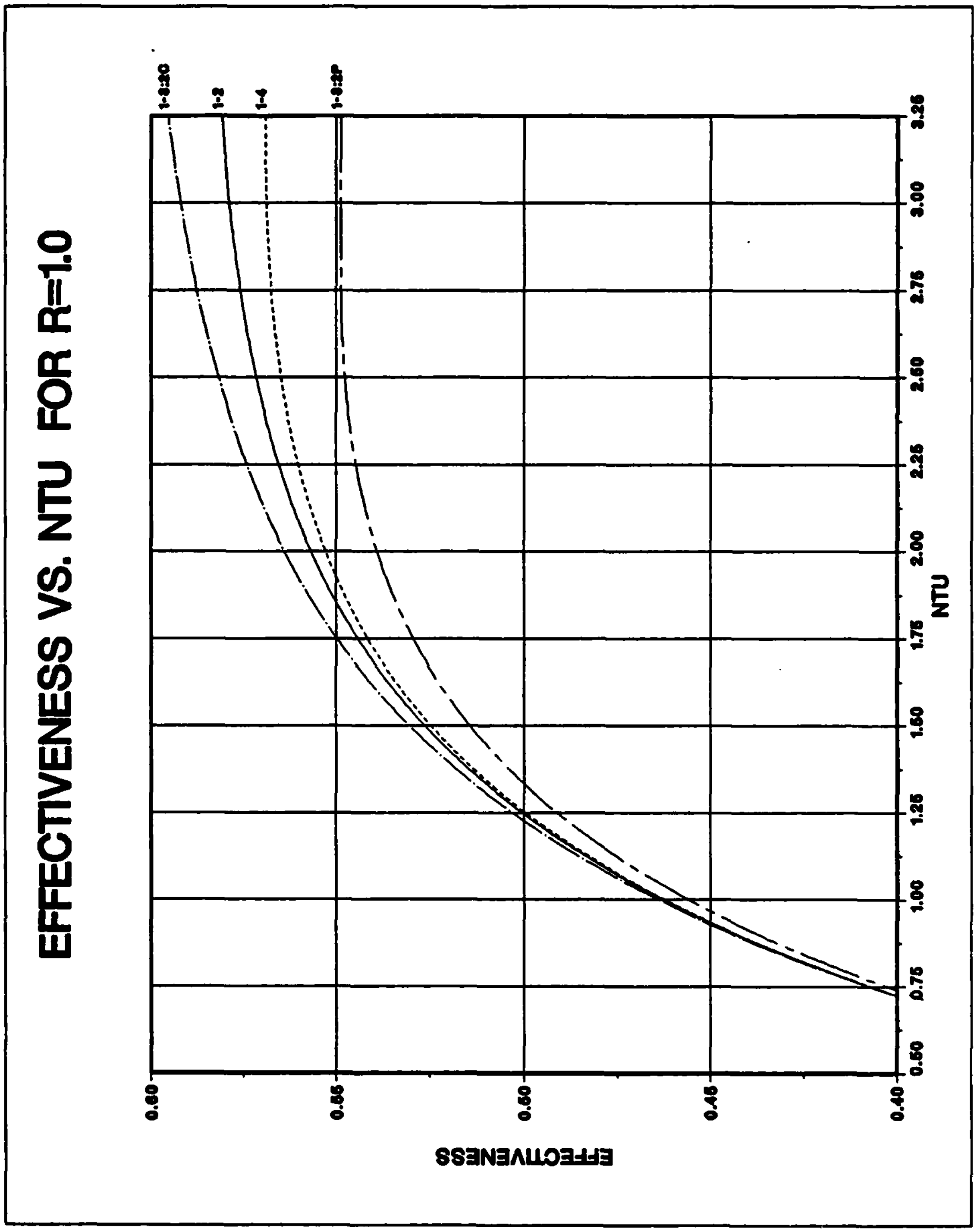


Figure 4.8 Comparison of Analytical (1-2 and 1-4) to Computer Results (1-3:2C and 1-3:2P) at R = 1.0

V. POLYNOMIAL REGRESSION

A. DEVELOPMENT OF POLYNOMIAL EQUATIONS

The empirical data obtained for the two 1-3 heat exchangers was designed to cover an extensive range of R values varying from 0.01 to 1.0 in increments of 0.01. As discussed earlier in Section IV, the data obtained at a specific value of R is the computer evaluated result of the effectiveness, for an associated N_{tu} value. With this accomplished, it then becomes possible to graph separate curves for each of the different R values as shown in Appendices M and N. Through a polynomial regression technique, as discussed in this section, it is also possible to develop implicit equations for the curves with $\varepsilon = f(N_{tu}, R)$. It is also apparent from an inspection of the graphical representation of the empirical data in Appendices M and N, that the curves conform to a high degree polynomial. However, further analytical investigation is needed to ascertain the exact order of the polynomial terms. This investigation will not only lead to the order of the polynomial, but to the specific equation for each curve.

By use of polynomial regression, the least-squares method can be readily extended to best fit the data to the m^{th}-degree for the polynomial

$$y = A_0 + A_1x + A_2x^2 + \ldots A_mx^m \tag{53}$$

with the error defined by

$$e_i = D_i - y_i = D_i - A_0 - A_1 x - A_2 x_i^2 - \ldots - A_m x_i^m$$

where D_i represents the empirical data value corresponding to x_i, x_i being free of error.

The objective is to minimize the sum of the squares of the residuals, S_r,

$$S_r = \sum_{i=1}^{m} e_i^2 = \sum^{m} (D_i - A_0 - A_1 x_1 - A_2 x_2^2 + \ldots A_m x^m)^2 \quad (54)$$

Because at a minimum, the partial derivatives $\partial S_r / \partial A_0$, $\partial S_r / \partial A_1 \ldots \partial S_r / \partial A_m$ vanish, after taking the derivative of S_r with respect to each of the coefficients of the polynomial, it can be seen that

$$\frac{\partial S_r}{\partial A_0} = 0 = -2 \sum (D_i - A_0 - A_1 x_i - A_2 x_i^2 - \ldots - A_m x_i^m)$$

$$\frac{\partial S_r}{\partial A_1} = 0 = -2 \sum x_i (D_i - A_0 - A_1 x_i - A_2 x_i^2 - \ldots - A_m x_i^m)$$

$$\frac{\partial S_r}{\partial A_2} = 0 = -2 \sum x_i^2 (D_i - A_0 - A_1 x_i - A_2 x_i^2 - \ldots - A_m x_i^m)$$

$$\vdots \qquad\qquad\qquad \vdots$$

$$\frac{\partial S_r}{\partial A_m} = 0 = -2 \sum x_i^m (D_i - A_0 - A_1 x_i - A_2 x_i^2 - \ldots - A_m x_i^m)$$

Then by dividing by -2 and rearranging we obtain

$$A_0M + A_1 \Sigma x_i + A_1x_i^2 + \ldots + A_m \Sigma x_i^m = \Sigma D_i$$

$$A_0 \Sigma x_i + A_1 \Sigma x_i^2 + A_2 \Sigma x_i^3 + \ldots + A_m \Sigma x_i^{m+1} = \Sigma x_iD_i$$

$$A_0 \Sigma x_i^2 + A_1 \Sigma x_i^3 + A_2 \Sigma x_i^4 + \ldots + A_m \Sigma x_i^{m+2} = \Sigma x_i^2D_i$$

$$\vdots$$

$$A_0 \Sigma x_i^m + A_1 \Sigma x_i^{m+1} + A_2 \Sigma x_i^{m+2} + \ldots + A_m \Sigma x_i^{2m} = \Sigma x_i^mD_i$$

where all summations are from i=1 through n. All of the foregoing m+1 equations are linear and have m+1 unknowns: A_0, A_1, A_2, ... A_m. The coefficients of the unknowns can be calculated directly from the observed data. Thus, the problem of determining a least-squares polynomial of degree m is equivalent to solving a system of m+1 simultaneous linear equations. Putting the equations in matrix form yields

$$\begin{bmatrix} N & \Sigma x_i & \Sigma x_i^2 & \Sigma x_i^3 & \ldots & \Sigma x_i^m \\ \Sigma x_i & \Sigma x_i^2 & \Sigma x_i^3 & \Sigma x_i^4 & \ldots & \Sigma x_i^{m+1} \\ \Sigma x_i^2 & \Sigma x_i^3 & \Sigma x_i^4 & \Sigma x_i^5 & \ldots & \Sigma x_i^{m+2} \\ \cdot & \cdot & \cdot & \cdot & & \cdot \\ \cdot & \cdot & \cdot & \cdot & & \cdot \\ \cdot & \cdot & \cdot & \cdot & & \cdot \\ \Sigma x_i^m & \Sigma x_i^{m+1} & \Sigma x_i^{m+2} & \Sigma x_i^{m+3} & \ldots & \Sigma x_i^{2m} \end{bmatrix} [A] = \begin{bmatrix} \Sigma D_i \\ \Sigma x_iD_i \\ \Sigma x_i^2D_i \\ \cdot \\ \cdot \\ \cdot \\ \Sigma x_i^mD_i \end{bmatrix}$$

[Ref. 15: pp. 302-309 and Ref. 16: 468-474].

From this point on, one finds that it is best to use a computer to assist in solving the simultaneous equations and this will also help alleviate any ill-conditioning that may otherwise occur. An existing "curvefit" program available through NON-IMSL [Ref. 16] and found in Appendix P was used although some modifications were made to the original program to best accommodate the goals of this work.

To determine the order of polynomial that should eventually be used, one increases the degree of the approximating polynomial as long as there is a statistically significant decrease in the variance σ^2, which is computed by

$$\sigma^2 = \frac{\Sigma e_i^2}{N - m - 1} \qquad (55)$$

In otherwords, the selection of the optimum degree polynomial is contingent upon a decreasing variance and once the variance begins to increase, the degree of the polynomial becomes too high. For all cases, it was found that the ε - N_{tu} developed curves are of the 5th order.

As shown in Figures 5.1 and 5.2 the computed values of effectiveness vs. N_{tu} for R = 0.1, 0.5 and 1.0 for both flow arrangements, (1-3:2P) and (1-3:2C), have been graphed and fitted by a 5th degree polynomial. Because all computed values for effectiveness follow a predictable trend, only a sample of the data covering the whole range of values of R have been shown. It is clear that the graphic interpretation strongly backs what is known analytically from the polynomial regression technique. Where the relationship for $\varepsilon = f(N_{tu}, R)$ is found explicitly from

$$\varepsilon = A_5 N_{tu}^5 + A_4 N_{tu}^4 + A_3 N_{tu}^3 + A_2 N_{tu}^2 + A_1 N_{tu} + A_0 \quad (56)$$

while the corresponding coefficients A_5, A_4, A_3, A_2, A_1, and A_0 relating to a specific value of R are found in Tables 2 and 3 for the (1-3:2P) and (1-3:2C) configurations. An example of how to use this equation in a heat exchanger problem now follows.

B. NUMERICAL EXAMPLE

Consider a heat exchanger containing 400 m^2 of (A = 400 m^2) of surface and operating with an overall heat transfer coefficient of 80 $W/m^2 \cdot C$. Cold fluid at a capacity rate of 10,000 W/°C enters the exchanger at 60°C. Hot fluid at a capacity rate of 20,000 W/°C enters the exchanger at 200°C.

1. Find

The effectiveness (ε) and compute the hot and cold fluid outlet temperatures for the (1-3:2C) shell tube pass arrangement.

2. Assumptions

1) Negligible heat loss to surroundings and kinetic and potential energy changes.

2) Constant thermal and fluid properties for both fluids.

3. Analysis

Here C_c = 10,000 W/°C and C_h = 20,000 W/°C this makes $R = C_{min}/C_{max} = C_c/C_h = (T_1 - T_2)/(t_2 - t_1)$

$$10{,}000/20{,}000 = 0.5.$$

and $N_{tu} = UA/C_c$

$$= 80\ (400)/(10{,}000) = 3.2$$

First, go to Table 2 (page 81) for the (1-3:2C) arrangement with R = 0.5 and find the coefficients

$$A_0 = 0.1294 \times 10^{-2}$$

$$A_1 = 0.98120$$

$$A_2 = -0.66161$$

$$A_3 = 0.27938$$

$$A_4 = -0.66456 \times 10^{-1}$$

$$A_5 = 0.66069 \times 10^{-2}$$

Then apply equation (56) for $N_{tu} = 3.2$

$$\varepsilon = A_5 N_{tu}^5 + A_4 N_{tu}^4 + A_3 N_{tu}^3 + A_2 N_{tu}^2 + A_1 N_{tu} + A_0 \quad (56)$$

$$\varepsilon = 0.769$$

Because $C_c < C_h$

$$\varepsilon = \frac{t_2 - t_1}{T_1 - t_1}$$

and with $T_1 - t_1 = 200 - 60 = 140°C$

$$t_2 - t_1 = \varepsilon(T_1 - t_1)$$
$$= 0.769\ (140)$$
$$= 107.7°C$$

Finally, the outlet cold fluid temperature t_2 is

$$t_2 = 107.7 + t_1$$
$$= 107.7 + 60$$
$$= 167.7°C$$

and the fluid temperature, T_2 is easily found

$$R = \frac{T_1 - T_2}{t_2 - t_1} = 0.5$$

$$T_2 = T_1 - 0.5\ (t_2 - t_1)$$
$$= 200 - 0.5\ (t_2 - t_1)$$
$$= 146.1°C$$

4. Observations

The primary observation made here is that by using the 5th order polynomial equation (56) with the appropriate coefficients found in Table 2 or 3, an accurate value for effectiveness can be found thus allowing one to solve for many more unknown values or prarameters of the heat exchanger (i.e., hot and cold outlet temperatures).

The other observation that is to be made is that when comparing the value for effectiveness computed here against the value for a 1-2 or 1-4 exchanger (0.745 and 0.740 respectively) under the same conditions, one finds that there is a significant difference in exchanger performance as a function of odd or even tube passes and that the 1-3:2C arrangement has a higher effectiveness than either the 1-2 or 1-4 arrangement. From inspection of the curves for the 1-3:2P exchanger at Figure N-1 or N-6 with $R = 0.5$ and $N_{tu} = 3.2$ an approximate value of $\varepsilon = .715$ is obtained. It is clear that this is also less than that of 1-3:2C arrangement. Therefore, it is evident that the 1-3:2C exchanger out-performs not only the 1-2 and 1-4 arrangement but also its counterpart the 1-3:2P exchanger by 3.1%, 3.8% and 7.0% respectfully.

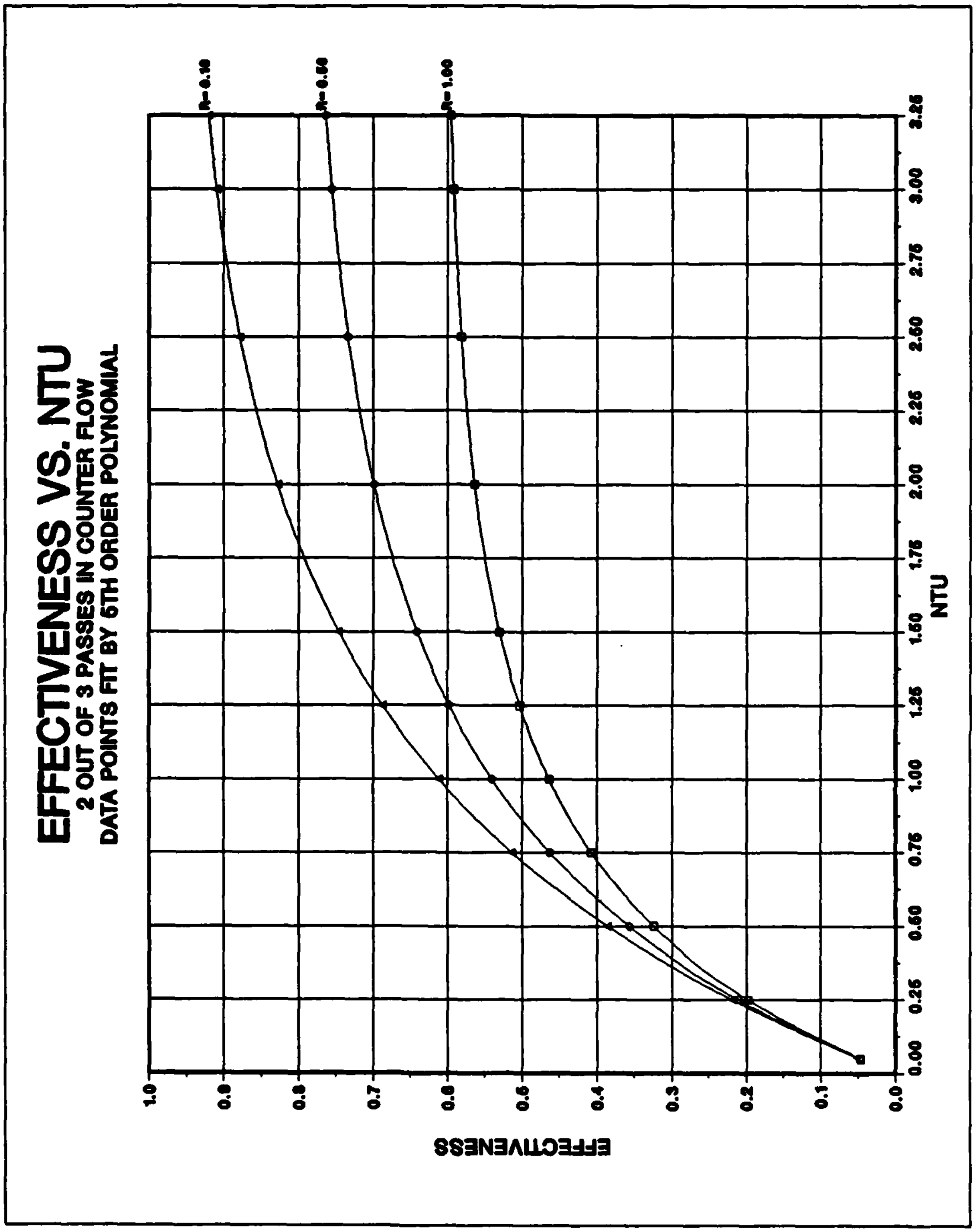


Figure 5.1 1-3:2C Data Fit by a 5th Order Polynomial

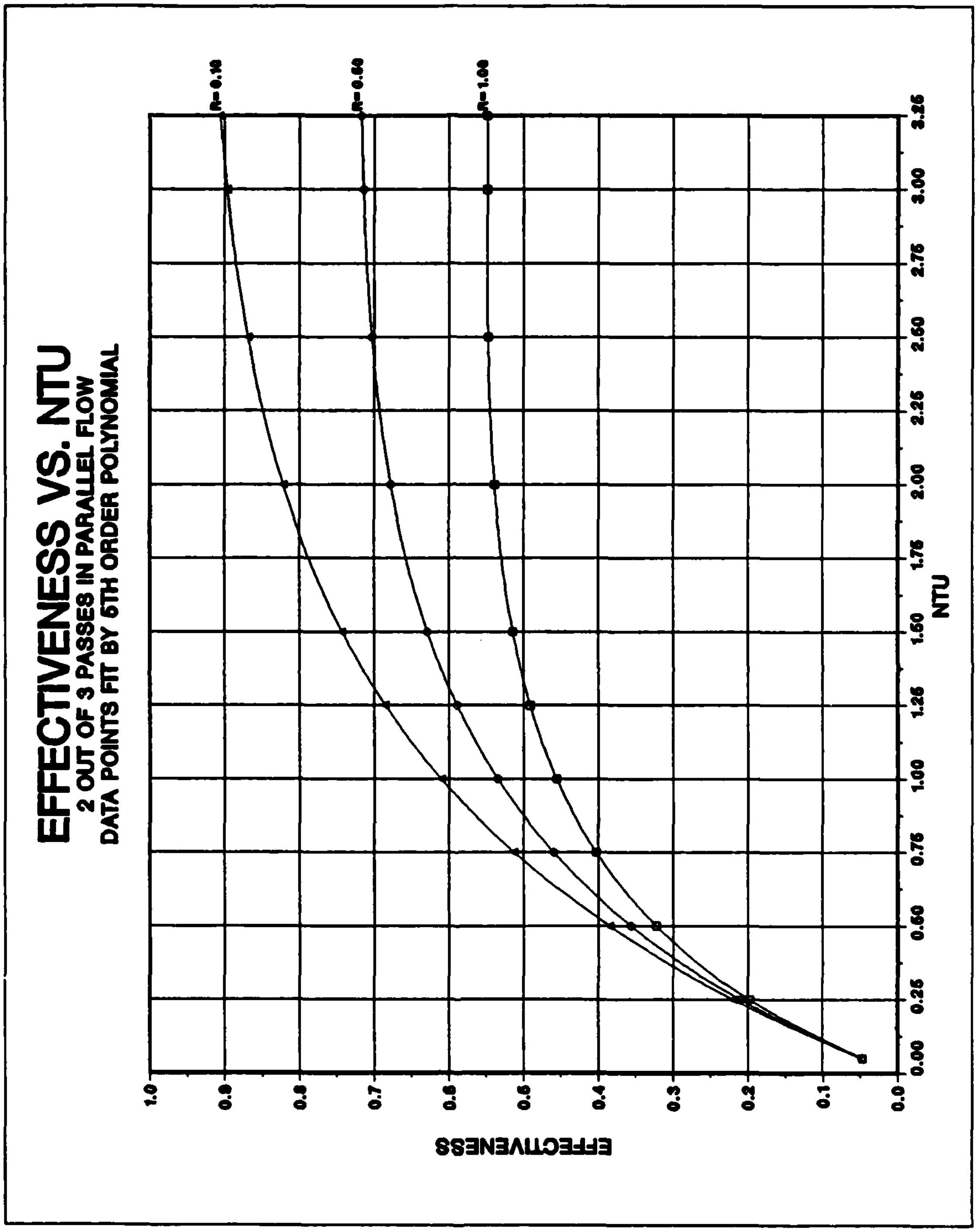


Figure 5.2 1-3:2P Data Fit by a 5th Order Polynomial

TABLE 2

1-3:2C COEFFICIENTS

R	A_0 x10^3	A_1	A_2	A_3	A_4 x10^1	A_5 x10^2
.01	-1.54650	1.00630	-0.51115	0.16418	-0.31787	0.27562
.02	1.09620	0.98977	-0.48530	0.14498	-0.25435	0.19915
.03	0.23928	0.99515	-0.49950	0.15565	-0.28857	0.23943
.04	-0.23948	0.99772	-0.50794	0.16165	-0.30646	0.25908
.05	0.70811	0.99188	-0.50226	0.15792	-0.29488	0.24608
.06	0.36469	0.99329	-0.50845	0.16207	-0.30594	0.25657
.07	0.17636	0.99546	-0.51698	0.16867	-0.32725	0.28161
.08	0.62843	0.99167	-0.51309	0.16524	-0.31420	0.26475
.09	0.41140	0.99354	-0.52158	0.17208	-0.33695	0.29217
.10	0.47085	0.99198	-0.52163	0.17131	-0.33137	0.28264
.11	0.68979	0.99116	-0.52589	0.17555	-0.34744	0.30429
.12	0.53544	0.99173	-0.53101	0.17938	-0.35920	0.31765
.13	0.38892	0.99173	-0.53462	0.18182	-0.36561	0.32373
.14	0.76975	0.98852	-0.53226	0.17971	-0.35696	0.31186
.15	0.60634	0.98885	-0.53636	0.18229	-0.36288	0.31623
.16	0.40649	0.98999	-0.54245	0.18673	-0.37606	0.33043
.17	0.73099	0.98713	-0.54081	0.18557	-0.37211	0.32587
.18	0.46417	0.98875	-0.54820	0.19109	-0.38883	0.34412
.19	0.90075	0.98061	-0.52634	0.16874	-0.30036	0.22473

TABLE 2 (cont'd)

1-3:2C COEFFICIENTS

R	A_0 x10^3	A_1	A_2	A_3	A_4 x10^1	A_5 x10^2
.20	0.71140	0.98703	-0.55346	0.19530	-0.40158	0.35808
.21	0.64893	0.98700	-0.5575	0.19879	-0.41317	0.37222
.22	0.56702	0.98762	-0.56291	0.20258	-0.42439	0.38422
.23	0.71515	0.98611	-0.56423	0.20381	-0.42850	0.38931
.24	0.61725	0.98692	-0.56969	0.20786	-0.44087	0.40315
.25	0.57119	0.98653	-0.57241	0.20960	-0.44493	0.40637
.26	0.79180	0.98547	-0.57511	0.21210	-0.45381	0.41785
.27	0.69607	0.98591	-0.58041	0.21654	-0.46901	0.43663
.28	0.62722	0.98553	-0.58350	0.21893	-0.47655	0.44549
.29	0.79437	0.98544	-0.58911	0.22410	-0.49504	0.46888
.30	0.67158	0.98625	-0.59584	0.23115	-0.52608	0.51644
.31	0.74865	0.98380	-0.59141	0.22464	-0.49281	0.46289
.32	0.92219	0.98226	-0.59256	0.22572	-0.49644	0.46752
.33	0.82224	0.98264	-0.59730	0.22942	-0.50820	0.48114
.34	0.88180	0.98243	-0.60118	0.23262	-0.51887	0.49414
.35	0.10262	0.98083	-0.60162	0.23267	-0.51751	0.49105
.36	0.99038	0.98036	-0.60424	0.23460	-0.52330	0.49765
.37	0.91970	0.98036	-0.60770	0.23702	-0.53023	0.50506
.38	1.08830	0.97884	-0.60875	0.23792	-0.53284	0.50798
.39	1.11470	0.97855	-0.61207	0.24047	-0.54058	0.51644

TABLE 2 (cont'd)

1-3:2C COEFFICIENTS

R	A_0 x10³	A_1	A_2	A_3	A_4 x10¹	A_5 x10²
.40	1.17740	0.97771	-0.61372	0.24125	-0.54130	0.51566
.41	1.21820	0.97688	-0.61606	0.24330	-0.54828	0.52441
.42	1.13560	0.97705	-0.61993	0.24606	-0.55618	0.53262
.43	1.17390	0.97657	-0.62278	0.24826	-0.56290	0.54017
.44	1.25770	0.97508	-0.62296	0.24798	-0.56028	0.53543
.45	1.30410	0.97449	-0.62533	0.24946	-0.56329	0.53694
.46	1.20250	0.97496	-0.63022	0.25345	-0.57673	0.55343
.47	1.39770	0.97310	-0.63045	0.25368	-0.57704	0.55347
.48	1.45510	0.97221	-0.63176	0.25421	-0.57681	0.55130
.49	2.63160	0.94641	-0.77159	0.35867	-0.89176	0.90142
.50	1.29400	0.98102	-0.66161	0.27938	-0.66456	0.66069
.51	1.46920	0.97076	-0.63972	0.26012	-0.59397	0.56948
.52	1.50140	0.97079	-0.64398	0.26376	-0.60667	0.58553
.53	1.54710	0.96986	-0.64538	0.26447	-0.60718	0.58416
.54	1.56460	0.96971	-0.64922	0.26778	-0.61866	0.59853
.55	1.46430	0.97013	-0.65396	0.27164	-0.63151	0.61405
.56	1.65860	0.96803	-0.65281	0.27025	-0.62482	0.60403
.57	1.68550	0.96775	-0.65632	0.27325	-0.63507	0.61674
.58	1.63300	0.96751	-0.65932	0.27553	-0.64200	0.62444

TABLE 2 (cont'd)

1-3:2C COEFFICIENTS

R	A_0 x10³	A_1	A_2	A_3	A_4 x10¹	A_5 x10²
.59	1.77010	0.96636	-0.66068	0.27651	-0.64435	0.62630
.60	1.85570	0.96466	-0.66124	0.27649	-0.64120	0.61866
.61	1.76480	0.96543	-0.66567	0.28014	-0.65464	0.63700
.62	1.74620	0.96535	-0.66936	0.28319	-0.66476	0.64910
.63	1.80880	0.96455	-0.67135	0.28478	-0.66975	0.65496
.64	1.72290	0.96471	-0.67523	0.28786	-0.67963	0.66649
.65	1.85600	0.96393	-0.67750	0.28960	-0.68455	0.67144
.66	1.81240	0.96311	-0.67878	0.29033	-0.68584	0.67199
.67	1.82510	0.96293	-0.68231	0.29330	-0.69586	0.68424
.68	1.87880	0.96231	-0.68466	0.29518	-0.70174	0.69110
.69	1.88370	0.96227	-0.68863	0.29860	-0.71344	0.70539
.70	1.96850	0.96109	-0.68955	0.29922	-0.71473	0.70618
.71	1.98700	0.96047	-0.69161	0.30065	-0.71834	0.70940
.72	2.00210	0.96018	-0.69436	0.30265	-0.72405	0.71533
.73	1.95690	0.95987	-0.69723	0.30501	-0.73180	0.72461
.74	2.08210	0.95882	-0.69884	0.30640	-0.73620	0.72971
.75	2.00140	0.95886	-0.70230	0.30908	-0.74462	0.73945
.76	2.07320	0.95796	-0.70372	0.31004	-0.74680	0.74106
.77	2.14030	0.95700	-0.70520	0.31115	-0.74972	0.74373
.78	2.10090	0.95754	-0.71036	0.31567	-0.76577	0.76409

TABLE 2 (cont'd)

1-3:2C COEFFICIENTS

R	A_0 x10^3	A_1	A_2	A_3	A_4 x10^1	A_5 x10^2
.79	2.17470	0.95647	-0.71149	0.31638	-0.76691	0.76409
.80	2.22410	0.95612	-0.71500	0.31955	-0.77811	0.77833
.81	2.29050	0.95454	-0.71442	0.31881	-0.77439	0.77261
.82	2.17700	0.95505	-0.71882	0.32224	-0.78513	0.78474
.83	2.36540	0.95352	-0.71902	0.32235	-0.78494	0.78416
.84	2.25330	0.95366	-0.72313	0.32610	-0.79853	0.80160
.85	2.31490	0.95324	-0.72564	0.32788	-0.80318	0.80576
.86	2.33970	0.95252	-0.72764	0.32960	-0.80894	0.81289
.87	2.42210	0.95135	-0.72849	0.33023	-0.81036	0.81397
.88	2.46190	0.95078	-0.73069	0.33197	-0.81554	0.81943
.89	2.50950	0.94972	-0.73156	0.33251	-0.81640	0.81960
.90	2.55370	0.94902	-0.73371	0.33440	-0.82268	0.82712
.91	2.48350	0.94906	-0.73726	0.33739	-0.83276	0.83937
.92	2.62780	0.94803	-0.73861	0.33847	-0.83576	0.84233
.93	2.53660	0.94826	-0.74226	0.34133	-0.84485	0.85293
.94	2.62050	0.94709	-0.74336	0.34240	-0.84835	0.85688
.95	2.63490	0.94638	-0.74494	0.34352	-0.85120	0.85945
.96	2.70740	0.94547	-0.74656	0.34495	-0.85586	0.86503
.97	2.72560	0.94513	-0.74924	0.34722	-0.86335	0.87396

TABLE 2 (cont'd)

1-3:2C COEFFICIENTS

R	A_0 x10^3	A_1	A_2	A_3	A_4 x10^1	A_5 x10^2
.98	2.75280	0.94473	-0.75178	0.34931	-0.86995	0.88147
.99	2.79380	0.94417	-0.75421	0.35148	-0.87737	0.89055
1.0	2.77220	0.94336	-0.75536	0.35222	-0.87876	0.89109

TABLE 3

1-3:2P COEFFICIENTS

R	A_0 x10³	A_1	A_2	A_3	A_4 x10¹	A_5 x10²
.01	-1.53750	1.00620	-0.51077	0.16346	-0.31456	0.27066
.02	1.11580	0.98955	-0.48504	0.14443	-0.25245	0.19708
.03	0.26058	0.99459	-0.49839	0.15419	-0.28333	0.23299
.04	-0.14339	0.99665	-0.50600	0.15936	-0.29827	0.24901
.05	0.71311	0.99151	-0.50132	0.15581	-0.28564	0.23286
.06	-0.76150	0.99957	-0.51965	0.16930	-0.32915	0.28450
.07	0.00791	0.99690	-0.52063	0.17043	-0.33265	0.28805
.08	0.53649	0.99321	-0.51784	0.16820	-0.32501	0.27915
.09	0.45711	0.99282	-0.52061	0.16963	-0.32731	0.27957
.10	0.32341	0.99298	-0.52472	0.17231	-0.33459	0.28716
.11	0.63371	0.99174	-0.52796	0.17532	-0.34560	0.30175
.12	0.44811	0.99423	-0.53763	0.18264	-0.36872	0.32827
.13	0.40379	0.99141	-0.53466	0.17935	-0.35466	0.30842
.14	0.80291	0.98808	-0.53222	0.17711	-0.34561	0.29604
.15	0.55870	0.98956	-0.53929	0.18226	-0.36119	0.31312
.16	0.39008	0.99034	-0.54496	0.18651	-0.37453	0.32851
.17	0.61833	0.98887	-0.54674	0.18791	-0.37882	0.33344
.18	0.37408	0.99011	-0.55323	0.19272	-0.39372	0.35028
.19	0.44322	0.98934	-0.55579	0.19446	-0.39841	0.35493

TABLE 3 (cont'd)

1-3:2P COEFFICIENTS

R	A_0 x10^3	A_1	A_2	A_3	A_4 x10^1	A_5 x10^2
.20	0.65879	0.98800	-0.55782	0.19608	-0.40326	0.36012
.21	0.62161	0.98815	-0.56240	0.19955	-0.41423	0.37300
.22	0.52051	0.98852	-0.56700	0.20282	-0.42390	0.38353
.23	0.70119	0.98663	-0.56742	0.20307	-0.42417	0.38350
.24	0.55381	0.98769	-0.57404	0.20845	-0.44276	0.40697
.25	0.50155	0.98755	-0.57772	0.21100	-0.44987	0.41422
.26	0.75200	0.98606	-0.57918	0.21217	-0.45355	0.41867
.27	0.67777	0.98627	-0.58389	0.21587	-0.46579	0.43366
.28	0.64963	0.98566	-0.58598	0.21689	-0.46711	0.43334
.29	0.89133	0.98422	-0.58768	0.21821	-0.47081	0.43704
.30	0.83692	0.98406	-0.59124	0.22081	-0.47884	0.44648
.31	0.72206	0.98469	-0.59633	0.22452	-0.49011	0.45909
.32	0.88562	0.98325	-0.59765	0.22550	-0.49292	0.46229
.33	0.81793	0.98323	-0.60125	0.22803	-0.50028	0.47027
.34	0.93507	0.98177	-0.60195	0.22811	-0.49868	0.46646
.35	1.02360	0.98140	-0.60574	0.23118	-0.50876	0.47864
.36	1.00020	0.98080	-0.60808	0.23263	-0.51221	0.48161
.37	0.93995	0.98064	-0.61149	0.23506	-0.51930	0.48934
.38	1.03810	0.98003	-0.61416	0.23693	-0.52453	0.49483
.39	1.12760	0.97855	-0.61487	0.23720	-0.52419	0.49327

TABLE 3 (cont'd)

1-3:2P COEFFICIENTS

R	A_0 x10^3	A_1	A_2	A_3	A_4 x10^1	A_5 x10^2
.40	1.11050	0.97929	-0.62069	0.24148	-0.53686	0.50692
.41	1.21970	0.97718	-0.61982	0.24070	-0.53399	0.50348
.42	0.13290	0.97741	-0.62437	0.24433	-0.54598	0.51805
.43	1.23020	0.97606	-0.62536	0.24480	-0.54612	0.51682
.44	1.19100	0.97648	-0.63029	0.24865	-0.55858	0.53160
.45	1.25920	0.97568	-0.63244	0.25010	-0.56229	0.53512
.46	1.16020	0.97599	-0.63644	0.25277	-0.56938	0.54187
.47	1.29110	0.97531	-0.63939	0.25522	-0.57754	0.55186
.48	1.22590	0.97482	-0.64198	0.25705	-0.58269	0.55729
.49	1.26780	0.97436	-0.64490	0.25916	-0.58859	0.56316
.50	0.88037	0.98069	-0.66207	0.27237	-0.63149	0.61370
.51	1.35630	0.97298	-0.64974	0.26273	-0.59899	0.57445
.52	1.39390	0.97251	-0.65249	0.26428	-0.60437	0.57989
.53	1.56550	0.97061	-0.65450	0.26750	-0.61662	0.59746
.54	1.45620	0.97180	-0.65881	0.26958	-0.61975	0.59779
.55	1.39610	0.97129	-0.66116	0.27113	-0.62353	0.60093
.56	1.52440	0.97070	-0.66454	0.27418	-0.63438	C.61487
.57	1.45420	0.97074	-0.66822	0.27690	-0.64250	0.62372
.58	1.51230	0.96951	-0.66891	0.27699	-0.64102	0.62022

TABLE 3 (cont'd)

1-3:2P COEFFICIENTS

R	A_0 x10^3	A_1	A_2	A_3	A_4 x10^1	A_5 x10^2
.59	1.54580	0.96909	-0.67179	0.27925	-0.64826	0.62890
.60	1.29670	0.97304	-0.68701	0.29329	-0.70007	0.69564
.61	1.58550	0.96830	-0.67761	0.28351	-0.66030	0.64118
.62	1.64580	0.96764	-0.68023	0.28565	-0.66713	0.64908
.63	1.69090	0.96705	-0.68272	0.28752	-0.67259	0.65501
.64	1.58780	0.96755	-0.68781	0.29187	-0.68781	0.67424
.65	1.78040	0.96555	-0.68713	0.29094	-0.68275	0.66610
.66	1.69280	0.96574	-0.69099	0.29381	-0.69143	0.67579
.67	1.72500	0.96543	-0.69417	0.29643	-0.70005	0.68609
.68	1.76970	0.96457	-0.69611	0.29800	-0.70500	0.69188
.69	1.78510	0.96428	-0.69931	0.30047	-0.71240	0.69995
.70	1.87300	0.96317	-0.70067	0.30149	-0.71503	0.70233
.71	1.89020	0.96250	-0.70246	0.30255	-0.71702	0.70328
.72	1.92640	0.96210	-0.70572	0.30541	-0.72691	0.71559
.73	1.85070	0.96211	-0.70916	0.30803	-0.73490	0.72452
.74	2.01410	0.96106	-0.71106	0.30976	-0.74079	0.73176
.75	1.91680	0.96092	-0.71407	0.31201	-0.74750	0.73913
.76	1.98610	0.95993	-0.71534	0.31279	-0.74887	0.73960
.77	2.00770	0.95981	-0.71914	0.31610	-0.76034	0.75395
.78	2.05850	0.95917	-0.72159	0.31810	-0.76647	0.76062

TABLE 3 (cont'd)

1-3:2P COEFFICIENTS

R	A_0 x10³	A_1	A_2	A_3	A_4 x10¹	A_5 x10²
.79	2.14880	0.95793	-0.72227	0.31840	-0.76612	0.75884
.80	2.14320	0.95825	-0.72763	0.32334	-0.78412	0.78210
.81	2.15380	0.95746	-0.72853	0.32352	-0.78281	0.77874
.82	2.11630	0.95704	-0.73116	0.32558	-0.78896	0.78526
.83	2.27160	0.95557	-0.73167	0.32598	-0.78968	0.78567
.84	2.19230	0.95559	-0.73515	0.32875	-0.79851	0.79585
.85	2.25020	0.95480	-0.73710	0.33036	-0.80345	0.80142
.86	2.28490	0.95444	-0.74010	0.33290	-0.81186	0.81156
.87	2.31220	0.95411	-0.74307	0.33537	-0.81990	0.82103
.88	2.39540	0.95293	-0.74399	0.33602	-0.82109	0.82142
.89	2.36970	0.95284	-0.74728	0.33869	-0.82970	0.83153
.90	2.47650	0.95131	-0.74725	0.33850	-0.82797	0.82834
.91	2.39460	0.95151	-0.75112	0.34167	-0.83821	0.84022
.92	2.53450	0.95060	-0.75308	0.34342	-0.84404	0.84728
.93	2.42020	0.95072	-0.75683	0.34662	-0.85500	0.86088
.94	2.49640	0.94967	-0.75778	0.34718	-0.85550	0.85991
.95	2.55890	0.94885	-0.75948	0.34854	-0.85936	0.86383
.96	2.54640	0.94891	-0.76309	0.35153	-0.86925	0.87578
.97	2.63240	0.94781	-0.76451	0.35284	-0.87333	0.88020

TABLE 3 (cont'd)

1-3:2P COEFFICIENTS

R	A_0 x10^3	A_1	A_2	A_3	A_4 x10^1	A_5 x10^2
.98	2.68180	0.94678	-0.76580	0.35409	-0.87773	0.88592
.99	2.69800	0.94670	-0.76845	0.35619	-0.88404	0.89273
1.0	2.63160	0.94641	-0.77159	0.35867	-0.89176	0.90142

VI. CONCLUSIONS AND RECOMMENDATIONS

A. CONCLUSIONS

Effectiveness values, though not analytically derived, can be determined for the 1-3:2C and 1-3:2P heat exchangers by utilizing a 5th order polynomial approximation. Sufficient data now exists to cover a complete range of capacity rate ratios for values of N_{tu} from 0.0 to 3.25. From the knowledge of a particular three tube pass heat exchanger arrangement and dimensions, fluid flow rates and temperature extremes, N_{tu} and R values may be computed. Then the effectiveness may be determined using the appropriate coefficients from the tables provided herein. After the effectiveness is obtained, there are two choices.

