PDR: per R. Emmt   Reference (1279)

0933

ADD: R. Emmt, RES 0/1

Department of Mechanical and Industrial Engineering

Method for Analysis of Ultimate Heat Sink Cooling Tower Performance

University of Illinois at Urbana-Champaign

by

S. M. Sullivan

and

W. E. Dunn

Performed for

Mr. Rex Wescott
U.S. Nuclear Regulatory Commission
Washington, DC

PDR: per R. Emmt
DF03 0/1

Reference (1279)

ADD: R. Emmt, RES 0/1

Department of Mechanical and Industrial Engineering

Method for Analysis of Ultimate Heat Sink Cooling Tower Performance

University of Illinois at Urbana-Champaign

by

S. M. Sullivan

and

W. E. Dunn

Performed for

Mr. Rex Wescott
U. S. Nuclear Regulatory Commission
Washington, DC

[illegible]

METHOD FOR ANALYSIS OF ULTIMATE HEAT SINK COOLING TOWER PERFORMANCE

by

S. M. Sullivan

and

W. E. Dunn

of the

Department of Mechanical and Industrial Engineering
University of Illinois at Urbana-Champaign
1206 W. Green Street
Urbana, Illinois 61801
217-333-1176

prepared for

Mr. Rex Wescott
U. S. Nuclear Regulatory Commission
Washington, DC 20555

April, 1986

ABSTRACT

This report developes computer models which can be used in the design and analysis of the cooling tower/basin systems of ultimate heat sinks within nuclear power plants, and demonstrates the ways in which these models are employed to determine the design basis required by the U. S. Nuclear Regulatory Commission Regulatory Guide 1.27.

The tower characteristic value is generated and the tower cold water temperature is predicted at the generated tower characteristic using a mathematical tower submodel. The predicted cold water temperature values are then used to evaluate the manufacturer's performance cold water temperature values.

The time constant of the tower/basin system is approximated by minimizing the sum of squares of a predicted basin temperature and a corresponding temperature value determined assuming a first-order system response. The time constant is used in a data scanning model which scans a long-term weather record from a representative meteorological station to determine the periods of most adverse meteorology for cooling or evaporation. The identified periods are used in an ultimate heat sink tower/basin simulation model to estimate design-basis basin temperature, basin mass, and basin salinity.

ACKNOWLEDGMENTS

The authors wish to express their sincere gratitude to Mr. Rex Wescott of the Nuclear Regulatory Commission for his contribution to the completion of this work. The assistance he provided during the progress of the study as well as the time he took to review the draft report are deeply appreciated.

The authors also wish to thank Dr. Anthony J. Policastro of the Environmental Research Division of Argonne National Laboratory for reviewing the draft report. His many helpful suggestions greatly improved the quality of the finished document.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

NOMENCLATURE

ENGLISH SYMBOLS

a	Surface area per unit volume of tower fill
A	Pond surface area (used in discussion of UHS cooling ponds)
c_a	Specific heat of dry air
c_L	Specific heat of water
c_v	Specific heat of water vapor
c_G	Specific heat at constant pressure of air and water-vapor mixture
e	$\Delta t/t$
E	Equilibrium temperature (used in dicussion of UHS cooling ponds)
f	Fraction of water evaporated
G	Mass flow rate of air
h	Heat transfer coefficient
h_a	Enthalpy of dry air
h_{fg}	Specific heat of vaporization at actual water temperature
h°_{fg}	Specific heat of vaporization at water temperature T_0
h_v	Enthalpy of water vapor
h_D	Mass transfer coefficient
h_G	Enthalpy of moist air
h_{G*}	Nondimensional enthalpy of moist air (used in finite

	difference approximation)
h_L	Enthalpy of liquid water
h_{L*}	Nondimensional enthalpy of water (used in finite difference approximation)
h_1	Enthalpy (at numbered entrance and exit states)
h'_L	Ratio of differential enthalpy to differential temperature $(dh_L(T_L)/dT_L)$
K	Effective heat transfer coefficient
KaV/L	Tower performance characteristic
K'	Effective overall (pond) surface heat transfer coefficient (used in discussion of UHS cooling ponds)
L	Mass flow rate of water
Le	Lewis number
LHF	Latent Heat Factor
L_i	Mass flow rate of water at step "i"
L/G	Liquid over gas flow rate ratio
m	Basin mass
m_i	Basin mass at step "i"
M	Pond mass (used in discussion of UHS cooling ponds)
Nu	Nusselt number
P	Total pressure
P_a	Partial pressure of air
P_v	Partial pressure of water vapor
Pr	Prandlt number
q	Heat transfer rate

q_i	Heat load at step "i"
q_p	Heat transfer rate of the plant
q_s	Rate of surface heat transfer (used in discussion of UHS cooling ponds)
q_t	Heat load of the tower
q_B	Heat load of the basin
q'	Plant heat load (used in discussion of UHS cooling ponds)
Q	Cooling tower heat load (plant heat load)
R	$h_L'(n)/c_L$ (used in tridiagonal matrix solution to finite difference approximation)
R_a	Perfect gas law constant for air
Re	Reynolds number
R_v	Perfect gas law constant for water vapor
s	Basin salinity
S	Sum of squares (used in calculating the system time constant)
Sc	Schmidt number
Sh	Sherwood number
t	Time
t'	Time (used in running wet-bulb temperature integration)
T	Pond temperature (used in discussion of UHS cooling ponds)
T_{act}	Tower cold water temperatures from manufacturerer's

	performance curves
T_i	Basin temperature predicted by program UHSSIM at step "i"
T_{in}	Tower inlet temperature
T_m	Mixed-mean temperature of dry air and water-vapor mixture
T_o	Enthalpy reference temperature
T_L	Temperature of water
$T_{L,i}$	Temperature of water at step "i"
T_{out}	Tower outlet temperature
T_{pred}	Predicted tower cold water temperature using mathematical tower submodel
T_{wb}	Wet-bulb temperature
$T_{wb,i}$	Wet-bulb temperature at step "i"
T_B	Basin temperature
$T_{B,wb}$	Basin temperature due to wet-bulb temperature effects
$T_{B,Q}$	Basin temperature due to plant heat load effects
T'	Pond temperature without heat load (used in dicussion of UHS cooling ponds)
T_i'	Basin temperature predicted by first-order response at step "i"
$\hat{T}_{wb}$	Running wet-bulb temperature (used in program NWS to follow changing wet-bulb temperature)
V	Volume of tower fill
V_*	Nondimensional volume (used in finite difference

approximation)

w Mass flow rate

GREEK SYMBOLS

Δq	Change in heat load
Δt	Change in time
ε	Efficiency $((T_{in}-T_{out})/(T_{in}-T_{wb}))$
ρ_v	Density of water vapor
$\rho_{v,sat}$	Density of saturated water vapor
ρ_G	Density of moist air
Σ	Summation
τ	Time constant of tower/basin system
θ	Excess temperature (used in discussion of UHS cooling ponds)
ω	Absolute humidity
ω_{sat}	Absolute saturation humidity

1. INTRODUCTION

1.1 Ultimate Heat Sink Cooling

This work centers on predicting the thermal performance of cooling towers used as the ultimate heat sink (UHS) of a proposed nuclear power plant. The purpose of the ultimate heat sink is to provide for the safe shutdown and cooldown of the nuclear reactor(s) in the event of a design-basis accident. Since issues of nuclear safety are involved, the analysis set forth herein is conservative in nature in that parameters are selected to provide the worst performance which may be reasonably expected. It is intended that this work can form the basis of a standardized analysis of UHS cooling tower performance.

Stringent standards for the design of the UHS have been set forth by the U. S. Nuclear Regulatory Commission (NRC) in Regulatory Guide 1.27 (1976). The three basic requirements set forth in that document are: (a) the UHS system should be able to dissipate the heat of a design-basis accident, such as a loss-of-coolant accident, of one unit while assuring the concurrent safe shutdown and cooldown of all remaining units; (b) the UHS system must provide a 30-day supply of cooling water at or below the design basis temperature for all safety related equipment; and (c) the UHS system must be capable of operating under two sets of meteorological conditions, the first resulting in the greatest water loss and the second resulting in the worst thermal performance.

The UHS cooling system differs from a conventional cooling system in several key respects, and, thus, the analysis must recognize and properly account for these differences. First, a UHS system must be imper-

vious to natural and man-made disasters. As a result, unusual designs are frequently employed. For example, UHS cooling tower structures must often be built to withstand earthquakes and tornadoes leading to complex air and water flow patterns unlike those of conventional cooling towers. Second, UHS cooling systems are heavily loaded relative to conventional cooling systems since they are intended to operate only under emergency conditions for which plant efficiency is not a consideration. Indeed, the thermal performance of the UHS system is measured in terms of its ability to maintain the temperature of cooling water entering the plant below the design basis limit of safety related equipment. This value is typically far above the temperature of cooling water required for efficient plant operation under normal conditions. Third, the UHS system must be able to operate for a period of at least 30 days without the addition of make-up water. Thus, the ability of the system to maintain an adequate water inventory is an additional constraint not normally employed in the design of conventional cooling systems. Fourth, the analysis must be able to handle off-design conditions such as a reduced number of fans, a reduced water flow rate (loss of a pump), or the effect of unusually high dissolved solids content resulting from the significant evaporative losses which may be expected. Lastly, since the UHS cooling system is intended for emergency use only, the UHS system must be designed to operate reliability on demand after a period of dormancy.

The NRC has found many types of UHS systems to be acceptable. Some examples of these types are: (a) a large body of water such as a river,

a lake, or an ocean, (b) a spray pond with a reservoir or other large body of water, (c) a submerged pond within a larger cooling lake which will remain filled even in the event of a major dam rupture, and (d) a specially designed mechanical-draft cooling tower. Cooling ponds, spray ponds, and cooling towers are the three major types of UHS systems in use today.

1.2 Objectives

The first of several objectives for this work is to establish a standardized method for analyzing the performance of UHS mechanical draft cooling towers. Standardized methods already exist for UHS cooling ponds and UHS spray ponds. The development and use of these latter methods is presented by Codell and Nuttle (1980) for the UHS cooling pond and by Codell (1981) for the UHS spray pond as is more fully discussed in Chapter 2. Standardized methods are valuable because uniformity can be established in determining the limits of system performance across the broad spectrum of possible UHS cooling tower designs. Like the methods developed for UHS cooling ponds and UHS spray ponds, the present method utilizes the entire set of data recorded by the National Weather Service at the site nearest to (and most representative of) the proposed plant. A major obstacle here is to identify, from within this typically massive amount of meteorological data, the periods of worst-case system performance. Development of a reliable selection algorithm is a major focus of the present work as is more fully described later.

The second objective of this work is to develop a realistically conservative analysis which is flexible enough to handle any proposed

UHS cooling tower design. The model must couple an analysis of tower heat and mass transfer characteristics with a prediction of basin temperature, basin mass, and basin dissolved solids content as functions of time. The model must incorporate the time-varying heat rejection rate of the plant and must allow for towers or tower cells to be configured in complex loop arrangements made necessary by the need for redundancy. The model must allow for variable air and water flow rates as well as operating conditions greatly different from the design point.

A third objective of this work is to apply the method to a real-world example. Application of a real-world example will verify the workability of the method and will also provide illustrative examples for other potential users.

The final objective of this work is to make the method easy to apply to many different sites. To accomplish this task, the input to the computer model must be structured in such a way that site dependent data can be used in a convenient manner. Moreover, the form of this data should be that traditionally available to the design engineer.

2. LITERATURE REVIEW

Because of the special considerations attendant with the UHS application, the scope of literature specifically dealing with the design of UHS cooling systems is limited. However, the U.S. Nuclear Regulatory Commission has published methods for the analysis of UHS cooling ponds (Codell and Nuttle, 1980) and methods for the analysis of UHS spray ponds (Codell, 1981). These methods address not only the proper modeling of heat transfer characteristics under accident conditions, but also provide procedures for selecting worst-case meteorological conditions. In addition, Battelle Pacific Northwest Laboratory has collected data by which to characterize the thermal performance and water loss of cooling ponds (Hadlock et al., 1978) and spray ponds (Hadlock et al., 1981) under heavily loaded conditions such as those which exist during UHS operation. Codell (1982) has compared these data with the NRC cooling pond and spray pond models. The methods for analysis of UHS cooling ponds are relevant to the present work not only in terms of general background for the UHS application, but more specifically because they illustrate the level of conservatism and sophistication inherent in the analyses applied to UHS systems other than cooling towers.

The literature dealing with the thermal performance of cooling towers in general is, on the other hand, extensive. As was previously noted, however, the designs of UHS cooling tower systems often differ markedly from conventional designs and the thermal performance analyses must be modified accordingly. A brief survey of the general literature base will be presented herein so as to provide a context for viewing the present work.

2.1 Ultimate Heat Sinks

2.1.1 Cooling Ponds

Codell and Nuttle (1980) describe an analysis of UHS cooling ponds with four major components:

(a) a method for treating surface heat transfer based on the equilibrium temperature concept of Brady, Graves, and Geyer (1969),

(b) three limiting-case models of pond hydrodynamics [fully mixed pond (zero dimensional), plug flow pond (one dimensional horizontally), and stratified pond model (one dimensional vertical)].

(c) a method for scanning a long record of meteorological data to determine the ambient conditions giving worst pond performance, and

(d) a method for comparing the typically long record of offsite data with the shorter but more detailed record of onsite data.

Of these several components, the method used to select the conditions of worst-case pond performance is of greatest relevance to the present work. This method hinges on the use of a linearized form of the surface heat transfer relationship as follows.

$$q_s = K' A (T - E), \tag{2.1}$$

where q_s is the rate of surface heat transfer, T is pond temperature, A is pond surface area, K' is the "effective overall (pond) surface heat transfer coefficient", and E is a parameter commonly known as the "equilibrium temperature". If the linearized form were exact and K were a constant, the "equilibrium temperature" would be the temperature that the pond would reach under steady state environmental conditions without external heat inputs or removals. Since the dependence of heat transfer rate on pond temperature is in fact not linear as supposed above, the equilibrium temperature is merely a device for separating the effects of heat load from the effects of changing ambient conditions. Under the assumptions of this analysis, the pond temperature at any given instant of time is simply the temperature the pond would have attained without heat input (governed by environmental conditions) plus the excess temperature produced by the heat rejection rate of the plant. To see that this is indeed the case, consider the equation of a fully mixed pond with mass M, specific heat c_L, and plant heat load q':

$$M\, c_L\, (dT/dt) = K'\, A\, (E - T) + q'. \qquad (2.2)$$

Since the equation is linear in T (assuming K, A and q are constants independent of T), the solution can written as the sum

$$T = T' + \theta, \qquad (2.3)$$

where T' is the pond temperature without heat load given by the solution of the equation

$$M c_L (dT'/dt) = K' A (E - T') \qquad (2.4)$$

and θ is the excess temperature given by the solution of

$$M c_L (d\theta/dt) = - K' A \theta + q'. \qquad (2.5)$$

The former equation can be solved to determine a value of T' during each hour for which a meteorological observation is available. The starting time of the accident is then selected so that the maximum value of T' and the maximum value of θ occur simultaneously. Thus, T' can be used to identify the period of worst-case thermal performance from within the long record of meteorological conditions.

Since the above analysis is only approximate, Codell recommends that a series of runs be made to study the sensitivity of the maximum pond temperature to the starting time of the accident.

2.1.2 Spray Ponds

The analysis of UHS spray ponds parallels that of the UHS cooling pond discussed above except that the enhanced heat transfer associated with the droplet spray must be added. In addition, the water loss due to drift (removal of spray droplets by the wind) must be accounted for. The basic concept of using a linear approximation to separate the effects of the changing environmental conditions from those of the time-varying heat rejection rate is retained.

In the case of the spray pond, the contribution to the overall heat and water loss arising from the pond surface is small relative to that from the spray. Also, due to the enhanced heat and mass transfer associated with the spray, the UHS spray pond is much smaller in size than a conventional UHS cooling pond. The spray pond is usually regarded as fully mixed.

2.1.3 Model Validation

Codell has compared the predictions of the NRC cooling pond and spray pond with data taken by Battelle Pacific Northwest Laboratory at two small geothermally heated ponds [1982]. These small heavily loaded ponds were considered to be appropriate analogs of UHS cooling ponds and spray ponds.

Battelle considered many sites as candidates for their study. Geothermal sites were chosen after careful consideration because of the availability of large quantities of heated water. Ponds were constructed at two geothermal sites, suitable for surface heat transfer measurements. A spray system was fitted to one of the ponds to allow spray data collection. A complete description of the data collected is given by Hadlock and Abbey [1978, 1981]. The comparisons presented by Codell demonstrate the reliabilility of the surface relationships used to predict surface heat transfer in the NRC models.

2.2 Evaporative Cooling Towers

For over 50 years, the performance of evaporative cooling towers has been estimated by the method of Merkel [1925]. Merkel's method is based on a control volume energy balance at an arbitrary height within

the tower. Merkel's method combines the equations of heat and mass transfer into an enthalpy-difference driving force to allow for both sensible and latent heat transfer. This assumption may be expressed mathematically as follows.

$$q = K\ a\ V\ (h_L - h_G), \qquad (2.6)$$

where q is the heat transfer rate, K is an effective transfer coefficient, a is surface area per unit volume of tower fill, V is the volume of tower fill, h_L is the enthalpy of liquid water, and h_G is the enthalpy of moist air. This relationship is combined with a statement of energy conservation (First Law of Thermodynamics)

$$q = G\ dh_G = L\ c_L\ dT_L, \qquad (2.7)$$

where G is the mass flow rate of air, L is the mass flow rate of water, c_L is the specific heat of water (often omitted since $c_L \cong 1$ Btu/lb_m-F), and T_L is the temperature of water.

Nahavandi and Oellinger [1977] point out that the theory of Merkel requires two major and four minor assumptions. The most important of these assumptions is that the reduction in the water flow rate due to evaporation is negligible. The second major assumption is that the Lewis Number for an air-water-vapor system is one. The four minor assumptions are: (a) the humidity ratio is negligible compared with the ratio of the air-to-vapor gas constants in the expression for the par-

tial pressure of vapor in the moist air, (b) the saturation humidity ratio is negligible compared with the ratio of the air-to-vapor gas constants in the expression for the partial pressure of vapor corresponding to the moist air at the interface temperature, (c) the moist air humid heat, which is the sum of the dry air specific heat and the product of the vapor specific heat and humidity ratio, is considered to be constant, and (d) the calculation of the moist air enthalpy correspon(, to the interface temperature is based on the bulk water temperature rather than the interface temperature.

The enthalpy theory of Merkel has been almost universally accepted due to the fact that dealing rigorously with heat and mass transfer requires a detailed numerical solution. Moreover, the analysis of cooling tower heat and mass transfer rests on the use of empirical data concerning fill performance. Merkel's method provides a simple means for interpreting these data in terms of two non-dimensional parameters KaV/L and L/G. With the availability of the high speed computer, accurate predictions of tower performance may be made and the effects of the assumptions may be accessed.

Nahavandi, et al. [1975] considered the effect of evaporative losses on the performance analysis of the counterflow cooling tower. Nahavandi compared KaV/L calculated using Merkel's method with a result including evaporative losses. Over the range of parameters considered in their study, a maximum relative error greater than 12 percent was found between the methods with Merkel's method giving the greater value. What is not immediately clear from this study however is the

magnitude of the error made in using Merkel's method both for the interpretation of empirical performance data and then in a predictive sense for other operating conditions. It may be argued that the effect would indeed be much less than indicated by the above study.

In a second paper, Nahavandi and Oellinger [1977] consider the effects of all six assumptions. They find that the four minor assumptions and the assumption of the Lewis Number equal to unity have less than 0.5 percent effect on the outcome of the analysis. Nahavandi and Oellinger conclude that the main consideration in improving Merkel's method is to include the effects of evaporative losses on the energy balance.

In the above studies by Nahavandi et al., a finite differnce method is used to solve the heat and mass transfer equations. The tower is divided into N control volumes. A set of finite difference equations are developed linking successive adjacent volumes. The finite difference equations are then solved iteratively.

Yadigaroglu and Pastor [1974] also investigated the effect of the approximations of the Merkel method on cooling tower performance. An error of less than 10 percent was found when comparing the Merkel solution to an "exact" formulation as described in their paper. Yadigaroglu and Pastor found that three assumptions, (i) Le = 1, (ii) the ratio of the gas constants being much greater than the specific humidity, and (iii) no evaporative losses in the water heat balance, were the main contributors to the error. The Le = 1 assumption tends to underestimate water cooling range whereas the other two approximations tend to overestimate water cooling range. The net overestimate of water cooling

range could lead to a non-conservative design, depending on how the method is implemented. Yadegaroglu and Pastor also used a finite difference method to integrate the governing equation across the tower packing.

The effects of the approximations of Merkel have been studied by other investigators as well. Sutherland [1983] found that tower volume could be underestimated by 5 to 15 percent by not considering the effects of evaporation within the tower. Poppe [1973] considered the deviations from Merkel's results in cases when equilibrium is almost reached within the system. Poppe extended the differential equations to take into account condensation when the air states passes through the saturation line into the fog region. Sutherland used a fourth-order Runge-Kutta based shooting method to solve the governing equations.

Electricité de France (see translated report published by the Electric Power Research Institute [1983]) developed a ditigal computer code that solves the formulation developed by Poppe. This computer code, entitled TEFERI, uses the fourth-order Runge-Kukta method of numerical interation. The model is constructed to calculate both the Poppe formulation and the simplified formulation of Merkel. The model was found to predict evaporated water flow rate well and the condensate content of the warm air with good accuracy in a validation study done by Electricité de France. The predictions were compared with full-scale test results from a cooling tower at the Neurath Power Station in the Federal Republic of Germany.

Majumdar et al. [1983] developed a mathematical model which com-

putes two-dimensional distributions of air velocity, air temperature, pressure, water vapor mass fraction and liquid water mass fraction. The local interface heat and mass transfer rates are calculated from empirical correlations using one of two methods. The first method includes one free parameter and follows the treatment of Merkel in which heat and mass transfer are combined; the second method employs two free parameters, treating heat and mass transfer separately. Majumdar et al. use a multi-dimensional finite difference method to solve the flow and energy equations in the tower. This code is licensed by the Electric Power Research Institute as the VERA2D software product.

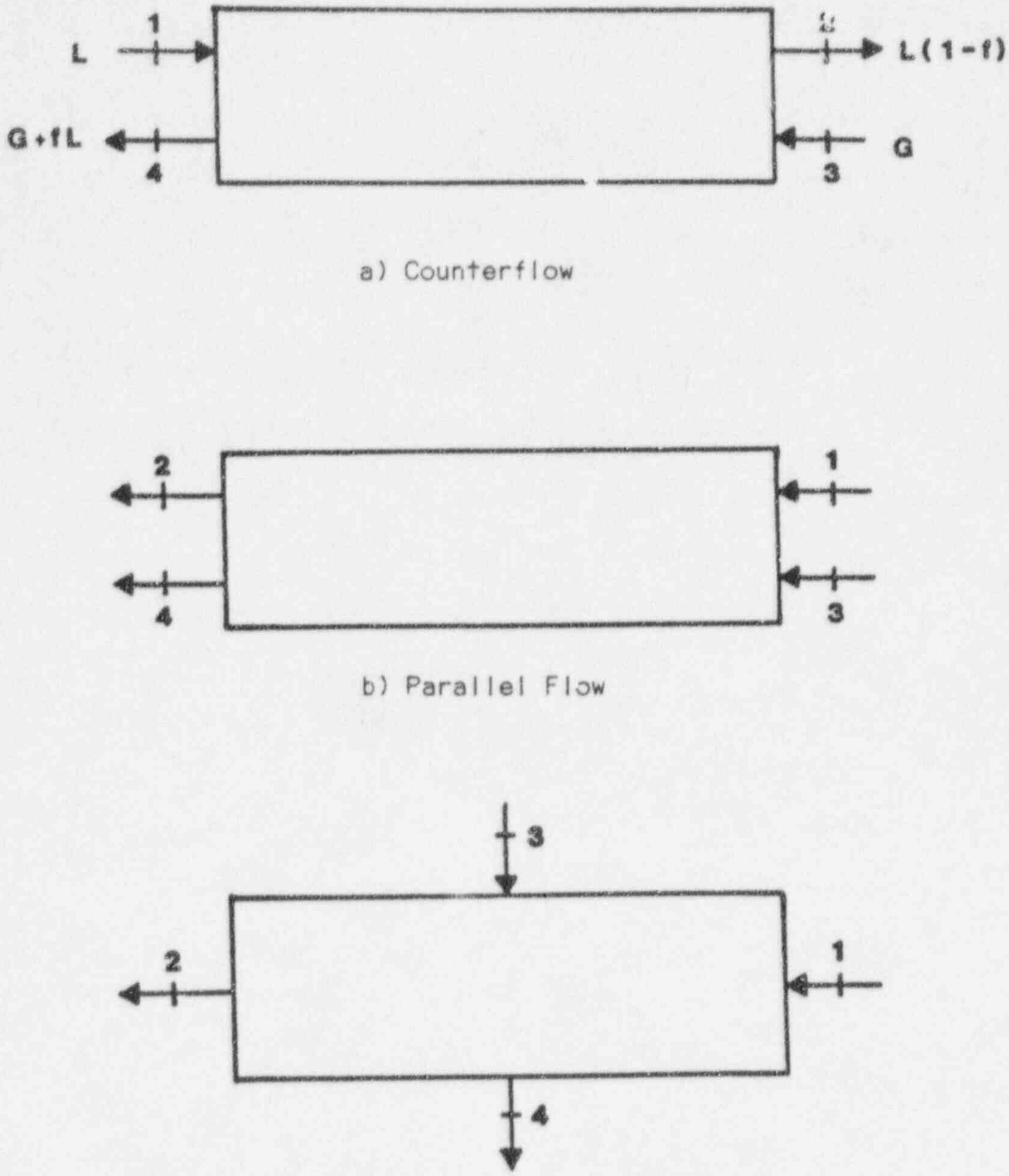
3. FORMULATION OF MODEL

As given by the objectives of the first chapter, the major purpose of this work is to develop a standardized method for analyzing the performance of the UHS cooling tower system that is also flexible enough to allow a realistically conservative analysis of any proposed UHS cooling tower design. This section will discuss the methodology backing the various computer programs which combine to evaluate the performance of a UHS cooling tower system. The descriptions that follow are intended only to give the underlying reasonings of the computer codes, actual application will be discussed in detail in Chapter 4.

The performance analysis can be divided into four distinct groups: the tower submodel, TOWER; the tower/basin model, UHSSIM; the worst-case selection algorithm, NWS; and the Performance Data Analysis Program, PDAP. More specifically, formulation of the tower analysis, the subroutine TOWER, will be developed rigorously from the enthalpy driving force concept of the Merkel Method discussed in the second chapter. The assumptions and approximations discussed earlier in the development of the Merkel equation will be presented as well as a discussion of the finite difference solution technique used in the calculation of the tower outlet temperature. A complete listing of the computer programs can be found in APPENDIX A.

3.1 Tower Submodel

The discussion of the cooling tower model will begin at its simplest form, as a basic heat exchanger. Three types of heat exchanger devices are shown in Figure 3.1. Any one of these three could represent



a) Counterflow

b) Parallel Flow

c) Crossflow

Figure 3.1 Heat Exchanger Devices

simplified cooling tower operation of two fluids transferring heat and mass. Though three different heat exchanger devices are shown, the tower analysis formulated here will be for counterflow direct contact water cooling by air as shown in Figure 3.1a.

3.1.1 Macroscopic Analysis

To begin this tower analysis, consider the cooling tower control volume of Figure 3.1a. Applying the conservation of mass yields

$$w_1 + w_3 = w_2 + w_4, \tag{3.1}$$

where w is the mass flow rate of the fluid at the respective entrance and exit states. The application of the first law of thermodynamics, the conservation of energy, to the control volume, assuming that adiabatic conditions exist at the walls and surfaces and that no work is being done on the system, gives

$$w_1 h_1 + w_3 h_3 = w_2 h_2 + w_4 h_4 , \tag{3.2}$$

where h is the enthalpy of the fluid at the respective entrance and exit states. The inlet conditions at states 1 and 3 are the known values and the exit conditions at states 2 and 4 are the values which must be determined. Consequently w_1, w_3, h_1 and h_3 are known quantities and w_2, w_4, h_2 and h_4 are unknown values. With two equations and four unknowns, two more constraints are needed to solve this problem. These constraints are found using a microscopic analysis of the heat and mass transfer within the tower.

3.1.2 Microscopic Analysis - Merkel Method

The microscopic analysis of this section will consist of developing the differential equations which describe the heat and mass transfer phenomena within an evaporative cooling tower. A segment of infinitesimal volume within the tower is shown in Figure 3.2. A heat balance equation for the water, assuming locally uniform water temperature T_L, is written in differential form as

$$dq = L\, c_L\, (T_L - T_o) - (L - dL)\, c_L\, (T_L - dT_L - T_o), \tag{3.3}$$

where q is the heat transfer rate, L is the mass flow rate of water, c_L is the specific heat of water, T_L is the temperature of water, and T_o is the enthalpy reference temperature. By expanding Equation (3.3) and neglecting products of differential quantities, the water heat-balance equation becomes

$$dq = L\, c_L\, dT_L + c_L\, (T_L - T_o)\, dL. \tag{3.4}$$

A heat balance equation for the air stream is given in differential form as

$$-dq = G\, c_a\, (T_m - T_o) + \omega\, G\, [c_v\, (T_m - T_o) + h^{\circ}_{fg}] - G\, c_a\, (T_m + dT_m - T_o) -$$

$$(\omega + d\omega)\, G\, [c_v\, (T_m - dT_m - T_o) + h^{\circ}_{fg}], \tag{3.5}$$

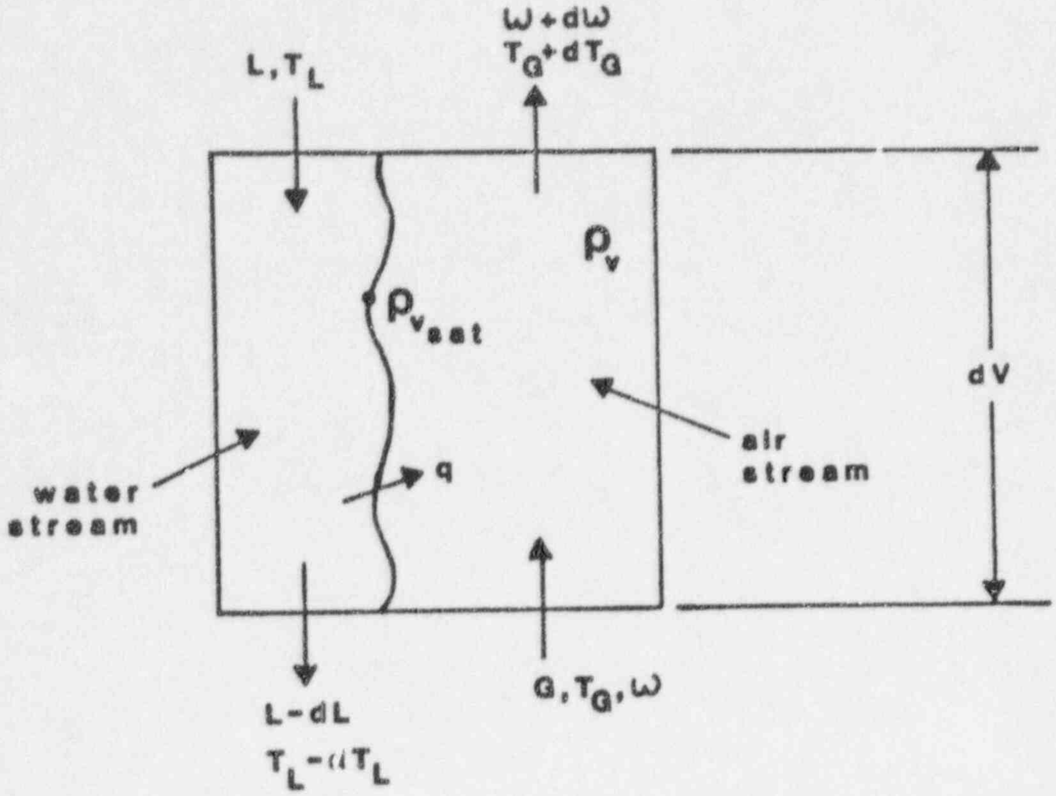


Figure 3.2 Infinitesimal Volume of Cooling Tower Packing

where G is the mass flow rate of air, c_a is the specific heat of dry air, T_m is the mixed-mean temperature of dry air and water-vapor mixture, ω is absolute humidity, c_v is the specific heat of water vapor, and h_{fg}^o is the specific heat of vaporization at temperature T_o. Again expanding and neglecting products of differential quantities, the air heat-balance equation becomes

$$dq = G\,(c_a + \omega\, c_v)\, dT_m + G\,[c_v\,(T_m - T_o) + h_{fg}^o]\, d\omega. \qquad (3.6)$$

The definition of humid air enthalpy,

$$h_G = h_a + \omega\, h_v = c_a\,(T_m - T_o) + \omega\,[c_v\,(T_m - T_o) + h_{fg}^o]\,, \qquad (3.7)$$

where h_G is the enthalpy of moist air, h_a the enthalpy of dry air, and h_v the enthalpy of water vapor, can be differentiated as

$$dh_G = c_a\, dT_m + \omega\, c_v\, dT_m + [c_v\,(T_m - T_o) + h_{fg}^o]\, d\omega \qquad (3.8)$$

and substituted into Equation (3.6) to compact the air stream heat balance equation to

$$dq = G\, dh_G. \qquad (3.9)$$

Equations (3.4) and (3.9) are the heat balance equations for the water and air streams respectively.

Heat is transformed at the interface shown in Figure 3.2 by two mechanisms. The first mechanism is sensible heat convection

$$dq_{conv} = h\ (T_L - T_m)\ a\ dV, \tag{3.10}$$

when h is the heat transfer coefficient, a is the surface air per unit volume of tower fill, and V is the volume of tower fill. The second mechanism is latent heat transported by evaporation

$$dq_{evap} = [h_{fg} + c_L\ (T_L - T_o)]\ dL, \tag{3.11}$$

where h_{fg} is the specific heat of vaporization at the actual water temperature. Equation (3.11) represents the amount of energy transferred during the evaporation of a mass of water dL. The summing of Equations (3.10) and (3.11) yields

$$dq = h\ (T_L - T_m)\ a\ dV + [h_{fg} + c_L\ (T_L - T_o)]\ dL, \tag{3.12}$$

the total heat transfer.

The mass transfer rate, or rate of evaporation, at the air water interface, assuming that at the interface the water vapor is saturated at the liquid temperature, may be written as

$$dL = h_D\ [\rho_{v,sat}(T_L) - \rho_v(T_m)]\ a\ dV\ , \tag{3.13}$$

where h_D is the mass transfer coefficient, $\rho_{v,sat}(T_L)$ is the density of saturated water vapor at temperature T_L, and $\rho_v(T_m)$ is the density of water vapor at temperature T_m.

In Equation (3.13), the driving force for mass transfer is the density difference between water vapor at the interface and water vapor in the humid air stream. In Equation (3.10) the driving force for sensible heat transfer is the temperature difference between the water stream and the air stream. The simplified formulation of Merkel combines the sensible heat transfer and the heat transfer due to evaporation to obtain a single driving force for total heat transfer. The single driving force proposed by Merkel is the difference between the enthalpy of the saturated air at the interface and the enthalpy of the humid air stream.

To begin the formulation of the single driving force concept, consider that since humid air is a mixture of dry air and water vapor both components are assumed to obey the ideal gas law

$$P_v = \rho_v R_v T_m \tag{3.14}$$

and

$$P_a = \rho_a R_a T_m, \tag{3.15}$$

where P_v and P_a are the partial pressures of water vapor and air respectively, ρ_v and ρ_a are the densities of water vapor and air respectively, and R is the perfect gas law constant. These two equations are related by Dalton's Law between total and partial pressures

$$P = P_a + P_v, \tag{3.16}$$

and by the definition of absolute humidity

$$\omega = \rho_v / \rho_a. \tag{3.17}$$

Using Equations (3.14) through (3.17), vapor density can be expressed as

$$\rho_v = \frac{P}{R_v T_m} \left(\frac{\omega}{\omega + R_a / R_v} \right). \tag{3.18}$$

Equation (3.18) can then be substituted into the equation for mass transfer, (3.13), to give

$$dL = \frac{h_D P}{R_v T_L} \left[\frac{\omega_{sat}(T_L)}{\omega_{sat}(T_L) + R_a/R_v} - \left(\frac{T_L}{T_m} \right) \frac{\omega(T_m)}{\omega(T_m) + R_a/R_v} \right] a \, dV , \tag{3.19}$$

where $\omega_{sat}(T_L)$ is the absolute saturation humidity at the water temperature T_L.

To continue, heat and mass transfer are related by the Chilton-Colburn analogy which uses familiar dimensionless groups to relate the heat and mass transfer coefficients,

$$\frac{Nu}{Re \; Pr^{1/3}} = \frac{Sh}{Re \; Sc^{1/3}} , \tag{3.20}$$

where Nu is the Nusselt number, Re the Reynolds number, Pr the Prandlt number, Sc the Schmidt number, and Sh the Sherwood number. Canceling the Reynolds number from each side of Equation (3.20) and manipulating allows the relationship between the heat transfer coefficient and mass transfer coefficient to be given as

$$h_D = \frac{h}{\rho_G c_G} \left(\frac{Pr}{Sc}\right)^{2/3} , \tag{3.21}$$

where ρ_G is the density of the moist air and c_G is the specific heat of moist air. Defining the Lewis number, Le, as the Prandlt number over the Schmidt number allows Equation (3.21) to be rewritten as

$$h_D = \frac{h\ Le^{2/3}}{\rho_G c_G} . \tag{3.22}$$

Substituting this expression into Equation (3.19) gives

$$dL = \left(\frac{h\ Le^{2/3}}{c_G}\right) (P/\rho_G R_v T_L) \left[\frac{\omega_{sat}(T_L)}{\omega_{sat}(T_L) + R_a/R_v} - \left(\frac{T_L}{T_m}\right) \frac{\omega(T_m)}{\omega(T_m) + R_a/R_v}\right] a\ dV. \tag{3.23}$$

To simplify this rather formidable equation four approximations are made. The first assumption is that for an air-water-vapor system the Lewis number is equal to unity. The second is that the temperature ratio T_L/T_m is assumed to be equal to unity. Third, since the ratio

R_a/R_v = 0.622 and a typical value for ω is 0.05, ω_{sat} and ω are eliminated from the denominators of Equation (3.23). And lastly, the group $P/(\rho_G R_v T_L)$ is considered to be equal to unity. With this approximation, Equation (3.23) reduces to

$$dL = \frac{h}{c_G} [\omega_{sat}(T_L) - \omega(T_m)] \text{ a } dV. \tag{3.24}$$

This equation can now be substituted into the equation for total interface heat transfer, Equation (3.12), to yield

$$dq = h\ (T_L - T_m) \text{ a } dV + [h_{fg} + c_L\ (T_L - T_o)] \frac{h}{c_G} [\omega_{sat}(T_L) - \omega(T)] \text{ a } dV, \tag{3.25}$$

which contains only the heat transfer coefficient h.