1) From a knowledge of q_{max} (see Section III), it is a simple matter to determine the actual heat transfer rate from the postulation that $q = \epsilon q_{max}$.

2) Both fluid outlet temperatures may be computed since $q = \omega_c C_{ph}(t_2 - t_1)$.

It is also apparent from the work done here that the 1-3:2C exchanger effectiveness outperforms that of the 1-2, 1-4 and its counter part, the 1-3:2P exchanger, as N_{tu} increases. This is true for any and all values of R. Therefore, because it is possible to determine the effectiveness of a 1-3 exchanger which has a higher effectiveness (the 1-3:2C arrangement) than that of the 1-2, 1-4 and

1-3:2P exchanger particularly at high N_{tu} levels, the 1-3 exchanger can now be given full consideration in heat exchanger design. This would be especially helpful where three tube passes could alleviate a configuration problem.

B. RECOMMENDATIONS

The following recommendations are provided for possible follow-on projects of a similar nature.

- Continue study of the ε - N_{tu} method for the 1-5 heat exchangers.
- Investigate the possibility of using a linear approximation from the 1-3 data to find a periodicity of R values at which to develop a 1-5 data base.
- Develop interactive software to be used on a microcomputer that contains both the 1-3 and 1-5 polynomial approximation coefficients. The procedure of entering the values for N_{tu}, R, and the type of heat exchanger when asked to do so in a menu-driven fashion so that effectiveness values can be readily obtained is self-evident.
- After results of 1-5 exchanger analyses are complete, investigate the effectiveness for both the 1-3 and 1-5 heat exchangers experimentally to confirm or refute the theoretical work done here.

APPENDIX A

THERMAL ANALYZER TVSSI COMPUTER PROGRAM

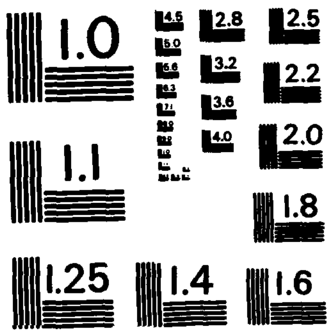
```
//OHARE JOB (2323,0267),'TVSSI',CLASS=C
//*MAIN  ORG=NPGVM1.2323P,SYSTEM=SY2
//  EXEC FORTVCG
//FORT.SYSPRINT DD DUMMY
//FORT.SYSIN DD *
C<<<<<<<<<<<<<<ZU0022PU     Z075     00511 <<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<
C  TVSSI IS A MODIFICATION OF TVSS2 TO HANDLE HEAT INPUT IN WATTS AND
C  TEMPERATURES IN DEGREES CENTIGRADE. CONDUCTANCES WILL BE IN WATTS/DEG
C                              PROGRAM TVSSI
C                  THIS PROGRAM WILL HANDLE UP TO 315 NODES
C ***********************************************************************
C *
C *  UPDATED APRIL 1985
C *  NOTE: IF DAMPING VALUES AND INITIAL TEMPS ARE THE SAME FOR ALL NODE
C *  ENTER 1 DAMPING VALUE COLS 25-32 CARD 3
C *  ENTER 1 INITIAL TEMP COLS 33-40 CARD 3
C *  CALL FOR DATA SET 6 ON CARD 2
C *
C ***********************************************************************
C
C ***********************************************************************
C *
C *      JCL NOTE: LISTING OF INPUT IS WRITTEN ON UNIT 3 BEFORE NORMAL
C *      OUTPUT ON UNIT 8. TO DELETE INPUT LIST, INSERT DUMMY
C *      DATA DEFINITION STATEMENT
C *
C ***********************************************************************
C
          IMPLICIT REAL*4 (A-H,O-Z)
      DIMENSION  TCO(5), CONTMP(50), ITAG(100), TCOEF(90),  COEF(100),
     1  H(316), A(316), TOLD(315), TOLD1(315), TOLD2(315), BETA(315),
     2  TITLE(20), INPTAG(10),EX(4), TMPHT(90), TMPHTV(5), NOCONT(6),
     3  HTR(42), BTUCRV(5), TMPCRV(5), TIMCO(5)
      CHARACTER *4 HEAD,HEAD1
      DATA HEAD,HEAD1/'BETA','TOLD'/
 1000 READ(4,500,END=999) TITLE
      IER = 0
      REWIND 1
      REWIND 2
      READ(4,501) N,NCT,NOHTRS,NOEXP,NOCASE,NTCOEF,NODCFH,NTMPHT
      NP1 = N + 1
      READ (4,501)  NTAG8,NTAG9,NTAG11
```

```
      READ(4,501) NONODS,NOCT,NOHTR,INPTAG
      READ(4,505) ERR,ALPHA,MNITS,BETA(1),TOLD(1)
      MAXNIT = IABS(MNITS)
      NIT = 0
      IF(BETA(1).NE.0.) GO TO 1010
      READ(4,502)  (BETA(I),I=1,N)
      GO TO 1030
 1010 DO 1020 I=2,N
 1020 BETA(I) = BETA(I-1)
C
 1030 DO 15 I=1,10
      INI = INPTAG(I)
      IF (INI)  17,17,6
    6 GO TO (1,2,3,7,10,12,14,808,809,810,811,810,810,810) , INI
    1 NTCIN = 18*NTCOEF
      READ(4, 502) (TCOEF(K), K=1,NTCIN)
      GO TO 15
    2 READ(4,  502)  (CONTMP(K), K=1,NCT)
      GO TO 15
    3 READ(4,    501)  NOCONT
      READ(4,    502)  (HTR(K), K=1,36)
      GO TO 15
    7 DO 9 L = 1,N
      ITAG(10) = 0
      READ(4,504)  NT, (ITAG(K),          K=1,9)
      READ(4,502) (COEF(M) , M=1, 9)
      IF (NT-9 )   8  , 8  ,710
  710 DO 715 K=10,NT,9
      ITAG (K+9) = 0
      KE = K+8
      READ(4,503)  (ITAG(M),          M=K ,KE)
  715 READ(4,502) (COEF(M) , M=K ,KE)
    8 WRITE (1) ITAG,COEF
      WRITE(3,556)(ITAG(KK),KK=1,30)
    9 CONTINUE
      ENDFILE 1
      REWIND 1
      GO TO 15
   10 READ(4, 502) EX
      GO TO 15
   12 IF (TOLD(1).NE.0.) GO TO 1201
      READ(4,502) (TOLD(K), K=1,N)
      GO TO 15
 1201 DO 1202 L=2,N
 1202 TOLD(L) = TOLD(L-1)
      GO TO 15
   14 K2 = 18*NTMPHT
      READ(4,502) (TMPHT(K), K=1,K2)
```

```
      GO TO 15
  808 READ(4,502)  BTUCRV
      GO TO 15
  809 READ(4,502)  TMPCRV
      GO TO 15
  810 WRITE (8,533)  INI
      STOP
  811 READ(4,502)  TIMCO
   15 CONTINUE
   17 CALL TVPAGE    (0,TITLE)
      CALL  TVSOUT(N, 1,TOLD,TOLD,TITLE)
C
      WRITE(3,551) TITLE
      WRITE(3,552)N,NCT,NOHTRS,NOEXP,NOCASE,NTCOEF,NODCFH,NTMPHT,NONODS,
     1 NOCT,NOHTR,MNITS
      WRITE(3,553)INPTAG,ERR,ALPHA
      WRITE(3,554)N,HEAD,(BETA(I),I=1,N)
      HEAD=HEAD1
      WRITE(3,554)N,HEAD,(TOLD(I),I=1,N)
      WRITE(3,555)(CONTMP(I),I=1,NCT)
   20 IF(NOHTRS) 24,24,21
   21 CALL TVSHTR(NOHTRS,HTR,CASBTU,NOCONT,TOLD,NODCFH)
   24 DO 107 NOD=1,N
      DO 25 I=1,NP1
   25 A(I) = 0.
      READ (1) ITAG, COEF
      DO 100  IWD =1,99
      IF (ITAG(IWD) .EQ. 0)  GO TO 105
      NODEI = ITAG(IWD) / 10
      METHI = MOD (ITAG(IWD) , 10)
      NTTH = NODEI - NONODS
      IF ( NTTH.LE.0 )  GO TO 55
      IF ( NODEI.EQ.999 )  GO TO 50
      NNODE = NTTH
      NTTH = NTTH - NOCT
      IF ( NTTH.LE.0 )  GO TO 60
      NNODE = NTTH
      NTTH = NTTH - NOHTR
      IF ( NTTH.LE.0 )  GO TO 49
      IF ( NTTH.GT.5 )  GO TO 820
      IF ( NTTH.LE.NTAG8 )  GO TO 815
      WRITE (9,610)  NOD,NODEI,NTAG8
      IER = 1
      GO TO 100
  815 A(NP1) = A(NP1) - BTUCRV(NTTH)*COEF(IWD)
      GO TO 100
  820 NTTH = NTTH - 5
      IF ( NTTH.GT.5 )  GO TO 830
```

AD-A159 706 THE EFFECTIVENESS OF HEAT EXCHANGERS WITH ONE SHELL PASS AND THREE TUBE PASSES(U) NAVAL POSTGRADUATE SCHOOL MONTEREY CA M S O'HARE JUN 85 2/3

UNCLASSIFIED F/G 20/13 NL


1.0
1.1
1.25
1.4
1.6
1.8
2.0
2.2
2.5
2.8
3.2
3.6
4.0
4.5

```
      IF ( NTTH.LE.NTAG9 ) GO TO 825
      WRITE (9,620)  NOD,NODEI,NTAG9
      IER = 1
      GO TO 100
  825 T1 = TMPCRV(NTTH)
      GO TO 65
  830 NTTH = NTTH - 5
      IF ( NTTH.LE.5 )  GO TO 48
      WRITE ( 9,570 ) NOD,NODEI
      IER = 1
      GO TO 100
   48 IF ( NTTH.LE.NTMPHT ) GO TO 480
      WRITE (9,630)  NOD,NODEI,NTMPHT
      IER = 1
      GO TO 100
  480 WCOEF = NTTH + 1100
      CALL TVFTMP (WCOEF, TOLD(NOD), TMPHT)
      A(NP1) = A(NP1)-WCOEF*COEF(IWD)
      TMPHTV(NTTH) = WCOEF
      GO TO 100
   49 IF ( NNODE.LE.NOHTRS )  GO TO 490
      WRITE (9,600)  NOD,NODEI,NOHTRS
      IER = 1
      GO TO 100
  490 LOCH = 36 + NNODE
      A(NP1) = A(NP1) - HTR(LOCH) * COEF(IWD)
      GO TO 100
   50 A(NP1) = A(NP1) - COEF(IWD)
      GO TO 100
   55 T1 = TOLD(NODEI)
      GO TO 65
   60 IF ( NNODE.LE.NCT ) GO TO 61
      WRITE (9,590)  NOD,NODEI,NCT
      IER = 1
      GO TO 100
   61 T1 = CONTMP(NNODE)
   65 T2 = TOLD(NOD)
      WCOEF = COEF(IWD)
C * * *
C         ASSUMING THE USER HAS RUN PROGRAM CHECK AND FIXED ANY WCOEFS
C         THAT MIGHT HAVE BEEN IN ERROR. TVSSI, THEREFORE, WILL NOT
C         CHECK AGAIN FOR A WCOEF > 100.  NO WARNING IS PRINTED.
C * * *
      IF ( WCOEF.LE.1000. )  GO TO 73
      IF ( WCOEF.LE.1005. )  GO TO 850
      IF ( WCOEF.LE.1100. )  GO TO 73
      IF ( WCOEF.GT.1105. )  GO TO 73
      ATEMP = (T1+T2) / 2.
```

```
      LC = WCOEF - 1099.99
      IF ( LC.LE.NTCOEF )   GO TO 840
      WRITE (9,640)  NOD,NODEI,WCOEF,NTCOEF
      IER = 1
      GO TO 100
  840 CALL TVFTMP (WCOEF, ATEMP, TCOEF)
      TCO(LC) = WCOEF
      GO TO 73
  850 LC = WCOEF - 999.99
      IF ( LC.LE.NTAG11 )  GO TO 855
      WRITE (9,650)  NOD,NODEI,WCOEF,NTAG11
      IER = 1
      GO TO 100
  855 WCOEF = TIMCO(LC)
C * * *
C         THE FOLLOWING COMPUTED GO TO WILL FALL THROUGH TO THE NEXT
C         STATEMENT FOR METHODS 6,7,8,& 9
C * * *
   73 GO TO (80,730,730,80,80), METHI
  730 TD = T1 - T2
      IF ( TD.NE.0. )  GO TO 731
      WCOEF = 0.
      GO TO 80
  731 IF ( ABS(TD).LT.1E-05 )  TD = SIGN( 1E-05,TD )
      ABSTD = ABS(TD)
      GO TO (80,74,75,80,80), METHI
      LC = METHI - 5
      IF ( LC.LE.NOEXP )  GO TO 77
      WRITE (9,660) NOD,NODEI,METHI,NOEXP
      IER = 1
      GO TO 100
   77 T3 = EX (LC)
C * *
C         THE FOLLOWING STATEMENT EQUATES TO
C         WCOEF = WCOEF * ((ABSTD ** T3)/ABSTD)
C * *
      WCOEF = WCOEF * ABSTD ** (T3-1.)
      GO TO 80
C * *
C        THE FOLLOWING STATEMENT EQUATES TO
C        WCOEF = WCOEF * (ABSTD ** 1.25)/ABSTD
C * *
   74 WCOEF = WCOEF * ABSTD ** .25
      GO TO 80
   75 T3 = (T1+273.) / 100.
      T4 = (T2+273.) / 100.
      WCOEF = WCOEF * ((T3**4 - T4**4) / TD )
   80 IF(NODEI - NONODS) 85,85,90
```

```
   85 A(NODEI) = A(NODEI) + WCOEF
      GO TO 95
   90 A(NP1) = A(NP1) - WCOEF*T1
   95 A(NOD) = A(NOD) - WCOEF
  100 CONTINUE
  105 IF(NOD-NOCASE)  107,106,107
  106 A(NP1) = A(NP1) - CASBTU
  107 WRITE (2) (A(K),K=1,NP1)
C
C * * *
C         IF IER.NE.0 AT THIS POINT, IT MEANS THERE ARE ERRORS IN THE
C         INPUT; TVSSI WILL SKIP THE CALCULATIONS AND RETURN TO
C         STATEMENT  1000  FOR ANOTHER PROBLEM.
C * * *
C
      IF ( IER.EQ.0 )  GO TO 110
      WRITE (9,670)
      GO TO 1000
C
  110 ENDFILE 2
      REWIND 1
      REWIND 2
      CALL CHOST (N, NP1, H)
      ERRTAG = 0.
      NIT = NIT + 1
      IF (NIT .LT. 3)  GO TO 120
      CALL CBETA (N, ALPHA, BETA, TOLD, TOLD1, TOLD2, ERR)
  120 DO 130 I=1,N
      IF(NIT .EQ. 1) TOLD1(I) = 0.0
      TOLD2(I) = TOLD1(I)
      TOLD1(I) = TOLD(I)
      TOLD(I) = TOLD(I) + BETA(I) * (  H(I)  -TOLD(I))
      A(I)    = TOLD(I) - TOLD1(I)
  130 IF (ABS(A(I)) .GT. ERR)  ERRTAG = 1.
      CALL TVSOUT (N, 2, TOLD, A, TITLE)
      IF (NOHTRS)  140,140,135
  135 LOCH = 36 + NOHTRS
      CALL  TVPAGE   (3,TITLE)
      WRITE (8,525)  (HTR(K), K=37,LOCH)
      WRITE (8,526)  CASBTU
  140 IF(NTCOEF) 150,150,145
  145 CALL  TVPAGE   (2,TITLE)
      WRITE (8,528)  (TCO  (K), K= 1,NTCOEF)
  150 IF (NTMPHT)  160,160,155
  155 CALL TVPAGE (2,TITLE)
      WRITE (8,529) (TMPHTV(K), K=1,NTMPHT)
  160 IF (NIT .GE. MAXNIT .AND. MAXNIT .GT. 0)  GO TO 175
      IF ( ERRTAG.GT.0. )  GO TO 20
```

```
      WRITE (10,500) TITLE
      WRITE (10,501) N
      WRITE (10,680) (TOLD(I),I=1,N)
      GO TO 1000
  175 DO 180 I=1,N
  180 IF (BETA(I) .LT. .0001)  BETA(I) = .0001
      WRITE (8,530)  MAXNIT, (BETA(I),I=1,N)
      GO TO 1000
C
  500 FORMAT (20A4)
  501 FORMAT (18I4)
  502 FORMAT (9G8.0)
  503 FORMAT (9I8)
  504 FORMAT (I2,I6,8I8)
  505 FORMAT (2G8.0, I8, 2G8.0)
  525 FORMAT(/12H HTRS WATTS   , 6G12.5)
  526 FORMAT (12H CASE WATTS   ,  G12.5)
  528 FORMAT (/12H TEMP COEFS   , 6G12.5)
  529 FORMAT ( /12H TEMP HEAT     , 6G12.5),
  530 FORMAT (,'0 OVER ',I5, ' ITERATIONS ' // '  BETAS' // (20F6.4) )
  533 FORMAT (' SET ',I5, ' IS NOT ACCEPTABLE.  FIX DECK AND RESUBMIT ' )
  551 FORMAT(1H ,20A4)
  552 FORMAT(//,' N   NCT NOHTRS NOEXP NOCASE NTCOEF NODCFH NTMPHT NONO
     1DS  NOCT  NOHTR  MNITS',/12I6,/)
  553 FORMAT(/,'          INPTAG 1 - 10
     1          ERR ,ALPHA',/,10I7,2(1X,F11.4))
  554 FORMAT(/,' ',I4,' ',A4,' 'S',(/,1X,16F8.4))
  555 FORMAT('    CONTMP(I),I=1,NCT)',(/,1X,16G8.2))
  556 FORMAT(15I8)
  570 FORMAT ( //1X,'* IMPOSSIBLE NODE NUMBER - YOU SPECIFIED AN ',
     1    'INTERACTION FROM NODE',I5,'   TO A NODE - - ',I5/)
  590 FORMAT (/1X,'* * * INVALID NODE - - YOU SPECIFIED AN ',
     1 'INTERACTION FROM NODE',I5,5X,'TO A NODE',I5,/15X,
     2 'BUT THERE ARE ONLY',I4,'    CONSTANT TEMPERATURES')
  600 FORMAT (/1X,'* * * INVALID NODE - - YOU SPECIFIED AN ',
     1 'INTERACTION FROM NODE',I5,5X,'TO A NODE',I5,/15X,
     2 'BUT THERE ARE ONLY ',I3,'    HEATERS')
  610 FORMAT (/1X,'* * * INVALID NODE - - YOU SPECIFIED AN ',
     1 'INTERACTION FROM NODE',I5,5X,'TO A NODE',I5,/15X,
     2 'BUT THERE ARE ONLY ',I3,'    WATT CURVES (TAG 8)')
  620 FORMAT (/1X,'* * * INVALID NODE - - YOU SPECIFIED AN ',
     1 'INTERACTION FROM NODE',I5,5X,'TO A NODE',I5,/15X,
     2 'BUT THERE ARE ONLY',I4,'    TEMPERATURE CURVES (SET 9)')
  630 FORMAT (/1X,'* * * INVALID NODE - - YOU SPECIFIED AN ',
     1 'INTERACTION FROM NODE',I5,5X,'TO A NODE',I5,/15X,
     2 'BUT THERE ARE ONLY',I4,'    TEMP-DEPENDENT WATT CURVES (SET 7)')
  640 FORMAT (/1X,'* * * INVALID CONDUCTANCE - - NODE',I5,5X,
     1 'TO NODE',I5,5X,'YOU SPECIFIED A CONDUCTANCE OF ',F7.1,/15X,
```

```
      2 'BUT THERE ARE ONLY ',I3,'    TEMP-DEPENDENT ',
      3 'COEFF CURVES (SET 1)')
  650 FORMAT (/1X,'* * * INVALID CONDUCTANCE - - NODE',I5,5X,
      1 'TO NODE',I5,5X,'YOU SPECIFIED A CONDUCTANCE OF ',F7.1,/15X,
      2 'BUT THERE ARE ONLY ',I3,'    TIME COEFFS (SET 11)')
  660 FORMAT (/1X,'* * * INVALID METHOD - - NODE',I5,5X,'TO NODE',
      1 I5,5X,'YOU SPECIFIED A METHOD OF ',I3,/15X,'BUT THERE ARE ONLY ',
      2 I3,'    UNIQUE EXPONENTS (SET 5)')
  670 FORMAT (////1X,'THERE ARE ERRORS IN YOUR INPUT FOR THIS PROBLEM.',
      1 //10X,'TVSSI WILL SKIP ANY FURTHER CALCULATIONS FOR THIS PROBLEM'
      2 ,/////)
  680 FORMAT (9F8.2)
  999 CONTINUE
      STOP
      END
C
      SUBROUTINE CBETA ( N, ALPHA, BETA, TN, TNM1, TNM2, ERR )
          IMPLICIT REAL*4 (A-H,O-Z)
      DIMENSION TN(315) , TNM1(315) , TNM2(315) , BETA(315)
      DO 50 I=1,N
      TMTM1 = TN(I)-TNM1(I)
      T12 = TNM1(I) - TNM2(I)
      IF ( ABS(T12).LT.1E-06 )  T12 = SIGN( 1E-06,T12 )
      GAMMA = TMTM1 /T12
      IF (GAMMA .GT. 0.)  GO TO 10
      IF ( ABS(TMTM1) .LE. ERR )  GO TO 50
      IF (GAMMA .LT. -ALPHA)  BETA(I) = -BETA(I)*ALPHA/GAMMA
      GO TO 50
   10 IF (GAMMA .GT. 1.)  GO TO 50
      BETA(I) = BETA(I) / ALPHA
   50 IF (BETA(I) .GT. 1.)  BETA(I) = 1.
      RETURN
      END
C
      SUBROUTINE  CHOST (N,NP1,EL)
          IMPLICIT REAL*4 (A-H,O-Z)
      DIMENSION EL(316) , LOCS(316) ,SAVE(49770)
      LOCS(1) = 1
      NM1 = N - 1
      I = 0
C
   10 I = I +1
      READ (2) (EL(K),K=1,NP1)
      IP1 = I + 1
      IF ( I .EQ. 1 )  GO TO 50
      DO 45 J=2,I
      LR = LOCS(J-1)
      IF ( EL(J-1) .EQ. 0. )  GO TO 45
```

```
      DO 40 JR=J,NP1
      IF ( SAVE(LR) .EQ. 0. )  GO TO 40
      EL(JR) = EL(JR) - EL(J-1)*SAVE(LR)
   40 LR = LR+1
   45 CONTINUE
   50 CONTINUE
   51 DO 60 K=IP1,NP1
      IF (EL(K) .EQ. 0.)  GO TO 60
      EL(K) = EL(K) / EL(I)
   60 CONTINUE
      IF (I .EQ. N)  GO TO 80
      LS = LOCS(I)
      LOCS(I+1) = LS+NP1-I
      DO 72 K = IP1,NP1
      SAVE(LS) = EL(K)
   72 LS = LS + 1
      GO TO 10
C
   80 REWIND 2
      EL(N) = EL(NP1)
      DO  90 I=1,NM1
      II = NP1 - I
      LF = N - I
      LR = LOCS(LF) + I
      EL(II-1) = SAVE(LR)
      DO 90 K = II,N
      LR = LOCS(LF) + K - II
      IF (SAVE(LR) .EQ. 0.)  GO TO 90
      EL(II-1) = EL(II-1) - SAVE(LR)*EL(K)
   90 CONTINUE
      RETURN
      END
C
      SUBROUTINE  TVFTMP  (CO, T, TCOEF)
          IMPLICIT REAL*4 (A-H,O-Z)
      DIMENSION TCOEF(90)
      NC =  CO - 1099.9
      NB =  18*NC - 17
      IF (T - TCOEF(NB))  2,2,6
    2 CO = TCOEF(NB+1)
      GO TO 60
    6 NE =  NB + 16
      NB = NB+2
      DO 50 K=NB,NE,2
      IF (T-TCOEF(K))   10,20,50
   10 TC = TCOEF(K)
      TCM2 = TCOEF(K-2)
      TCM1 = TCOEF(K-1)
```

```
      TCP1 = TCOEF(K+1)
      TCC = TC - TCM2
      IF ( ABS(TCC).LT.1E-06 )  TCC = SIGN( 1E-06,TCC )
      CO = ( (T - TCM2)/TCC ) * (TCP1 - TCM1) + TCM1
      GO TO 60
   20 CO = TCOEF (K+1)
      GO TO 60
   50 CONTINUE
      CO = TCOEF(NE+1)
   60 RETURN
      END
C
      SUBROUTINE  TVSOUT(N,NA,T1,T2,TITLE)
         IMPLICIT REAL*4 (A-H,O-Z)
      DIMENSION T1(315) ,T2(315) , ID(12) ,TITLE(20)
C
      CALL  TVPAGE (2,TITLE)
      WRITE (8,500)
      NL = NA+2
      DO 50 I=1,N,12
      CALL TVPAGE    (NL,TITLE)
      IF (I+11-N) 5,5,10
    5 N5=12
      GO TO 15
   10 N5=N-I+1
   15 DO 20 K=1,N5
   20 ID(K) = I+K-1
      WRITE (8,501)  (ID(K), K=1,N5)
      N1 = I + N5 - 1
      IF (NA-1)  25,25,30
   25 WRITE (8,504)  (T1(K), K= I,N1)
      GO TO 50
   30 WRITE (8,502)  (T1(K), K= I,N1)
      WRITE (8,503)  (T2(K), K= I,N1)
   50 CONTINUE
      RETURN
C
  500 FORMAT  (1H0)
  501 FORMAT  (/10H NODE NO.   ,12I9)
  502 FORMAT (12H NEW TEMPS    ,12F9.2)
  503 FORMAT (12H NEW - OLD    ,12F9.2)
  504 FORMAT (12H ORIG TEMPS   ,12F9.2)
      END
C
      SUBROUTINE  TVSHTR (NOHTRS,HTR,CASBTU,NOCONT,TOLD,NODCFH)
         IMPLICIT REAL*4 (A-H,O-Z)
      DIMENSION HTR(42) , NOCONT(6) , TOLD(315)
      CASBTU = 0.
```

```
      DO 25 K=1,NOHTRS
      LOK = 36+K
      IF (NOCONT(K))  24,24,43
   43 LOK1 = NOCONT(K)
      TEMP = TOLD(LOK1)
      IF (K-3)  45,45,50
   45 LOK2 = 0
      GO TO 55
   50 LOK2 = 18
   55 IF (NODCFH)   3,3,60
   60 IF (TEMP-HTR(2))  65,65,70
   65 HTR(LOK) = HTR(LOK2+1)
      CASBTU = CASBTU + HTR(LOK2+3)
      GO TO 25
   70 NODCFH = 0
    3 IF (TEMP - HTR(LOK2+4))  5,5,15
    5 HTR(LOK) = HTR(LOK2+9)
      CASBTU = CASBTU + HTR(LOK2+14)
      GO TO 25
   15 DO 16 K1=5,8
      LOK3 = LOK2+K1
      IF (TEMP - HTR(LOK3))  17,17,16
   16 CONTINUE
      HTR(LOK) = HTR(LOK2+13)
      CASBTU = CASBTU + HTR(LOK2+18)
      GO TO 25
   17 HTRL3 = HTR(LOK3) - HTR(LOK3 - 1)
      IF ( ABS(HTRL3).LT.1E-06 )  HTRL3 = SIGN( 1E-06,HTRL3 )
      FRAC = (TEMP - HTR(LOK3-1)) / HTRL3
      HTR(LOK) = (HTR(LOK3+5) - HTR(LOK3+4)) * FRAC + HTR(LOK3+4)
      CASBTU = CASBTU + (HTR(LOK3+10) - HTR(LOK3+9))*FRAC + HTR(LOK3+9)
      GO TO 25
   24 HTR(LOK) = 0.
   25 CONTINUE
      RETURN
      END
C
      SUBROUTINE  TVPAGE   (NL,FNAME)
          IMPLICIT REAL*4 (A-H,O-Z)
      DIMENSION FNAME (20)
      IF (NL)  10,10,20
   10 NPAGE = 0
      LINCNT = 75
   20 LINCNT = LINCNT + NL
      IF (LINCNT - 56)  40,40,30
   30 NPAGE = NPAGE + 1
      WRITE (8,50)  FNAME,NPAGE
      LINCNT = NL
```

```
   40 RETURN
   50 FORMAT (1H1,20X,20A4,8X,9HPAGE NO. ,I3/)
      END
/*
//GO.FT01F001 DD UNIT=SYSDA,SPACE=(CYL,(1,1))
//GO.FT02F001 DD UNIT=SYSDA,SPACE=(CYL,(1,1))
//GO.FT09F001 DD UNIT=SYSDA,SPACE=(CYL,(1,1))
//GO.FT10F001 DD UNIT=SYSDA,SPACE=(CYL,(1,1))
//GO.FT03F001 DD DUMMY
//GO.FT04F001 DD DISP=SHR,DSN=MSS.S2323.TVSSIV
//GO.FT08F001 DD SYSOUT=A,DCB=RECFM=FBA
//
```