Since

$$h_{fg} + c_L\ (T_L - T_o) = h^0_{fg} + c_v\ (T_L - T_o), \tag{3.26}$$

Equation (3.25) can be rewritten as

$$dq = \frac{h}{c_G} \{c_G\ (T_L - T_m) + [h^0_{fg} + c_v\ (T_L - T_o)][\omega_{sat}(T_L) - \omega(T_m)]\} \text{ a } dV. \tag{3.27}$$

The variable c_G, the specific heat at constant pressure of air and water vapor mixture, can be written

$$c_G = \frac{\rho_a c_a + \rho_v c_v}{\rho_a + \rho_v} \tag{3.28}$$

or

$$c_G = \frac{c_a + \omega c_v}{1 + \omega} . \tag{3.29}$$

The fifth approximation is to let $1 + \omega \cong 1$ so that

$$c_G \cong c_a + \omega c_v . \tag{3.30}$$

By substituting this expression into the bracketed term in Equation (3.27) and using the definition of specific enthalpy of the air-water-vapor mixture per unit mass of dry air, Equations (3.7) and (3.27) can be written as

$$dq = \frac{h}{c_G} [h_G(\omega_{sat}, T_L) - h_G(\omega, T_m)] \; a \; dV, \tag{3.31}$$

where h/c_G represents a mass transfer coefficient whi ı is usually denoted by K.

The sixth and final assumption to be made consists of neglecting the term $c_L(T_L - T_o)dL$, the change in water flow due to evaporation in the water heat-balance in Equation (3.4). Once this term is neglected,

the three equations of differential heat transfer can be equated, Equations (3.4), (3.9), and (3.31), to yield

$$c_L \; L \; dT_L = K \; [h_G(\omega_{sat}, T_L) - h_G(\omega, T_m)] \; a \; dV = G \; dh_G, \qquad (3.32)$$

which is the Merkel equation for total heat transfer in terms of the enthalpy driving force.

3.1.3 Enthalpy Driving Force Implementation

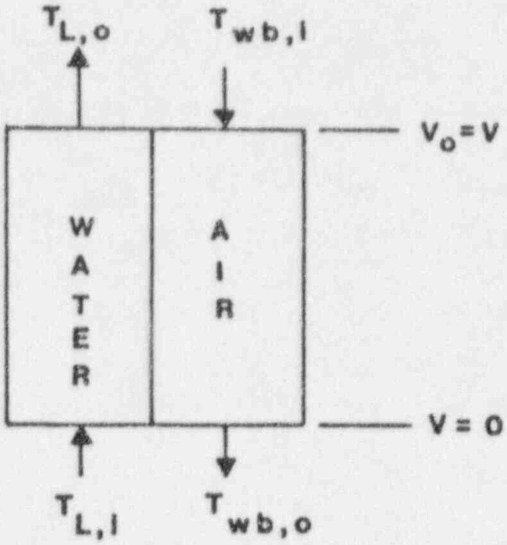
The enthalpy driving force concept derived in the previous section is used to solve for the outlet cold water temperature of the cooling tower in subroutine TOWER. This concept is implemented using a forward time, centered-space finite difference approximation.

The implementation begins with the governing equations of Merkel, Equation (3.32), given above. Figure 3.3a illustrates the essential features of the counterflow cooling tower. The enthalpy driving force for the finite difference analysis is considered to be vapor saturated at temperature T_L and vapor saturated at the wet-bulb temperature. Equation (3.32) can be rewritten as

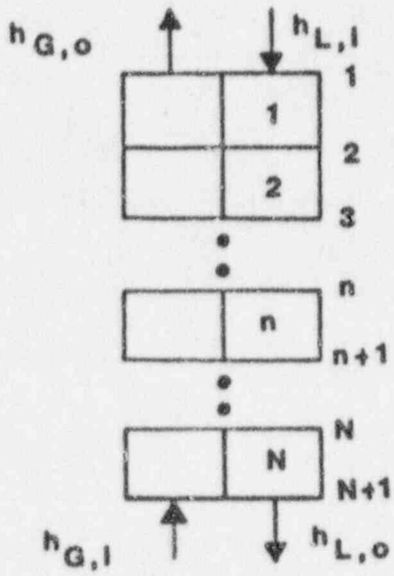
$$c_L \; L \; dT_L = K \; [h_L(T_L) - h_G(T_{wb})] \; a \; dV = G \; dh_G. \qquad (3.33)$$

Nondimensional variables in terms of enthalpies are introduced as

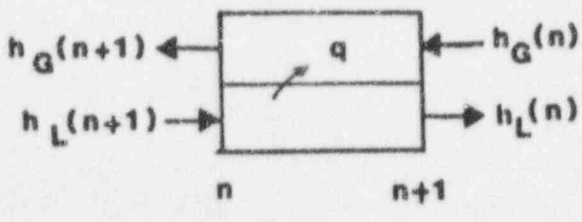
$$h_{G*} = [h_G(T_{wb}) - h_G(T_{wb,i})]/[h_L(T_{L,i}) - h_G(T_{wb,i})] \qquad (3.34)$$



a) Simplified Cooling Tower



b) Discrete Control Volumes



c) Generic Cell

Figure 3.3 Numerical Formulation of Cooling Tower Problem.

for the gas side and

$$h_{L*} = [h_L(T_{L,i}) - h_L(T_L)]/[h_L(T_{L,i}) - h_G(T_{wb,i})] \quad (3.35)$$

for the liquid side, allowing the enthalpy difference of Equation (3.33) to be expressed as

$$h_L(T_L) - h_G(T_{wb}) = (h_L(T_{L,i}) - h_G(T_{wb,i})](1 - h_{L*} - h_{G*}). \quad (3.36)$$

The expression dT_L is rewritten in terms of enthalpy as

$$dh_L(T_L) = (dh_L/dT)_{T_L} dT_L = h_L'(T_L)\ dT_L,$$

or

$$dT_L = dh_L(T_L)\ /\ h_L'(T_L). \quad (3.37)$$

The differential enthalpies are computed in terms of nondimensional parameters as

$$dh_L(T_L) = -[h_L(T_{L,i}) - h_G(T_{wb,i})]\ dh_{L*} \quad (3.38)$$

and

$$dh_G(T_{wb}) = [h_L(T_{L,i}) - h_G(T_{wb,i})]\ dh_{G*}, \quad (3.39)$$

and along with Equations (3.36) and (3.37) are substituted into Equation (3.33) to yield

$$-dh_{G\star} = (L\ c_L)/(G\ h_L'(T_L))\ dh_{L\star}$$

$$= (K\ a)/G\ (h_{L\star} + h_{G\star} - 1)\ dV_\star\ , \qquad (3.40)$$

where $V_\star = V_o/V$.

Equation (3.40) is now approximated numerically using discrete control volumes as shown in Figure 3.3b. Equation (3.33) becomes, using finite difference approximation,

$$\Delta q = K\ a\ [(h_L(n) + h_L(n+1))/2 - (h_G(n) + h_G(n+1))/2]$$

$$= G\ [h_G(n) - h_G(n+1)]$$

$$= L\ c_L\ [(h_L(n) - h_L(n+1))/h_L'(n)]\ , \qquad (3.41)$$

where $h_L'(n)$ is equal to h_L' at $(T_L(n) + T_L(n+1))/2$.

Nondimensionalizing and algebraically simplifying Equation (3.41) yields a tridiagonal set of equations of the form

$$A\ h_{L\star}(n-1) + B\ h_{L\star}(n) + C\ h_{L\star}(n+1) = D \qquad (3.42)$$

for $1 \leq n \leq N+1$, where N is equal to the number of discrete cells or

discrete control volumes. The values of the coefficients A, B, C, and D of Equation (3.42) vary depending on which cell is being considered. The boundry cells must be considered individually because of their initial conditions whereas the interior cells may be considered generically, as shown in Figure 3.3c. For interior cells 2 through N the coefficients for equation (3.42) are

$$A = (K\ a\ \Delta V_*)/(2\ L)[R(n) - (L\ R(n))/(G\ R(n-1))] - R(n)/R(n-1)\ ,$$

$$B = (K\ a\ \Delta V_*)/(2\ L)[R(n)/R(n-1) - L/G] + R(n)/R(n-1) + 1\ ,$$

$$C = (K\ a\ \Delta V_*)/(2\ L)[L/G - R(n)] - 1\ ,$$

and

$$D = 0\ , \qquad (3.43)$$

where $R(n) = h_L'(n)/c_L$ and $R(n-1) = h_L'(n-1)/c_L$. For boundary cells, when n is equal to N+1 the air inlet is known, therefore $h_G = h_{G,i}$ and $h_{G*}(N+1) = 0$. The coefficients for this case are

$$A = 0\ ,$$

$$B = (K\ a\ \Delta V_*)/(2\ L)[R(N) - L/G] - 1\ ,$$

$$C = (K\ a\ \Delta V_*)/(2\ L)[R(N) - L/G] - 1\ ,$$

and

$$D = (K\ a\ \Delta V_*\ R(N))/L. \quad (3.44)$$

When $n = 1$, the water inlet temperature is known, therefore $h_L = h_{L,i}$ and $h_{L*}(1) = 0$. The coefficients for this case are

$$A = 0\ ,$$

$$B = 2\ ,$$

$$C = 1\ ,$$

and $D = 0$. (3.45)

The tridiagonal set of equations generated using the finite difference approximation is solved using the tridiagonal matrix solver TDMS. A complete listing of the subroutine is listed in APPENDIX A for further reference.

3.2 Tower/Basin Model

The second major code to be discussed is the tower/basin model. This model, which includes subroutine TOWER, allows the entire cooling system to be analyzed. A simplified schematic of the plant/tower/basin

layout is shown in Figure 3.4. As can be seen, water is drawn from the basin, heated to a temperature T_{in} which depends on the heat load of the plant, and passed through the cooling towers where air is used in direct contact to remove heat from the water. The integration of each of these parts into a system for analysis is done in program UHSSIM. A flow chart of program UHSSIM showing this integration is given in Figure 3.5. As can be seen from the flow chart of Figure 3.5, program UHSSIM can be divided into three groups.

The first group consists of reading data and fitting some of the data to curves to allow discrete points to be represented as continuous curves. Time-varying values of the parameters KaV/L and L/G along with time-varying values of liquid flow rate and heat load are read and fit to linear curves to allow interpolation to be performed. Time-varying values of wet-bulb temperature are read and fit to a curve using a Hermite-Cubic Spline fit also for interpolation to be easily performed. Other set values such as the number of cooling towers, number of time steps, and step duration are also read in the initial portion of program UHSSIM. The second group in program UHSSIM is the manager. A time loop is started to simulate accident initiation and calls are made to subroutines as required.

The third group consists of subroutines BASIN and TOWER. Subroutine TOWER, as mentioned earlier, is needed to calculate the temperature of the water leaving the cooling towers. Subroutine BASIN simulates the changing basin conditions with time. Two ordinary differential equa-

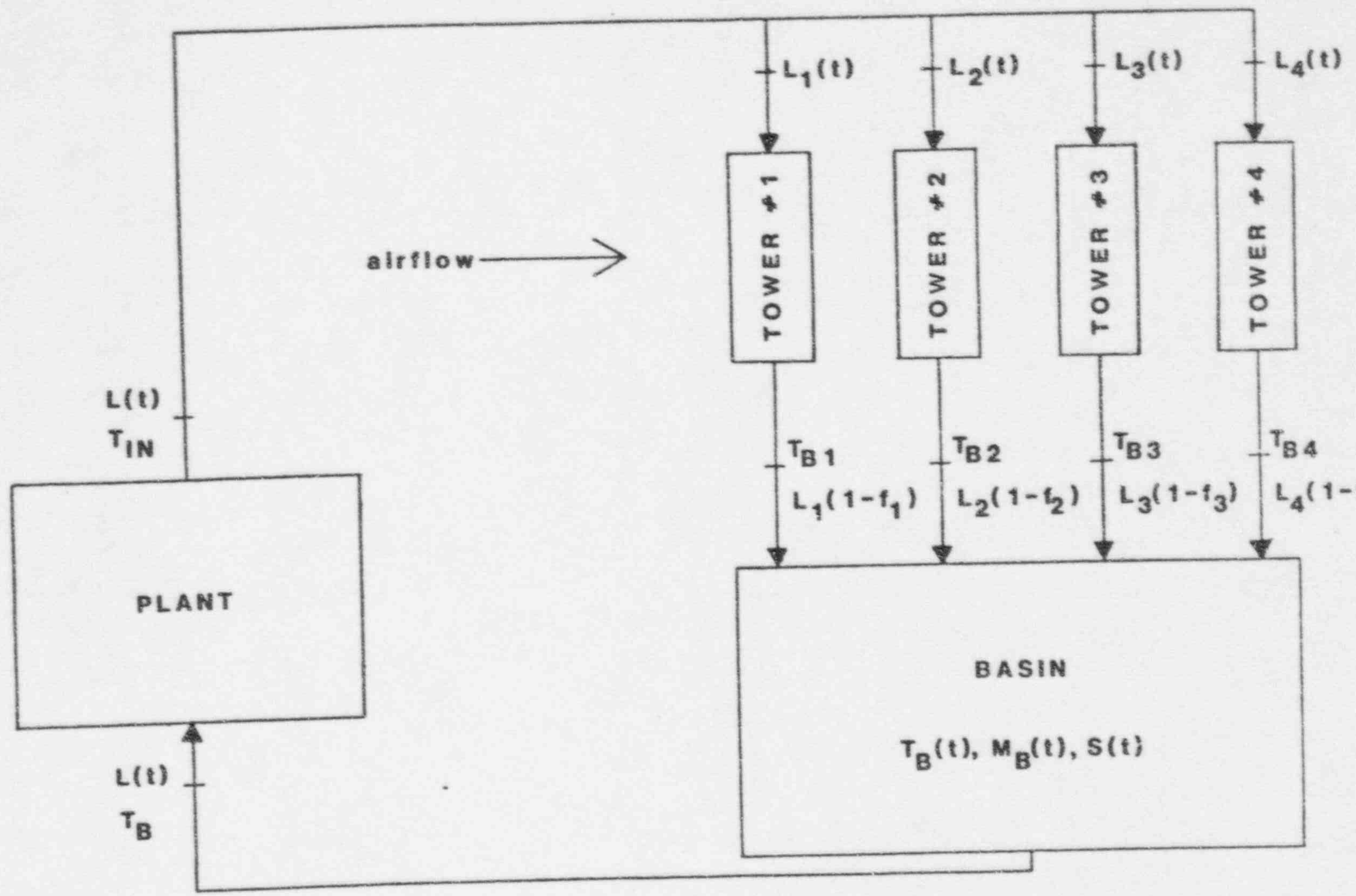
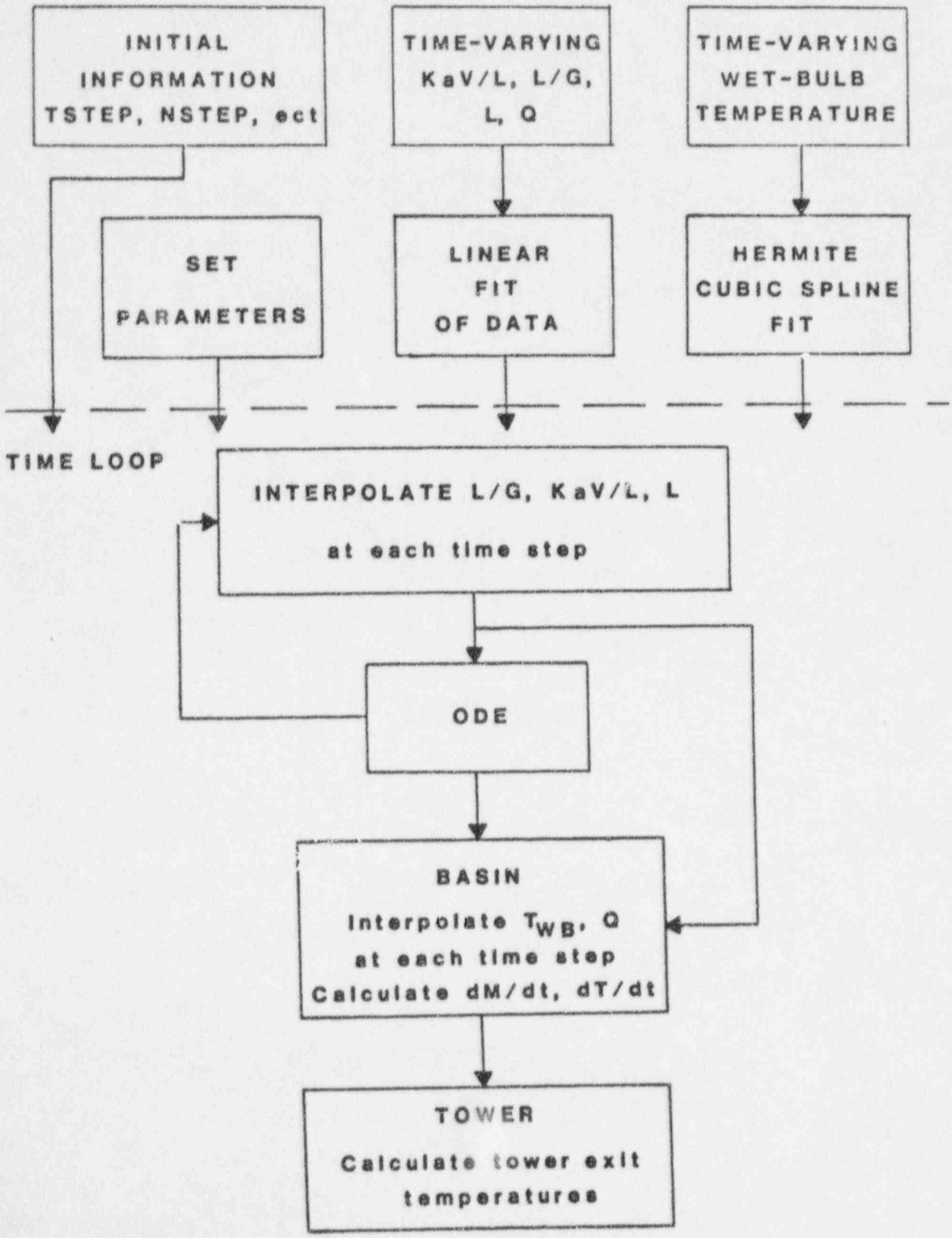


Figure 3.4 Schematic of Plant/Tower/Basin Simulation.

Figure 3.5 Flow Chart of Program UHSSIM.



tions are used to represent the changing basin temperature and changing basin mass at each time step. The changing basin temperature is given by the first law of thermodynamics as

$$\frac{d}{dt} (m\ c_L\ T_B) = \Sigma\ L_j\ c_L\ T_{out,j} - L\ c_L\ T_B\ , \qquad (3.46)$$

where m is basin mass, T_B is basin temperature, L_j is the liquid flow rates of each tower, j, $T_{out,j}$ is the outlet temperature for each tower, j, and L is the initial liquid flow rate. The changing basin mass is given as

$$\frac{dm}{dt} = \frac{q}{h_{fg}} LHF, \qquad (3.47)$$

where q is the heat transfer, h_{fg} is the specific heat of vaporization, and LHF is known as the Latent Heat Factor. This equation can be re-written as

$$\frac{dm}{dt} = \frac{L\ c_L\ (T_{in} - T_{out})\ LHF}{h_{fg}}\ , \qquad (3.48)$$

where T_{in} and T_{out} are tower inlet and outlet temperatures of the tower.

An ordinary differential equation solver called ODE is used to solve Equations (3.46) and (3.48). In this case ODE integrates from T_{in} to T_{out}. Once the differential equations are solved, updated values of basin temperature and basin mass are obtained and the time loop contin-

ues. The changing salinity and dissolved solids concentration is determined directly from the changing basin mass since as basin mass decreases, dissolved solid or salinity content increases. Therefore the changing salinity concentration is found by dividing the updated basin mass valve by the original salinity and dissolved solid content.

3.3 Worst-Case Meteorological Selection Algorithm

The program NWS (National Weather Service) reads meteorological data from National Weather Service Tape Data Family-14 (TDF-14) magnetic tapes. Hourly or three-hourly values of up to 48 meteorological variables are stored on these tapes in an alphanumeric format. This program interprets only the wet-bulb temperature values from the tape. Program NWS scans the tape, finding windows of worst-case meteorological conditions. A unique feature to this procedure is the fact that the windows fold to prevent windows from overlapping and limiting the data range.

Two terms used in the program NWS will be derived here and the remainder of the code will be discussed by way of application in Chapter 4.

3.3.1 Running Wet-Bulb Temperature

The occurrences of worst-case meteorological conditions are found by segregating periods, or windows, of maximum running wet-bulb temperatures on the tape. The running wet-bulb temperature, a weighted average of the present wet-bulb temperature with the previous running wet-bulb temperature, represents the time lag of the system to constantly changing wet-bulb temperatures.

The derivation of the running wet-bulb temperature begins with an energy balance

$$q_B(t) = q_p(t) - q_t(t) \quad (3.49)$$

of the plant/tower/basin system where $q_p(t)$ is the heat transfer of the plant. The heat load of the basin, $q_B(t)$, and the heat load of the tower, $q_t(t)$, can be defined as

$$q_B(t) = m\, c_L\, dT_B/dt \quad (3.50)$$

and

$$q_t(t) = L\, c_L\, \varepsilon\, (T_{in} - T_{wb}) \quad (3.51)$$

respectively, where

$$T_{in} = T_B + q_p(t)/L\, c_L, \quad (3.52)$$

T_{in} is the tower inlet temperature, T_{wb} is the wet-bulb temperature, T_B is the basin temperature, m is the basin mass, c_L is the specific heat of water, L is the liquid flow rate, and ε is the efficiency of the tower given by $(T_{in} - T_{out})/(T_{in}-T_{wb})$ where T_{out} is the tower outlet temperature. Substituting equations (3.50), (3.51), and (3.52) into equation (3.49) yields

$$m\ c_L\ dT_B/dt = q_p(t) - L\ c_L\ \varepsilon\ (T_B + \frac{q_p(t)}{L\ c_p} - T_{wb}). \tag{3.53}$$

The time-dependent basin temperature can be represented by combining plant heat load terms and dividing by $m\ c_L$

$$dT_B/dt = (T_B + T_{wb})/\tau + (1-\varepsilon)\ q_p(t)/(m\ c_L), \tag{3.54}$$

where τ is equal to $\tau = m/(L\ \varepsilon\ c_L)$.

Equation (3.54) shows that the basin temperature will be influenced by the changing ambient wet-bulb temperatures and the temperature due to plant heat rejection represented by the last term in the equation. Therefore, the basin temperature may be written as

$$T_B = T_{B,wb} + T_{B,Q}, \tag{3.55}$$

the sum of the two influences, where $T_{B,wb}$ is the basin temperature due to wet-bulb temperature effects and $T_{B,Q}$ is the basin temperature due to plant heat load effects. Therefore, by the principle of superposition, the basin temperature may be represented by

$$dT_{B,wb}/dt = -T_{B,wb}/\tau + T_{wb}/\tau \tag{3.56}$$

and

$$dT_{B,Q}/dt = -T_{B,Q}/\tau + (1-\varepsilon)\, q_p(t)/(m\, c_L) \qquad (3.57)$$

for wet-bulb temperature effect and for heat load effect, respectively.

To interpret more fully the effects of the varying ambient wet-bulb temperature, Equation (3.56), written in ordinary differential equation form,

$$dT_{B,wb}/dt + T_{B,wb}/\tau = T_{wb}/\tau \qquad (3.58)$$

is solved to yield

$$d/dt\, (\exp(t/\tau)\, T_{B,wb}) = \exp(t/\tau)\, T_{wb}/\tau. \qquad (3.59)$$

Integrating from time t to time zero, and solving for the basin temperature due to the wet-bulb temperature effects gives

$$T_{B,wb} = T_{B,wb}(0) \exp(-t/\tau) + \exp(-t/\tau) \int_0^t \exp(t'/\tau)\, \frac{T(t')}{\tau}\, dt'. \qquad (3.60)$$

From Equation (3.60), the running wet-bulb temperature

$$\hat{T}_{B,wb}(t) = \int_{-T}^{t} \exp\,[(t'-t)/\tau]\, \frac{T_{wb}(t')}{\tau}\, dt' \qquad (3.61)$$

is obtained. The running wet-bulb temperature is written as

$$\frac{d\hat{T}_{B,wb}}{dt} = - \frac{\hat{T}_{B,wb}}{\tau} + \frac{\hat{T}_{wb}}{\tau} \qquad (3.62)$$

for evaluation. Equation (3.62) is expressed in finite difference from as

$$\frac{\hat{T}_{B,wb}(n+1) - \hat{T}_{B,wb}(n)}{\Delta t} = T_{B,wb}(n+1) + \frac{1}{\tau} T_{wb}(n+1) \qquad (3.63)$$

by applying the implicit Backward Euler scheme. The n+1 values are those being calculated, and the n values are known quantities. Equation (3.63) is rewritten as

$$\hat{T}_{B,wb}(n+1) = [\hat{T}_{B,wb}(n) + \frac{\Delta t}{\tau} T_{wb}(n+1)]/(1 + \frac{\Delta t}{\tau}) \qquad (3.64)$$

to calculate the running wet-bulb temperature at each time step within program NWS.

3.3.2 Time Constant

The time constant of the tower/basin system represents the response of the basin temperature to external influences, namely wet-bulb temperature and plant heat load. This system response is assumed to be one of first-order.

The calculation of the time constant begins from an energy balance of the system as shown previously in Equations (3.49) through (3.54). From Equation (3.54) a recursive relationship can be written as

$$(T_{B,i} - T_{B,i-1})/\Delta t = (T_{B,i} + T_{wb,i})/\tau + (1-\varepsilon)\ q_i/(m_i\ c_L), \tag{3.65}$$

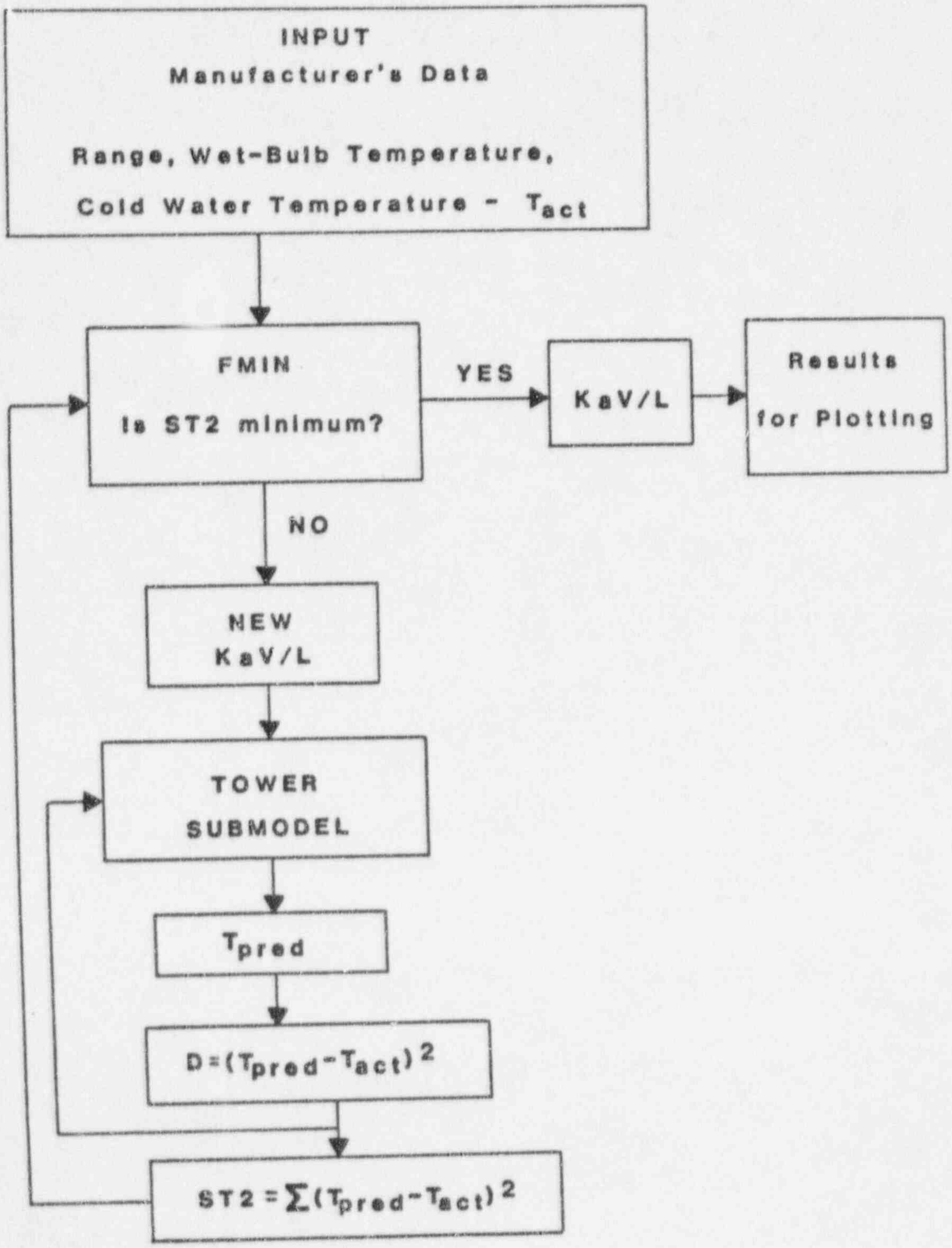
where i represents the current time step and i-1 the previous time step. To determine the time constant, the sum of the squares of the basin temperature $T_{B,i}$ and a temperature predicted using program UHSSIM is minimized. This minimizing procedure is explained more fully in Sections 3.4 and discussed via example in Chapter 4.

3.4 Performance Data Analysis

The Performance Data Analysis Program (PDAP) calculates the tower characteristic KaV/L values and, using these generated KaV/L values, predicts the tower cold water temperature. Wet-bulb temperature values, range values, and cold water temperature values are obtained from manufacturer supplied performance curves, and a comparison of the manufacturer supplied cold water temperature values with predicted cold water temperature values allows verification of the manufacturer's data. A flow diagram for program PDAP is given in Figure 3.6.

As shown on the flow chart of Figure 3.6 the tower submodel is used in program PDAP to predict the tower outlet temperature, represented as T_{pred}. Following the flow charts shows that the actual cold water temperature values, T_{act}, are subtracted from the predicted cold water temperature values and this difference is squared. This difference procedure is repeated for each cold water temperature value available from the manufacturer at a given design flow rate. A simple schematic given in Figure 3.7 shows the two methods of obtaining the cold water tempera-

Figure 3.6 Flow Chart of Performance Data Analysis Program.



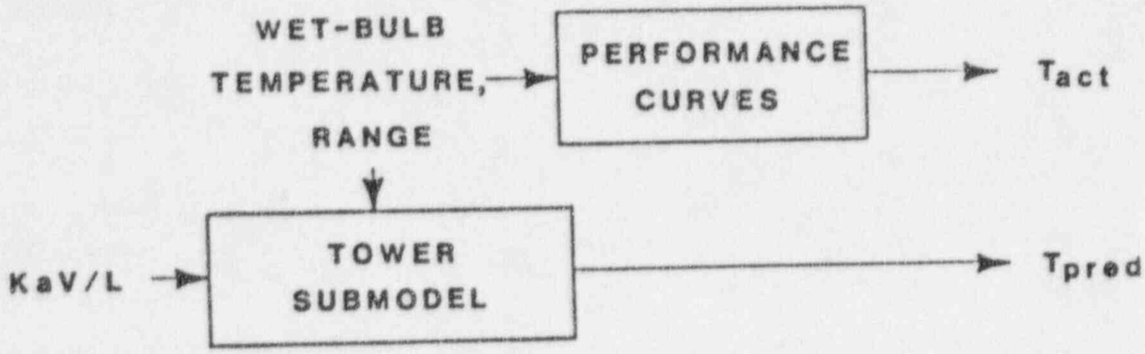


Figure 3.7 Methods of Obtaining Cooling Tower Cold Water Temperature.

Figure 3.8 Function FMIN · Minimum KaV/L Calculation.

ture values. The actual cold water temperature is obtained directly from manufacturers performance curves and the predicted cold water temperature from the tower submodel as stated earlier.

The tower characteristic KaV/L valve used in the tower submodel is shown in Figure 3.6. The function routine FMIN finds an approximate KaV/L valve when a second function, ST2, attains a minimum within a chosen interval. The function ST2 is the summation of the difference of the actual and predicted cold water temperature values as described above. When the minimum summed value is obtained, the error in the predicted cold water temperature is minimized. The KaV/L valves used to calculate the predicted cold water temperature when the minimum summation is obtained is defined as the KaV/L value for the tower system. Figure 3.8 illustrates this minimizing process. The interval of KaV/L value to be searched by function FMIN is determined by the user as a reasonable range for the KaV/L value to lie.

4. DESCRIPTION OF COMPUTER PROGRAMS AND SAMPLE PROBLEM

The purpose of this section is two-fold. First, the various computer programs written to implement the methodology set forth in Chapter 3 are described. Second, the application of the methodology is illustrated by way of a test case or example. The parameters for this test case were selected so as to be representative of an actual UHS design evaluation.

The following information is required to begin an evaluation of a proposed UHS cooling tower system.

1) The configuration of the UHS cooling tower system. This includes information on the number and size of towers to be employed, the liquid and air flow paths, and the basin volume and initial salinity.
2) The design-basis operating scenario for the towers. This includes a summary of the postulated operating conditions such as the number of pumps and fans assumed to be operable, the time-varying heat loads to the system, and the time-varying water and air flow rates to be employed.
3) The maximum allowable temperature of the cooling water returned to the plant. This value is determined based on the operational limits of safety-related equipment within the plant and, for the purposes of the UHS system performance evaluation, may be regarded as a given.
4) The full history of meteorological conditions representative of site. Typically, this consists of 30 or more years of observa-

tions at the nearest National Weather Service reporting station. This data is available on computer readable magnetic tape from the National Climatic Center. In choosing the NWS station, the primary consideration is usually proximity to the proposed plant; however, the similarity of ambient temperature and humidities is actually the important consideration. For example, coastal sites may be typically cooler and more humid than an inland station for which NWS data are available.

The application of the UHS system performance analysis is a step-by-step process with several important evaluation points along the way. Critical evaluation of the results at each of these junctures is essential to proper application of the methodology. Indeed, although the analysis presented herein is systematic and the computer codes are intended to be easy to apply, the importance of engineering judgment in evaluating the output of the various programs cannot be overstated.

A flowchart of the entire process is given in Figure 4.1. The left-hand side of Figure 4.1 shows the essential inputs described above. The various other boxes represent program modules, key intermediate and final results, and major evaluation points. The discussion of each module will focus first on describing the inputs and outputs to the module. This discussion will be followed in turn by the application of the corresponding program to the test case. Thus, the test case will serve to clarify any particularly confusing points which arise. The results of the test case will also be used to confirm the validity of the methodology used to determine the period of worst-case tower performance.

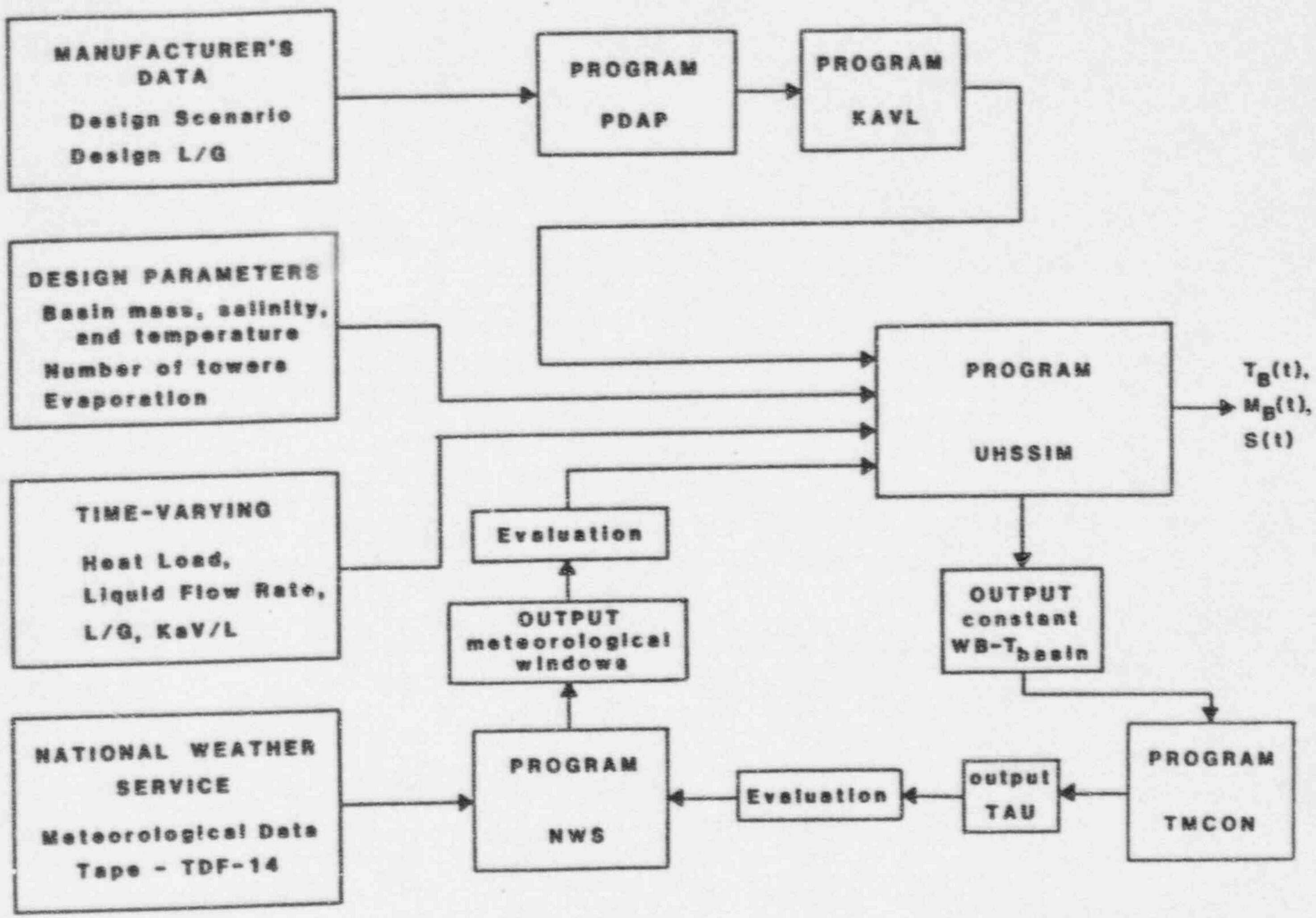


Figure 4.1 Flow Chart of UHS Cooling Tower Performance Analysis.

4.1 Description of the Sample Problem

The goal of a UHS performance analysis is to determine whether the proposed cooling system may be reasonably expected to provide (a) sufficient cooling that the temperature limit of safety-related equipment in the plant is not exceeded and (b) sufficient basin capacity that operation of the UHS system for a period of 30 days without make-up water is assured.

The hypothetical plant to be analyzed as our test case consists of a single nuclear unit of 1000 MW capacity. The UHS system consists of a single cooling tower of the design shown in Figure 4.2. The unusual physical structure of the tower reflects the fact that it is designed to meet seismic category 1 criterion. The tower consists of four cells, each of which employs five fans. The design air flow rate of each fan is 116,440 cubic feet per minute. The four cells are organized into two independent water loops, each loop with a design water flow rate of 16,500 gallons per minute. Although both loops can be operated simultaneously, the design-basis scenario postualted for the test case assumes that only two of the four cells are utilized. The design-basis accident scenario further assumes a water flow rate of 12,150 gallons per minute or 73.6 percent of design flow rate for the first 24 hours following an accident and a flow rate of 11,610 gallons per minute or 70.4 percent of design flow rate thereafter. The essential design parameters of the UHS cooling-tower/basin system are summarized in Table 4.1. The time-dependent heat loads and cooling water flow rates postulated for the accident are presented in Table 4.2.