APPENDIX B

NTU14 COMPUTER GENERATED INPUT ANALYZER PROGRAM

```
//QHARE    JOB (2323,0267),'NTU14',CLASS=B
//*MAIN    ORG=NPGVM1.2323P
   EXEC    FORTVCL,PARM.LKED='LIST,MAP'
//FORT.SYSIN DD*
C          THIS IS PROGRAM ENTU14
C
C
C          IT GENERATES AN INPUT FILE FOR THERMAL ANALYZER TO VERIFY
C          1-4 EFFECTIVENESS-NTU RELATIONSHIP THAT IS AVAILABLE IN
C          OPEN LITERATURE.
C
      DIMENSION COEF(250,5),KCON(250,5),L1(8),L2(3),L3(6),SET2(2),FL4(4)
C
      CHARACTER *79 TITLE
      CHARACTER *12 FNAME
C
      DATA IOT,IN,IPR,IWR/6,5,4,8/
C
C     OPEN PRINTER OUTPUT FILE
C
      OPEN(IPR,FILE='PRN',STATUS='NEW',FORM='FORMATTED',IOSTAT=ICK)
      IF(ICK.NE.0) WRITE(IOT,920)
  920 FORMAT(' Trouble opening printer output file' )
C
      WRITE(IOT,917)
  917 FORMAT(/' Input the title of this study - 79 columns only. ',
     &' This title will appear',/,' at the top of every printed page',
     &' of output:')
      READ(IN,918) TITLE
  918 FORMAT(A79)
C
      WRITE(IOT,901)
  901 FORMAT(/' INPUT HOT SIDE CAPACITY RATE:')
      READ(IN,902) CHOT
  902 FORMAT(BN,F10.0)
C
      WRITE(IOT,903)
  903 FORMAT(/' INPUT COLD SIDE CAPACITY RATE:')
      READ(IN,902) CCLD
C
      WRITE(IOT,904)
```

```
  904 FORMAT(/' INPUT OVERALL HEAT TRANSFER COEFFICIENT:')
      READ(IN,902) U
C
      WRITE(IOT,905)
  905 FORMAT(/' INPUT TOTAL HEAT TRANSFER SURFACE:')
      READ(IN,902) SURFTO
C
      WRITE(IOT,906)
  906 FORMAT(/' INPUT HOT SIDE INLET TEMPERATURE:')
      READ(IN,902) THOTIN
C
      WRITE(IOT,907)
  907 FORMAT(/' INPUT COLD SIDE INLET TEMPERATURE:')
      READ(IN,902) TCLDIN
C
      VALK1 = CHOT
      VALK2 = CCLD
60
      TINIT = 100.
C
C            FRONT END
C
      L1(1) = 250
      L1(2) = 2
      DO 10  I=3,8
   10 L1(I) = 0
C
      DO 20  I=1,3
   20 L2(I) = 0
C
      L3(1) = 300
      L3(2) = 50
      L3(3) = 6
      L3(4) = 2
      L3(5) = 4
      L3(6) = 6
C
      FL4(1) = .05
      FL4(2) = .66667
      FL4(3) = .8
      FL4(4) = TINIT
      L4 = 12
C
C   CONSTANT TEMPERATURES
C
      SET2(1) = THOTIN
      SET2(2) = TCLDIN
C
```

```
C     READY FOR INPUT SET 4
C
C     NODE 1
C
        KCON(1,1) = 514
        KCON(1,2) = 1504
        KCON(1,3) = 1514
        KCON(1,4) = 2504
        KCON(1,5) = 3015
        COEF(1,5) = VALK1
        DO 50 I = 1,4
   50   COEF(1,I) = VALK3
C
C     NODES 2 TO 50
C
        DO 75 I = 2,50
        J = I + 50
        K = 151 - I
        L = 150 + I
        M = 251 - I
        N = I - 1
        KCON(I,1) = 10*N + 5
        KCON(I,2) = 10*J + 4
        KCON(I,3) = 10*K + 4
        KCON(I,4) = 10*L + 4
        KCON(I,5) = 10*M + 4
        COEF(I,1) = VALK1
        DO 80  II = 2,5
        COEF(I,II) = VALK3
   80   CONTINUE
   75   CONTINUE
C
C     NODE 51
C
        KCON(51,1) = 3025
        KCON(51,2) = 14
        COEF(51,1) = VALK2
        COEF(51,2) = VALK3
C
C     NODES 52 TO 250
C
        DO 120  I = 52,250
        K = I - 1
        IF(I.GT.100) GO TO 122
        J = I - 50
        GO TO 135
  122   IF(I.GT.150) GO TO 124
        L = I - 100
```

```
      M = 2*L - 1
      N = M + 50
      J = I - N
      GO TO 135
  124 IF(I.GT.200) GO TO 126
      J = I - 150
      GO TO 135
  126 CONTINUE
      L = I - 200
      M = 2*L - 1
      N = M + 150
      J = I - N
  135 KCON(I,1) = 10*J + 4
      KCON(I,2) = 10*K + 5
      COEF(I,1) = VALK3
  120 COEF(I,2) = VALK2
C
C   END OF DATA SETUP
C   NOW CREATE INPUT FOR ANALYZER
C
      WRITE(IOT,922)
  922 FORMAT(/,' Enter name of input file, including drive',
     &  ' DESIGNATION: ')
      READ(IN,923) FNAME
  923 FORMAT(A12)
C
C     OPEN INPUT FILE
C
      OPEN(IWR,FILE=FNAME,STATUS='NEW',FORM='FORMATTED',IOSTAT=ICK)
      IF(ICK.GT.0)WRITE(IOT,924)
  924 FORMAT(/,' Trouble opening input file')
C
      WRITE(IOT,925) FNAME
  925 FORMAT(/,' Writing file :',A14)
      WRITE(IWR,919) TITLE
  919 FORMAT(1X,A79)
      WRITE(IWR,908)(L1(I),I=1,8)
  908 FORMAT(9I4)
      WRITE(IWR,908)(L2(I),I=1,3)
      WRITE(IWR,908)(L3(I),I=1,6)
      WRITE(IWR,911) FL4(1),FL4(2),L4,FL4(3),FL4(4)
  911 FORMAT(F8.3,F8.5,I8,2F8.1)
      WRITE(IWR,912) SET2(1),SET2(2)
  912 FORMAT(2F8.0)
C
      DO 200  I = 1,50
      WRITE(IWR,913)(KCON(I,J),J=1,5)
  913 FORMAT(9I8)
```

```
      WRITE(IWR,914)(COEF(I,J),J=1,5)
  914 FORMAT(5F8.4)
  200 CONTINUE
C
      DO 250 I=51,250
      WRITE(IWR,913) KCON(I,1),KCON(I,2)
      WRITE(IWR,914) COEF(I,1),COEF(I,2)
  250 CONTINUE
C
    7 CONTINUE
      STOP
      END
/*
//LKED.SYSLMOND DD DISP=SHR,DSNAME=MSS.S2323.LOAD
//LKED.SYSIN DD *
  NAME NTU14(R)
/*
//
```

APPENDIX C

NTU32C COMPUTER GENERATED INPUT ANALYZER PROGRAM

```
C          THIS IS PROGRAM NTU32C
C
C
C          IT GENERATES AN INPUT FILE FOR THERMAL ANALYZER TO OBTAIN
C          1-3 EFFECTIVENESS-NTU RELATIONSHIP THAT IS NOT AVAILABLE IN
C          OPEN LITERATURE (2C MEANING ONE PARALLEL PASS AND TWO
C          COUNTERFLOW PASSES).
C
      DIMENSION COEF(200,5),KCON(200,5),L1(8),L2(3),L3(6),SET2(2),FL4(4)
C
      CHARACTER *79 TITLE
      CHARACTER *12 FNAME
C
      DATA IOT,IN,IPR,IWR/6,5,4,8/
C
C     OPEN PRINTER OUTPUT FILE
C
      OPEN(IPR,FILE='PRN',STATUS='NEW',FORM='FORMATTED',IOSTAT=ICK)
      IF(ICK.NE.0) WRITE(IOT,920)
  920 FORMAT(' Trouble opening printer output file' )
C
      WRITE(IOT,917)
  917 FORMAT(/' Input the title of this study - 79 columns only. ',
     & ' This title will appear',/,' at the top of every printed page',
     & ' of output:')
      READ(IN,918) TITLE
  918 FORMAT(A79)
C
      WRITE(IOT,901)
  901 FORMAT(/' INPUT HOT SIDE CAPACITY RATE:')
      READ(IN,902) CHOT
  902 FORMAT(BN,F10.0)
C
      WRITE(IOT,903)
  903 FORMAT(/' INPUT COLD SIDE CAPACITY RATE:')
      READ(IN,902) CCLD
C
      WRITE(IOT,904)
  904 FORMAT(/' INPUT OVERALL HEAT TRANSFER COEFFICIENT:')
      READ(IN,902) U
C
```

```
      WRITE(IOT,905)
  905 FORMAT(/' INPUT TOTAL HEAT TRANSFER SURFACE:')
      READ(IN,902) SURFTO
C
      WRITE(IOT,906)
  906 FORMAT(/' INPUT HOT SIDE INLET TEMPERATURE:')
      READ(IN,902) THOTIN
C
      WRITE(IOT,907)
  907 FORMAT(/' INPUT COLD SIDE INLET TEMPERATURE:')
      READ(IN,902) TCLDIN
C
      VALK1 = CHOT
      VALK2 = CCLD
      VALK3 = U*SURFTO/150.
      TINIT = 125.
C
C              FRONT END
C
      L1(1) = 200
      L1(2) = 2
      DO 10  I=3,8
   10 L1(I) = 0
C
      DO 20  I=1,3
   20 L2(I) = 0
C
      L3(1) = 300
      L3(2) = 50
      L3(3) = 6
      L3(4) = 2
      L3(5) = 4
      L3(6) = 6
C
      FL4(1) = .05
      FL4(2) = .66667
      FL4(3) = .8
      FL4(4) = TINIT
      L4 = 12
C
C    CONSTANT TEMPERATURES
C
      SET2(1) = THOTIN
      SET2(2) = TCLDIN
C
C    READY FOR INPUT SET 4
C
C    NODE 1
```

```
C
      KCON(1,1) = 1004
      KCON(1,2) = 1014
      KCON(1,3) = 2004
      KCON(1,4) = 3015
      COEF(1,4) = VALK1
      DO 50 I = 1,3
   50 COEF(1,I) = VALK3
C
C   NODES 2 TO 50
C
      DO 75 I = 2,50
      J = 101 - I
      K = 100 + I
      L = 201 - I
      N = I - 1
      KCON(I,1) = 10*N + 5
      KCON(I,2) = 10*J + 4
      KCON(I,3) = 10*K + 4
      KCON(I,4) = 10*L + 4
      COEF(I,1) = VALK1
      DO 80  II = 2,4
      COEF(I,II) = VALK3
   80 CONTINUE
   75 CONTINUE
C
C  NODE 51
C
      KCON(51,1) = 3025
      KCON(51,2) = 504
      COEF(51,1) = VALK2
      COEF(51,2) = VALK3
C
C   NODES 52 TO 200
C
      DO 120  I = 52,200
      K = I - 1
      IF(I.GT.100) GO TO 122
      L = I - 50
      M = 2*L - 1
      J = I - M
      GO TO 135
  122 IF(I.GT.150) GO TO 124
      J = I - 100
      GO TO 135
  124 L = I - 150
      M = 2*L - 1
      J = I - M - 100
```

```
  135 KCON(I,1) = 10*J + 4
      KCON(I,2) = 10*K + 5
      COEF(I,1) = VALK3
  120 COEF(I,2) = VALK2
C
C    END OF DATA SETUP
C    NOW CREATE INPUT FOR ANALYZER
C
C     WRITE(IOT,922)
C 922 FORMAT(/,' ENTER NAME OF INPUT FILE, INCLUDING DRIVE',
C    &  ' DESIGNATION:')
C     READ(IN,923) FNAME
C 923 FORMAT(A12)
C
C     OPEN INPUT FILE
C
      OPEN(IWR,FILE='TVSSI',STATUS='NEW',FORM='FORMATTED',IOSTAT=ICK)
      IF(ICK.GT.0)WRITE(IOT,924)
  924 FORMAT(/,' Trouble opening input file')
C
C     WRITE(IOT,925) FNAME
      WRITE(IOT,925)
C 925 FORMAT(/,' WRITING FILE :',A14)
  925 FORMAT(/,' WRITING FILE : TVSSI')
      WRITE(IWR,919) TITLE
  919 FORMAT(1X,A79)
      WRITE(IWR,908)(L1(I),I=1,8)
  908 FORMAT(9I4)
      WRITE(IWR,908)(L2(I),I=1,3)
      WRITE(IWR,908)(L3(I),I=1,6)
      WRITE(IWR,911) FL4(1),FL4(2),L4,FL4(3),FL4(4)
  911 FORMAT(F8.3,F8.5,I8,2F8.1)
      WRITE(IWR,912) SET2(1),SET2(2)
  912 FORMAT(2F8.0)
C
      DO 200  I = 1,50
      WRITE(IWR,913)(KCON(I,J),J=1,4)
  913 FORMAT(9I8)
      WRITE(IWR,914)(COEF(I,J),J=1,4)
  914 FORMAT(4F8.4)
  200 CONTINUE
C
      DO 250 I=51,200
      WRITE(IWR,913) KCON(I,1),KCON(I,2)
      WRITE(IWR,914) COEF(I,1),COEF(I,2)
  250 CONTINUE
C
      CLOSE(IWR,IOSTAT=ICK)
```

```
      IF(ICK.NE.0) WRITE(IOT,921)EEX
  921 FORMAT(' Trouble closing printer output file'  )
C
      STOP
      END
```

APPENDIX D

NTU32P COMPUTER GENERATED INPUT ANALYZER PROGRAM

```
C           THIS IS PROGRAM NTU32P
C
C
C           IT GENERATES AN INPUT FILE FOR THERMAL ANALYZER TO OBTAIN
C           1-3 EFFECTIVENESS-NTU RELATIONSHIP THAT IS NOT AVAILABLE IN
C           OPEN LITERATURE (2P MEANING TWO PARALLEL PASSES AND ONE
C           COUNTERFLOW PASS).
C
      DIMENSION COEF(200,5),KCON(200,5),L1(8),L2(3),L3(6),SET2(2),FL4(4)
C
      CHARACTER *79 TITLE
      CHARACTER *12 FNAME
C
      DATA IOT,IN,IPR,IWR/6,5,4,8/
C
C     OPEN PRINTER OUTPUT FILE
C
      OPEN(IPR,FILE='PRN',STATUS='NEW',FORM='FORMATTED',IOSTAT=ICK)
      IF(ICK.NE.0) WRITE(IOT,920)
  920 FORMAT(' Trouble opening printer output file' )
C
      WRITE(IOT,917)
  917 FORMAT(/' Input the title of this study - 79 columns only. ',
     & ' This title will appear',/,' at the top of every printed page',
     & ' of output:')
      READ(IN,918) TITLE
  918 FORMAT(A79)
C
      WRITE(IOT,901)
  901 FORMAT(/' INPUT HOT SIDE CAPACITY RATE:')
      READ(IN,902) CHOT
  902 FORMAT(BN,F10.0)
C
      WRITE(IOT,903)
  903 FORMAT(/' INPUT COLD SIDE CAPACITY RATE:')
      READ(IN,902) CCLD
C
      WRITE(IOT,904)
  904 FORMAT(/' INPUT OVERALL HEAT TRANSFER COEFFICIENT:')
      READ(IN,902) U
C
```

```
      WRITE(IOT,905)
  905 FORMAT(/' INPUT TOTAL HEAT TRANSFER SURFACE:')
      READ(IN,902) SURFTO
C
      WRITE(IOT,906)
  906 FORMAT(/' INPUT HOT SIDE INLET TEMPERATURE:')
      READ(IN,902) THOTIN
C
      WRITE(IOT,907)
  907 FORMAT(/' INPUT COLD SIDE INLET TEMPERATURE:')
      READ(IN,902) TCLDIN
C
      VALK1 = CHOT
      VALK2 = CCLD
      VALK3 = U*SURFTO/150.
      TINIT = 125.
C
C              FRONT END
C
      L1(1) = 200
      L1(2) = 2
      DO 10  I=3,8
   10 L1(I) = 0
C
      DO 20  I=1,3
   20 L2(I) = 0
C
      L3(1) = 300
      L3(2) = 50
      L3(3) = 6
      L3(4) = 2
      L3(5) = 4
      L3(6) = 6
C
      FL4(1) = .05
      FL4(2) = .66667
      FL4(3) = .8
      FL4(4) = TINIT
      L4 = 12
C
C   CONSTANT TEMPERATURES
C
      SET2(1) = THOTIN
      SET2(2) = TCLDIN
C
C   READY FOR INPUT SET 4
C
C   NODE 1
```

```
C
      KCON(1,1) = 514
      KCON(1,2) = 1504
      KCON(1,3) = 1514
      KCON(1,4) = 3015
      COEF(1,4) = VALK1
      DO 50 I = 1,3
   50 COEF(1,I) = VALK3
C
C   NODES 2 TO 50
C
      DO 75 I = 2,50
      J = I + 50
      K = 151 - I
      L = 150 + I
      N = I - 1
      KCON(I,1) = 10*N + 5
      KCON(I,2) = 10*J + 4
      KCON(I,3) = 10*K + 4
      KCON(I,4) = 10*L + 4
      COEF(I,1) = VALK1
      DO 80  II = 2,4
      COEF(I,II) = VALK3
   80 CONTINUE
   75 CONTINUE
C
C  NODE 51
C
      KCON(51,1) = 3025
      KCON(51,2) = 14
      COEF(51,1) = VALK2
      COEF(51,2) = VALK3
C
C   NODES 52 TO 200
C
      DO 120  I = 52,200
      K = I - 1
      IF(I.GT.100) GO TO 122
      J = I - 50
      GO TO 135
  122 IF(I.GT.150) GO TO 124
      L = I - 100
      M = 2*L - 1
      N = M + 50
      J = I - N
      GO TO 135
  124 J = I - 150
  135 KCON(I,1) = 10*J + 4
```

```
      KCON(I,2) = 10*K + 5
      COEF(I,1) = VALK3
  120 COEF(I,2) = VALK2
C
C    END OF DATA SETUP
C    NOW CREATE INPUT FOR ANALYZER
C
      WRITE(IOT,922)
  922 FORMAT(/,' Enter name of input file, including drive',
     &  ' DESIGNATION:')
      READ(IN,923) FNAME
  923 FORMAT(A12)
C
C     OPEN INPUT FILE
C
      OPEN(IWR,FILE=FNAME,STATUS='NEW',FORM='FORMATTED',IOSTAT=ICK)
      IF(ICK.GT.0)WRITE(IOT,924)
  924 FORMAT(/,' Trouble opening input file')
C
      WRITE(IOT,925) FNAME
  925 FORMAT(/,' Writing file :',A14)
      WRITE(IWR,919) TITLE
  919 FORMAT(1X,A79)
      WRITE(IWR,908)(L1(I),I=1,8)
  908 FORMAT(9I4)
      WRITE(IWR,908)(L2(I),I=1,3)
      WRITE(IWR,908)(L3(I),I=1,6)
      WRITE(IWR,911) FL4(1),FL4(2),L4,FL4(3),FL4(4)
  911 FORMAT(F8.3,F8.5,I8,F8.5,F8.2)
      WRITE(IWR,912) SET2(1),SET2(2)
  912 FORMAT(2F8.0)
C
      DO 200  I = 1,50
      WRITE(IWR,913)(KCON(I,J),J=1,4)
  913 FORMAT(9I8)
      WRITE(IWR,914)(COEF(I,J),J=1,4)
  914 FORMAT(4F8.4)
  200 CONTINUE
C
      DO 250 I=51,200
      WRITE(IWR,913) KCON(I,1),KCON(I,2)
      WRITE(IWR,914) COEF(I,1),COEF(I,2)
  250 CONTINUE
C
      CLOSE(IWR,IOSTAT=ICK)
      IF(ICK.NE.0) WRITE(IOT,921)EEX
  921 FORMAT(' Trouble closing printer output file'  )
C
```

```
STOP
END
```

APPENDIX E

SAMPLE OUTPUT FROM NTU14 COMPUTER INPUT ANALYZER PROGRAM

```
 NTU=0.05  R=0.15  COUNTER
 200    2    0    0    0    0    0    0
   0    0    0
 300   50    6    2    4    6
   0.050 0.66667         12 0.80000   125.00
    200.       100.
    1004       1014       2004       3015
  0.0125     0.0125     0.0125250.0000
      15        994       1024       1994
250.0000     0.0125     0.0125     0.0125
      25        984       1034       1984
250.0000     0.0125     0.0125     0.0125
      35        974       1044       1974
250.0000     0.0125     0.0125     0.0125
      45        964       1054       1964
250.0000     0.0125     0.0125     0.0125
      55        954       1064       1954
250.0000     0.0125     0.0125     0.0125
      65        944       1074       1944
250.0000     0.0125     0.0125     0.0125
      75        934       1084       1934
250.0000     0.0125     0.0125     0.0125
      85        924       1094       1924
250.0000     0.0125     0.0125     0.0125
      95        914       1104       1914
250.0000     0.0125     0.0125     0.0125
     105        904       1114       1904
250.0000     0.0125     0.0125     0.0125
     115        894       1124       1894
250.0000     0.0125     0.0125     0.0125
     125        884       1134       1884
250.0000     0.0125     0.0125     0.0125
     135        874       1144       1874
250.0000     0.0125     0.0125     0.0125
     145        864       1154       1864
250.0000     0.0125     0.0125     0.0125
     155        854       1164       1854
250.0000     0.0125     0.0125     0.0125
     165        844       1174       1844
250.0000     0.0125     0.0125     0.0125
     175        834       1184       1834
250.0000     0.0125     0.0125     0.0125
```

185	824	1194	1824
250.0000	0.0125	0.0125	0.0125
195	814	1204	1814
250.0000	0.0125	0.0125	0.0125
205	804	1214	1804
250.0000	0.0125	0.0125	0.0125
215	794	1224	1794
250.0000	0.0125	0.0125	0.0125
225	784	1234	1784
250.0000	0.0125	0.0125	0.0125
235	774	1244	1774
250.0000	0.0125	0.0125	0.0125
245	764	1254	1764
250.0000	0.0125	0.0125	0.0125
255	754	1264	1754
250.0000	0.0125	0.0125	0.0125
265	744	1274	1744
250.0000	0.0125	0.0125	0.0125
275	734	1284	1734
250.0000	0.0125	0.0125	0.0125
285	724	1294	1724
250.0000	0.0125	0.0125	0.0125
295	714	1304	1714
250.0000	0.0125	0.0125	0.0125
305	704	1314	1704
250.0000	0.0125	0.0125	0.0125
315	694	1324	1694
250.0000	0.0125	0.0125	0.0125
325	684	1334	1684
250.0000	0.0125	0.0125	0.0125
335	674	1344	1674
250.0000	0.0125	0.0125	0.0125
345	664	1354	1664
250.0000	0.0125	0.0125	0.0125
355	654	1364	1654
250.0000	0.0125	0.0125	0.0125
365	644	1374	1644
250.0000	0.0125	0.0125	0.0125
375	634	1384	1634
250.0000	0.0125	0.0125	0.0125
385	624	1394	1624
250.0000	0.0125	0.0125	0.0125
395	614	1404	1614
250.0000	0.0125	0.0125	0.0125
405	604	1414	1604
250.0000	0.0125	0.0125	0.0125
415	594	1424	1594
250.0000	0.0125	0.0125	0.0125

425	584	1434	1584
250.0000	0.0125	0.0125	0.0125
435	574	1444	1574
250.0000	0.0125	0.0125	0.0125
445	564	1454	1564
250.0000	0.0125	0.0125	0.0125
455	554	1464	1554
250.0000	0.0125	0.0125	0.0125
465	544	1474	1544
250.0000	0.0125	0.0125	0.0125
475	534	1484	1534
250.0000	0.0125	0.0125	0.0125
485	524	1494	1524
250.0000	0.0125	0.0125	0.0125
495	514	1504	1514
250.0000	0.0125	0.0125	0.0125
3025	504		
37.5000	0.0125		
494	515		
0.0125	37.5000		
484	525		
0.0125	37.5000		
474	535		
0.0125	37.5000		
464	545		
0.0125	37.5000		
454	555		
0.0125	37.5000		
444	565		
0.0125	37.5000		
434	575		
0.0125	37.5000		
424	585		
0.0125	37.5000		
414	595		
0.0125	37.5000		
404	605		
0.0125	37.5000		
394	615		
0.0125	37.5000		
384	625		
0.0125	37.5000		
374	635		
0.0125	37.5000		
364	645		
0.0125	37.5000		
354	655		
0.0125	37.5000		

344	665
0.0125	37.5000
334	675
0.0125	37.5000
324	685
0.0125	37.5000
314	695
0.0125	37.5000
304	705
0.0125	37.5000
294	715
0.0125	37.5000
284	725
0.0125	37.5000
274	735
0.0125	37.5000
264	745
0.0125	37.5000
254	755
0.0125	37.5000
244	765
0.0125	37.5000
234	775
0.0125	37.5000
224	785
0.0125	37.5000
214	795
0.0125	37.5000
204	805
0.0125	37.5000
194	815
0.0125	37.5000
184	825
0.0125	37.5000
174	835
0.0125	37.5000
164	845
0.0125	37.5000
154	855
0.0125	37.5000
144	865
0.0125	37.5000
134	875
0.0125	37.5000
124	885
0.0125	37.5000
114	895
0.0125	37.5000

```
104 905
0.0125 37.5000
94 915
0.0125 37.5000
84 925
0.0125 37.5000
74 935
0.0125 37.5000
64 945
0.0125 37.5000
54 955
0.0125 37.5000
44 965
0.0125 37.5000
34 975
0.0125 37.5000
24 985
0.0125 37.5000
14 995
0.0125 37.5000
14 1005
0.0125 37.5000
24 1015
0.0125 37.5000
34 1025
0.0125 37.5000
44 1035
0.0125 37.5000
54 1045
0.0125 37.5000
64 1055
0.0125 37.5000
74 1065
0.0125 37.5000
84 1075
0.0125 37.5000
94 1085
0.0125 37.5000
104 1095
0.0125 37.5000
114 1105
0.0125 37.5000
124 1115
0.0125 37.5000
134 1125
0.0125 37.5000
144 1135
0.0125 37.5000
```

```
  154      1145
0.0125  37.5000
  164      1155
0.0125  37.5000
  174      1165
0.0125  37.5000
  184      1175
0.0125  37.5000
  194      1185
0.0125  37.5000
  204      1195
0.0125  37.5000
  214      1205
0.0125  37.5000
  224      1215
0.0125  37.5000
  234      1225
0.0125  37.5000
  244      1235
0.0125  37.5000
  254      1245
0.0125  37.5000
  264      1255
0.0125  37.5000
  274      1265
0.0125  37.5000
  284      1275
0.0125  37.5000
  294      1285
0.0125  37.5000
  304      1295
0.0125  37.5000
  314      1305
0.0125  37.5000
  324      1315
0.0125  37.5000
  334      1325
0.0125  37.5000
  344      1335
0.0125  37.5000
  354      1345
0.0125  37.5000
  364      1355
0.0125  37.5000
  374      1365
0.0125  37.5000
  384      1375
0.0125  37.5000
```

394	1385
0.0125	37.5000
404	1395
0.0125	37.5000
414	1405
0.0125	37.5000
424	1415
0.0125	37.5000
434	1425
0.0125	37.5000
444	1435
0.0125	37.5000
454	1445
0.0125	37.5000
464	1455
0.0125	37.5000
474	1465
0.0125	37.5000
484	1475
0.0125	37.5000
494	1485
0.0125	37.5000
504	1495
0.0125	37.5000
504	1505
0.0125	37.5000
494	1515
0.0125	37.5000
484	1525
0.0125	37.5000
474	1535
0.0125	37.5000
464	1545
0.0125	37.5000
454	1555
0.0125	37.5000
444	1565
0.0125	37.5000
434	1575
0.0125	37.5000
424	1585
0.0125	37.5000
414	1595
0.0125	37.5000
404	1605
0.0125	37.5000
394	1615
0.0125	37.5000

384	1625
0.0125	37.5000
374	1635
0.0125	37.5000
364	1645
0.0125	37.5000
354	1655
0.0125	37.5000
344	1665
0.0125	37.5000
334	1675
0.0125	37.5000
324	1685
0.0125	37.5000
314	1695
0.0125	37.5000
304	1705
0.0125	37.5000
294	1715
0.0125	37.5000
284	1725
0.0125	37.5000
274	1735
0.0125	37.5000
264	1745
0.0125	37.5000
254	1755
0.0125	37.5000
244	1765
0.0125	37.5000
234	1775
0.0125	37.5000
224	1785
0.0125	37.5000
214	1795
0.0125	37.5000
204	1805
0.0125	37.5000
194	1815
0.0125	37.5000
184	1825
0.0125	37.5000
174	1835
0.0125	37.5000
164	1845
0.0125	37.5000
154	1855
0.0125	37.5000

144	1865
0.0125	37.5000
134	1875
0.0125	37.5000
124	1885
0.0125	37.5000
114	1895
0.0125	37.5000
104	1905
0.0125	37.5000
94	1915
0.0125	37.5000
84	1925
0.0125	37.5000
74	1935
0.0125	37.5000
64	1945
0.0125	37.5000
54	1955
0.0125	37.5000
44	1965
0.0125	37.5000
34	1975
0.0125	37.5000
24	1985
0.0125	37.5000
14	1995
0.0125	37.5000

APPENDIX F

MODIFIED SECTIONS OF THERMAL ANALYZER TO RUN ON BATCH SYSTEM

```
//QHARE JOB (2323,0267),'TVCOUNT,CLASS=C
//*MAIN  ORG=NPGVM1.2323P,SYSTEM=SY2
//  EXEC FORTVCL,PARM.LKED='LIST,MAP'
//FORT.SYSIN DD *
C<<<<<<<<<<<<<ZU0022PU    Z075    00511 <<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<
c    THIS IS TVCOUNT WHICH IS A MODIFICATION TO TVSSI IN ORDER
c  TO RUN ON THE MULTIPLE VIRTUAL SYSTEM (MVS) WITH JOB ENTRY SUBSYSTEM
c  AND NETWORKING (JES3), MORE COMMONLY REFFERED TO AS THE BATCH SYSTEM.
C  NOTE THE ONLY CHANGES NEEDED WAS THE ADDITION OF JCL CODES.
c    TVCOUNT WILL ALSO HANDLE HEAT INPUT IN WATTS AND TEMPERATURES IN
C  DEGREES CENTIGRADE. CONDUCTANCES WILL BE IN WATTS/DEG
C
C                          PROGRAM TVCOUNT
C
C                 THIS PROGRAM WILL HANDLE UP TO 315 NODES
C *********************************************************************
C *
C *  UPDATED APRIL 1985
C *  NOTE: IF DAMPING VALUES AND INITIAL TEMPS ARE THE SAME FOR ALL NODE
C *  ENTER 1 DAMPING VALUE COLS 25-32 CARD 3
C *  ENTER 1 INITIAL TEMP COLS 33-40 CARD 3
C *  CALL FOR DATA SET 6 ON CARD 2
C *
C *********************************************************************
C
C *********************************************************************
C *
C *      JCL NOTE: LISTING OF INPUT IS WRITTEN ON UNIT 3 BEFORE NORMAL
C *      OUTPUT ON UNIT 8. TO DELETE INPUT LIST, INSERT DUMMY
C *      DATA DEFINITION STATEMENT
C *
C *********************************************************************
C
      IMPLICIT REAL*4 (A-H,O-Z)
      DIMENSION  TCO(5), CONTMP(50), ITAG(100), TCOEF(90),  COEF(100),
     1  H(316), A(316), TOLD(315), TOLD1(315), TOLD2(315), BETA(315),
     2  TITLE(20), INPTAG(10),EX(4), TMPHT(90), TMPHTV(5), NOCONT(6),
     3  HTR(42), BTUCRV(5), TMPCRV(5), TIMCO(5)
      CHARACTER *4 HEAD,HEAD1
      DATA HEAD,HEAD1/'BETA','TOLD'/
 1000 READ(4,500,END=999) TITLE
      IER = 0
```

```
      REWIND 1
      REWIND 2
      READ(4,501) N,NCT,NOHTRS,NOEXP,NOCASE,NTCOEF,NODCFH,NTMPHT
      NP1 = N + 1
      READ (4,501)  NTAG8,NTAG9,NTAG11
      READ(4,501) NONODS,NOCT,NOHTR,INPTAG
      READ(4,505) ERR,ALPHA,MNITS,BETA(1),TOLD(1)
      MAXNIT = IABS(MNITS)
      NIT = 0
      IF(BETA(1).NE.0.) GO TO 1010
      READ(4,502)   (BETA(I),I=1,N)
      GO TO 1030
 1010 DO 1020 I=2,N
 1020 BETA(I) = BETA(I-1)
C
 1030 DO 15 I=1,10
      INI = INPTAG(I)
      IF (INI)  17,17,6
    6 GO TO (1,2,3,7,10,12,14,808,809,810,811,810,810,810) , INI
    1 NTCIN = 18*NTCOEF
      READ(4, 502) (TCOEF(K), K=1,NTCIN)
      GO TO 15
    2 READ(4,  502)  (CONTMP(K), K=1,NCT)
      GO TO 15
    3 READ(4,    501)  NOCONT
      READ(4,    502)  (HTR(K), K=1,36)
      GO TO 15
    7 DO 9 L = 1,N
      ITAG(10) = 0
      READ(4,504)  NT, (ITAG(K),          K=1,9)
      READ(4,502) (COEF(M) , M=1, 9)
      IF (NT-9 )   8 , 8 ,710
  710 DO 715 K=10,NT,9
      ITAG (K+9) = 0
      KE = K+8
      READ(4,503)  (ITAG(M),          M=K ,KE)
  715 READ(4,502) (COEF(M) , M=K ,KE)
    8 WRITE (1) ITAG,COEF
      WRITE(3,556)(ITAG(KK),KK=1,30)
    9 CONTINUE
      ENDFILE 1
      REWIND 1
      GO TO 15
   10 READ(4, 502) EX
      GO TO 15
   12 IF (TOLD(1).NE.0.) GO TO 1201
      READ(4,502) (TOLD(K), K=1,N)
      GO TO 15
```

```
 1201 DO 1202 L=2,N
 1202 TOLD(L) = TOLD(L-1)
      GO TO 15
   14 K2 = 18*NTMPHT
      READ(4,502) (TMPHT(K), K=1,K2)
      GO TO 15
  808 READ(4,502)  BTUCRV
      GO TO 15
  809 READ(4,502)  TMPCRV
      GO TO 15
  810 WRITE (8,533)  INI
      STOP
  811 READ(4,502)  TIMCO
   15 CONTINUE
   17 CALL TVPAGE    (0,TITLE)
      CALL  TVSOUT(N, 1,TOLD,TOLD,TITLE)
C
      WRITE(3,551) TITLE
      WRITE(3,552)N,NCT,NOHTRS,NOEXP,NOCASE,NTCOEF,NODCFH,NTMPHT,NONODS,
     1 NOCT,NOHTR,MNITS
      WRITE(3,553)INPTAG,ERR,ALPHA
      WRITE(3,554)N,HEAD,(BETA(I),I=1,N)
      HEAD=HEAD1
      WRITE(3,554)N,HEAD,(TOLD(I),I=1,N)
      WRITE(3,555)(CONTMP(I),I=1,NCT)
   20 IF(NOHTRS) 24,24,21
   21 CALL TVSHTR(NOHTRS,HTR,CASBTU,NOCONT,TOLD,NODCFH)
   24 DO 107 NOD=1,N
      DO 25 I=1,NP1
   25 A(I) = 0.
      READ (1) ITAG, COEF
      DO 100  IWD =1,99
      IF (ITAG(IWD) .EQ. 0)  GO TO 105
      NODEI = ITAG(IWD) / 10
      METHI = MOD (ITAG(IWD) , 10)
      NTTH = NODEI - NONODS
      IF ( NTTH.LE.0 )  GO TO 55
      IF ( NODEI.EQ.999 )  GO TO 50
      NNODE = NTTH
      NTTH = NTTH - NOCT
      IF ( NTTH.LE.0 )  GO TO 60
      NNODE = NTTH
      NTTH = NTTH - NOHTR
      IF ( NTTH.LE.0 )  GO TO 49
      IF ( NTTH.GT.5 )  GO TO 820
      IF ( NTTH.LE.NTAG8 )  GO TO 815
      WRITE (9,610)  NOD,NODEI,NTAG8
      IER = 1
```

```
      GO TO 100
  815 A(NP1) = A(NP1) - BTUCRV(NTTH)*COEF(IWD)
      GO TO 100
  820 NTTH = NTTH - 5
      IF ( NTTH.GT.5 )  GO TO 830
      IF ( NTTH.LE.NTAG9 ) GO TO 825
      WRITE (9,620)  NOD,NODEI,NTAG9
      IER = 1
      GO TO 100
  825 T1 = TMPCRV(NTTH)
      GO TO 65
  830 NTTH = NTTH - 5
      IF ( NTTH.LE.5 )  GO TO 48
      WRITE ( 9,570 ) NOD,NODEI
      IER = 1
      GO TO 100
   48 IF ( NTTH.LE.NTMPHT ) GO TO 480
      WRITE (9,630)  NOD,NODEI,NTMPHT
      IER = 1
      GO TO 100
  480 WCOEF = NTTH + 1100
      CALL TVFTMP (WCOEF, TOLD(NOD), TMPHT)
      A(NP1) = A(NP1)-WCOEF*COEF(IWD)
      TMPHTV(NTTH) = WCOEF
      GO TO 100
   49 IF ( NNODE.LE.NOHTRS )  GO TO 490
      WRITE (9,600)  NOD,NODEI,NOHTRS
      IER = 1
      GO TO 100
  490 LOCH = 36 + NNODE
      A(NP1) = A(NP1) - HTR(LOCH) * COEF(IWD)
      GO TO 100
   50 A(NP1) = A(NP1) - COEF(IWD)
      GO TO 100
   55 T1 = TOLD(NODEI)
      GO TO 65
   60 IF ( NNODE.LE.NCT ) GO TO 61
      WRITE (9,590)  NOD,NODEI,NCT
      IER = 1
      GO TO 100
   61 T1 = CONTMP(NNODE)
   65 T2 = TOLD(NOD)
      WCOEF = COEF(IWD)
C * * *
C           ASSUMING THE USER HAS RUN PROGRAM CHECK AND FIXED ANY WCOEFS
C           THAT MIGHT HAVE BEEN IN ERROR. TVSSI, THEREFORE, WILL NOT
C           CHECK AGAIN FOR A WCOEF > 100.  NO WARNING IS PRINTED.
C * * *
```

```
      IF ( WCOEF.LE.1000. )  GO TO 73
      IF ( WCOEF.LE.1005. )  GO TO 850
      IF ( WCOEF.LE.1100. )  GO TO 73
      IF ( WCOEF.GT.1105. )  GO TO 73
      ATEMP = (T1+T2) / 2.
      LC = WCOEF - 1099.99
      IF ( LC.LE.NTCOEF )   GO TO 840
      WRITE (9,640)  NOD,NODEI,WCOEF,NTCOEF
      IER = 1
      GO TO 100
  840 CALL TVFTMP (WCOEF, ATEMP, TCOEF)
      TCO(LC) = WCOEF
      GO TO 73
  850 LC = WCOEF - 999.99
      IF ( LC.LE.NTAG11 )  GO TO 855
      WRITE (9,650)  NOD,NODEI,WCOEF,NTAG11
      IER = 1
      GO TO 100
  855 WCOEF = TIMCO(LC)
C * * *
C         THE FOLLOWING COMPUTED GO TO WILL FALL THROUGH TO THE NEXT
C         STATEMENT FOR METHODS 6,7,8,& 9
C * * *
   73 GO TO (80,730,730,80,80), METHI
  730 TD = T1 - T2
      IF ( TD.NE.0. )  GO TO 731
      WCOEF = 0.
      GO TO 80
  731 IF ( ABS(TD).LT.1E-05 )  TD = SIGN( 1E-05,TD )
      ABSTD = ABS(TD)
      GO TO (80,74,75,80,80), METHI
      LC = METHI - 5
      IF ( LC.LE.NOEXP )  GO TO 77
      WRITE (9,660) NOD,NODEI,METHI,NOEXP
      IER = 1
      GO TO 100
   77 T3 = EX (LC)
C * *
C         THE FOLLOWING STATEMENT EQUATES TO
C         WCOEF = WCOEF * ((ABSTD ** T3)/ABSTD)
C * *
      WCOEF = WCOEF * ABSTD ** (T3-1.)
      GO TO 80
C * *
C        THE FOLLOWING STATEMENT EQUATES TO
C        WCOEF = WCOEF * (ABSTD ** 1.25)/ABSTD
C * *
   74 WCOEF = WCOEF * ABSTD ** .25
```

```
      GO TO 80
   75 T3 = (T1+273.) / 100.
      T4 = (T2+273.) / 100.
      WCOEF = WCOEF * ((T3**4 - T4**4) / TD )
   80 IF(NODEI - NONODS) 85,85,90
   85 A(NODEI) = A(NODEI) + WCOEF
      GO TO 95
   90 A(NP1) = A(NP1) - WCOEF*T1
   95 A(NOD) = A(NOD) - WCOEF
  100 CONTINUE
  105 IF(NOD-NOCASE)  107,106,107
  106 A(NP1) = A(NP1) - CASBTU
  107 WRITE (2) (A(K),K=1,NP1)
C
C * * *
C          IF IER.NE.0 AT THIS POINT, IT MEANS THERE ARE ERRORS IN THE
C          INPUT; TVSSI WILL SKIP THE CALCULATIONS AND RETURN TO
C          STATEMENT  1000  FOR ANOTHER PROBLEM.
C * * *
C
      IF ( IER.EQ.0 )  GO TO 110
      WRITE (9,670)
      GO TO 1000
C
  110 ENDFILE 2
      REWIND 1
      REWIND 2
      CALL CHOST (N, NP1, H)
      ERRTAG = 0.
      NIT = NIT + 1
      IF (NIT .LT. 3)  GO TO 120
      CALL CBETA (N, ALPHA, BETA, TOLD, TOLD1, TOLD2, ERR)
  120 DO 130 I=1,N
      IF(NIT .EQ. 1) TOLD1(I) = 0.0
      TOLD2(I) = TOLD1(I)
      TOLD1(I) = TOLD(I)
      TOLD(I) = TOLD(I) + BETA(I) * (  H(I)  -TOLD(I))
      A(I)    = TOLD(I) - TOLD1(I)
  130 IF (ABS(A(I)) .GT. ERR)  ERRTAG = 1.
      CALL TVSOUT (N, 2, TOLD, A, TITLE)
      IF (NOHTRS)  140,140,135
  135 LOCH = 36 + NOHTRS
      CALL  TVPAGE   (3,TITLE)
      WRITE (8,525)  (HTR(K), K=37,LOCH)
      WRITE (8,526)  CASBTU
  140 IF(NTCOEF) 150,150,145
  145 CALL  TVPAGE   (2,TITLE)
      WRITE (8,528)  (TCO  (K), K= 1,NTCOEF)
```

```
  150 IF (NTMPHT)  160,160,155
  155 CALL TVPAGE (2,TITLE)
      WRITE (8,529) (TMPHTV(K), K=1,NTMPHT)
  160 IF (NIT .GE. MAXNIT .AND. MAXNIT .GT. 0)  GO TO 175
      IF ( ERRTAG.GT.0. )  GO TO 20
      WRITE (10,500) TITLE
      WRITE (10,501) N
      WRITE (10,680) (TOLD(I),I=1,N)
      GO TO 1000
  175 DO 180 I=1,N
  180 IF (BETA(I) .LT. .0001)  BETA(I) = .0001
      WRITE (8,530)  MAXNIT, (BETA(I),I=1,N)
      GO TO 1000
C
  500 FORMAT (20A4)
  501 FORMAT (18I4)
  502 FORMAT (9G8.0)
  503 FORMAT (9I8)
  504 FORMAT (I2,I6,8I8)
  505 FORMAT (2G8.0, I8, 2G8.0)
  525 FORMAT(/12H HTRS WATTS  , 6G12.5)
  526 FORMAT (12H CASE WATTS  ,  G12.5)
  528 FORMAT (/12H TEMP COEFS   , 6G12.5)
  529 FORMAT ( /12H TEMP HEAT     , 6G12.5),
  530 FORMAT ( '0 OVER ',I5, ' ITERATIONS ' // '  BETAS' // (20F6.4) )
  533 FORMAT (' SET ',I5, ' IS NOT ACCEPTABLE.  FIX DECK AND RESUBMIT ' )
  551 FORMAT(1H ,20A4)
  552 FORMAT(//,'  N   NCT NOHTRS NOEXP NOCASE NTCOEF NODCFH NTMPHT NONO
     1DS  NOCT  NOHTR  MNITS',/12I6,/)
  553 FORMAT(/,'           INPTAG 1 - 10
     1          ERR  ALPHA',/,10I7,2(1X,F11.4))
  554 FORMAT(/,' ',I4,'  ',A4,' 'S',(/,1X,16F8.4))
  555 FORMAT('   CONTMP(I),I=1,NCT)',(/,1X,16G8.2))
  556 FORMAT(15I8)
  570 FORMAT ( //1X,'* IMPOSSIBLE NODE NUMBER - YOU SPECIFIED AN ',
     1    'INTERACTION FROM NODE',I5,'   TO A NODE - - ',I5/),
  590 FORMAT (/1X,'* * * INVALID NODE - - YOU SPECIFIED AN ',
     1 'INTERACTION FROM NODE',I5,5X,'TO A NODE',I5,/15X,
     2 'BUT THERE ARE ONLY',I4,'   CONSTANT TEMPERATURES')
  600 FORMAT (/1X,'* * * INVALID NODE - - YOU SPECIFIED AN ',
     1 'INTERACTION FROM NODE',I5,5X,'TO A NODE',I5,/15X,
     2 'BUT THERE ARE ONLY ',I3,'   HEATERS')
  610 FORMAT (/1X,'* * * INVALID NODE - - YOU SPECIFIED AN ',
     1 'INTERACTION FROM NODE',I5,5X,'TO A NODE',I5,/15X,
     2 'BUT THERE ARE ONLY ',I3,'   WATT CURVES (TAG 8)')
  620 FORMAT (/1X,'* * * INVALID NODE - - YOU SPECIFIED AN ',
     1 'INTERACTION FROM NODE',I5,5X,'TO A NODE',I5,/15X,
     2 'BUT THERE ARE ONLY',I4,'   TEMPERATURE CURVES (SET 9)')
```

```
  630 FORMAT (/1X,'* * * INVALID NODE - - YOU SPECIFIED AN ',
     1 'INTERACTION FROM NODE',I5,5X,'TO A NODE',I5,/15X,
     2 'BUT THERE ARE ONLY',I4,'   TEMP-DEPENDENT WATT CURVES (SET 7)')
  640 FORMAT (/1X,'* * * INVALID CONDUCTANCE - - NODE',I5,5X,
     1 'TO NODE',I5,5X,'YOU SPECIFIED A CONDUCTANCE OF ',F7.1,/15X,
     2 'BUT THERE ARE ONLY ',I3,'   TEMP-DEPENDENT ',
     3 'COEFF CURVES (SET 1)')
  650 FORMAT (/1X,'* * * INVALID CONDUCTANCE - - NODE',I5,5X,
     1 'TO NODE',I5,5X,'YOU SPECIFIED A CONDUCTANCE OF ',F7.1,/15X,
     2 'BUT THERE ARE ONLY ',I3,'   TIME COEFFS (SET 11)')
  660 FORMAT (/1X,'* * * INVALID METHOD - - NODE',I5,5X,'TO NODE',
     1 I5,5X,'YOU SPECIFIED A METHOD OF ',I3,/15X,'BUT THERE ARE ONLY ',
     2 I3,'   UNIQUE EXPONENTS (SET 5)')
  670 FORMAT (////1X,'THERE ARE ERRORS IN YOUR INPUT FOR THIS PROBLEM.',
     1 //10X,'TVSSI WILL SKIP ANY FURTHER CALCULATIONS FOR THIS PROBLEM'
     2 ,/////)
  680 FORMAT (9F8.2)
  999 CONTINUE
      STOP
      END
C
      SUBROUTINE CBETA ( N, ALPHA, BETA, TN, TNM1, TNM2, ERR )
          IMPLICIT REAL*4 (A-H,O-Z)
      DIMENSION TN(315) , TNM1(315) , TNM2(315) , BETA(315)
      DO 50 I=1,N
      TMTM1 = TN(I)-TNM1(I)
      T12 = TNM1(I) - TNM2(I)
      IF ( ABS(T12).LT.1E-06 )  T12 = SIGN( 1E-06,T12 )
      GAMMA = TMTM1 /T12
      IF (GAMMA .GT. 0.)  GO TO 10
      IF ( ABS(TMTM1) .LE. ERR )  GO TO 50
      IF (GAMMA .LT. -ALPHA)  BETA(I) = -BETA(I)*ALPHA/GAMMA
      GO TO 50
   10 IF (GAMMA .GT. 1.)  GO TO 50
      BETA(I) = BETA(I) / ALPHA
   50 IF (BETA(I) .GT. 1.)  BETA(I) = 1.
      RETURN
      END
C
      SUBROUTINE  CHOST (N,NP1,EL)
          IMPLICIT REAL*4 (A-H,O-Z)
      DIMENSION EL(316) , LOCS(316) ,SAVE(49770)
      LOCS(1) = 1
      NM1 = N - 1
      I = 0
C
   10 I = I +1
      READ (2) (EL(K),K=1,NP1)
```

```
      IP1 = I + 1
      IF ( I .EQ. 1 )  GO TO 50
      DO 45 J=2,I
      LR = LOCS(J-1)
      IF ( EL(J-1) .EQ. 0. )  GO TO 45
      DO 40 JR=J,NP1
      IF ( SAVE(LR) .EQ. 0. )  GO TO 40
      EL(JR) = EL(JR) - EL(J-1)*SAVE(LR)
   40 LR = LR+1
   45 CONTINUE
   50 CONTINUE
   51 DO 60 K=IP1,NP1
      IF (EL(K) .EQ. 0.)  GO TO 60
      EL(K) = EL(K) / EL(I)
   60 CONTINUE
      IF (I .EQ. N)  GO TO 80
      LS = LOCS(I)
      LOCS(I+1) = LS+NP1-I
      DO 72 K = IP1,NP1
      SAVE(LS) = EL(K)
   72 LS = LS + 1
      GO TO 10
C
   80 REWIND 2
      EL(N) = EL(NP1)
      DO  90 I=1,NM1
      II = NP1 - I
      LF = N - I
      LR = LOCS(LF) + I
      EL(II-1) = SAVE(LR)
      DO 90 K = II,N
      LR = LOCS(LF) + K - II
      IF (SAVE(LR) .EQ. 0.)  GO TO 90
      EL(II-1) = EL(II-1) - SAVE(LR)*EL(K)
   90 CONTINUE
      RETURN
      END
C
      SUBROUTINE  TVFTMP  (CO, T, TCOEF)
          IMPLICIT REAL*4 (A-H,O-Z)
      DIMENSION TCOEF(90)
      NC =  CO - 1099.9
      NB =  18*NC - 17
      IF (T - TCOEF(NB))  2,2,6
    2 CO = TCOEF(NB+1)
      GO TO 60
    6 NE =  NB + 16
      NB = NB+2
```

```
      DO 50 K=NB,NE,2
      IF (T-TCOEF(K))   10,20,50
   10 TC = TCOEF(K)
      TCM2 = TCOEF(K-2)
      TCM1 = TCOEF(K-1)
      TCP1 = TCOEF(K+1)
      TCC = TC - TCM2
      IF ( ABS(TCC).LT.1E-06 ), TCC = SIGN( 1E-06,TCC )
      CO = ( (T - TCM2)/TCC ) * (TCP1 - TCM1) + TCM1
      GO TO 60
   20 CO = TCOEF (K+1)
      GO TO 60
   50 CONTINUE
      CO = TCOEF(NE+1)
   60 RETURN
      END
C
      SUBROUTINE  TVSOUT(N,NA,T1,T2,TITLE)
         IMPLICIT REAL*4 (A-H,O-Z)
      DIMENSION T1(315) ,T2(315) , ID(12) ,TITLE(20)
C
      CALL  TVPAGE (2,TITLE)
      WRITE (8,500)
      NL = NA+2
      DO 50 I=1,N,12
      CALL TVPAGE    (NL,TITLE)
      IF (I+11-N) 5,5,10
    5 N5=12
      GO TO 15
   10 N5=N-I+1
   15 DO 20 K=1,N5
   20 ID(K) = I+K-1
      WRITE (8,501)  (ID(K), K=1,N5)
      N1 = I + N5 - 1
      IF (NA-1)  25,25,30
   25 WRITE (8,504)  (T1(K), K= I,N1)
      GO TO 50
   30 WRITE (8,502)  (T1(K), K= I,N1)
      WRITE (8,503)  (T2(K), K= I,N1)
   50 CONTINUE
      RETURN
C
  500 FORMAT  (1H0)
  501 FORMAT  (/10H NODE NO.    ,12I9)
  502 FORMAT (12H NEW TEMPS     ,12F9.2)
  503 FORMAT (12H NEW - OLD     ,12F9.2)
  504 FORMAT (12H ORIG TEMPS    ,12F9.2)
      END
```

```
C
      SUBROUTINE  TVSHTR (NOHTRS,HTR,CASBTU,NOCONT,TOLD,NODCFH)
          IMPLICIT REAL*4 (A-H,O-Z)
      DIMENSION HTR(42) , NOCONT(6) , TOLD(315)
      CASBTU = 0.
      DO 25 K=1,NOHTRS
      LOK = 36+K
      IF (NOCONT(K))  24,24,43
   43 LOK1 = NOCONT(K)
      TEMP = TOLD(LOK1)
      IF (K-3)  45,45,50
   45 LOK2 = 0
      GO TO 55
   50 LOK2 = 18
   55 IF (NODCFH)   3,3,60
   60 IF (TEMP-HTR(2))  65,65,70
   65 HTR(LOK) = HTR(LOK2+1)
      CASBTU = CASBTU + HTR(LOK2+3)
      GO TO 25
   70 NODCFH = 0
    3 IF (TEMP - HTR(LOK2+4))  5,5,15
    5 HTR(LOK) = HTR(LOK2+9)
      CASBTU = CASBTU + HTR(LOK2+14)
      GO TO 25
   15 DO 16 K1=5,8
      LOK3 = LOK2+K1
      IF (TEMP - HTR(LOK3))  17,17,16
   16 CONTINUE
      HTR(LOK) = HTR(LOK2+13)
      CASBTU = CASBTU + HTR(LOK2+18)
      GO TO 25
   17 HTRL3 = HTR(LOK3) - HTR(LOK3 - 1)
      IF ( ABS(HTRL3).LT.1E-06 )  HTRL3 = SIGN( 1E-06,HTRL3 )
      FRAC = (TEMP - HTR(LOK3-1)) / HTRL3
      HTR(LOK) = (HTR(LOK3+5) - HTR(LOK3+4)) * FRAC + HTR(LOK3+4)
      CASBTU = CASBTU + (HTR(LOK3+10) - HTR(LOK3+9))*FRAC + HTR(LOK3+9)
      GO TO 25
   24 HTR(LOK) = 0.
   25 CONTINUE
      RETURN
      END
C
      SUBROUTINE  TVPAGE   (NL,FNAME)
          IMPLICIT REAL*4 (A-H,O-Z)
      DIMENSION FNAME (20)
      IF (NL)  10,10,20
   10 NPAGE = 0
      LINCNT = 75
```

```
   20 LINCNT = LINCNT + NL
      IF (LINCNT - 56)  40,40,30
   30 NPAGE = NPAGE + 1
      WRITE (8,50)  FNAME,NPAGE
      LINCNT = NL
   40 RETURN
   50 FORMAT (1H1,20X,20A4,8X,9HPAGE NO.  ,I3/)
      END
/*
//LKED.SYSLMOD DD DISP=SHR,DSNAME=MSS.S2323.LOADLIB
//LKED.SYSIN DD *
  NAME TVCOUNT(R)
/*
//
```

APPENDIX G

MODIFIED NTU14 PROGRAM NTU14BC USED TO RUN ON BATCH SYSTEM

```
//OHARE   JOB (2323,0267),'NTUI4BC',CLASS=B
//*MAIN   ORG=NPGVM1.2323P
//   EXEC FORTVCL,PARM.LKED='LIST,MAP'
//FORT.SYSIN DD *
C          THIS IS PROGRAM ENTU14
C
C
C          IT GENERATES AN INPUT FILE FOR THERMAL ANALYZER TO VERIFY
C          1-4 EFFECTIVENESS-NTU RELATIONSHIP THAT IS AVAILABLE IN
C          OPEN LITERATURE.
C
      INTEGER O,P
      DIMENSION COEF(250,5),KCON(250,5),L1(8),L2(3),L3(6),SET2(2),FL4(4)
     1,CHOT(22,22),CCLD(22,22),U(22,22),SURFTO(22,22),THOTIN(22,22),
     2TCLDIN(22,22),TITLE(22,22),FNAME(22,22)
C
      CHARACTER *25 TITLE
      CHARACTER *25 FNAME
C
      DO 7 O=1,21,2
      P=O+1
      READ(1,900) CHOT(O,1),CCLD(O,2),U(O,3),SURFTO(O,4),THOTIN(O,5),TCL
     1DIN(O,6)
  900 FORMAT(2F10.0,F10.5,3F10.0)
      READ(1,918) TITLE(P,1),FNAME(P,2)
  918 FORMAT(2A25)
C
C     OPEN OUTPUT FILE
C
      OPEN(2,FILE=FNAME(P,2),FORM='FORMATTED')
C
      VALK1 = CHOT(O,1)
      VALK2 = CCLD(O,2)
      VALK3 = U(O,3)*SURFTO(O,4)/200.
      TINIT = 125.
C
C            FRONT END
C
      L1(1) = 250
      L1(2) = 2
      DO 10  I=3,8
   10 L1(I) = 0
```

```
C
      DO 20  I=1,3
   20 L2(I) = 0
C
      L3(1) = 300
      L3(2) = 50
      L3(3) = 6
      L3(4) = 2
      L3(5) = 4
      L3(6) = 6
C
      FL4(1) = .05
      FL4(2) = .66667
      FL4(3) = .8
      FL4(4) = TINIT
      L4 = 12
C
C    CONSTANT TEMPERATURES
C
      SET2(1) = THOTIN(0,5)
      SET2(2) = TCLDIN(0,6)
C
C    READY FOR INPUT SET 4
C
C    NODE 1
C
      KCON(1,1) = 514
      KCON(1,2) = 1504
      KCON(1,3) = 1514
      KCON(1,4) = 2504
      KCON(1,5) = 3015
      COEF(1,5) = VALK1
      DO 50 I = 1,4
   50 COEF(1,I) = VALK3
C
C    NODES 2 TO 50
C
      DO 75 I = 2,50
      J = I + 50
      K = 151 - I
      L = 150 + I
      M = 251 - I
      N = I - 1
      KCON(I,1) = 10*N + 5
      KCON(I,2) = 10*J + 4
      KCON(I,3) = 10*K + 4
      KCON(I,4) = 10*L + 4
      KCON(I,5) = 10*M + 4
```

```
      COEF(I,1) = VALK1
      DO 80  II = 2,5
      COEF(I,II) = VALK3
   80 CONTINUE
   75 CONTINUE
C
C  NODE 51
C
      KCON(51,1) = 3025
      KCON(51,2) = 14
      COEF(51,1) = VALK2
      COEF(51,2) = VALK3
C
C   NODES 52 TO 250
C
      DO 120  I = 52,250
      K = I - 1
      IF(I.GT.100) GO TO 122
      J = I - 50
      GO TO 135
  122 IF(I.GT.150) GO TO 124
      L = I - 100
      M = 2*L - 1
      N = M + 50
      J = I - N
      GO TO 135
  124 IF(I.GT.200) GO TO 126
      J = I - 150
      GO TO 135
  126 CONTINUE
      L = I - 200
      M = 2*L - 1
      N = M + 150
      J = I - N
  135 KCON(I,1) = 10*J + 4
      KCON(I,2) = 10*K + 5
      COEF(I,1) = VALK3
  120 COEF(I,2) = VALK2
C
C   END OF DATA SETUP
C   NOW CREATE INPUT FOR ANALYZER
C
      WRITE(2,919) TITLE(P,1)
  919 FORMAT(1X,A25)
      WRITE(2,908)(L1(I),I=1,8)
  908 FORMAT(9I4)
      WRITE(2,908)(L2(I),I=1,3)
      WRITE(2,908)(L3(I),I=1,6)
```

```
      WRITE(2,911) FL4(1),FL4(2),L4,FL4(3),FL4(4)
  911 FORMAT(F8.3,F8.5,I8,2F8.1)
      WRITE(2,912) SET2(1),SET2(2)
  912 FORMAT(2F8.0)
C
      DO 200  I = 1,50
      WRITE(2,913)(KCON(I,J),J=1,5)
  913 FORMAT(9I8)
      WRITE(2,914)(COEF(I,J),J=1,5)
  914 FORMAT(5F8.4)
  200 CONTINUE
C
      DO 250 I=51,250
      WRITE(2,913) KCON(I,1),KCON(I,2)
      WRITE(2,914) COEF(I,1),COEF(I,2)
  250 CONTINUE
C
    7 CONTINUE
      STOP
      END
/*
//LKED.SYSLMOD DD DISP=SHR,DSNAME=MSS.S2323.LOAD
//LKED.SYSIN DD *
  NAME NTU14(R)
/*
//
```

APPENDIX H

NTU14BL LIBRARY BATCH PROGRAM

```
//QHN01 JOB (2323,0267),'NTU14bL',CLASS=P
//*MAIN   ORG=NPGVM1.2323P
//*FORMAT PR,DDNAME=,DEST=LOCAL
//   EXEC FORTVG,PROG=NTU14,LIB='MSS.S2323.LOAD'
//GO.TVSSIA    DD DISP=(NEW,PASS),DSN=&TVSSIA,
//             UNIT=SYSDA,SPACE=(CYL,(2,2)),
//             DCB=(RECFM=FB,LRECL=80,BLKSIZE=6160)
//GO.TVSSIB    DD DISP=(NEW,PASS),DSN=&TVSSIB,
//             UNIT=SYSDA,SPACE=(CYL,(2,2)),
//             DCB=(RECFM=FB,LRECL=80,BLKSIZE=6160)
//GO.TVSSIC    DD DISP=(NEW,PASS),DSN=&TVSSIC,
//             UNIT=SYSDA,SPACE=(CYL,(2,2)),
//             DCB=(RECFM=FB,LRECL=80,BLKSIZE=6160)
//GO.TVSSID    DD DISP=(NEW,PASS),DSN=&TVSSID,
//             UNIT=SYSDA,SPACE=(CYL,(2,2)),
//             DCB=(RECFM=FB,LRECL=80,BLKSIZE=6160)
//GO.TVSSIE    DD DISP=(NEW,PASS),DSN=&TVSSIE,
//             UNIT=SYSDA,SPACE=(CYL,(2,2)),
//             DCB=(RECFM=FB,LRECL=80,BLKSIZE=6160)
//GO.TVSSIF    DD DISP=(NEW,PASS),DSN=&TVSSIF,
//             UNIT=SYSDA,SPACE=(CYL,(2,2)),
//             DCB=(RECFM=FB,LRECL=80,BLKSIZE=6160)
//GO.TVSSIG    DD DISP=(NEW,PASS),DSN=&TVSSIG,
//             UNIT=SYSDA,SPACE=(CYL,(2,2)),
//             DCB=(RECFM=FB,LRECL=80,BLKSIZE=6160)
//GO.TVSSIH    DD DISP=(NEW,PASS),DSN=&TVSSIH,
//             UNIT=SYSDA,SPACE=(CYL,(2,2)),
//             DCB=(RECFM=FB,LRECL=80,BLKSIZE=6160)
//GO.TVSSII    DD DISP=(NEW,PASS),DSN=&TVSSII,
//             UNIT=SYSDA,SPACE=(CYL,(2,2)),
//             DCB=(RECFM=FB,LRECL=80,BLKSIZE=6160)
//GO.TVSSIJ    DD DISP=(NEW,PASS),DSN=&TVSSIJ,
//             UNIT=SYSDA,SPACE=(CYL,(2,2)),
//             DCB=(RECFM=FB,LRECL=80,BLKSIZE=6160)
//GO.TVSSIK    DD DISP=(NEW,PASS),DSN=&TVSSIK,
//             UNIT=SYSDA,SPACE=(CYL,(2,2)),
//             DCB=(RECFM=FB,LRECL=80,BLKSIZE=6160)
//GO.FT01F001 DD *
     250.        25.0   0.08333       15.       200.       100.
NTU=0.05 R=0.10 NTU14     TVSSIA
     250.        25.0   0.41667       15.       200.       100.
NTU=0.25 R=0.10 NTU14     TVSSIB
```

```
         250.        25.0    0.83333         15.        200.        100.
NTU=0.50 R=0.10 NTU14        TVSSIC
         250.        25.0    1.25000         15.        200.        100.
NTU=0.75 R=0.10 NTU14        TVSSID
         250.        25.0    1.66667         15.        200.        100.
NTU=1.00 R=0.10 NTU14        TVSSIE
         250.        25.0    2.08333         15.        200.        100.
NTU=1.25 R=0.10 NTU14        TVSSIF
         250.        25.0    2.50000         15.        200.        100.
NTU=1.50 R=0.10 NTU14        TVSSIG
         250.        25.0    3.33333         15.        200.        100.
NTU=2.00 R=0.10 NTU14        TVSSIH
         250.        25.0    4.16667         15.        200.        100.
NTU=2.50 R=0.10 NTU14        TVSSII
         250.        25.0    5.00000         15.        200.        100.
NTU=3.00 R=0.10 NTU14        TVSSIJ
         250.        25.0    5.41667         15.        200.        100.
NTU=3.25 R=0.10 NTU14        TVSSIK
/*
//STEPA   EXEC FORTVG,PROG=TVCOUNT,LIB='MSS.S2323.LOADLIB'
//GO.FT01F001 DD UNIT=SYSDA,SPACE=(CYL,(1,1))
//GO.FT02F001 DD UNIT=SYSDA,SPACE=(CYL,(1,1))
//GO.FT09F001 DD UNIT=SYSDA,SPACE=(CYL,(1,1))
//GO.FT10F001 DD UNIT=SYSDA,SPACE=(CYL,(1,1))
//GO.FT03F001 DD DUMMY
//GO.FT04F001 DD DISP=(OLD,DELETE),UNIT=SYSDA,DSN=&TVSSIA
//GO.FT08F001 DD SYSOUT=A,DCB=RECFM=FBA
//STEPB   EXEC FORTVG,PROG=TVCOUNT,LIB='MSS.S2323.LOADLIB'
//GO.FT01F001 DD UNIT=SYSDA,SPACE=(CYL,(1,1))
//GO.FT02F001 DD UNIT=SYSDA,SPACE=(CYL,(1,1))
//GO.FT09F001 DD UNIT=SYSDA,SPACE=(CYL,(1,1))
//GO.FT10F001 DD UNIT=SYSDA,SPACE=(CYL,(1,1))
//GO.FT03F001 DD DUMMY
//GO.FT04F001 DD DISP=(OLD,DELETE),UNIT=SYSDA,DSN=&TVSSIB
//GO.FT08F001 DD SYSOUT=A,DCB=RECFM=FBA
//STEPC   EXEC FORTVG,PROG=TVCOUNT,LIB='MSS.S2323.LOADLIB'
//GO.FT01F001 DD UNIT=SYSDA,SPACE=(CYL,(1,1))
//GO.FT02F001 DD UNIT=SYSDA,SPACE=(CYL,(1,1))
//GO.FT09F001 DD UNIT=SYSDA,SPACE=(CYL,(1,1))
//GO.FT10F001 DD UNIT=SYSDA,SPACE=(CYL,(1,1))
//GO.FT03F001 DD DUMMY
//GO.FT04F001 DD DISP=(OLD,DELETE),UNIT=SYSDA,DSN=&TVSSIC
//GO.FT08F001 DD SYSOUT=A,DCB=RECFM=FBA
//STEPD   EXEC FORTVG,PROG=TVCOUNT,LIB='MSS.S2323.LOADLIB'
//GO.FT01F001 DD UNIT=SYSDA,SPACE=(CYL,(1,1))
//GO.FT02F001 DD UNIT=SYSDA,SPACE=(CYL,(1,1))
//GO.FT09F001 DD UNIT=SYSDA,SPACE=(CYL,(1,1))
//GO.FT10F001 DD UNIT=SYSDA,SPACE=(CYL,(1,1))
```

```
//GO.FT03F001 DD DUMMY
//GO.FT04F001 DD DISP=(OLD,DELETE),UNIT=SYSDA,DSN=&TVSSID
//GO.FT08F001 DD SYSOUT=A,DCB=RECFM=FBA
//STEPE   EXEC FORTVG,PROG=TVCOUNT,LIB='MSS.S2323.LOADLIB'
//GO.FT01F001 DD UNIT=SYSDA,SPACE=(CYL,(1,1))
//GO.FT02F001 DD UNIT=SYSDA,SPACE=(CYL,(1,1))
//GO.FT09F001 DD UNIT=SYSDA,SPACE=(CYL,(1,1))
//GO.FT10F001 DD UNIT=SYSDA,SPACE=(CYL,(1,1))
//GO.FT03F001 DD DUMMY
//GO.FT04F001 DD DISP=(OLD,DELETE),UNIT=SYSDA,DSN=&TVSSIE
//GO.FT08F001 DD SYSOUT=A,DCB=RECFM=FBA
//STEPF   EXEC FORTVG,PROG=TVCOUNT,LIB='MSS.S2323.LOADLIB'
//GO.FT01F001 DD UNIT=SYSDA,SPACE=(CYL,(1,1))
//GO.FT02F001 DD UNIT=SYSDA,SPACE=(CYL,(1,1))
//GO.FT09F001 DD UNIT=SYSDA,SPACE=(CYL,(1,1))
//GO.FT10F001 DD UNIT=SYSDA,SPACE=(CYL,(1,1))
//GO.FT03F001 DD DUMMY
//GO.FT04F001 DD DISP=(OLD,DELETE),UNIT=SYSDA,DSN=&TVSSIF
//GO.FT08F001 DD SYSOUT=A,DCB=RECFM=FBA
//STEPG   EXEC FORTVG,PROG=TVCOUNT,LIB='MSS.S2323.LOADLIB'
//GO.FT01F001 DD UNIT=SYSDA,SPACE=(CYL,(1,1))
//GO.FT02F001 DD UNIT=SYSDA,SPACE=(CYL,(1,1))
//GO.FT09F001 DD UNIT=SYSDA,SPACE=(CYL,(1,1))
//GO.FT10F001 DD UNIT=SYSDA,SPACE=(CYL,(1,1))
//GO.FT03F001 DD DUMMY
//GO.FT04F001 DD DISP=(OLD,DELETE),UNIT=SYSDA,DSN=&TVSSIG
//GO.FT08F001 DD SYSOUT=A,DCB=RECFM=FBA
//STEPH   EXEC FORTVG,PROG=TVCOUNT,LIB='MSS.S2323.LOADLIB'
//GO.FT01F001 DD UNIT=SYSDA,SPACE=(CYL,(1,1))
//GO.FT02F001 DD UNIT=SYSDA,SPACE=(CYL,(1,1))
//GO.FT09F001 DD UNIT=SYSDA,SPACE=(CYL,(1,1))
//GO.FT10F001 DD UNIT=SYSDA,SPACE=(CYL,(1,1))
//GO.FT03F001 DD DUMMY
//GO.FT04F001 DD DISP=(OLD,DELETE),UNIT=SYSDA,DSN=&TVSSIH
//GO.FT08F001 DD SYSOUT=A,DCB=RECFM=FBA
//STEPI   EXEC FORTVG,PROG=TVCOUNT,LIB='MSS.S2323.LOADLIB'
//GO.FT01F001 DD UNIT=SYSDA,SPACE=(CYL,(1,1))
//GO.FT02F001 DD UNIT=SYSDA,SPACE=(CYL,(1,1))
//GO.FT09F001 DD UNIT=SYSDA,SPACE=(CYL,(1,1))
//GO.FT10F001 DD UNIT=SYSDA,SPACE=(CYL,(1,1))
//GO.FT03F001 DD DUMMY
//GO.FT04F001 DD DISP=(OLD,DELETE),UNIT=SYSDA,DSN=&TVSSII
//GO.FT08F001 DD SYSOUT=A,DCB=RECFM=FBA
//STEPJ   EXEC FORTVG,PROG=TVCOUNT,LIB='MSS.S2323.LOADLIB'
//GO.FT01F001 DD UNIT=SYSDA,SPACE=(CYL,(1,1))
//GO.FT02F001 DD UNIT=SYSDA,SPACE=(CYL,(1,1))
//GO.FT09F001 DD UNIT=SYSDA,SPACE=(CYL,(1,1))
//GO.FT10F001 DD UNIT=SYSDA,SPACE=(CYL,(1,1))
```

```
//GO.FT03F001 DD DUMMY
//GO.FT04F001 DD DISP=(OLD,DELETE),UNIT=SYSDA,DSN=&TVSSIJ
//GO.FT08F001 DD SYSOUT=A,DCB=RECFM=FBA
//STEPK   EXEC FORTVG,PROG=TVCOUNT,LIB='MSS.S2323.LOADLIB'
//GO.FT01F001 DD UNIT=SYSDA,SPACE=(CYL,(1,1))
//GO.FT02F001 DD UNIT=SYSDA,SPACE=(CYL,(1,1))
//GO.FT09F001 DD UNIT=SYSDA,SPACE=(CYL,(1,1))
//GO.FT10F001 DD UNIT=SYSDA,SPACE=(CYL,(1,1))
//GO.FT03F001 DD DUMMY
//GO.FT04F001 DD DISP=(OLD,DELETE),UNIT=SYSDA,DSN=&TVSSIK
//GO.FT08F001 DD SYSOUT=A,DCB=RECFM=FBA
//
```