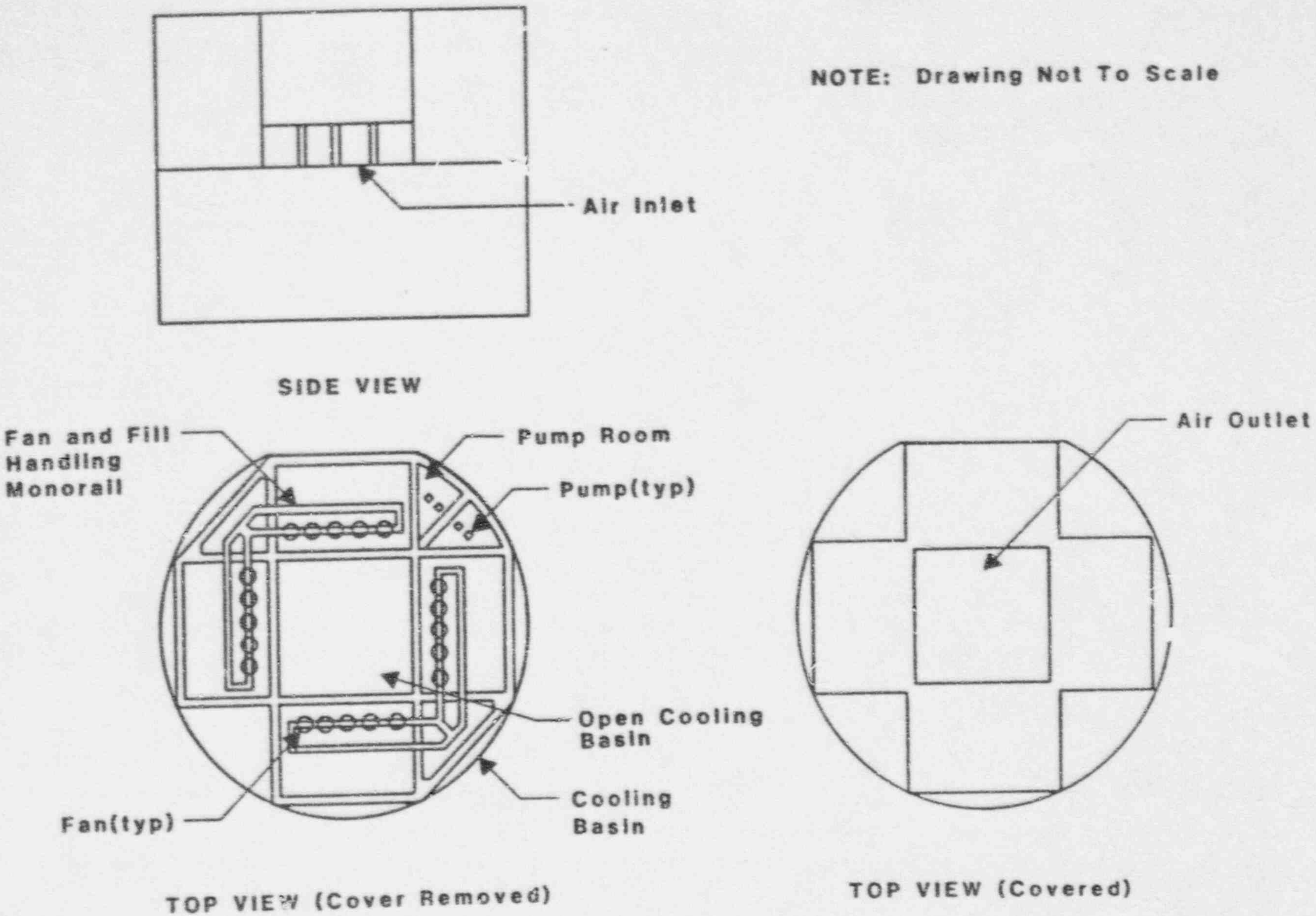


Figure 4.2 Schematic View of UHS Cooling Tower for Test Case.

Table 4.1 Design Parameters for Test Case UHS Cooling Tower

Design Circulating Water Flow Rate	33,000 gpm total in two loops 16,500 gpm in each loop
L/G	1.59*
KAV/L	1.30*
Design Wet Bulb Temperature	81 °F
Design Approach	12 °F
Design Range	23 °F
Number of Cells	4
Number of Fans per Cell	5
Air Flow Rate per Fan	116,440 cfm
Water Volume in Basin (initial)	6,500,000 gallons
Water Mass in Basin (initial)	54,200,000 lb_m
Dissolved Solids Content of Basin Water	Mississippi River Water
Maximum Allowable Return Temperature to Plant	93 °F

---

* Determined from manufacturer's performance curves.

Table 4.2 Heat Load and Cooling Water Flow Rate During LOCA.
(Adapted From Final Safety Analysis Report.)

Time	Time (sec)	Heat Load (Btu/hr)	Service Water Flow (gpm)
0	0	6.62×10^7	12150
0.5	1.8×10^3	1.93×10^8	12150
1.0	3.6×10^3	1.93×10^8	12150
1.5	5.4×10^3	1.93×10^8	12150
2.0	7.2×10^3	1.93×10^8	12150
2.5	9.0×10^3	1.93×10^8	12150
3	1.08×10^4	1.93×10^8	12150
4	1.44×10^4	1.93×10^8	12150
5	1.80×10^4	1.93×10^8	12150
6	2.16×10^4	1.93×10^8	12150
8	2.88×10^4	1.93×10^8	12150
10	3.6[illegible] $\times 10^4$	1.93×10^8	12150
12	4.32×10^4	1.90×10^8	12150
16	5.76×10^4	1.82×10^8	12150
20	7.20×10^4	1.78×10^8	12150
24	8.64×10^4	1.31×10^8	12150
2 days	1.73×10^5	9.54×10^7	11610
3	2.59×10^5	8.86×10^7	11610
4	3.46×10^5	8.41×10^7	11610
5	4.32×10^5	8.11×10^7	11610
6	5.18×10^5	7.76×10^7	11610
7	6.05×10^5	7.62×10^7	11610
8	6.91×10^5	7.46×10^7	11610
9	7.78×10^5	7.31×10^7	11610
10	8.64×10^5	7.12×10^7	11610
15	1.29×10^6	6.70×10^7	11610
20	1.73×10^6	6.40×10^7	11610
25	2.16×10^6	6.16×10^7	11610
30	2.59×10^6	6.04×10^7	11610

4.2 Performance Data Analysis Program

The purpose of the performance data analysis program (PDAP) is to determine from the manufacturer's data the characteristic parameter values for the tower(s) to be used. The program also allows a comparison between manufacturer's performance predictions for off-design operation with the prediction of the numerical tower performance simulation used in the UHS system analysis procedure. The program determines KaV/L values for each of the operating conditions for which performance data are available. Ideally, such data should be available for a full spectrum of water and air flow rates but the degree to which such information is available varies from case to case. In some cases, the information presented is merely predictions by the manufacturer using proprietary data on fill characteristics and, in many instances, proprietary methods of analysis. For this reason, it is important that the performance of the system be re-evaluated when cooling tower acceptance data are available. When only KaV/L values or single "design point" data are presented, it may be generally assumed that KaV remains relatively constant for a reasonable range of near-design-point conditions. If flow rates (water and/or air) during accident operation fall below design conditions, a penalty on KaV/L is appropriate. The magnitude of this penalty must, unfortunately, be based on best engineering judgment.

Inputs to the performance data analysis program are listed in Table 4.3 in terms of variable name, variable definition, and variable units. The first line of the input file is a alphanumeric "case" label used to identify the output. The remainder of the input data are broken

Table 4.3 Descri ·tion of Input Variables for Program PDAP

VARIABLE NAME	DESCRIPTION AND UNITS
CASE	80-Column Header Used to Identify Output File
FLOW	80-Column Header Used to Identify Liquid Flow Rates Within the Output
LOVERG	Liquid to Gas Flow Rate Ratio
PERC	Data Liquid Flow Rate Percent
S	Initial Basin Salinity
NRAN	Number of Range Values Within a Flow Rate
RLABEL	80-Column Header Used to Identify the Temperature Values Obtained at a Specific Range Value
RAN	Array Containing the Range Values (F)
NWB	Array Containing the Number of Wet-Bulb Temperature Values Within Each Range Value
WBD	Wet-Bulb Temperature Data (F)
TCOLD	Cold Water Temperature (F)

into groups according to percent flow rate. Within each "flow rate" grouping, the first entry is the alphanumeric "flow rate" label used to identify the flow rate on output. The "flow rate" label is then followed by three numerical entries: the L/G value, the percent flow rate, and the salinity of the cooling water under test conditions. Following the conventional practice in giving tower performance data, the values are further subdivided into groups according to cooling range, with cold water temperature tabulated against ambient wet-bulb temperature within each given range. Program PDAP uses the total body of data for all ranges at a given flow rate to determine a corresponding KaV/L value. This value of KaV/L is used, in turn, to predict cold water temperature at each of the operating points (range and ambient wet-bulb temperature) given in the input. A comparison between reported and predicted cold water temperatures provides a test of the ability of the tower model to reproduce the performance characteristics of the tower.

The performance data available for the hypothetical test case are given in Table 4.4 with the corresponding input file to program PDAP shown in Figure 4.3. The input file shown in Figure 4.3 is divided across three pages for the purpose of presentation here. As indicated, data are available for water flow rates ranging from 50 percent to 110 percent of the design value.

Figure 4.4 shows the printed output of the PDAP run for the test case; comparisons of predicted and reported cold water temperatures are shown in Figure 4.5 - 4.11. As was done for the input file, the output is continued over four pages for the sake of presentation. The numeri-

Table 4.4 Performance Data for Test Case. (WB - Wet-Bulb Temperature; CW - Cold Water Temperature)

	PERCENT LIQUID FLOW RATE													
	50		60		70		80		90		100		110	
L (gpm)	8250		9900		11550		13200		14850		16500		18150	
L/G	0.793		0.952		1.111		1.270		1.428		1.578		1.745	
Range(F)	WB	CW	WB	CW	WB	CW	WB	CW	WB	CW	WB	CW	WB	CW
15	40	56.0	40	58.6	40	61.2	40	63.5	40	65.8	40	67.9	40	69.9
	45	59.1	45	61.4	45	63.9	45	66.0	45	68.1	45	70.1	45	72.0
	50	62.1	50	64.4	50	66.8	50	68.8	50	70.8	50	72.5	50	74.4
	55	65.75	55	67.7	55	69.7	55	71.6	55	73.3	55	75.0	55	76.7
	60	69.0	60	71.0	60	72.8	60	74.5	60	76.1	60	77.8	60	79.3
	65	72.7	65	74.5	65	76.1	65	77.8	65	79.0	65	80.4	65	82.1
	70	76.3	70	77.9	70	79.4	70	80.8	70	82.0	70	83.2	70	84.9
	75	80.05	75	81.7	75	83.0	75	84.3	75	85.3	75	86.4	75	87.9
	80	84.1	80	85.5	80	86.6	80	87.7	80	88.7	80	89.9	80	91.0
	85	88.5	85	89.5	85	90.4	85	91.6	85	92.2	85	93.2	85	94.3
20	40	59.1	40	62.2	40	65.0	40	67.7	40	70.1	40	72.2	40	74.5
	45	61.8	45	64.7	45	67.4	45	69.8	45	72.1	45	74.2	45	76.4
	50	64.7	50	67.4	50	69.9	50	72.2	50	74.3	50	76.2	50	78.3
	55	67.9	55	70.4	55	72.5	55	74.7	55	76.7	55	78.8	55	80.3
	60	70.9	60	73.2	60	75.3	60	77.2	60	79.2	60	81.0	60	82.7
	65	74.2	65	76.3	65	78.3	65	80.0	65	81.8	65	83.2	65	84.8
	70	77.7	70	79.4	70	81.3	70	82.9	70	84.5	70	85.9	70	87.5
	75	81.2	75	83.0	75	84.5	75	85.9	75	87.5	75	88.5	75	90.1
	80	84.95	80	86.5	80	87.9	80	89.1	80	90.7	80	91.4	80	93.0
	85	88.9	85	90.4	85	91.7	85	92.8	85	93.7	85	94.8	85	96.0
25	40	61.5	40	64.9	40	68.0	40	70.8	40	73.3	40	75.6	40	78.0
	45	64.0	45	67.2	45	70.2	45	72.7	45	75.2	45	77.5	45	79.7
	50	66.7	50	69.6	50	72.3	50	74.9	50	77.2	50	79.2	50	81.2
	55	69.7	55	72.3	55	74.9	55	77.2	55	79.4	55	81.2	55	83.2
	60	72.4	60	75.0	60	77.4	60	79.4	60	81.4	60	83.2	60	85.2
	65	75.5	65	77.8	65	80.1	65	82.1	65	83.9	65	85.3	65	87.3
	70	78.8	70	80.7	70	82.9	70	84.7	70	86.3	70	87.8	70	89.7
	75	82.1	75	83.9	75	86.0	75	87.3	75	89.1	75	90.2	75	91.9
	80	85.9	80	87.3	80	89.0	80	90.3	80	91.8	80	92.9	80	94.6
	85	89.7	85	91.0	85	92.4	85	93.9	85	94.7	85	96.1	85	97.2
30	40	63.5	40	67.2	40	70.3	40	73.3	40	75.9	40	78.3	40	80.7
	45	65.9	45	69.2	45	72.2	45	75.1	45	77.5	45	79.9	45	82.2
	50	68.4	50	71.5	50	74.4	50	77.0	50	79.4	50	81.5	50	83.7
	55	71.1	55	73.9	55	76.7	55	79.0	55	81.3	55	83.1	55	85.3
	60	73.7	60	76.3	60	78.9	60	81.0	60	83.3	60	85.05	60	87.2
	65	76.7	65	79.1	65	81.5	65	83.4	65	85.4	65	87.0	65	88.9
	70	79.8	70	81.8	70	84.0	70	85.9	70	87.8	70	89.1	70	91.2
	75	83.0	75	84.9	75	86.8	75	88.6	75	90.3	75	91.5	75	93.4
	80	86.3	80	88.1	80	89.9	80	91.3	80	92.9	80	94.0	80	95.9
	85	90.2	85	91.6	85	93.2	85	94.5	85	95.7	85	97.0	85	98.3

```
TEST CASE
50 % OF DESIGN FLOW RATE (8250 GPM)
0.7935,50,0.0
4
RANGE = 15
15 10
40 56.0
45 59.1
50 62.1
55 65.75
60 69.0
65 72.7
70 76.3
75 80.05
80 84.1
85 88.5
RANGE = 20
20 10
40 59.1
45 61.8
50 64.7
55 67.9
60 70.9
65 74.2
70 77.7
75 81.2
80 84.95
85 88.9
RANGE = 25
25 10
40 61.5
45 64.0
50 66.7
55 69.7
60 72.4
65 75.5
70 78.8
75 82.1
80 85.9
85 89.7
RANGE = 30
30 10
40 63.5
45 65.9
50 68.4
55 71.1
60 73.7
65 76.7
70 79.8
75 83.0
80 86.3
85 90.2
60 % OF DESIGN FLOW RATE (9900 GPM)
0.9522,60,0.0
4
RANGE = 15
15 10
40 58.6
45 61.4
50 64.4
55 67.7
60 71.0
65 74.5
70 77.9
75 81.7
80 85.5
85 89.5
RANGE = 20
20 10
40 62.2
45 64.7
50 67.4
55 70.4
60 73.2
65 76.3
70 79.4
75 83.0
80 86.5
85 90.4
RANGE = 25
25 10
40 64.9
45 67.2
50 69.6
55 72.3
60 75.0
65 77.8
70 80.7
75 83.9
80 87.3
85 91.0
RANGE = 30
30 10
40 67.2
45 69.2
50 71.5
55 73.9
60 76.3
65 79.1
70 81.8
75 84.9
80 88.1
85 91.6
70 % OF DESIGN FLOW RATE (11550 GPM)
1.111,70,0.0
4
RANGE = 15
15 10
40 61.2
45 63.9
50 66.8
55 69.7
60 72.8
65 76.1
70 79.4
75 83.0
80 86.6
85 90.4
RANGE = 20
20 10
40 65.0
45 67.4
50 69.9
55 72.5
60 75.3
65 78.3
70 81.3
75 84.5
80 87.9
85 91.7
RANGE = 25
25 10
40 68.0
```

Figure 4.3 Input Listing for Program PDAP.

```
45 70.2
50 72.3
55 74.9
60 77.4
65 80.1
70 82.9
75 86.0
80 89.0
85 92.4
RANGE = 30
30 10
40 7[illegible].3
45 72.2
50 74.4
55 76.7
60 78.9
65 81.5
70 84.0
75 86.8
80 89.9
85 93.2
80 % OF DESIGN FLOW RATE (13200 GPM)
1.270,80,0.0
4
RANGE = 15
15 10
40 63.5
45 66.0
50 68.8
55 71.6
60 74.5
65 77.8
70 80.8
75 84.3
80 87.7
85 91.6
RANGE = 20
20 10
40 67.7
45 69.8
50 72.2
55 74.7
60 77.2
65 80.0
70 82.9
75 85.9
80 89.1
85 92.8
RANGE = 25
25 10
40 70.8
45 72.7
50 74.9
55 77.2
60 79.4
65 82.1
70 84.7
75 87.3
80 90.3
85 93.9
RANGE = 30
30 10
40 73.3
45 75.1
50 77.0
55 79.0
60 81.0
65 83.4
70 85.9
75 88.6
80 91.3
85 94.5
90 % OF DESIGN FLOW RATE (14850 GPM)
1.428,90,0.0
4
RANGE = 15
15 10
40 65.8
45 68.1
50 70.8
55 73.3
60 76.1
65 79.0
70 82.0
75 85.3
80 88.7
85 92.2
RANGE = 20
20 10
40 70.1
45 72.1
50 74.3
55 76.7
60 79.2
65 81.8
70 84.5
75 87.5
80 90.7
85 93.7
RANGE = 25
25 10
40 73.3
45 75.2
50 77.2
55 79.4
60 81.4
65 83.9
70 86.3
75 89.1
80 91.8
85 94.7
RANGE = 30
30 10
40 75.9
45 77.5
50 79.4
55 81.3
60 83.3
65 85.4
70 87.8
75 90.3
80 92.9
85 95.7
100 % OF DESIGN FLOW RATE (16500 GPM)
1.578,100,0.0
4
RANGE = 15
15 10
40 67.9
45 70.1
50 72.5
55 75.0
60 77.8
65 80.4
```

Figure 4.3 Continued

```
70 83.2
75 86.4
80 89.9
85 93.2
RANGE = 20
20 10
40 72.2
45 74.2
50 76.2
55 78.8
60 81.0
65 83.2
70 85.9
75 88.5
80 91.4
85 94.8
RANGE = 25
25 10
40 75.6
45 77.5
50 79.2
55 81.2
60 83.2
65 85.3
70 87.8
75 90.2
80 92.9
85 96.1
RANGE = 30
30 10
40 78.3
45 79.9
50 81.5
55 83.1
60 85.05
65 87.0
70 89.1
75 91.5
80 94.0
85 97.0
110 % OF DESIGN FLOW RATE (18150 GPM)
1.745,110,0.0
4
RANGE = 15
15 10
40 69.9
45 72.0
50 74.4
55 76.7
60 79.3
65 82.1
70 84.9
75 87.9
80 91.0
85 94.3
RANGE = 20
20 10
40 74.5
45 76.4
50 78.3
55 80.3
60 82.7
65 84.8
70 87.5
75 90.1
80 93.0
85 96.0
RANGE = 25
25 10
40 78.0
45 79.7
50 81.2
55 83.2
60 85.2
65 87.3
70 89.7
75 91.9
80 94.6
85 97.2
RANGE = 30
30 10
40 80.7
45 82.2
50 83.7
55 85.3
60 87.2
65 88.9
70 91.2
75 93.4
80 95.9
85 98.3
```