APPENDIX I

MODIFIED NTU32C PROGRAM NTU32CC USED TO RUN ON BATCH SYSTEM

```
//OHARE   JOB (2323,0267),'NTU32CC',CLASS=G
//*MAIN   ORG=NPGVM1.2323P,SYSTEM=SY2
//   EXEC FORTVCL,PARM.LKED='LIST.MAP'
//FORT.SYSIN DD *
C          THIS IS PROGRAM NTU32C
C
C
C          IT GENERATES AN INPUT FILE FOR THERMAL ANALYZER TO OBTAIN
C          1-3 EFFECTIVENESS-NTU RELATIONSHIP THAT IS NOT AVAILABLE IN
C          OPEN LITERATURE (2C MEANING ONE PARALLEL PASS AND TWO
C          COUNTERFLOW PASSES).
C
      INTEGER O,P
      DIMENSION COEF(200,5),KCON(200,5),L1(8),L2(3),L3(6),SET2(2),FL4(4)
     1,CHOT(22,22),CCLD(22,22),U(22,22),SURFTO(22,22),THOTIN(22,22),
     2TCLDIN(22,22),TITLE(22,22),FNAME(22,22)
C
      CHARACTER *25 TITLE
      CHARACTER *25 FNAME
C
      DO 7 O=1,21,2
      P=O+1
      READ(1,900) CHOT(O,1),CCLD(O,2),U(O,3),SURFTO(O,4),THOTIN(O,5),TCL
     1DIN(O,6)
  900 FORMAT(2F10.0,F10.5,3F10.0)
      READ(1,918) TITLE(P,1),FNAME(P,2)
  918 FORMAT(2A25)
C
C     OPEN OUTPUT FILE
C
      OPEN(2,FILE=FNAME(P,2),FORM='FORMATTED')
C
      VALK1 = CHOT(O,1)
      VALK2 = CCLD(O,2)
      VALK3 = U(O,3)*SURFTO(O,4)/150.
      TINIT = 125.
C
C            FRONT END
C
      L1(1) = 200
      L1(2) = 2
      DO 10  I=3,8
```

```
   10 L1(I) = 0
C
      DO 20  I=1,3
   20 L2(I) = 0
C
      L3(1) = 300
      L3(2) = 50
      L3(3) = 6
      L3(4) = 2
      L3(5) = 4
      L3(6) = 6
C
      FL4(1) = .05
      FL4(2) = .66667
      FL4(3) = .8
      FL4(4) = TINIT
      L4 = 12
C
C    CONSTANT TEMPERATURES
C
      SET2(1) = THOTIN(0,5)
      SET2(2) = TCLDIN(0,6)
C
C    READY FOR INPUT SET 4
C
C    NODE 1
C
      KCON(1,1) = 1004
      KCON(1,2) = 1014
      KCON(1,3) = 2004
      KCON(1,4) = 3015
      COEF(1,4) = VALK1
      DO 50 I = 1,3
   50 COEF(1,I) = VALK3
C
C    NODES 2 TO 50
C
      DO 75 I = 2,50
      J = 101 - I
      K = 100 + I
      L = 201 - I
      N = I - 1
      KCON(I,1) = 10*N + 5
      KCON(I,2) = 10*J + 4
      KCON(I,3) = 10*K + 4
      KCON(I,4) = 10*L + 4
      COEF(I,1) = VALK1
      DO 80  II = 2,4
```

```
      COEF(I,II) = VALK3
   80 CONTINUE
   75 CONTINUE
C
C  NODE 51
C
      KCON(51,1) = 3025
      KCON(51,2) = 504
      COEF(51,1) = VALK2
      COEF(51,2) = VALK3
C
C   NODES 52 TO 200
C
      DO 120  I = 52,200
      K = I - 1
      IF(I.GT.100) GO TO 122
      L = I - 50
      M = 2*L - 1
      J = I - M
      GO TO 135
  122 IF(I.GT.150) GO TO 124
      J = I - 100
      GO TO 135
  124 L = I - 150
      M = 2*L - 1
      J = I - M - 100
  135 KCON(I,1) = 10*J + 4
      KCON(I,2) = 10*K + 5
      COEF(I,1) = VALK3
  120 COEF(I,2) = VALK2
C
C   END OF DATA SETUP
C   NOW CREATE INPUT FOR ANALYZER
C
      WRITE(2,919) TITLE(P,1)
  919 FORMAT(1X,A25)
      WRITE(2,908)(L1(I),I=1,8)
  908 FORMAT(9I4)
      WRITE(2,908)(L2(I),I=1,3)
      WRITE(2,908)(L3(I),I=1,6)
      WRITE(2,911) FL4(1),FL4(2),L4,FL4(3),FL4(4)
  911 FORMAT(F8.3,F8.5,I8,F8.5,F8.2)
      WRITE(2,912) SET2(1),SET2(2)
  912 FORMAT(2F8.0)
C
      DO 200  I = 1,50
      WRITE(2,913)(KCON(I,J),J=1,4)
  913 FORMAT(9I8)
```

```
      WRITE(2,914)(COEF(I,J),J=1,4)
  914 FORMAT(4F8.4)
  200 CONTINUE
C
      DO 250 I=51,200
      WRITE(2,913) KCON(I,1),KCON(I,2)
      WRITE(2,914) COEF(I,1),COEF(I,2)
  250 CONTINUE
C
    7 CONTINUE
      STOP
      END
/*
//LKED.SYSLMOD DD DISP=SHR,DSNAME=MSS.S2323.LOADLIB
//LKED.SYSIN DD *
  NAME COUNTER(R)
/*
//
```

APPENDIX J

NTU32CL LIBRARY BATCH PROGRAM

```
//QH4CL JOB (2323,0267),'NTU32CL',CLASS=P
//*MAIN  ORG=NPGVM1.2323P
//*FORMAT PR,DDNAME=,DEST=LOCAL
//  EXEC FORTVG,PROG=COUNTER,LIB='MSS.S2323.LOADLIB'
//GO.TVSSIA    DD DISP=(NEW,PASS),DSN=&TVSSIA,
//             UNIT=SYSDA,SPACE=(CYL,(2,2)),
//             DCB=(RECFM=FB,LRECL=80,BLKSIZE=6160)
//GO.TVSSIB    DD DISP=(NEW,PASS),DSN=&TVSSIB,
//             UNIT=SYSDA,SPACE=(CYL,(2,2)),
//             DCB=(RECFM=FB,LRECL=80,BLKSIZE=6160)
//GO.TVSSIC    DD DISP=(NEW,PASS),DSN=&TVSSIC,
//             UNIT=SYSDA,SPACE=(CYL,(2,2)),
//             DCB=(RECFM=FB,LRECL=80,BLKSIZE=6160)
//GO.TVSSID    DD DISP=(NEW,PASS),DSN=&TVSSID,
//             UNIT=SYSDA,SPACE=(CYL,(2,2)),
//             DCB=(RECFM=FB,LRECL=80,BLKSIZE=6160)
//GO.TVSSIE    DD DISP=(NEW,PASS),DSN=&TVSSIE,
//             UNIT=SYSDA,SPACE=(CYL,(2,2)),
//             DCB=(RECFM=FB,LRECL=80,BLKSIZE=6160)
//GO.TVSSIF    DD DISP=(NEW,PASS),DSN=&TVSSIF,
//             UNIT=SYSDA,SPACE=(CYL,(2,2)),
//             DCB=(RECFM=FB,LRECL=80,BLKSIZE=6160)
//GO.TVSSIG    DD DISP=(NEW,PASS),DSN=&TVSSIG,
//             UNIT=SYSDA,SPACE=(CYL,(2,2)),
//             DCB=(RECFM=FB,LRECL=80,BLKSIZE=6160)
//GO.TVSSIH    DD DISP=(NEW,PASS),DSN=&TVSSIH,
//             UNIT=SYSDA,SPACE=(CYL,(2,2)),
//             DCB=(RECFM=FB,LRECL=80,BLKSIZE=6160)
//GO.TVSSII    DD DISP=(NEW,PASS),DSN=&TVSSII,
//             UNIT=SYSDA,SPACE=(CYL,(2,2)),
//             DCB=(RECFM=FB,LRECL=80,BLKSIZE=6160)
//GO.TVSSIJ    DD DISP=(NEW,PASS),DSN=&TVSSIJ,
//             UNIT=SYSDA,SPACE=(CYL,(2,2)),
//             DCB=(RECFM=FB,LRECL=80,BLKSIZE=6160)
//GO.TVSSIK    DD DISP=(NEW,PASS),DSN=&TVSSIK,
//             UNIT=SYSDA,SPACE=(CYL,(2,2)),
//             DCB=(RECFM=FB,LRECL=80,BLKSIZE=6160)
//GO.FT01F001 DD *
      250.      100.0   0.33333       15.       200.       100.
NTU=0.05 R=0.40 COUNTER  TVSSIA
      250.      100.0   1.66667       15.       200.       100.
NTU=0.25 R=0.40 COUNTER  TVSSIB
```

```
        250.         100.0    3.33333        15.        200.        100.
NTU=0.50 R=0.40 COUNTER   TVSSIC
        250.         100.0    5.00000        15.        200.        100.
NTU=0.75 R=0.40 COUNTER   TVSSID
        250.         100.0    6.66667        15.        200.        100.
NTU=1.00 R=0.40 COUNTER   TVSSIE
        250.         100.0    8.33333        15.        200.        100.
NTU=1.25 R=0.40 COUNTER   TVSSIF
        250.         100.0   10.00000        15.        200.        100.
NTU=1.50 R=0.40 COUNTER   TVSSIG
        250.         100.0   13.33333        15.        200.        100.
NTU=2.00 R=0.40 COUNTER   TVSSIH
        250.         100.0   16.66666        15.        200.        100.
NTU=2.50 R=0.40 COUNTER   TVSSII
        250.         100.0   20.00000        15.        200.        100.
NTU=3.00 R=0.40 COUNTER   TVSSIJ
        250.         100.0   21.66666        15.        200.        100.
NTU=3.25 R=0.40 COUNTER   TVSSIK
/*
//STEPA   EXEC  FORTVG,PROG=TVCOUNT,LIB='MSS.S2323.LOADLIB'
//GO.FT01F001 DD UNIT=SYSDA,SPACE=(CYL,(1,1))
//GO.FT02F001 DD UNIT=SYSDA,SPACE=(CYL,(1,1))
//GO.FT09F001 DD UNIT=SYSDA,SPACE=(CYL,(1,1))
//GO.FT10F001 DD UNIT=SYSDA,SPACE=(CYL,(1,1))
//GO.FT03F001 DD DUMMY
//GO.FT04F001 DD DISP=(OLD,DELETE),UNIT=SYSDA,DSN=&TVSSIA
//GO.FT08F001 DD SYSOUT=A,DCB=RECFM=FBA
//STEPB   EXEC  FORTVG,PROG=TVCOUNT,LIB='MSS.S2323.LOADLIB'
//GO.FT01F001 DD UNIT=SYSDA,SPACE=(CYL,(1,1))
//GO.FT02F001 DD UNIT=SYSDA,SPACE=(CYL,(1,1))
//GO.FT09F001 DD UNIT=SYSDA,SPACE=(CYL,(1,1))
//GO.FT10F001 DD UNIT=SYSDA,SPACE=(CYL,(1,1))
//GO.FT03F001 DD DUMMY
//GO.FT04F001 DD DISP=(OLD,DELETE),UNIT=SYSDA,DSN=&TVSSIB
//GO.FT08F001 DD SYSOUT=A,DCB=RECFM=FBA
//STEPC   EXEC  FORTVG,PROG=TVCOUNT,LIB='MSS.S2323.LOADLIB'
//GO.FT01F001 DD UNIT=SYSDA,SPACE=(CYL,(1,1))
//GO.FT02F001 DD UNIT=SYSDA,SPACE=(CYL,(1,1))
//GO.FT09F001 DD UNIT=SYSDA,SPACE=(CYL,(1,1))
//GO.FT10F001 DD UNIT=SYSDA,SPACE=(CYL,(1,1))
//GO.FT03F001 DD DUMMY
//GO.FT04F001 DD DISP=(OLD,DELETE),UNIT=SYSDA,DSN=&TVSSIC
//GO.FT08F001 DD SYSOUT=A,DCB=RECFM=FBA
//STEPD   EXEC  FORTVG,PROG=TVCOUNT,LIB='MSS.S2323.LOADLIB'
//GO.FT01F001 DD UNIT=SYSDA,SPACE=(CYL,(1,1))
//GO.FT02F001 DD UNIT=SYSDA,SPACE=(CYL,(1,1))
//GO.FT09F001 DD UNIT=SYSDA,SPACE=(CYL,(1,1))
//GO.FT10F001 DD UNIT=SYSDA,SPACE=(CYL,(1,1))
```

```
//GO.FT03F001 DD DUMMY
//GO.FT04F001 DD DISP=(OLD,DELETE),UNIT=SYSDA,DSN=&TVSSID
//GO.FT08F001 DD SYSOUT=A,DCB=RECFM=FBA
//STEPE   EXEC FORTVG,PROG=TVCOUNT,LIB='MSS.S2323.LOADLIB'
//GO.FT01F001 DD UNIT=SYSDA,SPACE=(CYL,(1,1))
//GO.FT02F001 DD UNIT=SYSDA,SPACE=(CYL,(1,1))
//GO.FT09F001 DD UNIT=SYSDA,SPACE=(CYL,(1,1))
//GO.FT10F001 DD UNIT=SYSDA,SPACE=(CYL,(1,1))
//GO.FT03F001 DD DUMMY
//GO.FT04F001 DD DISP=(OLD,DELETE),UNIT=SYSDA,DSN=&TVSSIE
//GO.FT08F001 DD SYSOUT=A,DCB=RECFM=FBA
//STEPF   EXEC FORTVG,PROG=TVCOUNT,LIB='MSS.S2323.LOADLIB'
//GO.FT01F001 DD UNIT=SYSDA,SPACE=(CYL,(1,1))
//GO.FT02F001 DD UNIT=SYSDA,SPACE=(CYL,(1,1))
//GO.FT09F001 DD UNIT=SYSDA,SPACE=(CYL,(1,1))
//GO.FT10F001 DD UNIT=SYSDA,SPACE=(CYL,(1,1))
//GO.FT03F001 DD DUMMY
//GO.FT04F001 DD DISP=(OLD,DELETE),UNIT=SYSDA,DSN=&TVSSIF
//GO.FT08F001 DD SYSOUT=A,DCB=RECFM=FBA
//STEPG   EXEC FORTVG,PROG=TVCOUNT,LIB='MSS.S2323.LOADLIB'
//GO.FT01F001 DD UNIT=SYSDA,SPACE=(CYL,(1,1))
//GO.FT02F001 DD UNIT=SYSDA,SPACE=(CYL,(1,1))
//GO.FT09F001 DD UNIT=SYSDA,SPACE=(CYL,(1,1))
//GO.FT10F001 DD UNIT=SYSDA,SPACE=(CYL,(1,1))
//GO.FT03F001 DD DUMMY
//GO.FT04F001 DD DISP=(OLD,DELETE),UNIT=SYSDA,DSN=&TVSSIG
//GO.FT08F001 DD SYSOUT=A,DCB=RECFM=FBA
//STEPH   EXEC FORTVG,PROG=TVCOUNT,LIB='MSS.S2323.LOADLIB'
//GO.FT01F001 DD UNIT=SYSDA,SPACE=(CYL,(1,1))
//GO.FT02F001 DD UNIT=SYSDA,SPACE=(CYL,(1,1))
//GO.FT09F001 DD UNIT=SYSDA,SPACE=(CYL,(1,1))
//GO.FT10F001 DD UNIT=SYSDA,SPACE=(CYL,(1,1))
//GO.FT03F001 DD DUMMY
//GO.FT04F001 DD DISP=(OLD,DELETE),UNIT=SYSDA,DSN=&TVSSIH
//GO.FT08F001 DD SYSOUT=A,DCB=RECFM=FBA
//STEPI   EXEC FORTVG,PROG=TVCOUNT,LIB='MSS.S2323.LOADLIB'
//GO.FT01F001 DD UNIT=SYSDA,SPACE=(CYL,(1,1))
//GO.FT02F001 DD UNIT=SYSDA,SPACE=(CYL,(1,1))
//GO.FT09F001 DD UNIT=SYSDA,SPACE=(CYL,(1,1))
//GO.FT10F001 DD UNIT=SYSDA,SPACE=(CYL,(1,1))
//GO.FT03F001 DD DUMMY
//GO.FT04F001 DD DISP=(OLD,DELETE),UNIT=SYSDA,DSN=&TVSSII
//GO.FT08F001 DD SYSOUT=A,DCB=RECFM=FBA
//STEPJ   EXEC FORTVG,PROG=TVCOUNT,LIB='MSS.S2323.LOADLIB'
//GO.FT01F001 DD UNIT=SYSDA,SPACE=(CYL,(1,1))
//GO.FT02F001 DD UNIT=SYSDA,SPACE=(CYL,(1,1))
//GO.FT09F001 DD UNIT=SYSDA,SPACE=(CYL,(1,1))
//GO.FT10F001 DD UNIT=SYSDA,SPACE=(CYL,(1,1))
```

```
//GO.FT03F001 DD DUMMY
//GO.FT04F001 DD DISP=(OLD,DELETE),UNIT=SYSDA,DSN=&TVSSIJ
//GO.FT08F001 DD SYSOUT=A,DCB=RECFM=FBA
//STEPK   EXEC FORTVG,PROG=TVCOUNT,LIB='MSS.S2323.LOADLIB'
//GO.FT01F001 DD UNIT=SYSDA,SPACE=(CYL,(1,1))
//GO.FT02F001 DD UNIT=SYSDA,SPACE=(CYL,(1,1))
//GO.FT09F001 DD UNIT=SYSDA,SPACE=(CYL,(1,1))
//GO.FT10F001 DD UNIT=SYSDA,SPACE=(CYL,(1,1))
//GO.FT03F001 DD DUMMY
//GO.FT04F001 DD DISP=(OLD,DELETE),UNIT=SYSDA,DSN=&TVSSIK
//GO.FT08F001 DD SYSOUT=A,DCB=RECFM=FBA
//
```

APPENDIX K

MODIFIED NTU32P PROGRAM NTU32PC USED TO RUN ON BATCH SYSTEM

```
//OH32PC JOB (2323,0267),'NTU32PC',CLASS=G
//*MAIN  ORG=NPGVM1.2323P,SYSTEM=SY2
//  EXEC FORTVCL,PARM.LKED='LIST,MAP'
//FORT.SYSIN DD *
C          THIS IS PROGRAM NTU32P
C
C
C          IT GENERATES AN INPUT FILE FOR THERMAL ANALYZER TO OBTAIN
C          1-3 EFFECTIVENESS-NTU RELATIONSHIP THAT IS NOT AVAILABLE IN
C          OPEN LITERATURE (2P MEANING TWO PARALLEL PASSES AND ONE
C          COUNTERFLOW PASS).
C
      INTEGER O,P
      DIMENSION COEF(200,5),KCON(200,5),L1(8),L2(3),L3(6),SET2(2),FL4(4)
     1,CHOT(22,22),CCLD(22,22),U(22,22),SURFTO(22,22),THOTIN(22,22),
     2TCLDIN(22,22),TITLE(22,22),FNAME(22,22)
C
      CHARACTER *25 TITLE
      CHARACTER *25 FNAME
C
      DO 7 O=1,21,2
      P=O+1
      READ(1,900) CHOT(O,1),CCLD(O,2),U(O,3),SURFTO(O,4),THOTIN(O,5),TCL
     1DIN(O,6)
  900 FORMAT(2F10.0,F10.5,3F10.0)
      READ(1,918) TITLE(P,1),FNAME(P,2)
  918 FORMAT(2A25)
C
C     OPEN OUTPUT FILE
C
      OPEN(2,FILE=FNAME(P,2),FORM='FORMATTED')
C
      VALK1 = CHOT(O,1)
      VALK2 = CCLD(O,2)
      VALK3 = U(O,3)*SURFTO(O,4)/150.
      TINIT = 125.
C
C            FRONT END
C
      L1(1) = 200
      L1(2) = 2
      DO 10  I=3,8
```

```
   10 L1(I) = 0
C
      DO 20  I=1,3
   20 L2(I) = 0
C
      L3(1) = 300
      L3(2) = 50
      L3(3) = 6
      L3(4) = 2
      L3(5) = 4
      L3(6) = 6
C
      FL4(1) = .05
      FL4(2) = .66667
      FL4(3) = .8
      FL4(4) = TINIT
      L4 = 12
C
C    CONSTANT TEMPERATURES
C
      SET2(1) = THOTIN(0,5)
      SET2(2) = TCLDIN(0,6)
C
C    READY FOR INPUT SET 4
C
C    NODE 1
C
      KCON(1,1) = 514
      KCON(1,2) = 1504
      KCON(1,3) = 1514
      KCON(1,4) = 3015
      COEF(1,4) = VALK1
      DO 50 I = 1,3
   50 COEF(1,I) = VALK3
C
C    NODES 2 TO 50
C
      DO 75 I = 2,50
      J = I + 50
      K = 151 - I
      L = 150 + I
      N = I - 1
      KCON(I,1) = 10*N + 5
      KCON(I,2) = 10*J + 4
      KCON(I,3) = 10*K + 4
      KCON(I,4) = 10*L + 4
      COEF(I,1) = VALK1
      DO 80  II = 2,4
```

```
      COEF(I,II) = VALK3
   80 CONTINUE
   75 CONTINUE
C
C  NODE 51
C
      KCON(51,1) = 3025
      KCON(51,2) = 14
      COEF(51,1) = VALK2
      COEF(51,2) = VALK3
C
C   NODES 52 TO 200
C
      DO 120  I = 52,200
      K = I - 1
      IF(I.GT.100) GO TO 122
      J = I - 50
      GO TO 135
  122 IF(I.GT.150) GO TO 124
      L = I - 100
      M = 2*L - 1
      N = M + 50
      J = I - N
      GO TO 135
  124 J = I - 150
  135 KCON(I,1) = 10*J + 4
      KCON(I,2) = 10*K + 5
      COEF(I,1) = VALK3
  120 COEF(I,2) = VALK2
C
C   END OF DATA SETUP
C   NOW CREATE INPUT FOR ANALYZER
C
      WRITE(2,919) TITLE(P,1)
  919 FORMAT(1X,A25)
      WRITE(2,908)(L1(I),I=1,8)
  908 FORMAT(9I4)
      WRITE(2,908)(L2(I),I=1,3)
      WRITE(2,908)(L3(I),I=1,6)
      WRITE(2,911) FL4(1),FL4(2),L4,FL4(3),FL4(4)
  911 FORMAT(F8.3,F8.5,I8,F8.5,F8.2)
      WRITE(2,912) SET2(1),SET2(2)
  912 FORMAT(2F8.0)
C
      DO 200  I = 1,50
      WRITE(2,913)(KCON(I,J),J=1,4)
  913 FORMAT(9I8)
      WRITE(2,914)(COEF(I,J),J=1,4)
```

```
  914 FORMAT(4F8.4)
  200 CONTINUE
C
      DO 250 I=51,200
      WRITE(2,913) KCON(I,1),KCON(I,2)
      WRITE(2,914) COEF(I,1),COEF(I,2)
  250 CONTINUE
C
  7   CONTINUE
      STOP
      END
/*
//LKED.SYSLMOD DD DISP=SHR,DSNAME=MSS.S2323.LOADLIB
//LKED.SYSIN   DD  *
    NAME PARALLEL(R)
/*
//
```

APPENDIX L

NTU32PL LIBRARY BATCH PROGRAM

```
//QH2PL JOB (2323,0267),'NTU32PL',CLASS=P
//*MAIN  ORG=NPGVM1.2323P
//*FORMAT PR,DDNAME=,DEST=LOCAL
//  EXEC FORTVG,PROG=PARALLEL,LIB='MSS.S2323.LOADLIB'
//GO.TVSSIL    DD DISP=(NEW,PASS),DSN=&TVSSIL,
//             UNIT=SYSDA,SPACE=(CYL,(2,2)),
//             DCB=(RECFM=FB,LRECL=80,BLKSIZE=6160)
//GO.TVSSIM    DD DISP=(NEW,PASS),DSN=&TVSSIM,
//             UNIT=SYSDA,SPACE=(CYL,(2,2)),
//             DCB=(RECFM=FB,LRECL=80,BLKSIZE=6160)
//GO.TVSSIN    DD DISP=(NEW,PASS),DSN=&TVSSIN,
//             UNIT=SYSDA,SPACE=(CYL,(2,2)),
//             DCB=(RECFM=FB,LRECL=80,BLKSIZE=6160)
//GO.TVSSIO    DD DISP=(NEW,PASS),DSN=&TVSSIO,
//             UNIT=SYSDA,SPACE=(CYL,(2,2)),
//             DCB=(RECFM=FB,LRECL=80,BLKSIZE=6160)
//GO.TVSSIP    DD DISP=(NEW,PASS),DSN=&TVSSIP,
//             UNIT=SYSDA,SPACE=(CYL,(2,2)),
//             DCB=(RECFM=FB,LRECL=80,BLKSIZE=6160)
//GO.TVSSIQ    DD DISP=(NEW,PASS),DSN=&TVSSIQ,
//             UNIT=SYSDA,SPACE=(CYL,(2,2)),
//             DCB=(RECFM=FB,LRECL=80,BLKSIZE=6160)
//GO.TVSSIR    DD DISP=(NEW,PASS),DSN=&TVSSIR,
//             UNIT=SYSDA,SPACE=(CYL,(2,2)),
//             DCB=(RECFM=FB,LRECL=80,BLKSIZE=6160)
//GO.TVSSIS    DD DISP=(NEW,PASS),DSN=&TVSSIS,
//             UNIT=SYSDA,SPACE=(CYL,(2,2)),
//             DCB=(RECFM=FB,LRECL=80,BLKSIZE=6160)
//GO.TVSSIT    DD DISP=(NEW,PASS),DSN=&TVSSIT,
//             UNIT=SYSDA,SPACE=(CYL,(2,2)),
//             DCB=(RECFM=FB,LRECL=80,BLKSIZE=6160)
//GO.TVSSIU    DD DISP=(NEW,PASS),DSN=&TVSSIU,
//             UNIT=SYSDA,SPACE=(CYL,(2,2)),
//             DCB=(RECFM=FB,LRECL=80,BLKSIZE=6160)
//GO.TVSSIV    DD DISP=(NEW,PASS),DSN=&TVSSIV,
//             UNIT=SYSDA,SPACE=(CYL,(2,2)),
//             DCB=(RECFM=FB,LRECL=80,BLKSIZE=6160)
//GO.FT01F001 DD *
     250.      150.0    0.50000        15.        200.        100.
NTU=0.05 R=0.60 COUNTER  TVSSIA
     250.      150.0    2.50000        15.        200.        100.
NTU=0.25 R=0.60 COUNTER  TVSSIB
```

```
          250.      150.0    5.00000        15.        200.        100.
NTU=0.50 R=0.60 COUNTER  TVSSIC
          250.      150.0    7.50000        15.        200.        100.
NTU=0.75 R=0.60 COUNTER  TVSSID
          250.      150.0   10.00000        15.        200.        100.
NTU=1.00 R=0.60 COUNTER  TVSSIE
          250.      150.0   12.50000        15.        200.        100.
NTU=1.25 R=0.60 COUNTER  TVSSIF
          250.      150.0   15.00000        15.        200.        100.
NTU=1.50 R=0.60 COUNTER  TVSSIG
          250.      150.0   20.00000        15.        200.        100.
NTU=2.00 R=0.60 COUNTER  TVSSIH
          250.      150.0   25.00000        15.        200.        100.
NTU=2.50 R=0.60 COUNTER  TVSSII
          250.      150.0   30.00000        15.        200.        100.
NTU=3.00 R=0.60 COUNTER  TVSSIJ
          250.      150.0   32.50000        15.        200.        100.
NTU=3.25 R=0.60 COUNTER  TVSSIK
/*
//STEPL   EXEC  FORTVG,PROG=TVCOUNT,LIB='MSS.S2323.LOADLIB'
//GO.FT01F001 DD UNIT=SYSDA,SPACE=(CYL,(1,1))
//GO.FT02F001 DD UNIT=SYSDA,SPACE=(CYL,(1,1))
//GO.FT09F001 DD UNIT=SYSDA,SPACE=(CYL,(1,1))
//GO.FT10F001 DD UNIT=SYSDA,SPACE=(CYL,(1,1))
//GO.FT03F001 DD DUMMY
//GO.FT04F001 DD DISP=(OLD,DELETE),UNIT=SYSDA,DSN=&TVSSIL
//GO.FT08F001 DD SYSOUT=A,DCB=RECFM=FBA
//STEPM   EXEC  FORTVG,PROG=TVCOUNT,LIB='MSS.S2323.LOADLIB'
//GO.FT01F001 DD UNIT=SYSDA,SPACE=(CYL,(1,1))
//GO.FT02F001 DD UNIT=SYSDA,SPACE=(CYL,(1,1))
//GO.FT09F001 DD UNIT=SYSDA,SPACE=(CYL,(1,1))
//GO.FT10F001 DD UNIT=SYSDA,SPACE=(CYL,(1,1))
//GO.FT03F001 DD DUMMY
//GO.FT04F001 DD DISP=(OLD,DELETE),UNIT=SYSDA,DSN=&TVSSIM
//GO.FT08F001 DD SYSOUT=A,DCB=RECFM=FBA
//STEPN   EXEC  FORTVG,PROG=TVCOUNT,LIB='MSS.S2323.LOADLIB'
//GO.FT01F001 DD UNIT=SYSDA,SPACE=(CYL,(1,1))
//GO.FT02F001 DD UNIT=SYSDA,SPACE=(CYL,(1,1))
//GO.FT09F001 DD UNIT=SYSDA,SPACE=(CYL,(1,1))
//GO.FT10F001 DD UNIT=SYSDA,SPACE=(CYL,(1,1))
//GO.FT03F001 DD DUMMY
//GO.FT04F001 DD DISP=(OLD,DELETE),UNIT=SYSDA,DSN=&TVSSIN
//GO.FT08F001 DD SYSOUT=A,DCB=RECFM=FBA
//STEPO   EXEC  FORTVG,PROG=TVCOUNT,LIB='MSS.S2323.LOADLIB'
//GO.FT01F001 DD UNIT=SYSDA,SPACE=(CYL,(1,1))
//GO.FT02F001 DD UNIT=SYSDA,SPACE=(CYL,(1,1))
//GO.FT09F001 DD UNIT=SYSDA,SPACE=(CYL,(1,1))
//GO.FT10F001 DD UNIT=SYSDA,SPACE=(CYL,(1,1))
```

```
//GO.FT03F001 DD DUMMY
//GO.FT04F001 DD DISP=(OLD,DELETE),UNIT=SYSDA,DSN=&TVSSIO
//GO.FT08F001 DD SYSOUT=A,DCB=RECFM=FBA
//STEPP   EXEC FORTVG,PROG=TVCOUNT,LIB='MSS.S2323.LOADLIB'
//GO.FT01F001 DD UNIT=SYSDA,SPACE=(CYL,(1,1))
//GO.FT02F001 DD UNIT=SYSDA,SPACE=(CYL,(1,1))
//GO.FT09F001 DD UNIT=SYSDA,SPACE=(CYL,(1,1))
//GO.FT10F001 DD UNIT=SYSDA,SPACE=(CYL,(1,1))
//GO.FT03F001 DD DUMMY
//GO.FT04F001 DD DISP=(OLD,DELETE),UNIT=SYSDA,DSN=&TVSSIP
//GO.FT08F001 DD SYSOUT=A,DCB=RECFM=FBA
//STEPQ   EXEC FORTVG,PROG=TVCOUNT,LIB='MSS.S2323.LOADLIB'
//GO.FT01F001 DD UNIT=SYSDA,SPACE=(CYL,(1,1))
//GO.FT02F001 DD UNIT=SYSDA,SPACE=(CYL,(1,1))
//GO.FT09F001 DD UNIT=SYSDA,SPACE=(CYL,(1,1))
//GO.FT10F001 DD UNIT=SYSDA,SPACE=(CYL,(1,1))
//GO.FT03F001 DD DUMMY
//GO.FT04F001 DD DISP=(OLD,DELETE),UNIT=SYSDA,DSN=&TVSSIQ
//GO.FT08F001 DD SYSOUT=A,DCB=RECFM=FBA
//STEPR   EXEC FORTVG,PROG=TVCOUNT,LIB='MSS.S2323.LOADLIB'
//GO.FT01F001 DD UNIT=SYSDA,SPACE=(CYL,(1,1))
//GO.FT02F001 DD UNIT=SYSDA,SPACE=(CYL,(1,1))
//GO.FT09F001 DD UNIT=SYSDA,SPACE=(CYL,(1,1))
//GO.FT10F001 DD UNIT=SYSDA,SPACE=(CYL,(1,1))
//GO.FT03F001 DD DUMMY
//GO.FT04F001 DD DISP=(OLD,DELETE),UNIT=SYSDA,DSN=&TVSSIR
//GO.FT08F001 DD SYSOUT=A,DCB=RECFM=FBA
//STEPS   EXEC FORTVG,PROG=TVCOUNT,LIB='MSS.S2323.LOADLIB'
//GO.FT01F001 DD UNIT=SYSDA,SPACE=(CYL,(1,1))
//GO.FT02F001 DD UNIT=SYSDA,SPACE=(CYL,(1,1))
//GO.FT09F001 DD UNIT=SYSDA,SPACE=(CYL,(1,1))
//GO.FT10F001 DD UNIT=SYSDA,SPACE=(CYL,(1,1))
//GO.FT03F001 DD DUMMY
//GO.FT04F001 DD DISP=(OLD,DELETE),UNIT=SYSDA,DSN=&TVSSIS
//GO.FT08F001 DD SYSOUT=A,DCB=RECFM=FBA
//STEPT   EXEC FORTVG,PROG=TVCOUNT,LIB='MSS.S2323.LOADLIB'
//GO.FT01F001 DD UNIT=SYSDA,SPACE=(CYL,(1,1))
//GO.FT02F001 DD UNIT=SYSDA,SPACE=(CYL,(1,1))
//GO.FT09F001 DD UNIT=SYSDA,SPACE=(CYL,(1,1))
//GO.FT10F001 DD UNIT=SYSDA,SPACE=(CYL,(1,1))
//GO.FT03F001 DD DUMMY
//GO.FT04F001 DD DISP=(OLD,DELETE),UNIT=SYSDA,DSN=&TVSSIT
//GO.FT08F001 DD SYSOUT=A,DCB=RECFM=FBA
//STEPU   EXEC FORTVG,PROG=TVCOUNT,LIB='MSS.S2323.LOADLIB'
//GO.FT01F001 DD UNIT=SYSDA,SPACE=(CYL,(1,1))
//GO.FT02F001 DD UNIT=SYSDA,SPACE=(CYL,(1,1))
//GO.FT09F001 DD UNIT=SYSDA,SPACE=(CYL,(1,1))
//GO.FT10F001 DD UNIT=SYSDA,SPACE=(CYL,(1,1))
```

```
//GO.FT03F001 DD DUMMY
//GO.FT04F001 DD DISP=(OLD,DELETE),UNIT=SYSDA,DSN=&TVSSIU
//GO.FT08F001 DD SYSOUT=A,DCB=RECFM=FBA
//STEPV  EXEC FORTVG,PROG=TVCOUNT,LIB='MSS.S2323.LOADLIB'
//GO.FT01F001 DD UNIT=SYSDA,SPACE=(CYL,(1,1))
//GO.FT02F001 DD UNIT=SYSDA,SPACE=(CYL,(1,1))
//GO.FT09F001 DD UNIT=SYSDA,SPACE=(CYL,(1,1))
//GO.FT10F001 DD UNIT=SYSDA,SPACE=(CYL,(1,1))
//GO.FT03F001 DD DUMMY
//GO.FT04F001 DD DISP=(OLD,DELETE),UNIT=SYSDA,DSN=&TVSSIV
//GO.FT08F001 DD SYSOUT=A,DCB=RECFM=FBA
//
```

APPENDIX M

1-3:2C EFFECTIVENESS VS. N_{tu} GRAPHS AT VARIOUS R VALUES

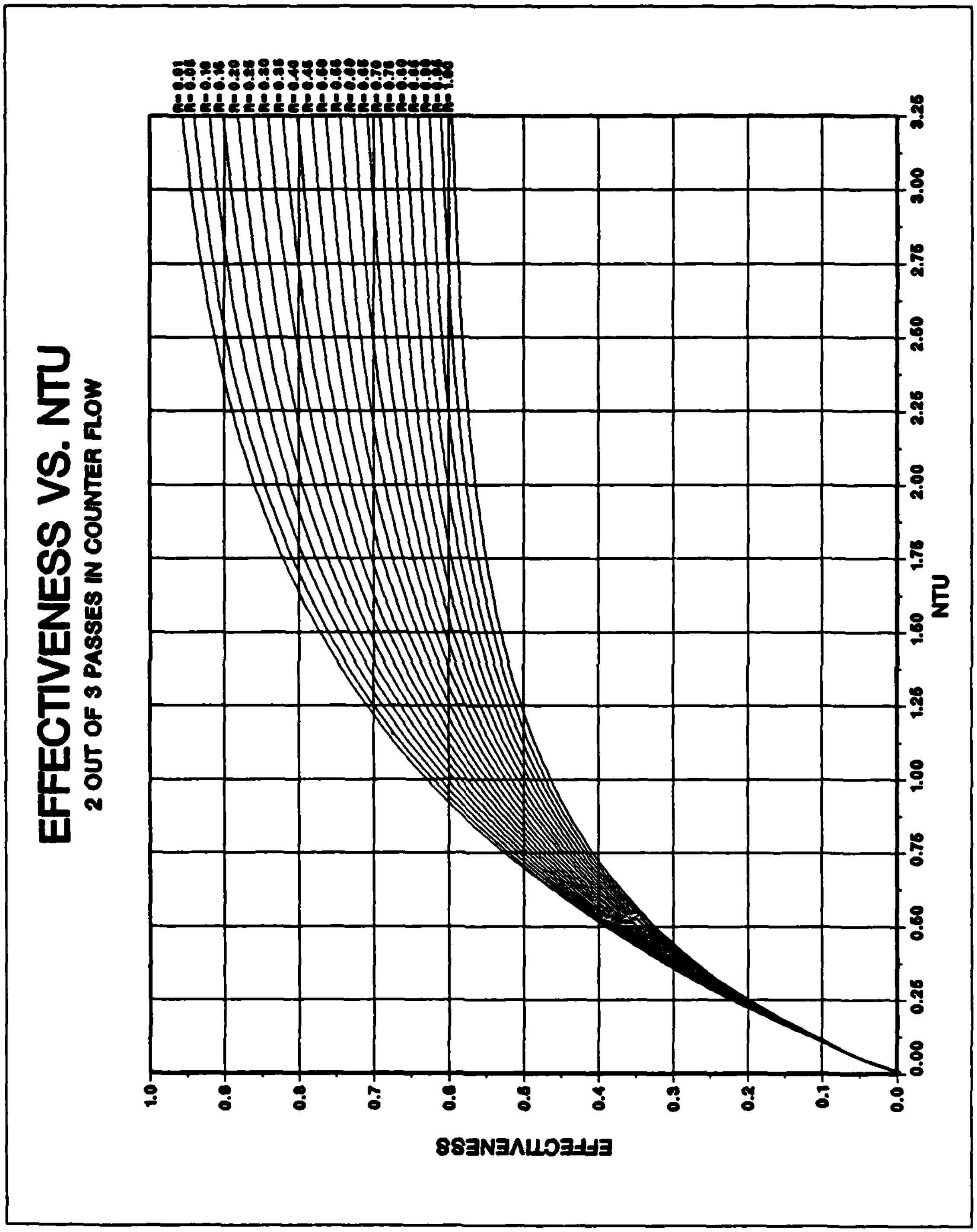


Figure M.1 1-3:2C Effectiveness vs. N_{tu} over Range of R from 0.01 to 1.0

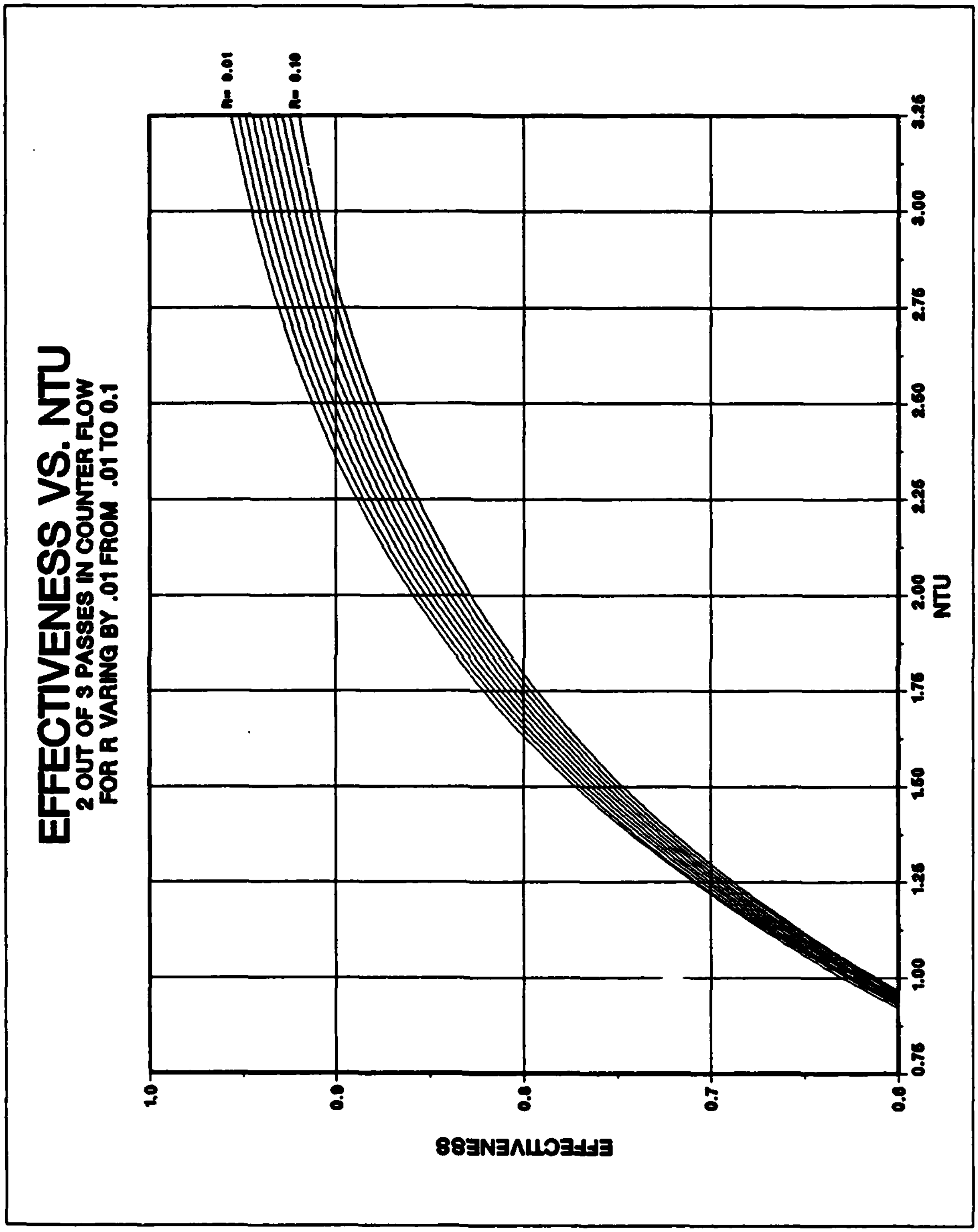


Figure M.2 1-3:2C Effectiveness vs. N_{tu} over Range of R from 0.01 to 0.10

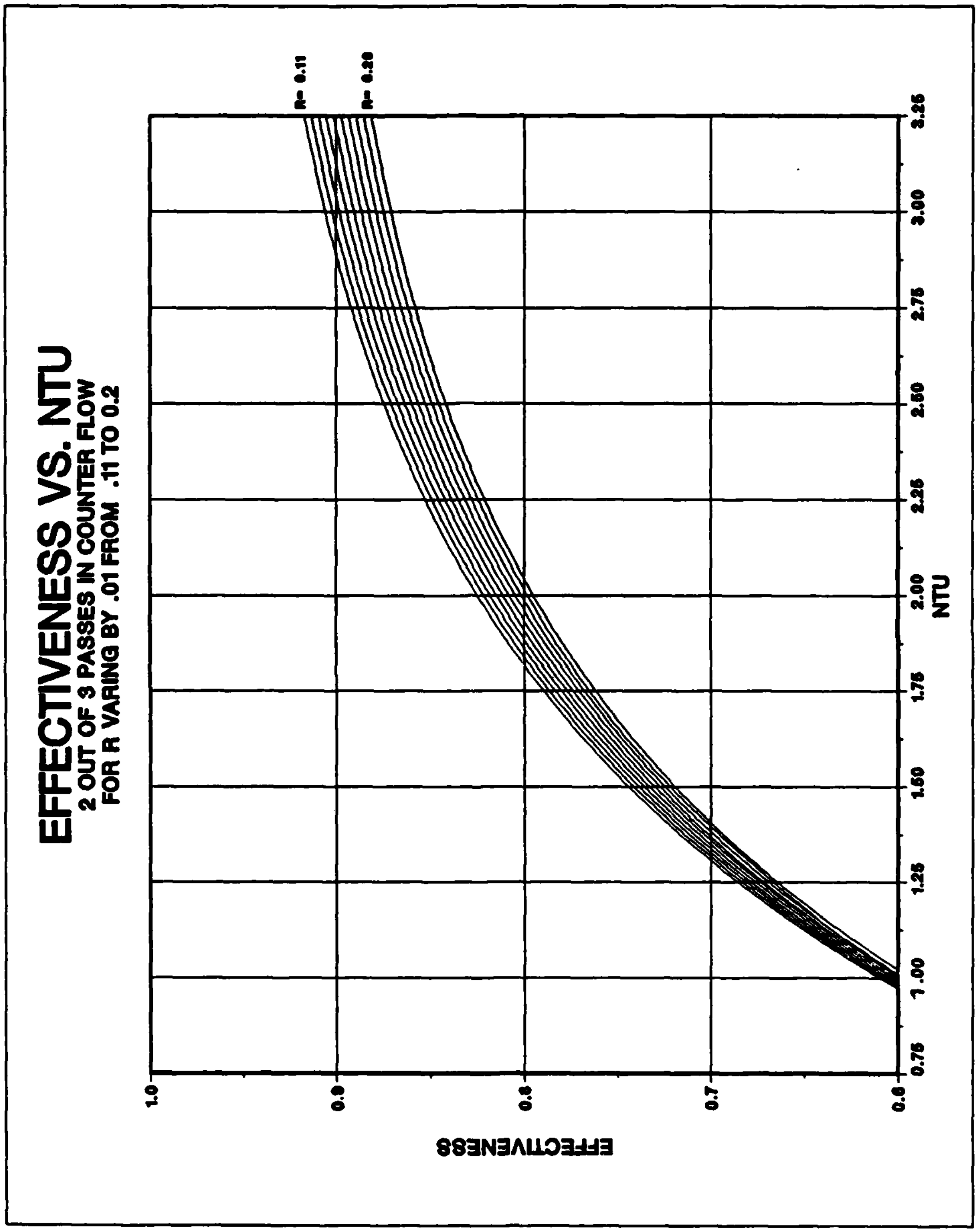


Figure M.3 1-3:2C Effectiveness vs. N_{tu} over Range of R from 0.11 to 0.2

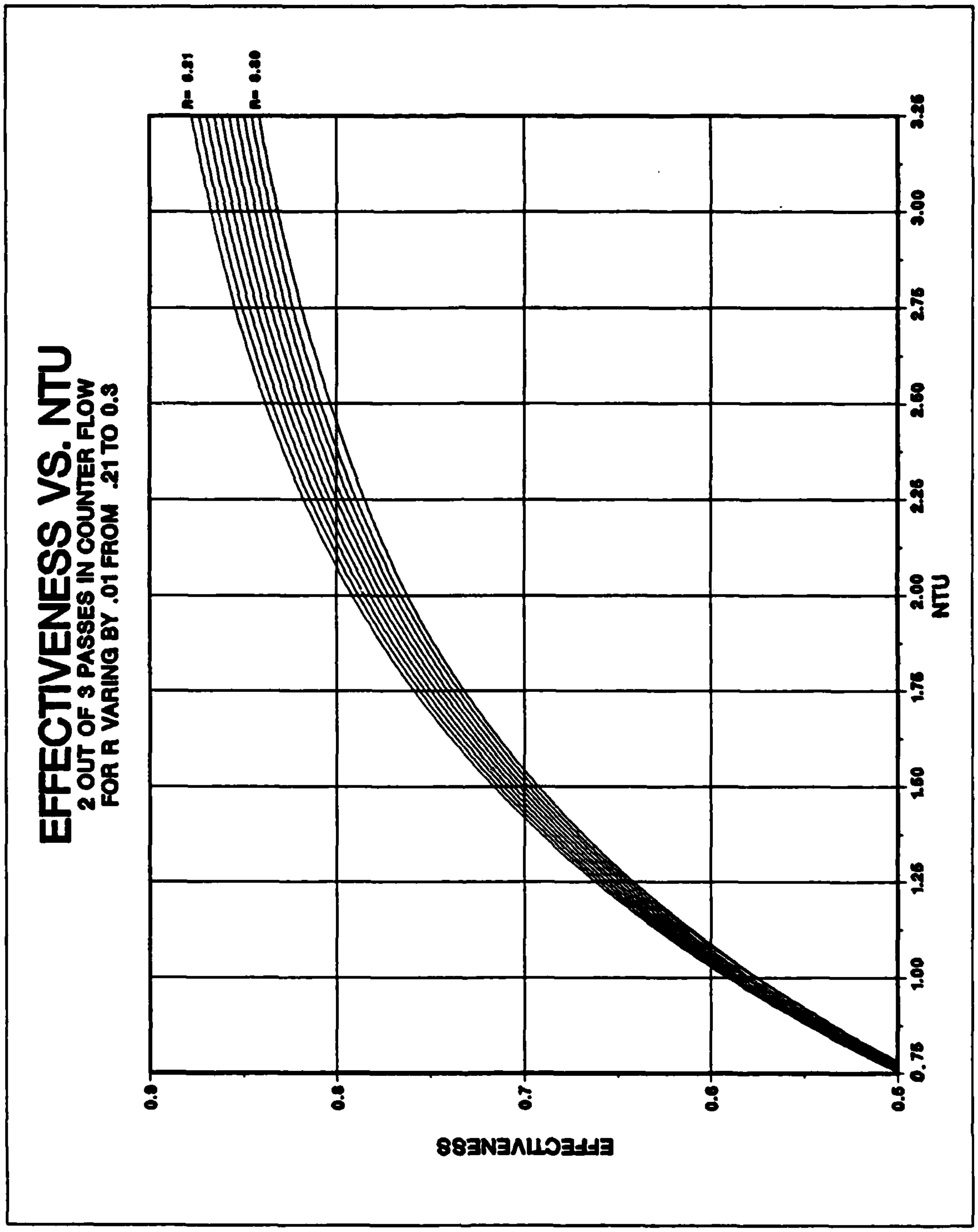


Figure M.4 1-3:2C Effectiveness vs. N_{tu} over Range of R from 0.21 to 0.3

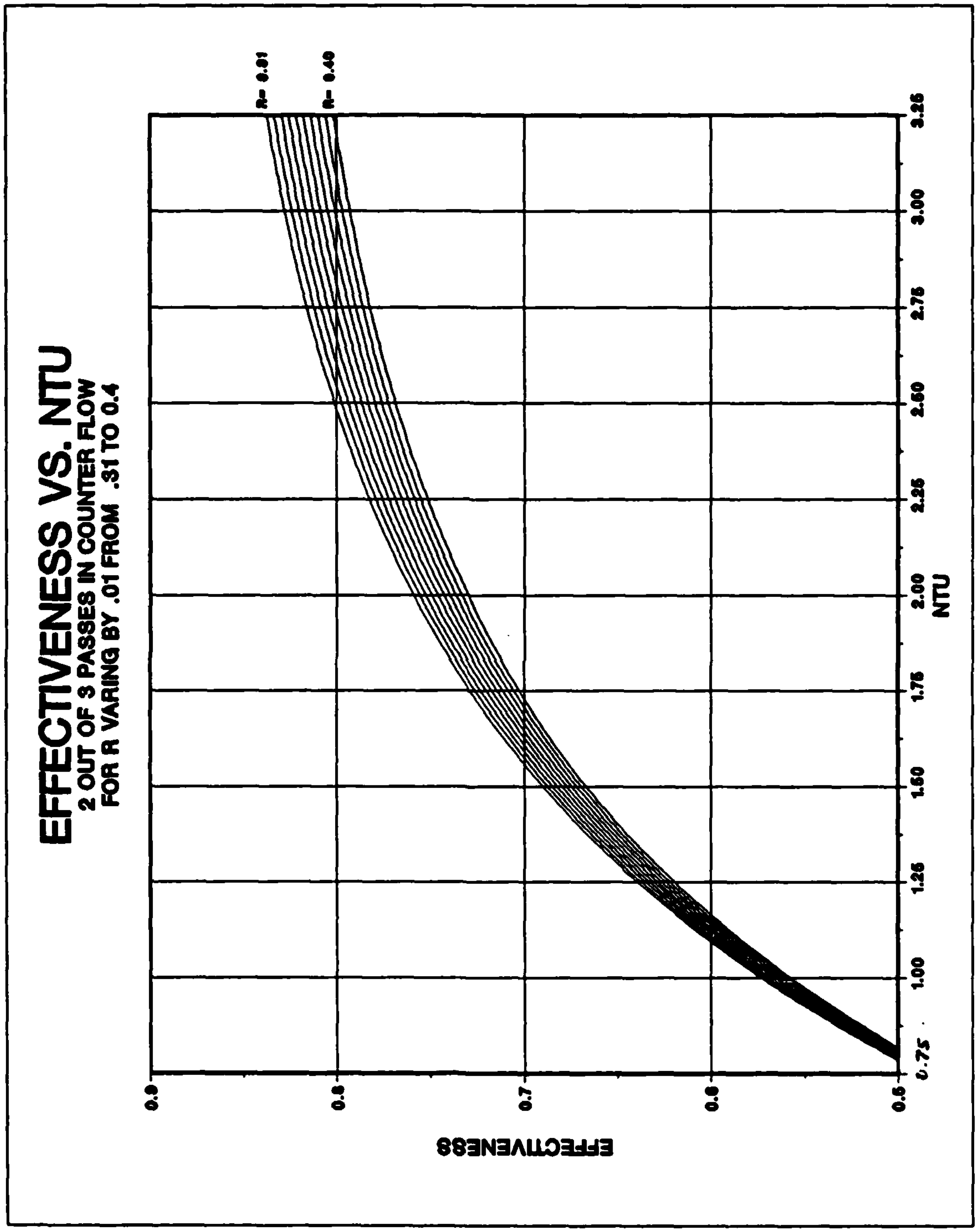


Figure M.5 1-3:2C Effectiveness vs. N_{tu} over Range of R from 0.31 to 0.4

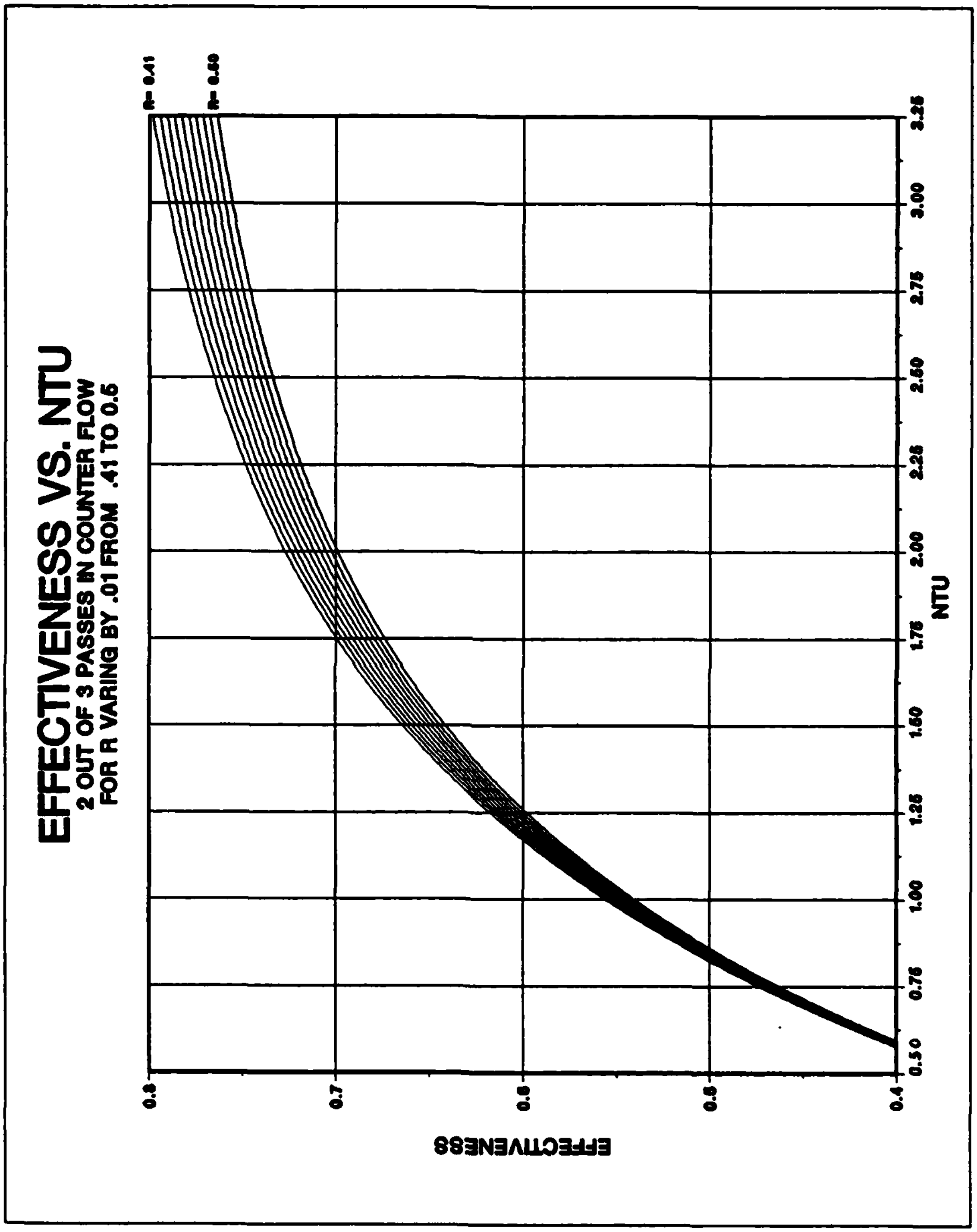


Figure M.6 1-3:2C Effectiveness vs. N_{tu} over Range of R from 0.41 to 0.5

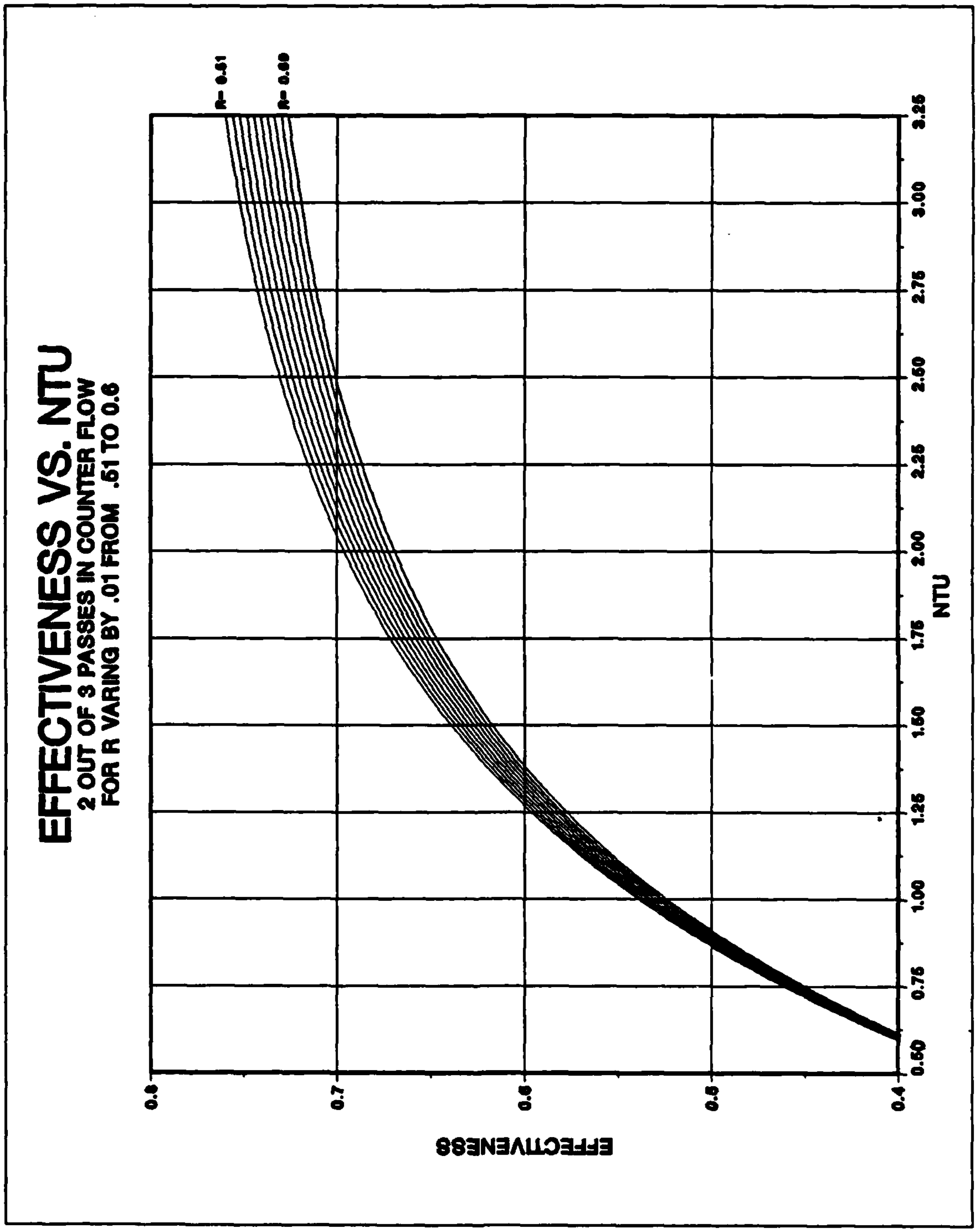


Figure M.7 1-3:2C Effectiveness vs. N_{tu} over Range of R from 0.51 to 0.6

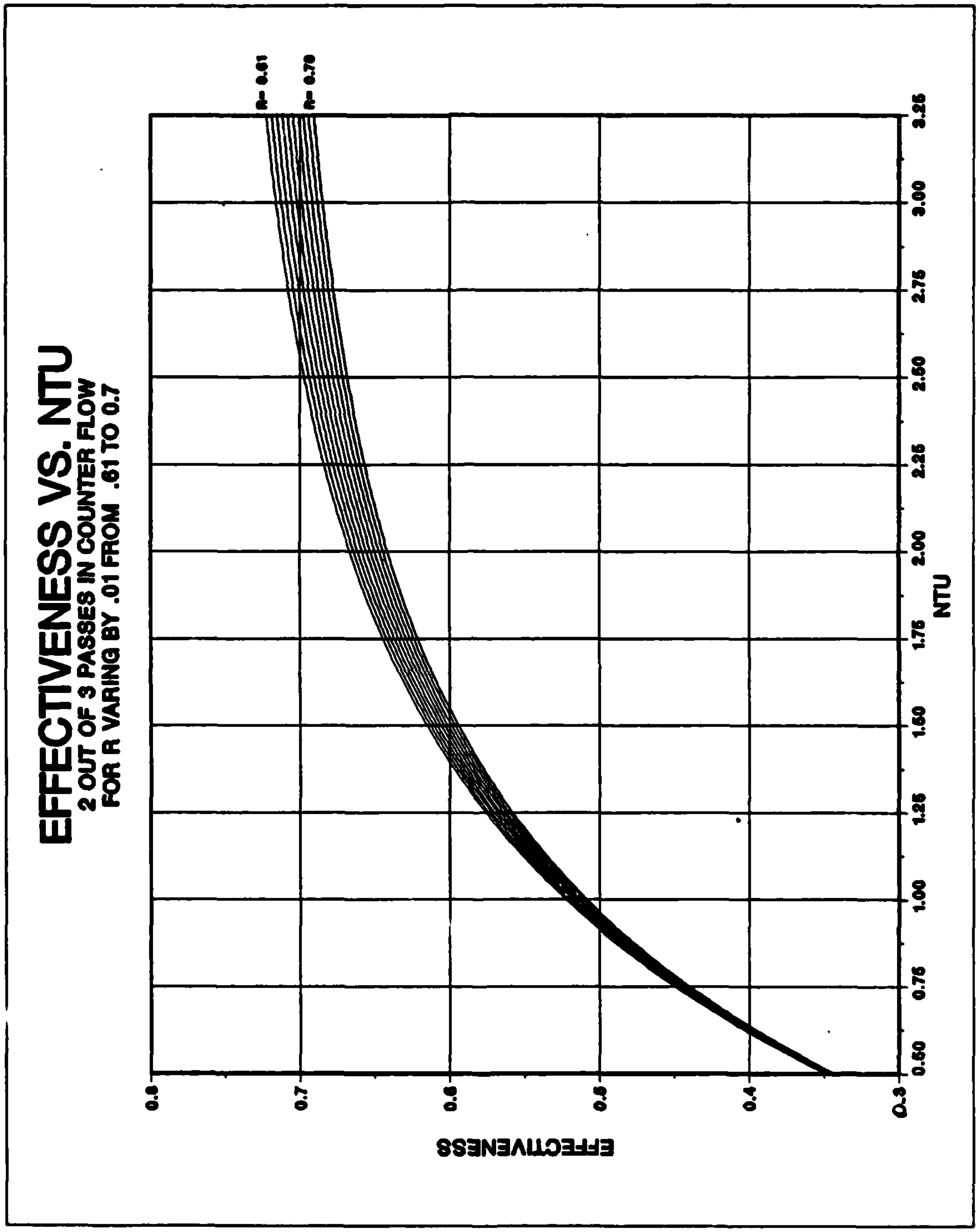


Figure M.8 1-3:2C Effectiveness vs. N_{tu} over Range of R from 0.61 to 0.7

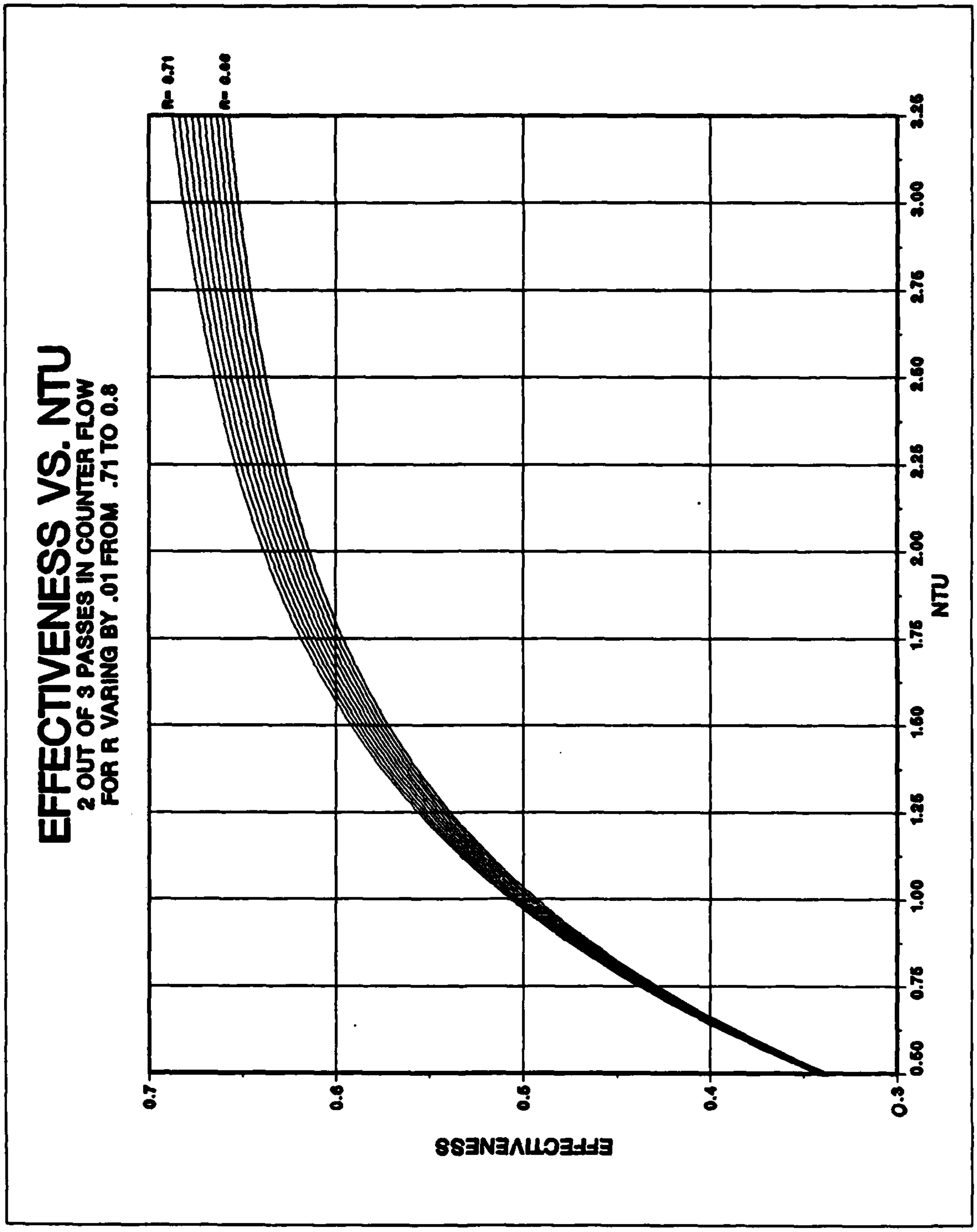


Figure M.9 1-3:2C Effectiveness vs. N_{tu} over Range of R from 0.71 to 0.8

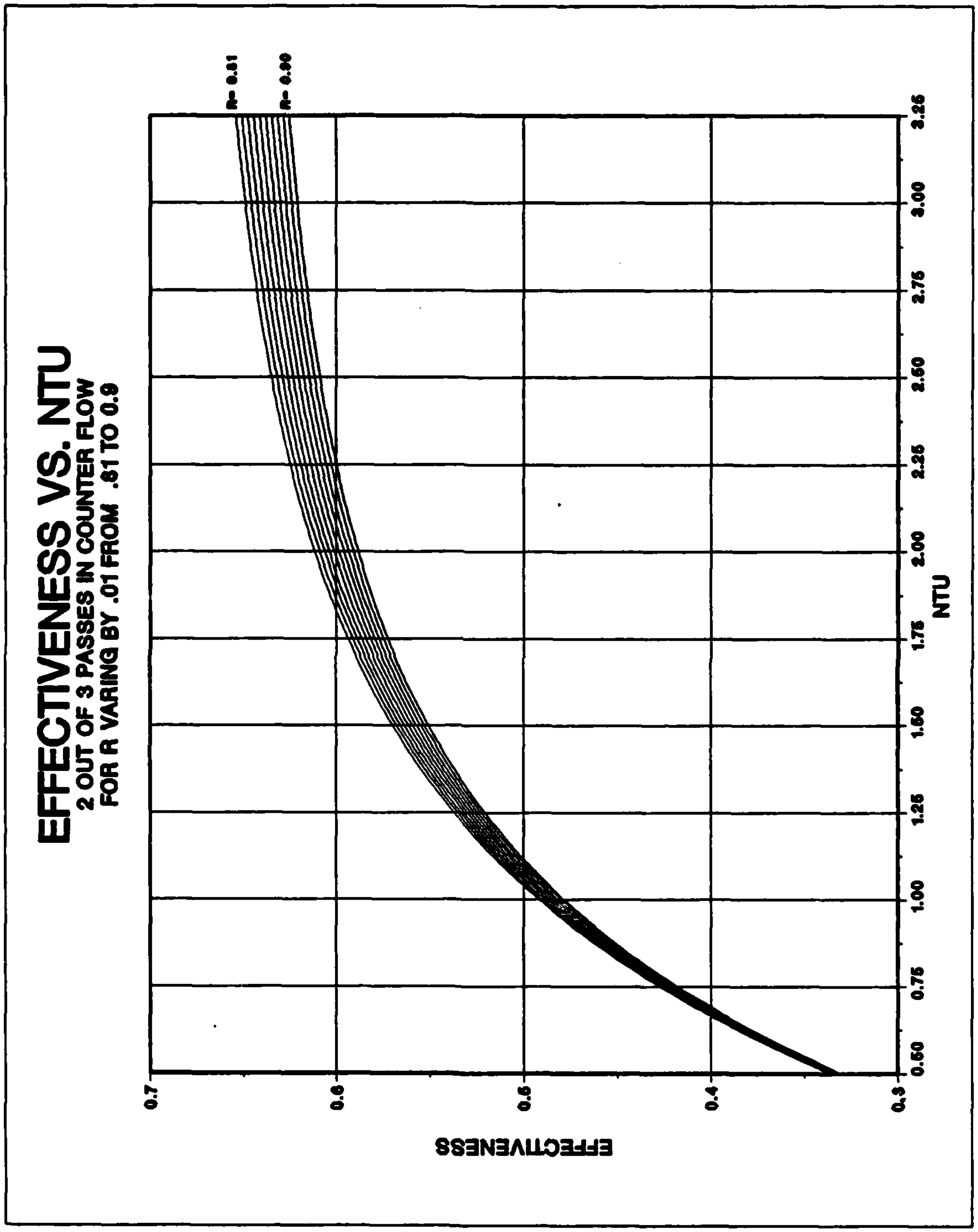


Figure M.10 1-3:2C Effectiveness vs. Ntu over Range of R from 0.81 to 0.9

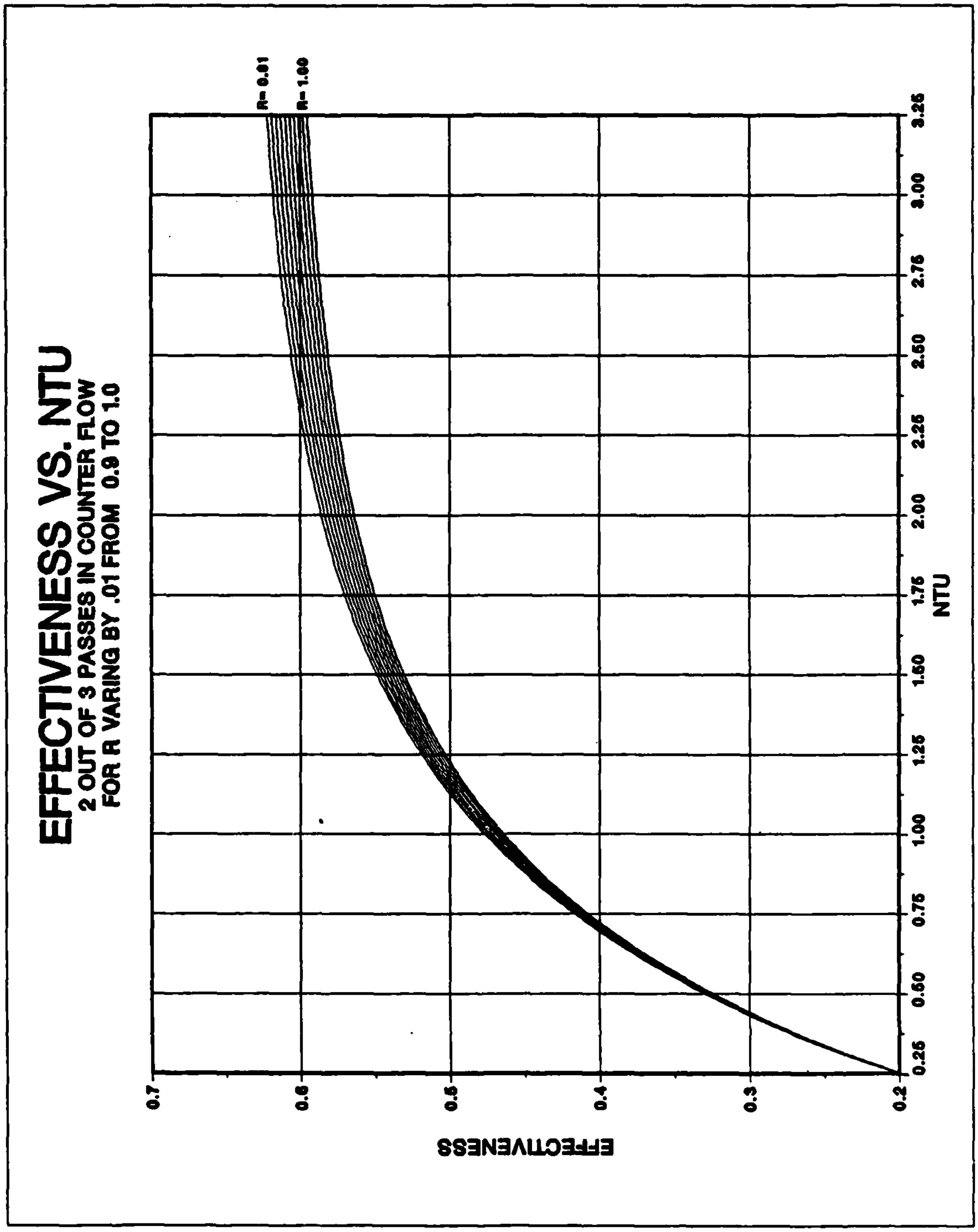


Figure M.11 1-3:2C Effectiveness vs. N_{tu} over Range of R from 0.9 to 1.0

APPENDIX N

1-3:2P EFFECTIVENESS VS. N_{tu} GRAPHS AT VARIOUS R VALUES

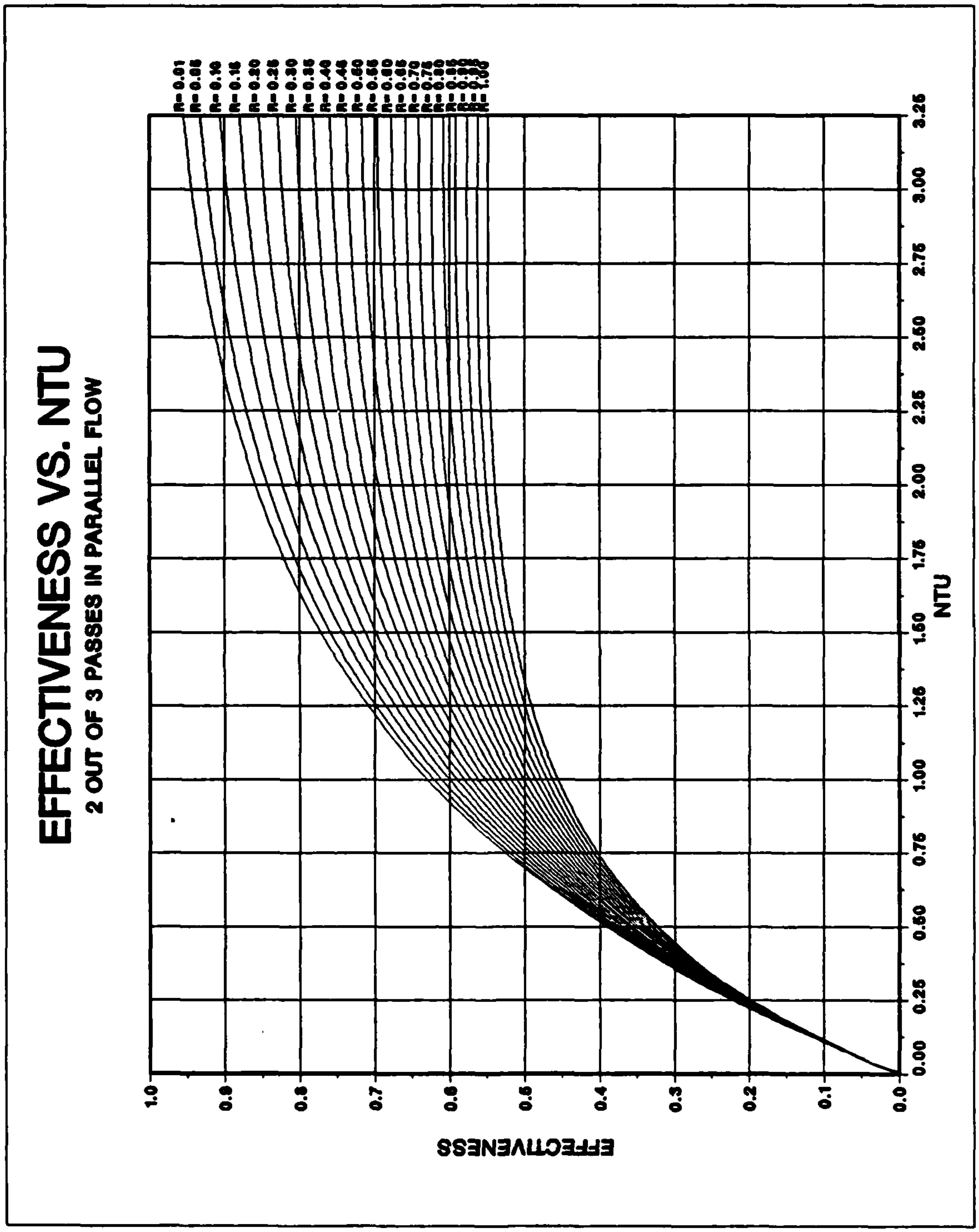


Figure N.1 1-3:2P Effectiveness vs. N_{tu} over Range of R from 0.01 to 1.0

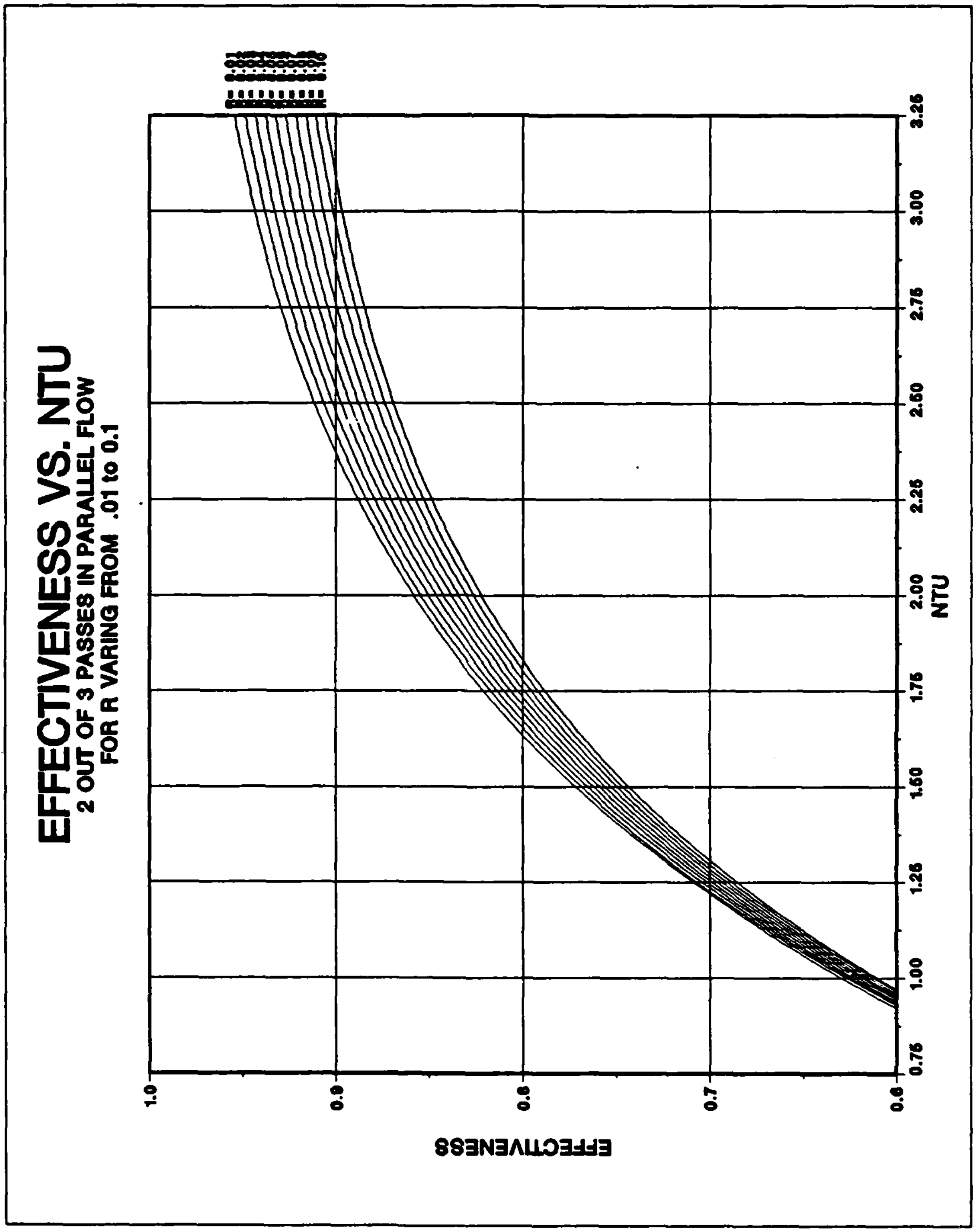


Figure N.2 1-3:2P Effectiveness vs. N_{tu} over Range of R from 0.01 to 0.1

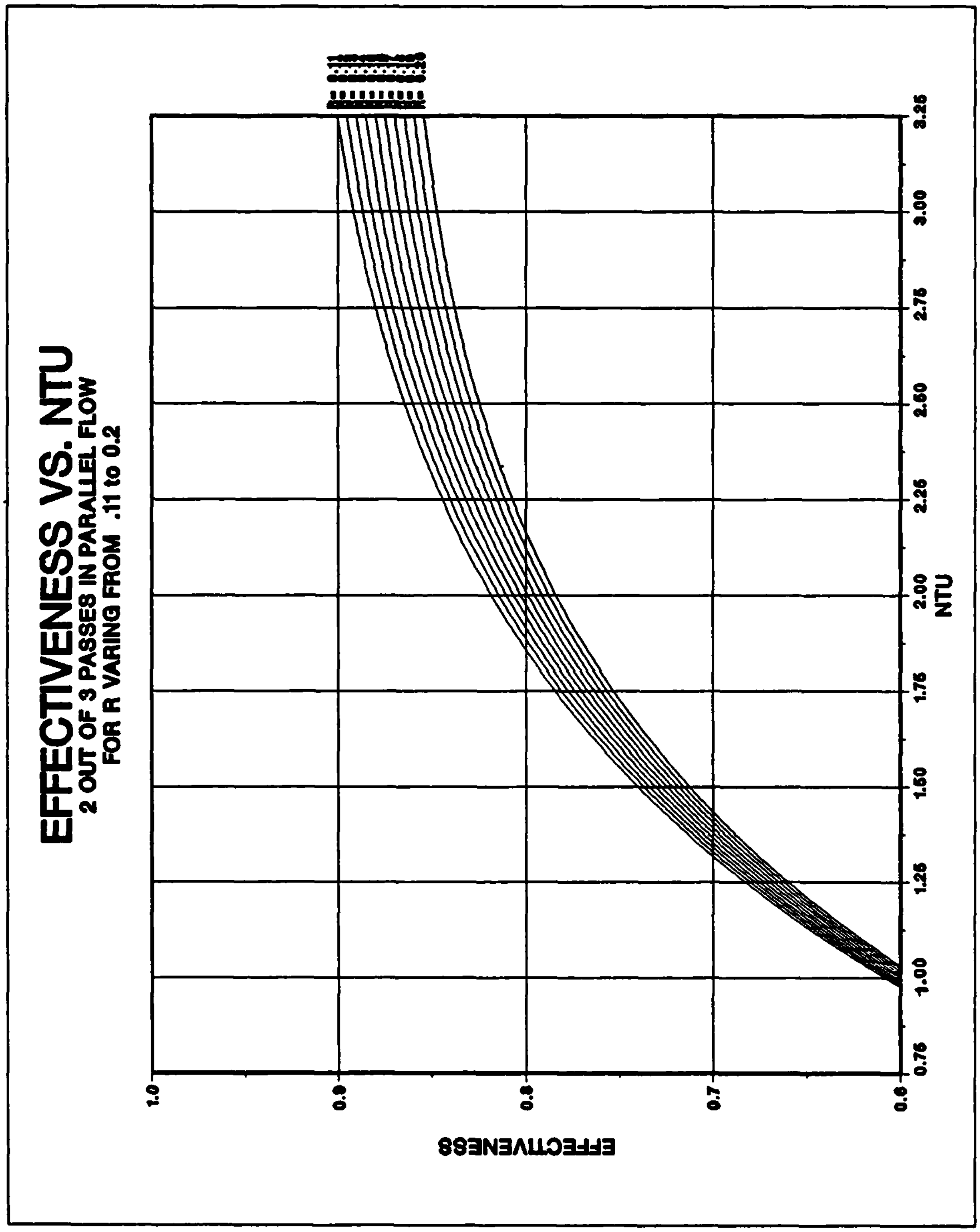


Figure N.3 1-3:2P Effectiveness vs. N_{tu} over Range of R from 0.11 to 0.2

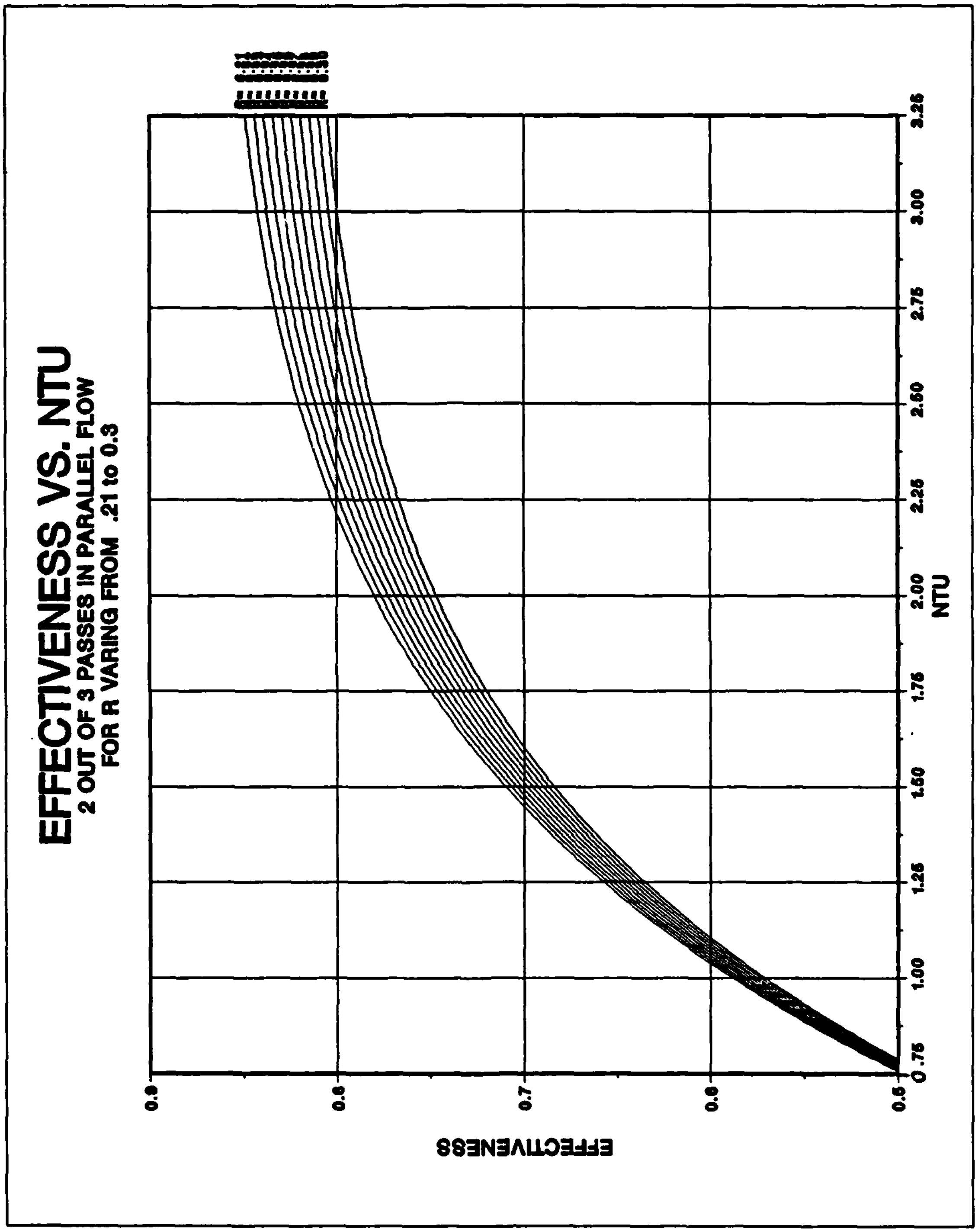


Figure N.4 1-3:2P Effectiveness vs. N_{tu} over Range of R from 0.21 to 0.3

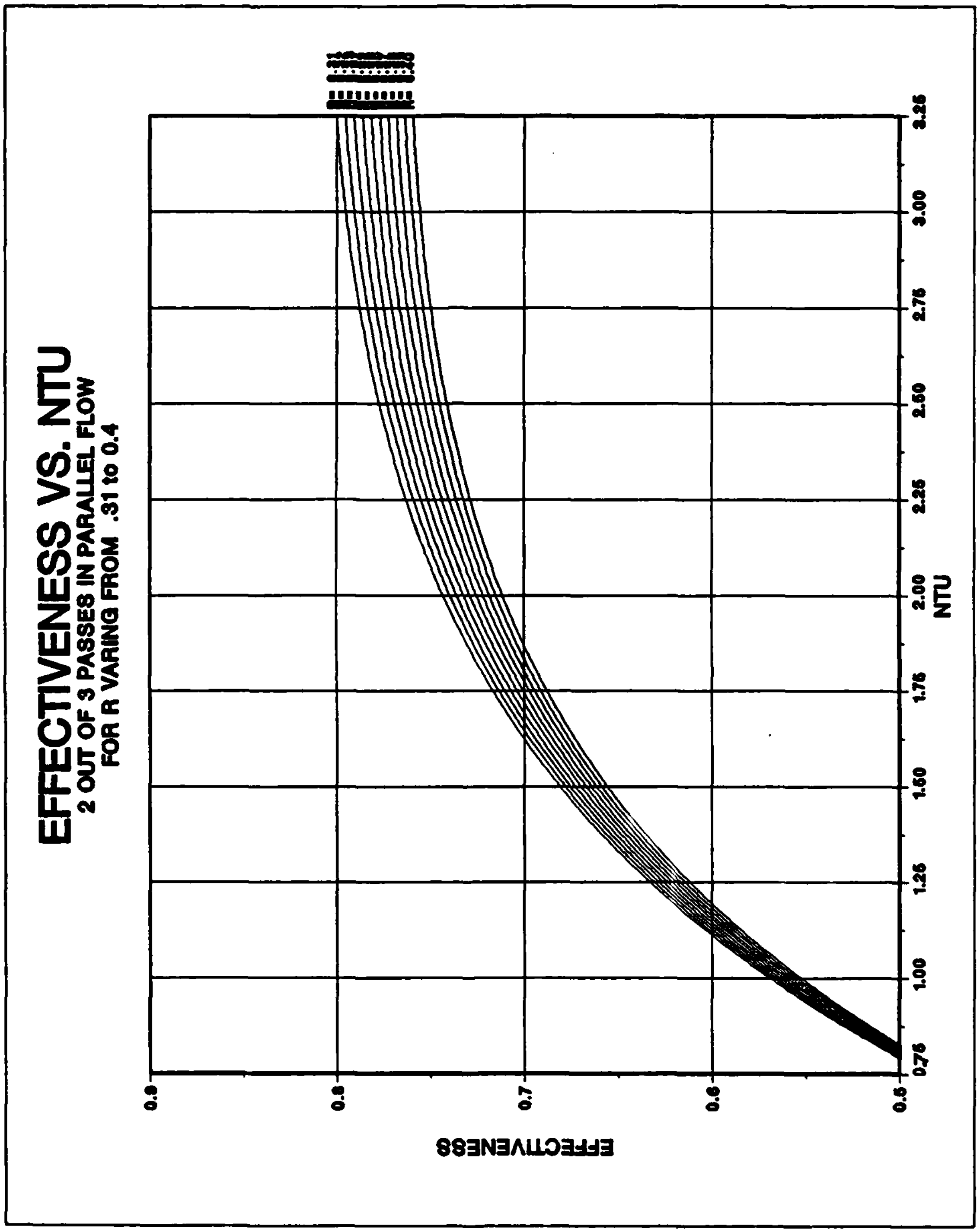


Figure N.5 1-3:2P Effectiveness vs. N_{tu} over Range of R from 0.31 to 0.4

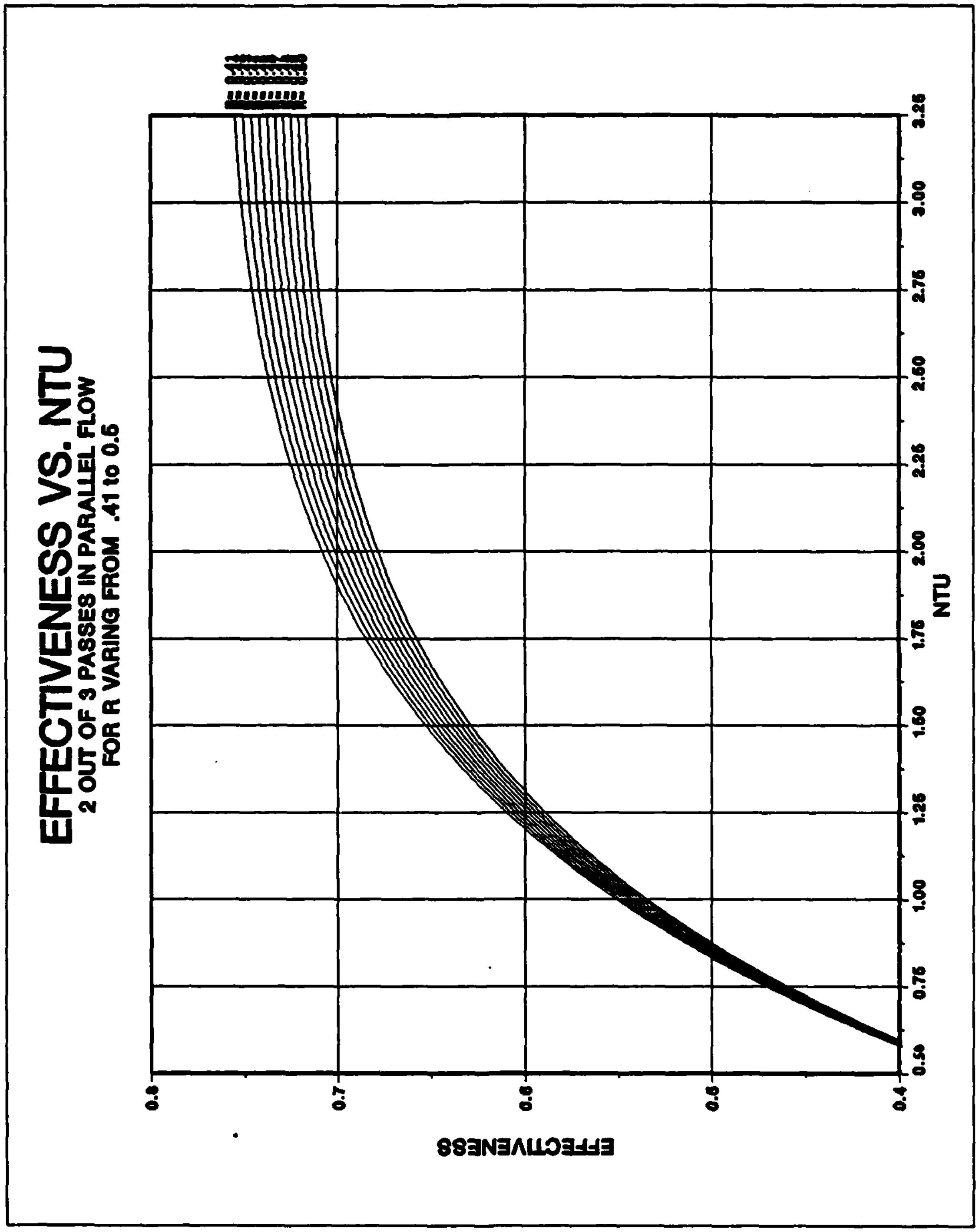


Figure N.6 1-3:2P Effectiveness vs. N_{tu} over Range of R from 0.41 to 0.5

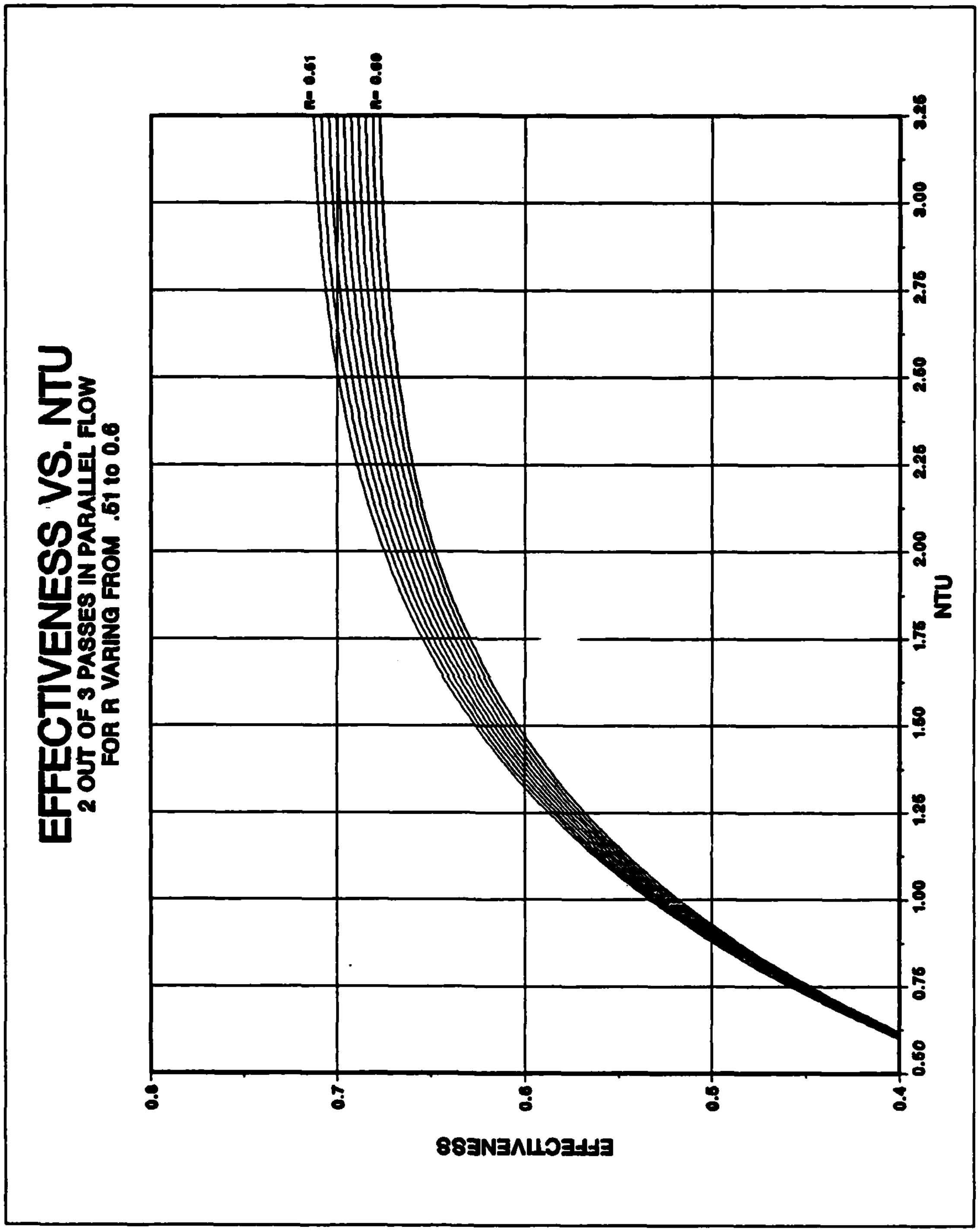


Figure N.7 1-3:2P Effectiveness vs. N_{tu} over Range of R from 0.51 to 0.6

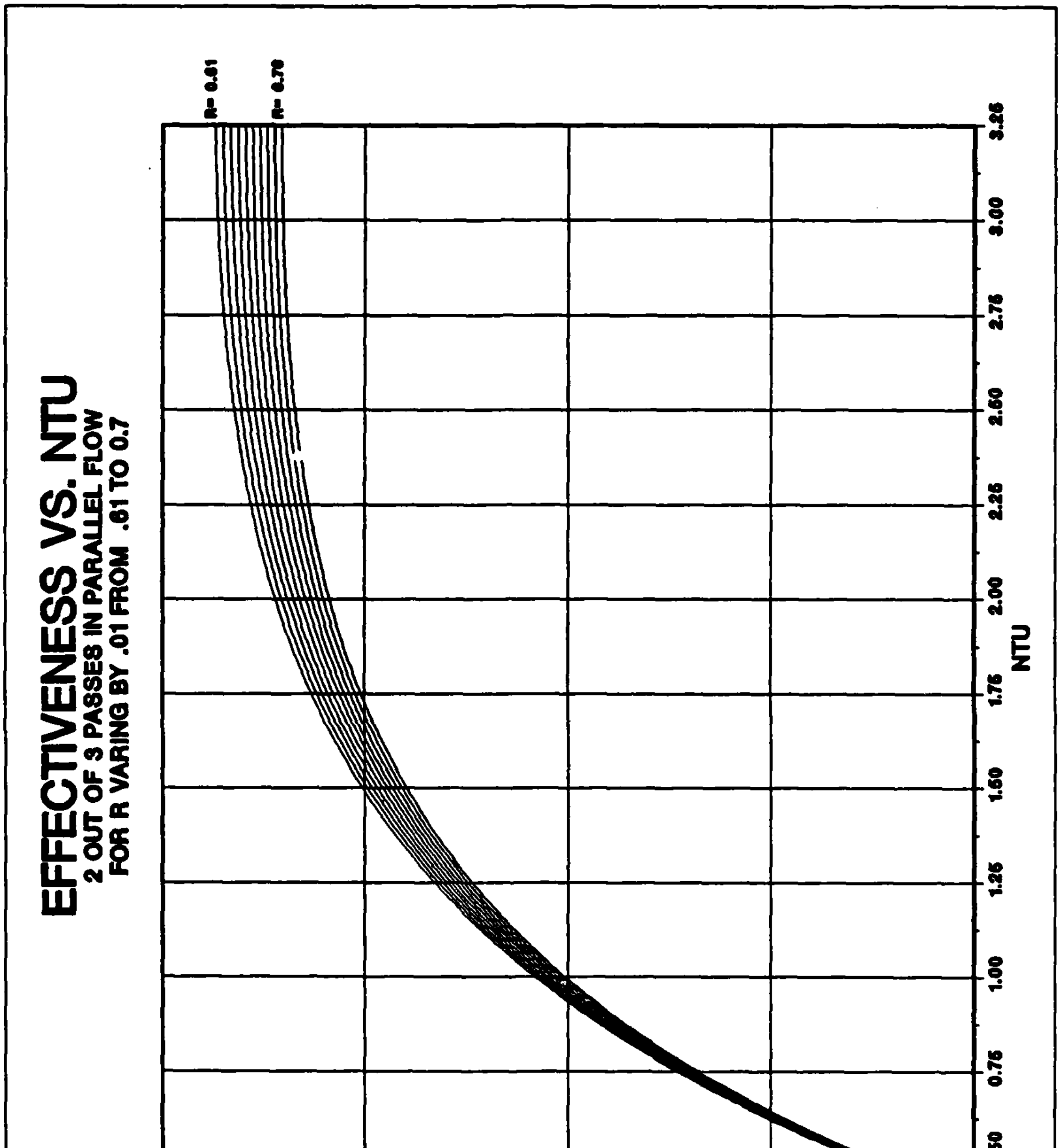


Figure N.8 1-3:2P Effectiveness vs. N_{tu} over Range of R from 0.61 to 0.7

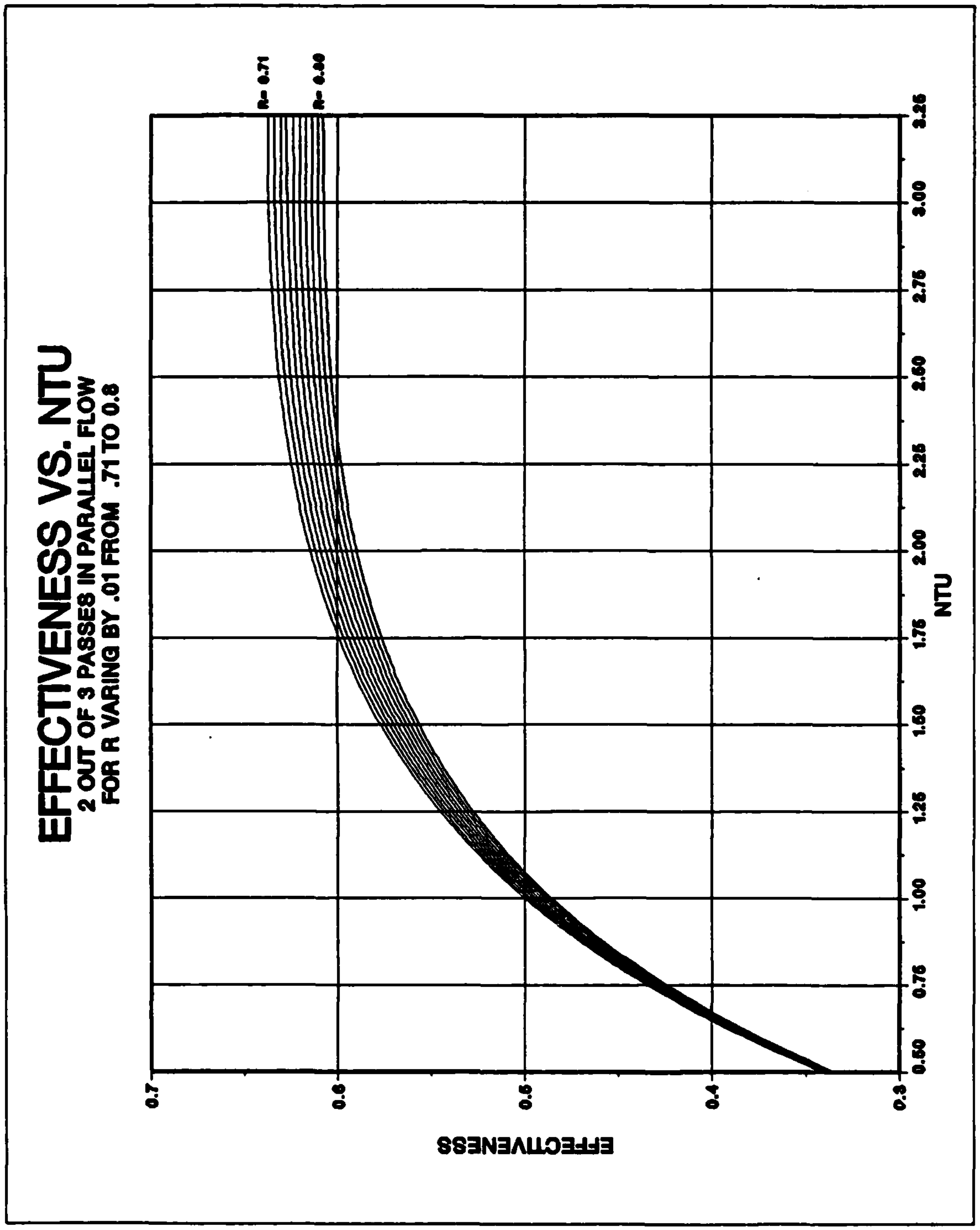


Figure N.9 1-3:2P Effectiveness vs. N_{tu} over Range of R from 0.71 to 0.8

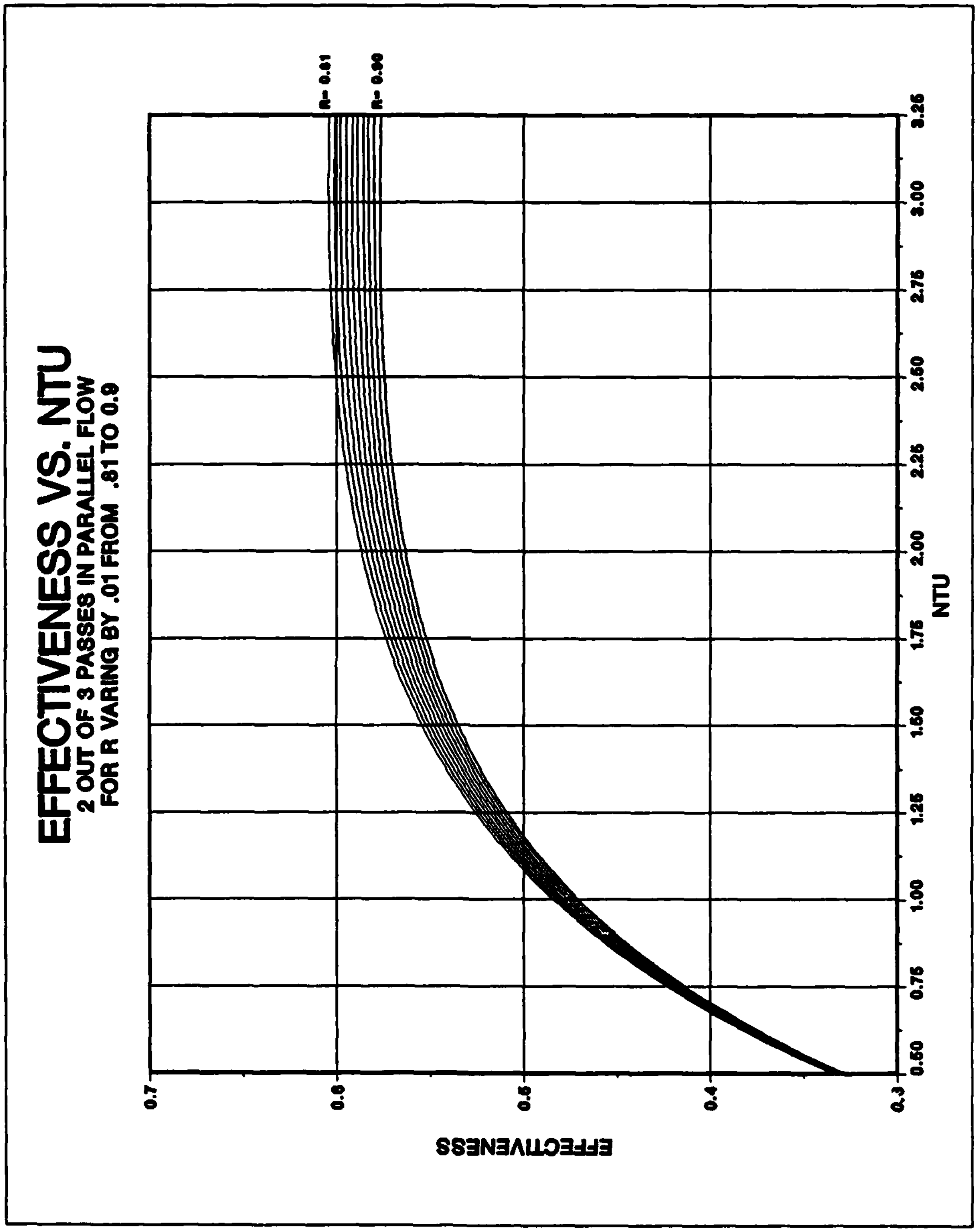


Figure N.10 1-3:2P Effectiveness vs. N_{tu} over Range of R from 0.81 to 0.9

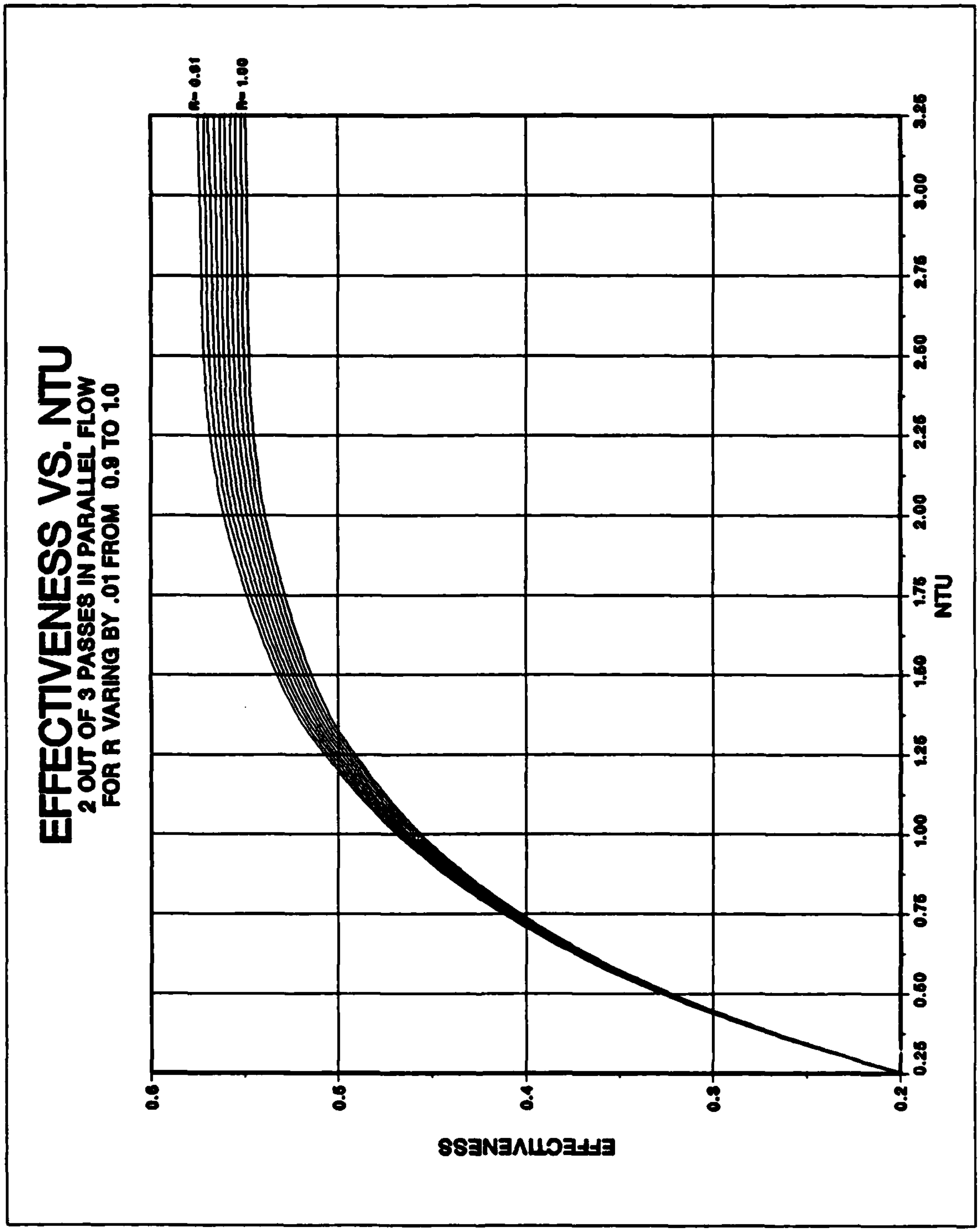


Figure N.11 1-3:2P Effectiveness vs. N_{tu} over Range of R from 0.91 to 1.0

APPENDIX O

SAMPLE DISSPLAY PROGRAM USED FOR GRAPHING DATA

```
C ********************************************************************
C *                                                                  *
C *                 DISSPLAY GRAPHING ROUTINE                        *
C *                            by                                    *
C *                 LCDR MARK S O'HARE U.S.N                         *
C *                        8 MAY 1985                                *
C *          NAVAL POSTGRADUATE SCHOOL, MONTEREY, CA                 *
C *                                                                  *
C ********************************************************************
C
C CALL DISSPLA GRAPHICS PACKAGE
      DIMENSION X1(12),Y1(12)
      DIMENSION X2(12),Y2(12),X3(12),Y3(12),X4(12),Y4(12)
      DIMENSION X5(12),Y5(12),X6(12),Y6(12)
      DIMENSION X7(12),X8(12),X9(12),X10(12),X11(12),X12(12)
      DIMENSION Y7(12),Y8(12),Y9(12),Y10(12),Y11(12),Y12(12)
      DIMENSION X13(12),X14(12),X15(12),X16(12),X17(12),X18(12)
      DIMENSION Y13(12),Y14(12),Y15(12),Y16(12),Y17(12),Y18(12)
      DIMENSION X19(12),X20(12),X21(12)
      DIMENSION Y19(12),Y20(12),Y21(12)
      DIMENSION WW(21)
C THIS IS THE DISTANCE OF THE VALUE FOR R
      TT=3.315
C THIS IS THE DISTANCE OF 'R='
      T=3.265
C     CALL TEK618
      CALL COMPRS
C     ******* START LEVEL 1 WORK *********
C     CALL BCOMON (10*21+2)
      DATA WW/0.01,0.05,0.10,0.15,0.20,0.25,0.30,0.35,0.4,0.45,0.5,
     *0.55,0.6,0.65,0.7,0.75,0.8,0.85,0.9,0.95,1.0/
1000  FORMAT (6X,F4.2,3X,F6.4)
      DO 20 I=1,11
      READ (21,1000) X1(I),Y1(I)
20    CONTINUE
      DO 30 I=1,11
      READ (22,1000) X2(I),Y2(I)
30    CONTINUE
      DO 40 I=1,11
      READ (23,1000) X3(I),Y3(I)
40    CONTINUE
      DO 50 I=1,11
```

```
      READ (24,1000) X4(I),Y4(I)
50    CONTINUE
      DO 60 I=1,11
      READ (25,1000) X5(I),Y5(I)
60    CONTINUE
      DO 70 I=1,11
      READ (26,1000) X6(I),Y6(I)
70    CONTINUE
      DO 80 I=1,11
      READ (27,1000) X7(I),Y7(I)
80    CONTINUE
      DO 90 I=1,11
      READ (28,1000) X8(I),Y8(I)
90    CONTINUE
      DO100 I=1,11
      READ (29,1000) X9(I),Y9(I)
100   CONTINUE
      DO 110 I=1,11
      READ (30,1000) X10(I),Y10(I)
110   CONTINUE
      DO120 I=1,11
      READ (31,1000) X11(I),Y11(I)
120   CONTINUE
      DO 130 I=1,11
      READ (32,1000) X12(I),Y12(I)
130   CONTINUE
      DO 140 I=1,11
      READ (33,1000) X13(I),Y13(I)
140   CONTINUE
      DO 150 I=1,11
      READ (34,1000) X14(I),Y14(I)
150   CONTINUE
      DO 160 I=1,11
      READ (35,1000) X15(I),Y15(I)
160   CONTINUE
      DO 170 I=1,11
      READ (36,1000) X16(I),Y16(I)
170   CONTINUE
      DO 180 I=1,11
      READ (37,1000) X17(I),Y17(I)
180   CONTINUE
      DO 190 I=1,11
      READ (38,1000) X18(I),Y18(I)
190   CONTINUE
      DO 200 I=1,11
      READ (39,1000) X19(I),Y19(I)
200   CONTINUE
      DO 210 I=1,11
```

```
      READ (40,1000) X20(I),Y20(I)
210   CONTINUE
      DO 220 I=1,11
      READ (41,1000) X21(I),Y21(I)
220   CONTINUE
      CALL NOBRDR
      CALL SWISSM
      CALL BASALF('STANDARD')
      CALL MIXALF('L/CSTD')
      CALL HWROT('COMIC')
      CALL SHDCHR(90.,1,0.005,1)
      CALL HEIGHT (.17)
      CALL HWSCAL('SCREEN')
      CALL PAGE(14.0,11.0)
      CALL XNAME ('NTU$',100)
      CALL YNAME ('EFFECTIVENESS$',100)
      CALL AREA2D(11.0,8.5)
      CALL HEADIN ('EFFECTIVENESS VS. NTU$',100,2.0,2)
      CALL HEADIN ('2 OUT OF 3 PASSES IN COUNTER FLOW$',100,1.0,2)
      CALL YTICKS (2)
      CALL XTICKS (2)
      CALL YAXANG (180)
      CALL GRAF (0.0,0.25,3.25,0.0,0.10,1.0)
      CALL HEIGHT (.10)
      CALL POLY5
      CALL CURVE (X1,Y1,11,0)
      W=WW(1)
      Z=Y1(11)
      CALL RLREAL (W,2,TT,Z)
      CALL RLMESS ('R=$',2,T,Z)
      CALL CURVE (X2,Y2,11,0)
      W=WW(2)
      Z=Y2(11)
      CALL RLREAL (W,2,TT,Z)
      CALL RLMESS ('R=$',2,T,Z)