Figure 4.3 Continued

```
 TEST CASE
50 % OF DESIGN FLOW RATE (8250 GPM)
  1.9040    .7935    50.0  0.0000
4
RANGE = 15
   15.0         10.0
  40.0000    56.0000    56.0151      .0151
  45.0000    59.1000    59.1455      .0455
  50.0000    62.1000    62.2948      .1948
  55.0000    65.7500    65.7348     -.0152
  60.0000    69.0000    69.1075      .1075
  65.0000    72.7000    72.6933     -.0067
  70.0000    76.3000    76.3383      .0383
  75.0000    80.0500    80.1192      .0692
  80.0000    84.1000    84.0622     -.0378
  85.0000    88.5000    88.1604     -.3396
RANGE = 20
   20.0         10.0
  40.0000    59.1000    59.1671      .0671
  45.0000    61.8000    61.9394      .1394
  50.0000    64.7000    64.8431      .1431
  55.0000    67.9000    67.9080      .0080
  60.0000    70.9000    71.0004      .1004
  65.0000    74.2000    74.2679      .0679
  70.0000    77.7000    77.6840     -.0160
  75.0000    81.2000    81.2182      .0182
  80.0000    84.9500    84.9232     -.0268
  85.0000    88.9000    88.7908     -.1092
RANGE = 25
   25.0         10.0
  40.0000    61.5000    61.6569      .1569
  45.0000    64.0000    64.1905      .1905
  50.0000    66.7000    66.8527      .1527
  55.0000    69.7000    69.6726     -.0274
  60.0000    72.4000    72.5180      .1180
  65.0000    75.5000    75.5636      .0636
  70.0000    78.8000    78.7666     -.0334
  75.0000    82.1000    82.1050      .0050
  80.0000    85.9000    85.6554     -.2446
  85.0000    89.7000    89.3568     -.3432
RANGE = 30
   30.0         10.0
  40.0000    63.5000    63.6764      .1764
  45.0000    65.9000    66.0269      .1269
  50.0000    68.4000    68.4824      .0824
  55.0000    71.1000    71.0741     -.0259
  60.0000    73.7000    73.7544      .0544
  65.0000    76.7000    76.6289     -.0711
  70.0000    79.8000    79.6519     -.1481
  75.0000    83.0000    82.8382     -.1618
  80.0000    86.3000    86.2025     -.0975
  85.0000    90.2000    89.7944     -.4056
60 % OF DESIGN FLOW RATE (9900 GPM)
  1.7140    .9522    60.0  0.0000
4
RANGE = 15
   15.0         10.0
  40.0000    58.6000    58.7121      .1121
  45.0000    61.4000    61.5978      .1978
  50.0000    64.4000    64.6204      .2204
  55.0000    67.7000    67.8148      .1148
  60.0000    71.0000    71.0694      .0694
  65.0000    74.5000    74.4602     -.0398
  70.0000    77.9000    77.9056      .0056
  75.0000    81.7000    81.5487     -.1513
  80.0000    85.5000    85.2885     -.2115
  85.0000    89.5000    89.1755     -.3245
RANGE = 20
   20.0         10.0
  40.0000    62.2000    62.2992      .0992
  45.0000    64.7000    64.8520      .1520
  50.0000    67.4000    67.5412      .1412
  55.0000    70.4000    70.3967     -.0033
  60.0000    73.2000    73.2707      .0707
  65.0000    76.3000    76.3243      .0243
  70.0000    79.4000    79.4842      .0842
  75.0000    83.0000    82.8665     -.1335
  80.0000    86.5000    86.3497     -.1503
  85.0000    90.4000    90.0286     -.3714
RANGE = 25
   25.0         10.0
  40.0000    64.9000    65.0547      .1547
  45.0000    67.2000    67.3701      .1701
  50.0000    69.6000    69.7848      .1848
  55.0000    72.3000    72.3644      .0644
  60.0000    75.0000    75.0337      .0337
  65.0000    77.8000    77.8331      .0331
  70.0000    80.7000    80.7755      .0755
  75.0000    83.9000    83.9064      .0064
```

Figure 4.4 Output Listing for Program PDAP.

80.0000	87.3000	87.2105	-.0895
85.0000	91.0000	90.7048	-.2952

RANGE = 30

30.0	10.0		
40.0000	67.2000	67.2825	.0825
45.0000	69.2000	69.3659	.1659
50.0000	71.5000	71.6033	.1033
55.0000	73.9000	73.9528	.0528
60.0000	76.3000	76.4066	.1066
65.0000	79.1000	79.0557	-.0443
70.0000	81.8000	81.8165	.0165
75.0000	84.9000	84.7842	-.1158
80.0000	88.1000	87.9209	-.1791
85.0000	91.6000	91.2602	-.3398

70 % OF DESIGN FLOW RATE (11550 GPM)

1.5595 1.1110 70.0 0.0000

4

RANGE = 15

15.0	10.0		
40.0000	61.2000	61.3346	.1346
45.0000	63.9000	64.0688	.1688
50.0000	66.8000	66.9391	.1391
55.0000	69.7000	69.8658	.1658
60.0000	72.8000	72.9367	.1367
65.0000	76.1000	76.1499	.0499
70.0000	79.4000	79.4436	.0436
75.0000	83.0000	82.9146	-.0854
80.0000	86.6000	86.4824	-.1176
85.0000	90.4000	90.2042	-.1958

RANGE = 20

20.0	10.0		
40.0000	65.0000	65.1985	.1985
45.0000	67.4000	67.5946	.1946
50.0000	69.9000	70.0896	.1896
55.0000	72.5000	72.6897	.1897
60.0000	75.3000	75.4340	.1340
65.0000	78.3000	78.3240	.0240
70.0000	81.3000	81.3118	.0118
75.0000	84.5000	84.4607	-.0393
80.0000	87.9000	87.7733	-.1267
85.0000	91.7000	91.2835	-.4165

RANGE = 25

25.0	10.0		
40.0000	68.0000	68.2030	.2030
45.0000	70.2000	70.3581	.1581
50.0000	72.3000	72.5453	.2453
55.0000	74.9000	74.9608	.0608
60.0000	77.4000	77.4282	.0282
65.0000	80.1000	80.0502	-.0498
70.0000	82.9000	82.8076	-.0924
75.0000	86.0000	85.7495	-.2505
80.0000	89.0000	88.8130	-.1870
85.0000	92.4000	92.0954	-.3046

RANGE = 30

30.0	10.0		
40.0000	70.3000	70.5395	.2395
45.0000	72.2000	72.4578	.2578
50.0000	74.4000	74.5307	.1307
55.0000	76.7000	76.7090	.0090
60.0000	78.9000	78.9611	.0611
65.0000	81.5000	81.4093	-.0907
70.0000	84.0000	83 9615	-.0385
75.0000	86.8000	86.[illegible]061	-.0939
80.0000	89.9000	89.6452	-.2548
85.0000	93.2000	92.7711	-.4289

80 % OF DESIGN FLOW RATE (13200 GPM)

1.4549 1.2700 80.0 0.0000

4

RANGE = 15

15.0	10.0		
40.0000	63.5000	63.6260	.1260
45.0000	66.0000	66.1816	.1816
50.0000	68.8000	68.9223	.1223
55.0000	71.6000	71.7155	.1155
60.0000	74.5000	74.6144	.1144
65.0000	77.8000	77.7317	-.0683
70.0000	80.8000	80.8229	.0229
75.0000	84.3000	84.1638	-.1362
80.0000	87.7000	87.5720	-.1280
85.0000	91.6000	91.2035	-.3965

RANGE = 20

20.0	10.0		
40.0000	67.7000	67.7956	.0956
45.0000	69.8000	69.9727	.1727
50.0000	72.2000	72.3302	.1302
55.0000	74.7000	74.7896	.0896
60.0000	77.2000	77.3268	.1268
65.0000	80.0000	80.0479	.0479
70.0000	82.9000	82.8931	-.0069

Figure 4.4 Continued

75.0000	85.9000	85.8735	-.0265
80.0000	89.1000	89.0214	-.0786
85.0000	92.8000	92.3953	-.4047

RANGE = 25

25.0	10.0		
40.0000	70.8000	70.9228	.1228
45.0000	72.7000	72.8698	.1698
50.0000	74.9000	74.9768	.0768
55.0000	77.2000	77.1899	-.0101
60.0000	79.4000	79.4613	.0613
65.0000	82.1000	81.9632	-.1368
70.0000	84.7000	84.5464	-.1536
75.0000	87.3000	87.2579	-.0421
80.0000	90.3000	90.1875	-.1125
85.0000	93.9000	93.3551	-.5449

RANGE = 30

30.0	10.0		
40.0000	73.3000	73.3725	.0725
45.0000	75.1000	75.1515	.0515
50.0000	77.0000	77.0297	.0297
55.0000	79.0000	79.0117	.0117
60.0000	81.0000	81.0955	.0955
65.0000	83.4000	83.3765	-.0235
70.0000	85.9000	85.7946	-.1054
75.0000	88.6000	88.3802	-.2198
80.0000	91.3000	91.1145	-.1855
85.0000	94.5000	94.0789	-.4211

90 % OF DESIGN FLOW RATE (14850 GPM)

1.3633 1.4280 90.0 0.0000

4

RANGE = 15

15.0	10.0		
40.0000	65.8000	65.8905	.0905
45.0000	68.1000	68.2720	.1720
50.0000	70.8000	70.8903	.0903
55.0000	73.3000	73.4721	.1721
60.0000	76.1000	76.2491	.1491
65.0000	79.0000	79.1370	.1370
70.0000	82.0000	82.1436	.1436
75.0000	85.3000	85.3367	.0367
80.0000	88.7000	88.6541	-.0459
85.0000	92.2000	92.1027	-.0973

RANGE = 20

20.0	10.0		
40.0000	70.1000	70.2224	.1224
45.0000	72.1000	72.2732	.1732
50.0000	74.3000	74.4663	.1663
55.0000	76.7000	76.7996	.0996
60.0000	79.2000	79.2400	.0400
65.0000	81.8000	81.7995	-.0005
70.0000	84.5000	84.4819	-.0181
75.0000	87.5000	87.3518	-.1482
80.0000	90.7000	90.3807	-.3193
85.0000	93.7000	93.4947	-.2053

RANGE = 25

25.0	10.0		
40.0000	73.3000	73.4707	.1707
45.0000	75.2000	75.3243	.1243
50.0000	77.2000	77.2758	.0758
55.0000	79.4000	79.3609	-.0391
60.0000	81.4000	81.4702	.0702
65.0000	83.9000	83.8148	-.0852
70.0000	86.3000	86.2381	-.0619
75.0000	89.1000	88.8716	-.2284
80.0000	91.8000	91.6150	-.1850
85.0000	94.7000	94.5455	-.1545

RANGE = 30

30.0	10.0		
40.0000	75.9000	76.0042	.1042
45.0000	77.5000	77.6336	.1336
50.0000	79.4000	79.4158	.0158
55.0000	81.3000	81.2741	-.0259
60.0000	83.3000	83.2473	-.0527
65.0000	85.4000	85.3472	-.0528
70.0000	87.8000	87.6249	-.1751
75.0000	90.3000	90.0480	-.2520
80.0000	92.9000	92.6316	-.2684
85.0000	95.7000	95.4048	-.2952

100 % OF DESIGN FLOW RATE (16500 GPM)

1.2979 1.5780 100.0 0.00000000

4

RANGE = 15

15.0	10.0		
40.0000	67.9000	67.9101	.0101
45.0000	70.1000	70.1761	.0761
50.0000	72.5000	72.5875	.0875
55.0000	75.0000	75.1006	.1006
60.0000	77.8000	77.8051	.0051
65.0000	80.4000	80.4974	.0974
70.0000	83.2000	83.3525	.1525

Figure 4.4 Continued

75.0000	86.4000	86.4394	.0394
80.0000	89.9000	89.7053	-.1947
85.0000	93.2000	93.0116	-.1884
RANGE = 20			
20.0	10.0		
40.0000	72.2000	72.3226	.1226
45.0000	74.2000	74.3036	.1036
50.0000	76.2000	76.3437	.1437
55.0000	78.8000	78.6797	-.1203
60.0000	81.0000	80.9298	-.0702
65.0000	83.2000	83.2711	.0711
70.0000	85.9000	85.8703	-.0297
75.0000	88.5000	88.5442	.0442
80.0000	91.4000	91.4177	.0177
85.0000	94.8000	94.5271	-.2729
RANGE = 25			
25.0	10.0		
40.0000	75.6000	75.7030	.1030
45.0000	77.5000	77.4816	-.0184
50.0000	79.2000	79.2544	.0544
55.0000	81.2000	81.1982	-.0018
60.0000	83.2000	83.2232	.0232
65.0000	85.3000	85.3725	.0725
70.0000	87.8000	87.7314	-.0686
75.0000	90.2000	90.1807	-.0193
80.0000	92.9000	92.8266	-.0734
85.0000	96.1000	95.7022	-.3978
RANGE = 30			
30.0	10.0		
40.0000	78.3000	78.2947	-.0053
45.0000	79.9000	79.8474	-.0526
50.0000	81.5000	81.4654	-.0346
55.0000	83.1000	83.1629	.0629
60.0000	85.0500	85.0391	-.0109
65.0000	87.0000	87.0144	.0144
70.0000	89.1000	89.1369	.0369
75.0000	91.5000	91.4447	-.0553
80.0000	94.0000	93.9083	-.0917
85.0000	97.0000	96.6038	-.3962

110 % OF DESIGN FLOW RATE (18150 GPM)

1.2187 1.7450 110.0 0.0000

4

RANGE = 15			
15.0	10.0		
40.0000	69.9000	70.0305	.1305
45.0000	72.0000	72.1850	.1850
50.0000	74.4000	74.5336	.1336
55.0000	76.7000	76.8861	.1861
60.0000	79.3000	79.4351	.1351
65.0000	82.1000	82.1346	.0346
70.0000	84.9000	84.9060	.0060
75.0000	87.9000	87.8349	-.0651
80.0000	91.0000	90.8893	-.1107
85.0000	94.3000	94.1060	-.1940
RANGE = 20			
20.0	10.0		
40.0000	74.5000	74.6979	.1979
45.0000	76.4000	76.5671	.1671
50.0000	78.3000	78.4960	.1960
55.0000	80.3000	80.5264	.2264
60.0000	82.7000	82.7781	.0781
65.0000	84.8000	85.0004	.2004
70.0000	87.5000	87.5149	.0149
75.0000	90.1000	90.0916	-.0084
80.0000	93.0000	92.8611	-.1389
85.0000	96.0000	95.7725	-.2275
RANGE = 25			
25.0	10.0		
40.0000	78.0000	78.1935	.1935
45.0000	79.7000	79.8284	.1284
50.0000	81.2000	81.4567	.2567
55.0000	83.2000	83.3233	.1233
60.0000	85.2000	85.2629	.0629
65.0000	87.3000	87.3188	.0188
70.0000	89.7000	89.5526	-.1474
75.0000	91.9000	91.8464	-.0536
80.0000	94.6000	94.3841	-.2159
85.0000	97.2000	97.0287	-.1713
RANGE = 30			
30.0	10.0		
40.0000	80.7000	80.8424	.1424
45.0000	82.2000	82.2859	.0859
50.0000	83.7000	83.7967	.0967
55.0000	85.3000	85.4145	.1145
60.0000	87.2000	87.1886	-.0114
65.0000	88.9000	89.0076	.1076
70.0000	91.2000	91.0703	-.1297
75.0000	93.4000	93.2255	-.1745
80.0000	95.9000	95.5680	-.3320
85.0000	98.3000	98.0414	-.2586

Figure 4.4 Continued

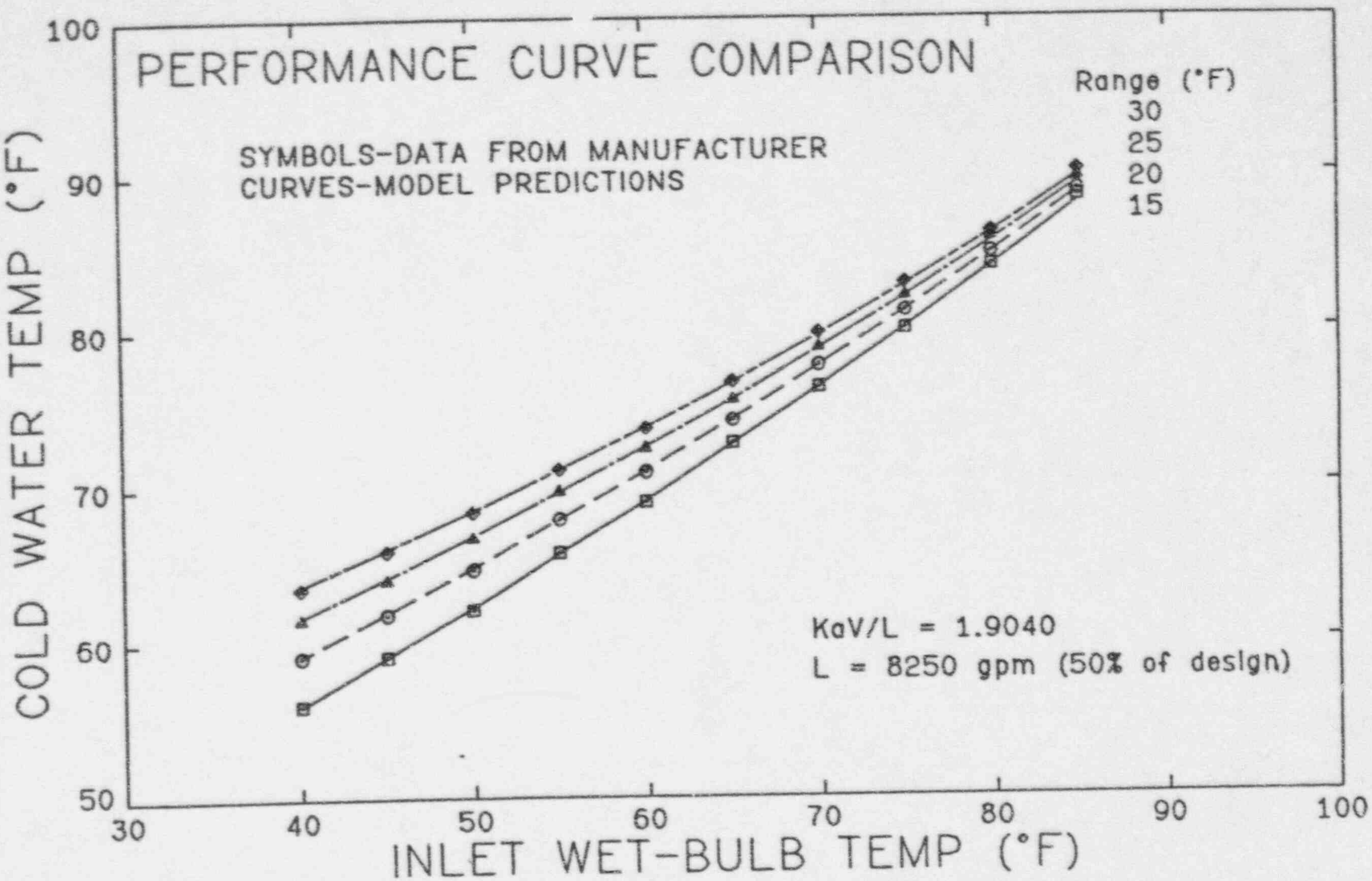


Figure 4.5 Comparison of Manufacturer's Performance Data with Predictions of Mathematical Model for 50% of Design Flow Rate.