      CALL CURVE (X3,Y3,11,0)
      W=WW(3)
      Z=Y3(11)
      CALL RLREAL (W,2,TT,Z)
      CALL RLMESS ('R=$',2,T,Z)
      CALL CURVE (X4,Y4,11,0)
      W=WW(4)
      Z=Y4(11)
      CALL RLREAL (W,2,TT,Z)
      CALL RLMESS ('R=$',2,T,Z)
      CALL CURVE (X5,Y5,11,0)
      W=WW(5)
      Z=Y5(11)
```

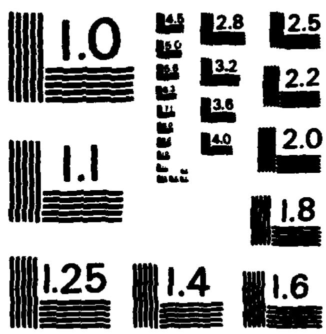
```
CALL RLREAL (W,2,TT,Z)
CALL RLMESS ('R=$',2,T,Z)
CALL CURVE (X6,Y6,11,0)
W=WW(6)
Z=Y6(11)
CALL RLREAL (W,2,TT,Z)
CALL RLMESS ('R=$',2,T,Z)
CALL CURVE (X7,Y7,11,0)
W=WW(7)
Z=Y7(11)
CALL RLREAL (W,2,TT,Z)
CALL RLMESS ('R=$',2,T,Z)
CALL CURVE (X8,Y8,11,0)
W=WW(8)
Z=Y8(11)
CALL RLREAL (W,2,TT,Z)
CALL RLMESS ('R=$',2,T,Z)
CALL CURVE (X9,Y9,11,0)
W=WW(9)
Z=Y9(11)
CALL RLREAL (W,2,TT,Z)
CALL RLMESS ('R=$',2,T,Z)
CALL CURVE (X10,Y10,11,0)
W=WW(10)
Z=Y10(11)
CALL RLREAL (W,2,TT,Z)
CALL RLMESS ('R=$',2,T,Z)
CALL CURVE (X11,Y11,11,0)
W=WW(11)
Z=Y11(11)
CALL RLREAL (W,2,TT,Z)
CALL RLMESS ('R=$',2,T,Z)
CALL CURVE (X12,Y12,11,0)
W=WW(12)
Z=Y12(11)
CALL RLREAL (W,2,TT,Z)
CALL RLMESS ('R=$',2,T,Z)
CALL CURVE (X13,Y13,11,0)
W=WW(13)
Z=Y13(11)
CALL RLREAL (W,2,TT,Z)
CALL RLMESS ('R=$',2,T,Z)
CALL CURVE (X14,Y14,11,0)
W=WW(14)
Z=Y14(11)
CALL RLREAL (W,2,TT,Z)
CALL RLMESS ('R=$',2,T,Z)
CALL CURVE (X15,Y15,11,0)
```

```
      W=WW(15)
      Z=Y15(11)
      CALL RLREAL (W,2,TT,Z)
      CALL RLMESS ('R=$',2,T,Z)
      CALL CURVE (X16,Y16,11,0)
      W=WW(16)
      Z=Y16(11)
      CALL RLREAL (W,2,TT,Z)
      CALL RLMESS ('R=$',2,T,Z)
      CALL CURVE (X17,Y17,11,0)
      W=WW(17)
      Z=Y17(11)
      CALL RLREAL (W,2,TT,Z)
      CALL RLMESS ('R=$',2,T,Z)
      CALL CURVE (X18,Y18,11,0)
      W=WW(18)
      Z=Y18(11)
      CALL RLREAL (W,2,TT,Z)
      CALL RLMESS ('R=$',2,T,Z)
      CALL CURVE (X19,Y19,11,0)
      W=WW(19)
      Z=Y19(11)
      CALL RLREAL (W,2,TT,Z)
      CALL RLMESS ('R=$',2,T,Z)
      CALL CURVE (X20,Y20,11,0)
      W=WW(20)
      Z=Y20(11)
      CALL RLREAL (W,2,TT,Z)
      CALL RLMESS ('R=$',2,T,Z)
      CALL CURVE (X21,Y21,11,0)
      W=WW(21)
      Z=Y21(11)
      CALL RLREAL (W,2,TT,Z)
      CALL RLMESS ('R=$',2,T,Z)
C     ******** START LEVEL 2 WORK *********
      CALL FRAME
C     ******* START LEVEL 3 WORK ********
      CALL GRID (1,1)
      CALL ENDPL(C)
      CALL DONEPL
      STOP
      END
```

AD-A159 706 THE EFFECTIVENESS OF HEAT EXCHANGERS WITH ONE SHELL PASS AND THREE TUBE PASSES(U) NAVAL POSTGRADUATE SCHOOL MONTEREY CA M S O'HARE JUN 85 3/3

UNCLASSIFIED F/G 20/13 NL

END
FILMED
DTIC



MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

APPENDIX P

POLYNOMIAL REGRESSION CURVEFIT PROGRAM

```
C
C
C
C       PROGRAM POLYFIT
C
C       PURPOSE:   DO AN NTH ORDER LEAST-SQUARES FIT OF THE INPUT VALUES
C                  "X" AND "Y".
C
C       INPUT:     I)  X.
C                  II)  Y.
C
C       OUTPUT:    I)  MEAN OF X.
C                  II)  RANGE OF X.
C                  III)  ORDER OF THE FIT.
C                  IV)  COEFFICIENTS IN DOUBLE PRECISION.
C                  V)  R.M.S. ERROR OF THE RESIDUALS.
C
C       CALL:      I) CURFIT(IORDER,NITEMS,X,Y,AA)
C                  II)  SOLVE(NR,C)
C                  III)  FUNCTION AMAX(X,NITEMS)
C                  IV)  FUNCTION AMIN(X,NITEMS)
C
      INTEGER I,NITEMS,MAXORD,J,K,L,M,N
      DOUBLE PRECISION AA(40)
      DIMENSION X(11),Y(11),YCOMP(11),ONE(11,2),TWO(11,2)
      COMMON IDIAGN
C
C
C
C       1.  ZERO ALL VALUES THAT WILL BE USED TO COMPUTE SUMS.
C
C
      RMSSUM = 0.0
C
C
C       2.  INPUT THE DATA.
C
C
      IDIAGN=0
C      WRITE(6,100)
C100   FORMAT(2X,'MAXIMUM ORDER OF THE FIT?')
```

```
 120  FORMAT(2X,' ')
C     READ(5,*) MAXORD
      MAXORD=5
C
C
      DO 10 I=1,11
         READ(1,*) ONE(I,1),TWO(I,2)
         X(I) = ONE(I,1)
         Y(I) = TWO(I,2)
 10   CONTINUE
C
C
      NITEMS = I - 2
C
C
C      3.  COMPUTE THE LEAST-SQUARES COEFFICIENTS BY CALLING "CURFIT()".
C          THE COEFFICIENTS ARE RETURNED IN "AA()" SUCH THAT:
C
C                Y=AA(1)+AA(2)*X+...+AA(N)*(X**N)
C
C
      CALL CURFIT(MAXORD,NITEMS,X,Y,AA)
C
C
C      4.  COMPUTE THE RESIDUALS AND THE R.M.S. ERROR OF THE RESIDUALS.
C
C
      DO 90 I = 1,NITEMS
          YCOMP(I) = 0.0
          IJUNK = MAXORD + 1
          DO 91 J = 1,IJUNK
             YCOMP(I) = YCOMP(I) + AA(J)*(X(I)**(J-1))
             IF(X(I) .EQ. 0.0) YCOMP(I) = AA(1)
 91       CONTINUE
C
C
          RMSSUM = RMSSUM + (Y(I)-YCOMP(I))*(Y(I)-YCOMP(I))
 90   CONTINUE
C
C
      RMS = SQRT(RMSSUM/FLOAT(NITEMS))
C
C
C      5.  PRINT OUT THE RESULTS.
C
C
      WRITE(7,120)
      WRITE(7,120)
```

```
      WRITE(7,120)
      WRITE(7,102) MAXORD
102   FORMAT(2X,'LEAST SQUARES FIT OF ORDER: ',I5)
      WRITE(7,120)
      WRITE(7,103) RMS
103   FORMAT(2X,'R.M.S. ERROR = ',1F15.5)
      WRITE(7,120)
      WRITE(7,120)
      WRITE(7,104)
      WRITE(7,105)
104   FORMAT(2X,'ORDER, COEFFICIENT')
105   FORMAT(2X,'____________________')
C
C
      IJUNK = MAXORD + 1
      DO 92 I=1,IJUNK
            WRITE(7,*) I,AA(I)
92    CONTINUE
C
C
      WRITE(7,120)
      WRITE(7,120)
      WRITE(7,110)
      WRITE(7,111)
110   FORMAT(2X,'I, X, Y, Y(COMPUTED)')
111   FORMAT(2X,'______________________')
      DO 93 I=1,NITEMS
            WRITE(7,*) I,X(I),Y(I),YCOMP(I)
93    CONTINUE
      STOP
      END
C
C
C
C
C
C
C
C
C
      SUBROUTINE CURFIT(IORDER,NITEMS,X,Z,AA)
C
C     PURPOSE:  FIT Z(X) USING AN "IORDER" ORDER
C               LEAST-SQUARES REGRESSION FIT FOR
C               "NITEMS" OF DATA.
C
C
      EXTERNAL AMAX,AMIN
```

```
      DIMENSION X(NITEMS),Z(NITEMS)
      DIMENSION XX(500),ZZ(500)
      DIMENSION XXX(2),ZZZ(2)
      DOUBLE PRECISION SMATRX(100,100),AA(40)
      DIMENSION SUBSUM(40)
      COMMON IDIAGN
C
C
C      1.  NORMALIZE "X" AND "Z" BETWEEN "-1" AND "1".
C
C
      XMAX = AMAX(X,NITEMS)
      XMIN = AMIN(X,NITEMS)
      ZMAX = AMAX(Z,NITEMS)
      ZMIN = AMIN(Z,NITEMS)
C
C
      XXX(1) = XMAX
      XXX(2) = ABS(XMIN)
      ZZZ(1) = ZMAX
      ZZZ(2) = ABS(ZMIN)
      XNORM = AMAX(XXX,2)
      ZNORM = AMAX(ZZZ,2)
C
C
      IF(IDIAGN .EQ. 3) WRITE(7,107) XNORM,ZNORM
107   FORMAT(2X,'CURFIT:  XNORM,ZNORM = ',3F15.5)
C
C
      DO 96 I=1,NITEMS
             XX(I) = X(I)/XNORM
             ZZ(I) = Z(I)/ZNORM
             IF(IDIAGN .EQ. 4) WRITE(7,108) I,XX(I),ZZ(I)
96    CONTINUE
C
108   FORMAT(2X,'CURFIT:  I,XX,ZZ = ',I5,3F12.4)
C
C
C      2.  COMPUTE ALL NECESSARY SUMS.
C
C
      ICOEF = IORDER + 1
C
C
      IF(IDIAGN .EQ. 3) WRITE(7,111) ICOEF,NITEMS
111   FORMAT(2X,'CURFIT:  ICOEF,NTIEMS = ',2I10)
C
C
```

```
      DO 97 I=1,ICOEF
         IJUNK = ICOEF + 1
         DO 97 J=1,IJUNK
            SMATRX(I,J) = 0.0
97    CONTINUE
C
C
      DO 900 N=1,NITEMS
         SUBSUM(1) = 1.0