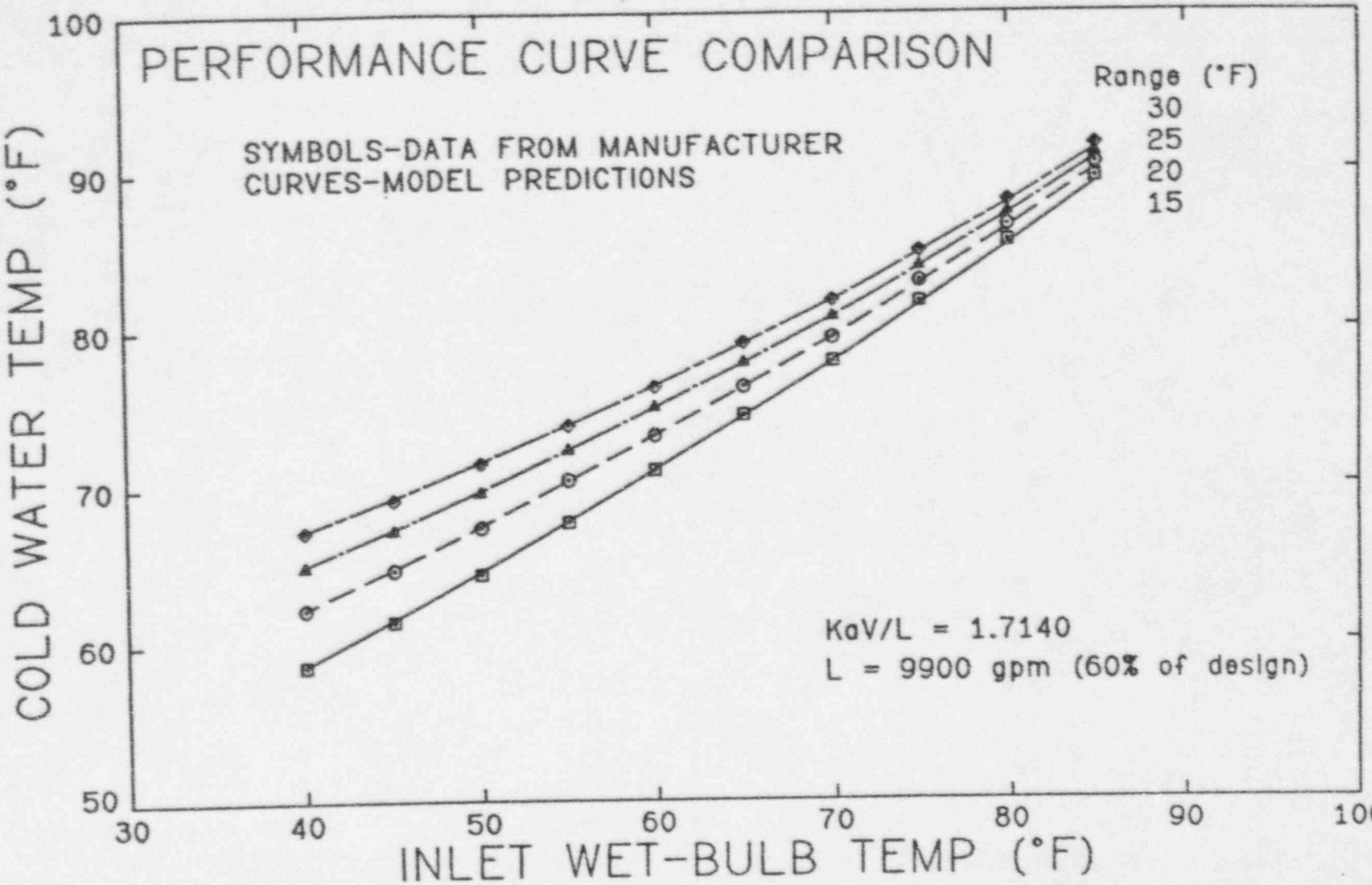


Figure 4.6 Comparison of Manufacturer's Performance Data with Predictions of Mathematical Model for 60% of Design Flow Rate.

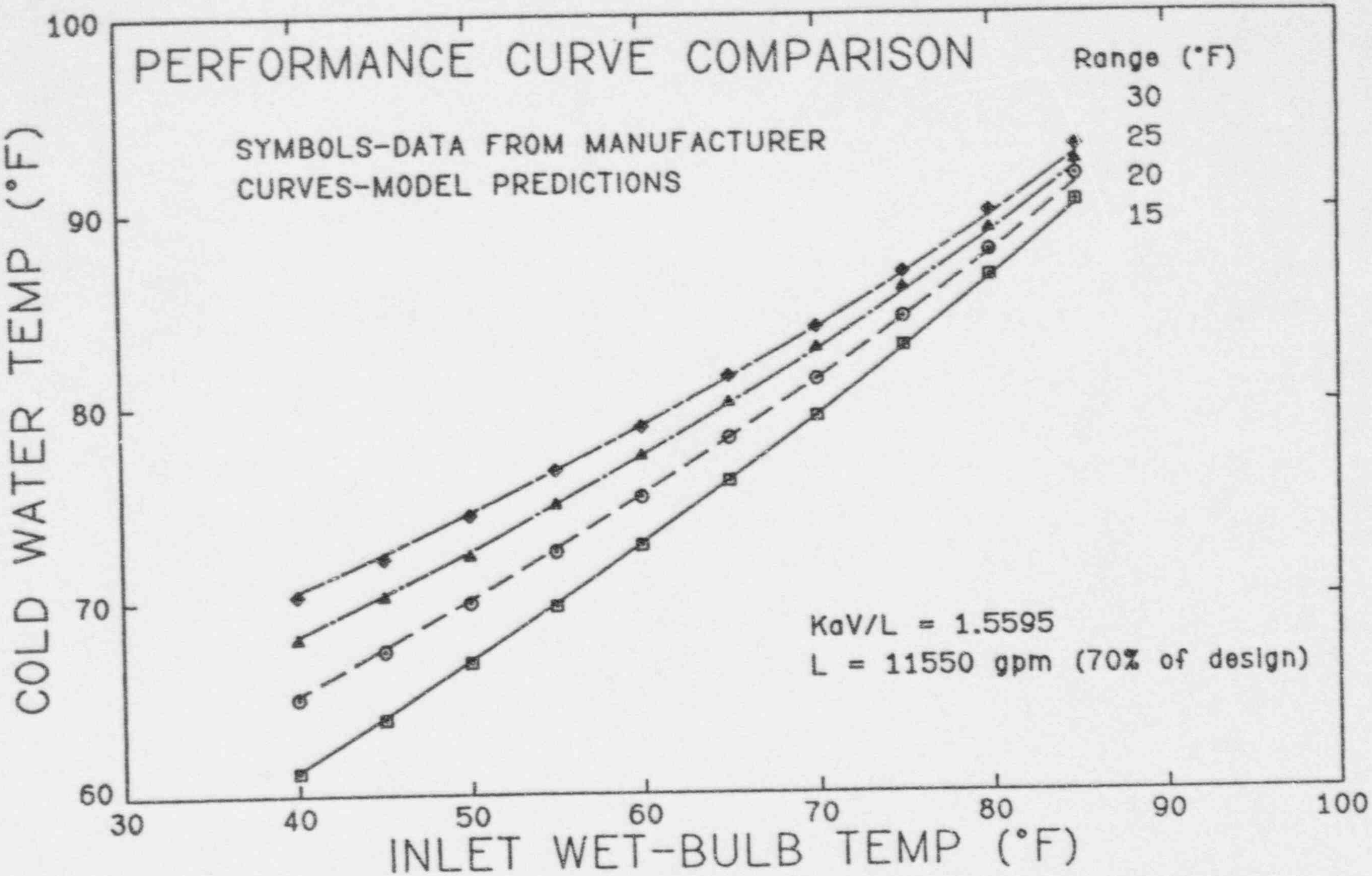


Figure 4.7 Comparison of Manufacturer's Performance Data with Predictions of Mathematical Model for 70% of Design Flow Rate.

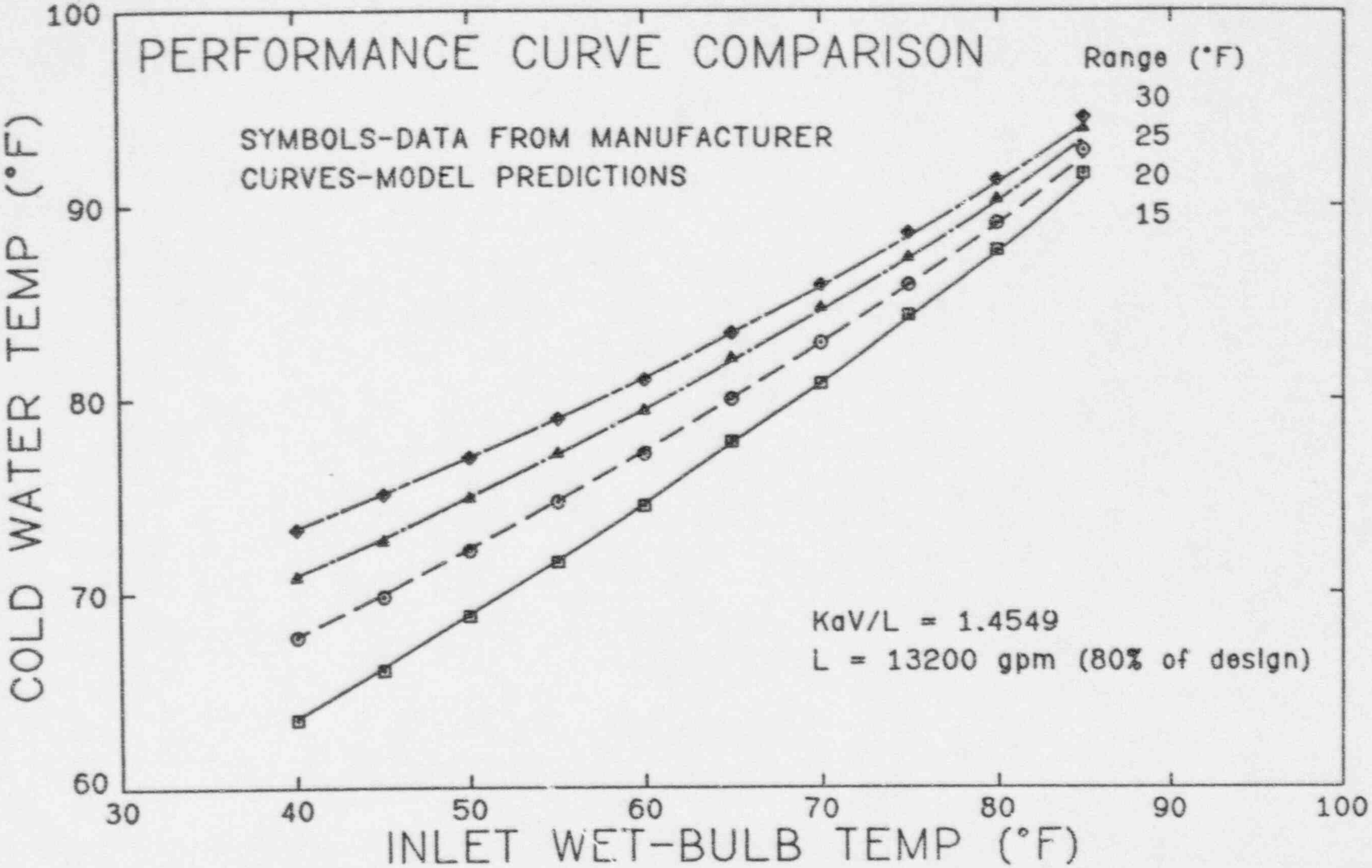


Figure 4.8 Comparison of Manufacturer's Performance Data with Predictions of Mathematical Model for 80% of Design Flow Rate.

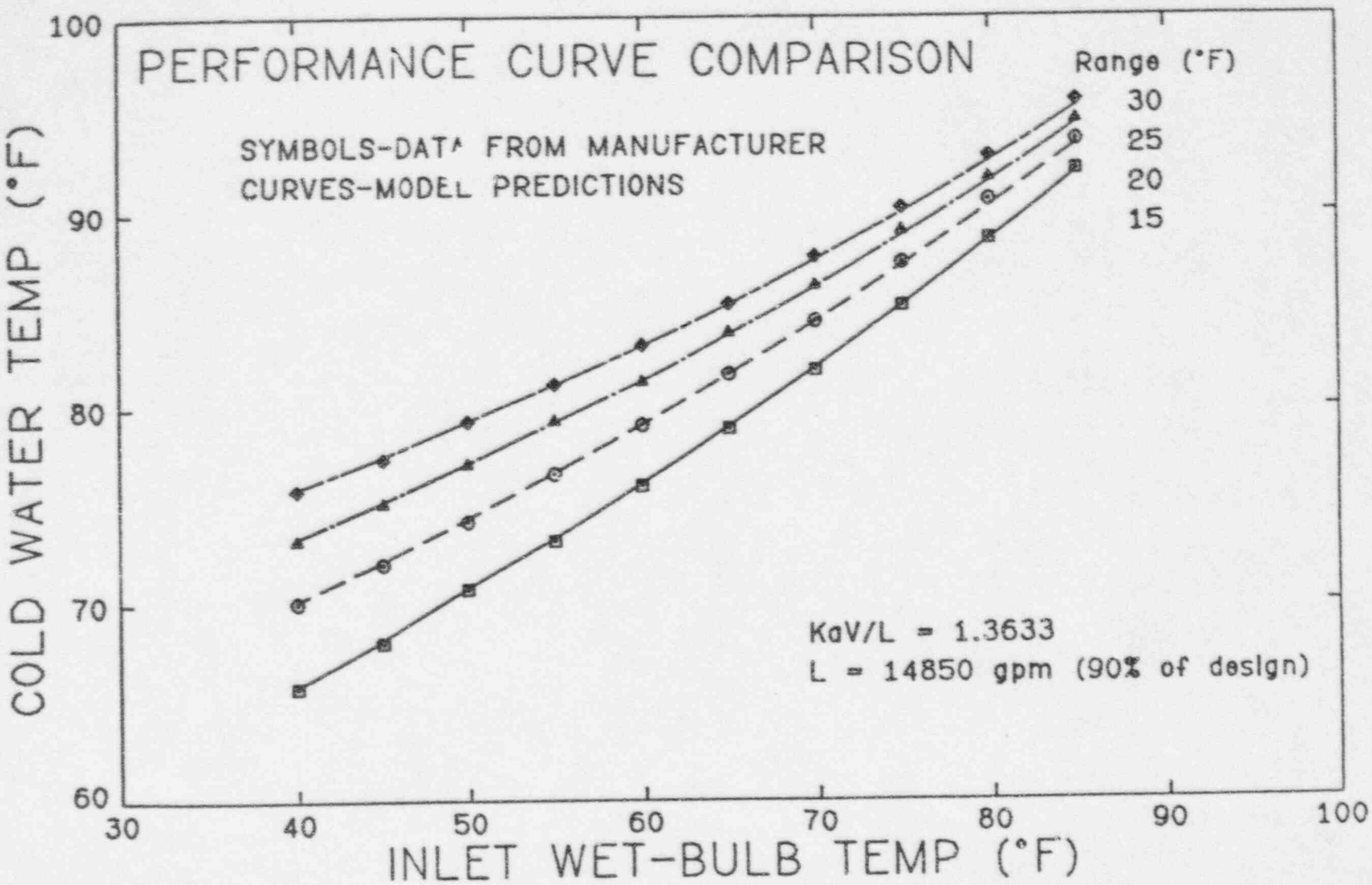


Figure 4.9 Comparison of Manufacturer's Performance Data with Predictions of Mathematical Model for 90% of Design Flow Rate.

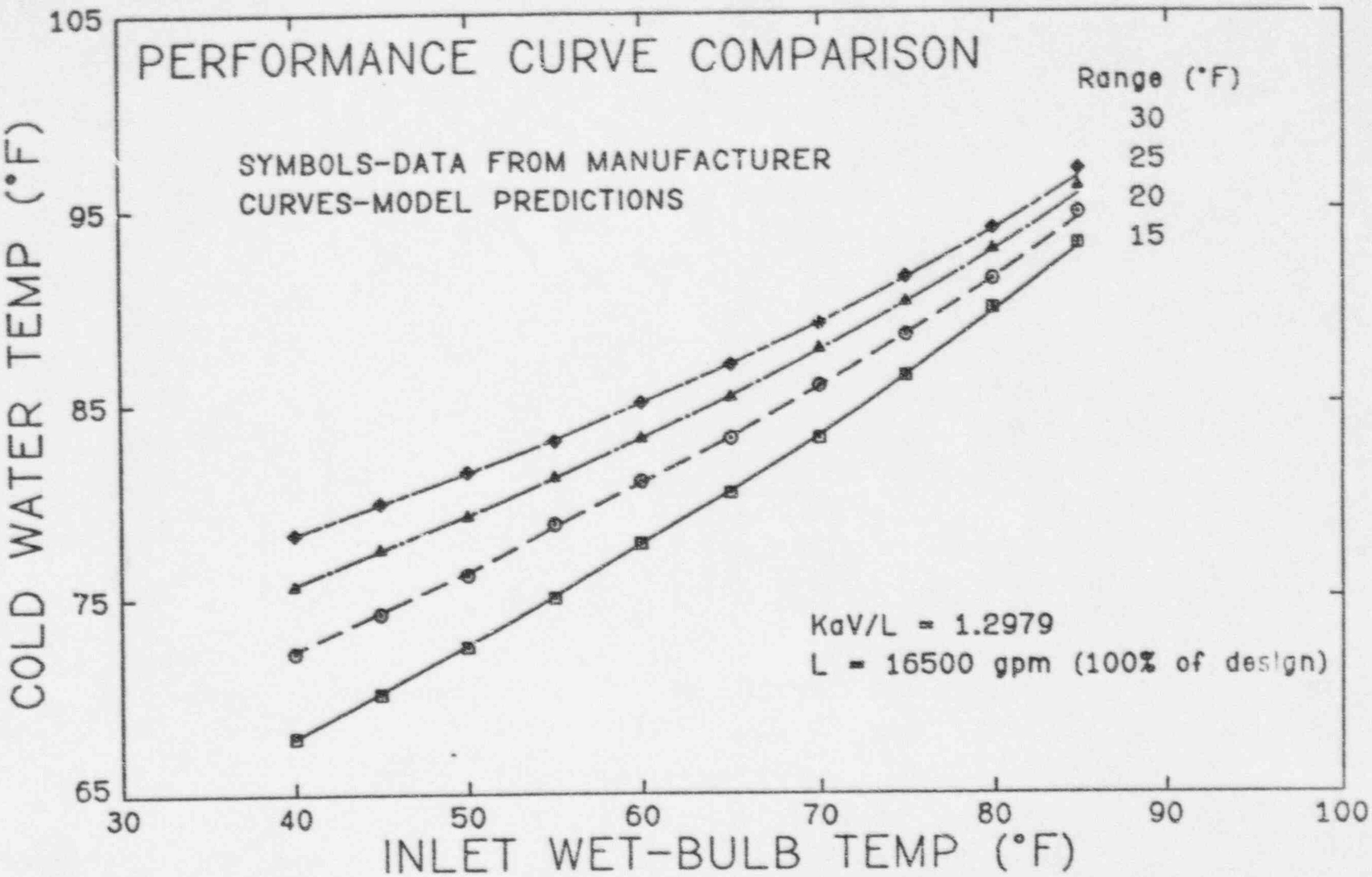


Figure 4.10 Comparison of Manufacturer's Performance Data with Predictions of Mathematical Model for 100% of Design Flow Rate.