         DO 98 I=1,IORDER
            SUBSUM(I+1) = XX(N)**(I)
98       CONTINUE
      DO 900 J=1,ICOEF
         DO 99 I=1,ICOEF
            SMATRX(J,I) = SMATRX(J,I)+SUBSUM(I)*SUBSUM(J)
99       CONTINUE
         SMATRX(J,ICOEF+1)=SMATRX(J,ICOEF+1)+SUBSUM(J)*ZZ(N)
900   CONTINUE
C
C
      IF(IDIAGN .NE. 6) GO TO 22
         DO 906 I=1,ICOEF
            IJUNK = ICOEF + 1
            DO 906 J=1,IJUNK
               WRITE(7,110) I,J,SMATRX(I,J)
906      CONTINUE
22    CONTINUE
C
110   FORMAT(2X,'CURFIT:  SMATRX(',I3,',',I3,') = ',1F15.5)
C
C
C      3.  NOW, INVERT THE MATRIX.  (SOLVE THE SET OF
C          SIMULTANEOUS EQUATIONS.)
C
C
      CALL SOLVE(SMATRX,ICOEF)
C
C
      DO 903 I=1,ICOEF
         AA(I) = SMATRX(I,ICOEF+1)
         IF(IDIAGN.EQ.8) WRITE(7,112) I,AA(I)
903   CONTINUE
C
112   FORMAT(2X,'CURFIT:  AA(',I3,') = ',1F15.5)
C
C
C      4.  "UNNORMALIZE" THE RETURNED COEFFICIENTS.
C
```

```
C
      DO 901 I=2,ICOEF
         AA(I) = AA(I)/XNORM**(I-1)
901   CONTINUE
C
C
      DO 902 I=1,ICOEF
         AA(I) = AA(I) * ZNORM
         IF(IDIAGN.EQ.9) WRITE(7,113) I,AA(I)
902   CONTINUE
C
113   FORMAT(2X,'CURFIT:  AA(',I3,') = ',1F15.5)
      RETURN
      END
C
C
C
C
C
C
      SUBROUTINE SOLVE(C,NR)
C
C     PURPOSE:  SOLVE A SET OF "NR" SIMULTANEOUS EQUATIONS
C               USING GAUSSIAN ELIMINATION.
C
C
      DOUBLE PRECISION C(100,100),SAVE
      INTEGER NR
C
C
      NC = NR + 1
      DO 303 I=1,NR
            KEXCH = 1
            M = I + 1
306         IF(DABS(C(I,I))-1.D-5) 308,308,307
307            DO 301 J=M,NC
301               C(I,J)=C(I,J)/C(I,I)
               DO 303 J=1,NR
                  IF(J-I) 322,303,322
322               DO 302 K=M,NC
302                  C(J,K)=C(J,K)-C(J,I)*C(I,K)
303   CONTINUE
      RETURN
C
C
C
308   L = I + KEXCH
      IF(L-NR) 309,309,330
```

```
309   DO 311 N=I,NC
             SAVE = C(I,N)
             C(I,N) = C(L,N)
311          C(L,N) = SAVE
C
      KEXCH = KEXCH + 1
      GO TO 306
C
C
C
330   WRITE(7,331)
331   FORMAT(2X,'EQUATIONS CANNOT BE SOLVED...')
      RETURN
      END
C
C
C
C
C
C
      REAL FUNCTION AMAX(X,NITEMS)
C
C     PURPOSE:  FIND THE MAXIMUM VALUE IN AN ARRAY.
C
C
      DIMENSION X(500)
      INTEGER NITEMS
C
C
      XMAX = X(1)
      DO 340 I=1,NITEMS
             IF(X(I) .GT. XMAX)  XMAX = X(I)
340   CONTINUE
      AMAX = XMAX
      RETURN
      END
C
C
C
C
C
C
      REAL FUNCTION AMIN(X,NITEMS)
C
C     PURPOSE:  FIND THE MINIMUM VALUE IN AN ARRAY.
C
C
      DIMENSION X(500)
```

```
      INTEGER NITEMS
C
C
      XMIN = X(1)
      DO 340 I=1,NITEMS
         IF(X(I) .LT. XMIN)  XMIN = X(I)
340   CONTINUE
      AMIN = XMIN
      RETURN
      END
```

LIST OF REFERENCES

1. Kern, D. Q., Process Heat Transfer, McGraw-Hill Book Company, New York, NY, 1950.

2. Kays, W. M. and London, A. L., Compact Heat Exchangers, Third Edition, McGraw-Hill Book Company, New York, NY, 1985.

3. Nagle, W. M., "Mean Temperature Differences in Multipass Heat Exchangers," Industrial and Engineering Chemistry, Vol. 25, 1933.

4. Davis, F. K., Ross Heater and Manufacturing Company Bulletin, 350, 72, 1931.

5. Standards of the Tubular Heat Exchanger Manufacturers Association (TEMA), Sixth Edition, New York, NY, 1978.

6. Underwood, A. J. V., "The Calculation of the Mean Temperature Difference in Multipass Heat Exchangers," Journal of Institute of Petroleum Technologists, Vol. 20, 1934.

7. Bowman, R. A., "Mean Temperature Difference Correction in Multipass Exchangers," Industrial and Engineering Chemistry, Vol. 28, 1936, pp. 541-544.

8. Kraus, A. D. and Kern, D. Q., ASME Paper 65-HT-18, presented at the ASME-AIChE Heat Transfer Conference, Los Angeles, CA, 1965.

9. Bowman, R. A., Mueller, A. C. and Nagle, W. M., "Mean Temperature Difference in Design," Trans. ASME, Vol. 62, 1940.

10. Ten Broeck, H. A., "Multipass Exchanger Calculations," Industrial and Engineering Chemistry, Vol. 30, 1938, pp. 1041-1042.

11. Fischer, F. K., "Mean Temperature Difference Correction in Multipass Exchangers," Industrial and Engineering Chemistry, Vol. 30, 1938, pp. 377-383.

12. Kern, D. Q. and Kraus, A. D., Extended Surface Heat Transfer, McGraw-Hill Book Company, New York, NY, 1972.

13. Stewart, G. W., Introduction to Matrix Computations, pp. 130-150, Academic-Press, New York, NY, 1973.

14. Chapra, S. C. and Canale, R. P., Numerical Methods for Engineers with Personal Computer Applications, McGraw-Hill Book Company, New York, NY, 1985.

15. Gerald, C. F., Applied Numerical Analysis, 2nd edition, Addison-Wesley Publishing Company, Reading, Massachusetts, 1980.

16. Non-International Mathematical and Statistical Library (IMSL), Curvefit Program, Computer Program on Library IBM 3033 AP System, Naval Postgraduate School, Monterey, CA, 1983.

INITIAL DISTRIBUTION LIST

		No. Copies
1.	Defense Technical Information Center Cameron Station Alexandria, Virginia 22304-6145	2
2.	Library, Code 0142 Naval Postgraduate School Monterey, California 93943-5100	2
3.	Department Chairman, Code 69 Department of Mechanical Engineering Naval Postgraduate School Monterey, California 93943-5100	1
4.	CAPT T. Albee (USN, Ret.) 2503 Rock Branch Road Vienna, RI 22180	1
5.	LCDR R. Feckler, USN 1643 Rivergate Trail Jacksonville, FL 32223	1
6.	LCDR M. P. Golden, USNR 125 Woodridge Drive, South Stanford, CT 06902	1
7.	CAPT B. Green, USN 10821 Colton Street Burke, VA 22015	1
8.	LCDR L. Hess, USN 10 Gillespie Lane Monterey, CA 93940	1
9.	Professor A. D. Kraus, Code 69Ks Department of Mechanical Engineering Naval Postgraduate School Monterey, California 93943-5100	2
10.	CAPT B. Lee, USN Chief of Staff COMSIXTH FLT FPO New York 09501-6002	1

11. CAPT T. Leshko, USN 1
9010 Ft. Craig Drive
Burke, VA 22015

12. Professor Paul J. Marto, Code 69 1
Department of Mechanical Engineering
Naval Postgraduate School
Monterey, CA 93943-5100

13. Mr. Robert McCarthy 1
10200 South 52th Avenue
Oaklawn, IL 60453

14. Mr. John McDonald 1
819 Oak Trail
Crownsville, MD 21032

15. Mr. Robert McDonald 1
RFD 2 Box #320
Stevensville, MD 21666

16. LCDR J. E. McClanahan, USN 1
16206 Dartmoor Drive
Dumfries, VA

17. LCDR Mark S. O'Hare, USN 3
107 Peyton Drive
Chicago Hgts, IL 60411

18. Mr. M. O'Hare 1
c/o D. Davidson
320 East 54th Street
New York, NY 10022

19. Mr. Robert M. O'Hare, Sr. 3
107 Peyton Drive
Chicago Hgts, IL 60411

20. CAPT R. Schlenzig (USN, Ret.) 1
1042 Encino Road
Coronado, California 92118

21. LCDR M. Smith, USN 1
10025 Branford
San Diego, CA 92129

22. CDR P. Snyder, USN 1
4818 Mary Mead Drive
Fairfax, VA 22030

23. Professor King-Mon Tu, Code 69 1
Department of Mechanical Engineering
Naval Postgraduate School
Monterey, California 93943-5100

END

FILMED

11-85

DTIC