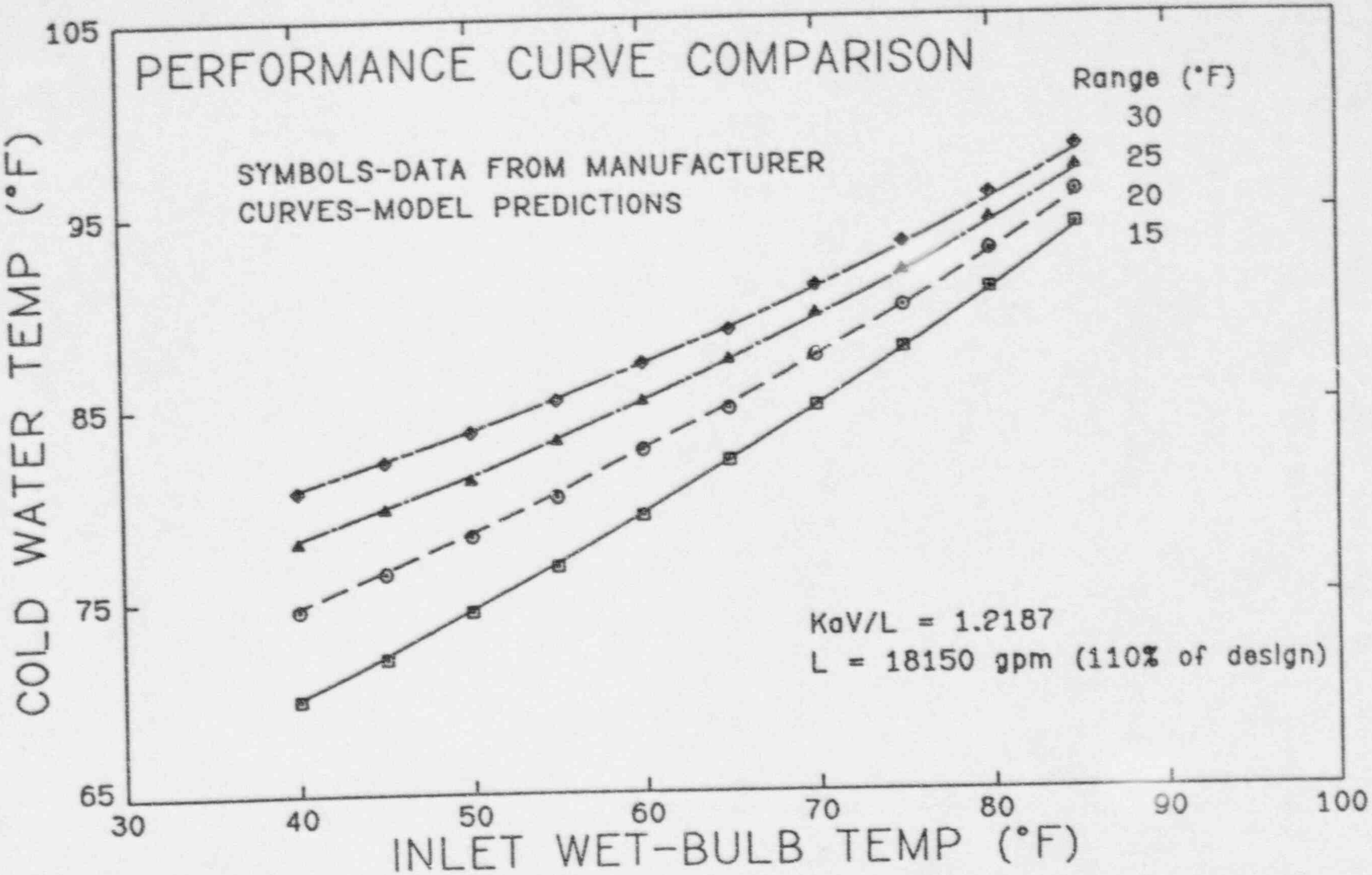


Figure 4.11 Comparison of Manufacturer's Performance Data with Predictions of Mathematical Model for 110% of Design Flow Rate.

cal difference between the predicted and reported temperatures is shown in the fourth column of Figure 4.4. As is readily observed from Figures 4.5 - 4.11 as well as the tabulated output given in Figure 4.4, the agreement is excellent, indicating that the tower simulation is indeed able to reproduce the tower performance characteristics. Figure 4.12 compares relative KaV (i.e., the ratio of KaV at the given flow rate to KaV at the design flow rate) to relative flow rate (i.e., the ratio of the given flow rate to the design flow rate). This plot illustrates that KaV declines with declining flow rate as may be expected.

Since the design-basis operating points (73.6 percent flow rate and 70.4 percent flow rate) do not coincide with any of the flow rates for which performance data are available, a simple program entitled KAVL is provided to interpolate by Hermite cubic spline the values of KaV/L at the actual operating points. The inputs to program KAVL are taken directly from the output of program PDAP.

4.3 Time Constant Evaluation

Programs UHSSIM and TMCON (TIME CONSTANT) are needed to determine the approximate time constant of the tower/basin system for use in the data scanning procedure to be described later. Recall that, for the purpose of determining the period of worst-case tower performance, the response of the basin temperature is approximated as being exponential in time. Although the assumption of a first-order time response is borne out well in practice, the value of the time constant may vary over the range of possible operating temperatures for the tower/basin system. Thus, the value of the time constant employed must be representa-

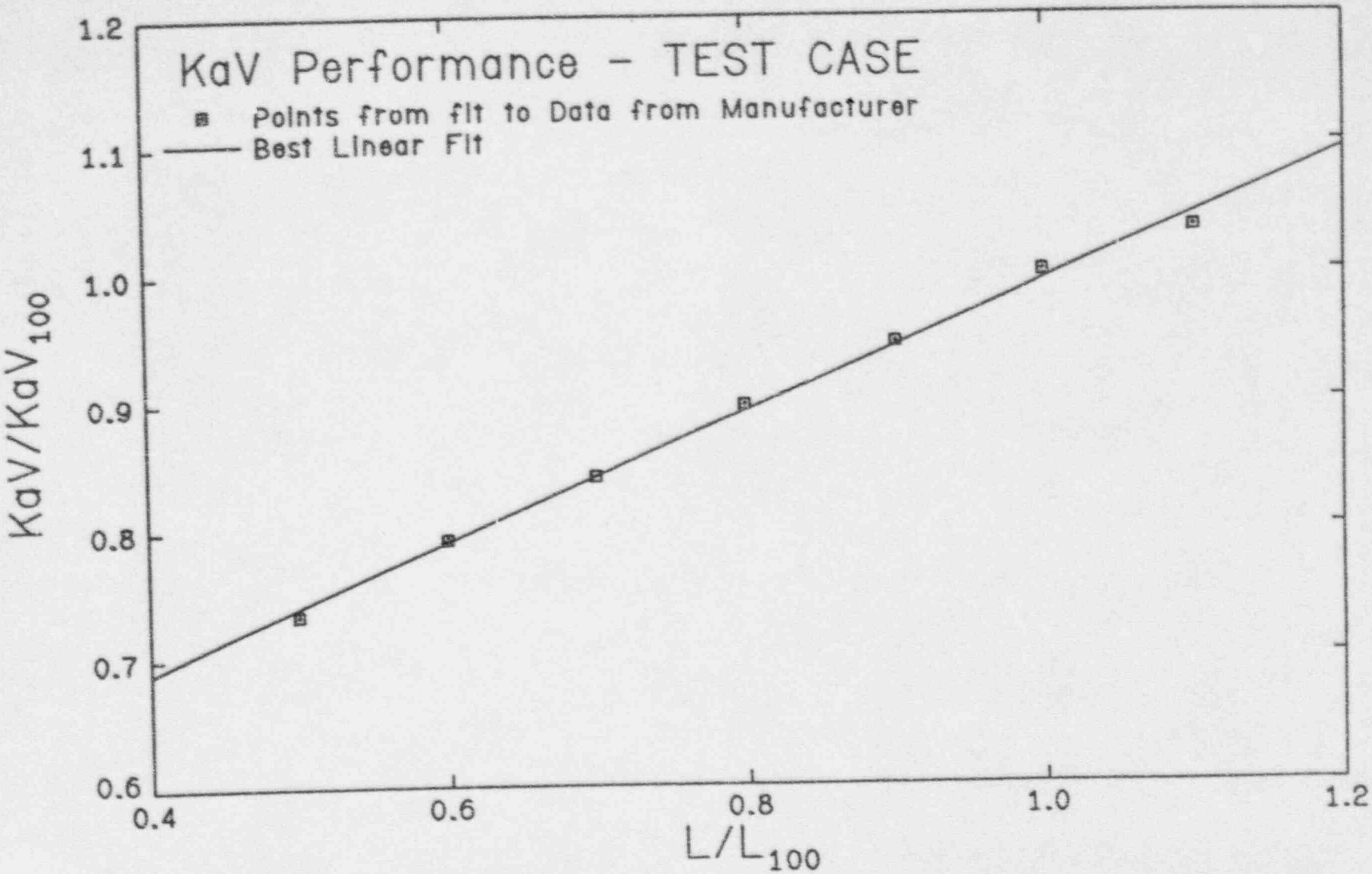


Figure 4.12 Variation of Relative Tower Parameter KaV (Ratio of KaV at the Given Flow Rate to KaV at Design Flow Rate) to Relative Flow Rate (Ratio of Given Flow Rate to Design Flow Rate).

tive of the specific accident conditions hypothesized. For the purpose of determining the time constant of the tower/basin system, a constant ambient wet-bulb temperature is used. Also, a reasonable choice for initial basin temperature is necessary. The exact values selected for these inputs are not crucial to the time-constant determination; rather, it is merely necessary that the operating temperature of the tower(s) be in the proper range.

The required input for program UHSSIM is set forth in Table 4.5 and Figures 4.13 and 4.14. Specifically, Table 4.5 defines each FORTRAN variable name, Figure 4.13 illustrates the format in which the input is expected, and Figure 4.14 shows the actual input file used in the test case simulation. As shown in Figure 4.13, the input is divided into six sections. The first section gives the number of towers, the initial mass of water in the basin and the initial salinity of the basin. For the test case, only a single tower is present, the mass of water initially in the basin is 5.42×10^7 lb_m, and the initial salinity is taken to be 5 parts per thousand (ppt).

The second section lists the control variables for the simulation: the time increment to be used in printing the results, the number of such time steps to be simulated, a logical value indicating whether evaporation is to be allowed, and a logical value indicating whether a constant wet-bulb temperature is to be employed. In the sample case, the time step is set to one hour, the number of steps is set to 100, and evaporation is allowed. As noted earlier, a user-specified constant wet-bulb temperature is used for the purpose of determining the tir

Table 4.5 Description of Input Variables for Program UHSSIM

VARIABLE NAME	DESCRIPTION AND UNITS
NTOWER	Number of Towers
BASMAS	Initial Basin Mass (lbm)
S	Initial Basin Salinity
TSTEP	Printing Time Step Size (hr)
NSTEP	Number of Steps to be Taken
NOEVAP	Logical Variable Set .True. for No Evaporation .False. Otherwise
CONWB	Logical Variable Set .True. for Constant Wet-Bulb Temperature .False. Otherwise
NL	Number of Flow Rate Values Per Tower
TE	Array Containing Times of Corresponding Flow Rate Values (hr)
LOVRGS	Array Containing L/G Values
KAVLS	Array Containing KaV/L Values
LS	Array Containing Liquid Flow Rate Values (lbm/hr)
NH	Number of Heat Load Values Per Tower
TME	Array Containing Times of Corresponding Heat Load Values (Btu/hr)
WB	Constant Wet-Bulb Temperature Value (F)
BASTMP	Initial Basin Temperature (F)
NWINDW	Number of Meteorological Window to be Read
NSTART	Position Within Meteorological Window to Begin Reading Wet-Bulb Temperature Values

Figure 4.13 Input Format for Program UHSSIM

```
Section 1        NTOWER,BASMAS,S

Section 2        TSTEP,NSTEP,NOEVAP,CONWB

Section 3        NL(1),NL(2),...
                 TE(1,1),LOVRGS(1,1),KAVLS(1,1),LS(1,1) }
                 TE(2,1),LOVRGS(2,1),KAVLS(2,1),LS(2,1) }  Tower #1
                 .                                      }
                 .

                 TE(1,2),LOVRGS(1,2),KAVLS(1,2),LS(1,2) }
                 TE(2,2),LOVRGS(2,2),KAVLS(2,2),LS(2,2) }  Tower #2
                 .                                      }
                 .                                         etc...

Section 4        NH(1),NH(2),...
                 TME(1,1),Q(1,1) }
                 TME(2,1),Q(2,1) }  Tower #1
                 .               }
                 .

                 TME(1,2),Q(1,2) }
                 TME(2,2),Q(2,2) }  Tower #2
                 .               }
                 .                  etc...

Section 5        WB,BASTMP

Section 6        NWINDW,NSTART
```

Figure 4.14 Input Listing for Program UHSSIM.

```
1,5.42E7,0.005
1.0,100,F,T
4
0.0,1.1687,1.5189,6.0661E6
25.0,1.1687,1.5189,6.0661E6
25.0,1.1167,1.5551,5.796E6
100.0,1.1167,1.5551,5.796E6
29
0.0,66.2E6
0.5,193.0E6
1.0,193.0E6
1.5,193.0E6
2.0,193.0E6
2.5,193.0E6
3.0,193.0E6
4.0,193.0E6
5.0,193.0E6
6.0,193.0E6
8.0,193.0E6
10.0,193.0E6
12.0,190.0E6
16.0,182.0E6
20.0,178.0E6
24.0,131.0E6
48.0,95.4E6
72.0,88.6E6
96.0,84.1E6
120.0,81.1E6
144.0,77.6E6
168.0,76.2E6
192.0,74.6E6
216.0,73.1E6
240.0,71.2E6
360.0,67.0E6
480.0,64.0E6
600.0,61.6E6
720.0,60.4E6
81.,81.
1,79
```

constant of the tower/basin system, so the variable CONWB is set to be true.

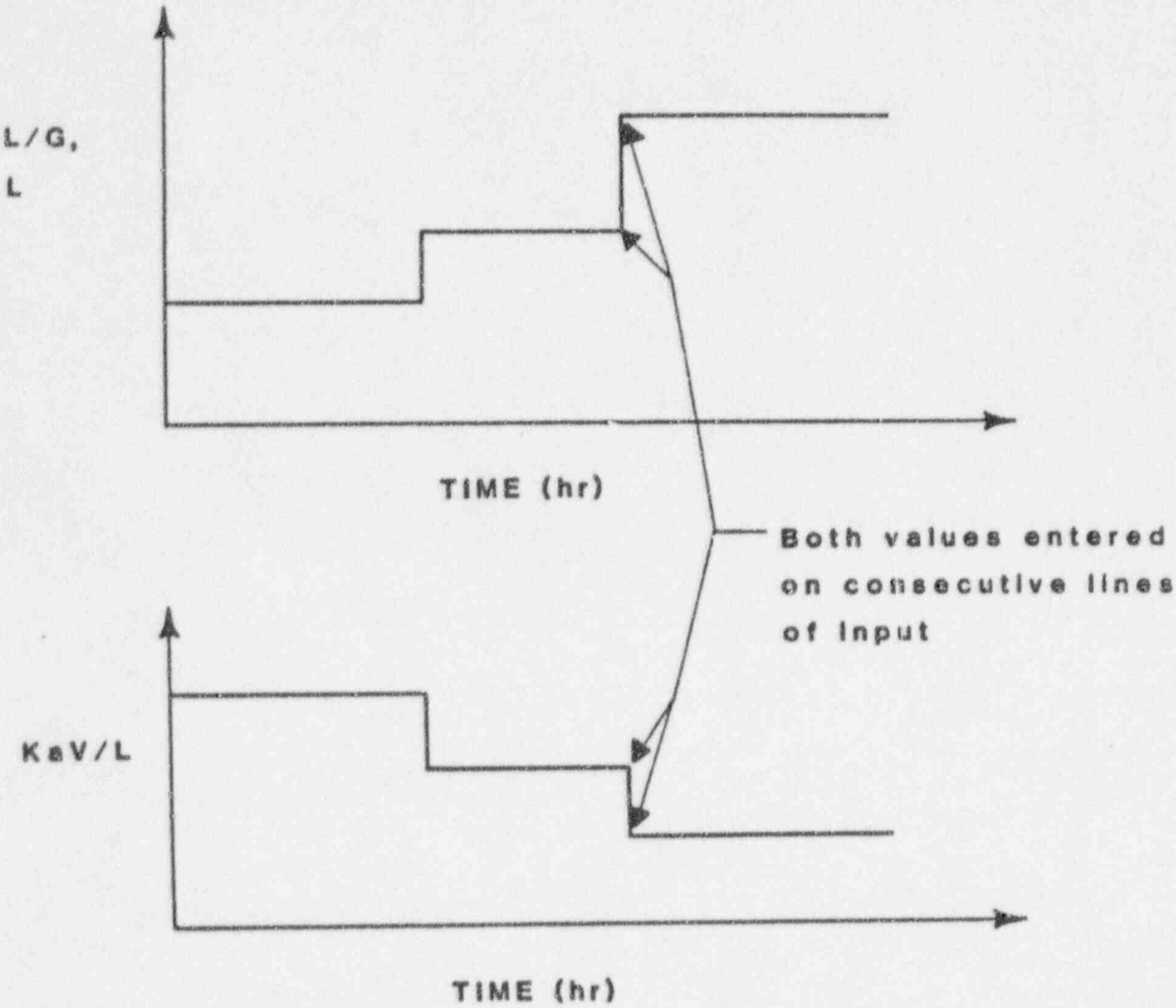
The third section provides information on how the various tower performance parameters change with time for each tower during the simulated accident. The specific parameters required include the two characteristic values L/G and KaV/L as well as the actual water flow rate, all as functions of time for each tower separately. The first line in Section 3 gives the number of lines of input required for each tower. Thus, in the case of two towers, the value of NTOWER set in Section 1 will be 2, and two values of NL (first line of Section 3) are expected. If these values are 10 and 15, respectively, then 10 lines of input are correspondingly expected for tower 1 followed by 15 lines of input for tower 2.

For the test case, only one tower is present; thus, only one value of NL, namely 4, is read, indicating that four lines of input are to be given for the tower. A close examination of the input shows that two consecutive lines are given for a time of 25 hours after start of the accident. This is done to correctly represent the step change in the flow rate at that time as Figure 4.15 illustrates.

Section 4 provides the time-dependent accident heat loads for each tower in a manner paralleling that used to specify tower performance data in Section 3. For the test case, 29 entries are made as shown in Figure 4.14.

Section 5 provides the values of the constant wet-bulb temperature and initial basin temperature used in the time constant determination.

Figure 4.15 Step Change in L, KaV/L, and L/G Values.



Since Section 6 is required only for the final simulation using the meteorological data windows extracted by the data scanning procedure, discussion of it will be deferred. For the test case, the wet-bulb temperature and initial basin temperature were both set to 81 °F.

The actual time constant is now calculated using program TMCON. Program UHSSIM creates an output file which is directly input to program TMCON as well as an interpreted output file for printing as shown in Table 4.6. The time constant is determined by minimizing the sum of squares S given by

$$S = \sum (T_i - T_i')^2, \qquad (4.1)$$

where T_i 's the basin temperature at step "i" predicted by UHSSIM and T_i' is the corresponding value determined assuming first-order system response. The values of T_i' are given by the recursive relationship

$$T_i' = (T_{i-1}' + e\, T_{wb,i} + (1 - \varepsilon_i)q_i \Delta t/(m_i c_L))/(1 + e), \qquad (4.2)$$

where e is $\Delta t/\tau$, $T_{wb,i}$ is the wet-bulb temperature at step "i", q_i is the heat load at step "i", L_i is the flow rate at step "i", m_i is the basin mass at step "i", ε_i is $(T_{in,i} - T_{out,i})/(T_{in,i} - T_{wb,i})$, $T_{in,i}$ is the tower inlet temperature at step "i", $T_{out,i}$ is the tower exit temperature at step "i", and c_L is the specific heat of water.

The time constant value found by program TMCON should be evaluated as to whether it appears reasonable for the system under consideration. Typical values lie in the range of several hours to several tens of hours. For the test case, the time constant was found to be 11.46

Table 4.6 Output Listing from Program UHSSIM for Constant Wet-Bulb Temperature.

TIME (HR)	TBASIN (F)	BMASS (LBM)	S (PPT)	TWB (F)	Q (BTU/HR)	L (LBM/HR)
0.0	81.00	5.42E+07	5.0	81.0	6.62E+07	6.07E+06
1.0	81.80	5.41E+07	5.0	81.0	1.93E+08	6.07E+06
2.0	82.63	5.39E+07	5.0	81.0	1.93E+08	6.07E+06
3.0	83.39	5.38E+07	5.0	81.0	1.93E+08	6.07E+06
4.0	84.09	5.37E+07	5.1	81.0	1.93E+08	6.07E+06
5.0	84.72	5.35E+07	5.1	81.0	1.93E+08	6.07E+06
6.0	85.30	5.34E+07	5.1	81.0	1.93E+08	6.07E+06
7.0	85.83	5.32E+07	5.1	81.0	1.93E+08	6.07E+06
8.0	86.31	5.30E+07	5.1	81.0	1.93E+08	6.07E+06
9.0	86.75	5.29E+07	5.1	81.0	1.93E+08	6.07E+06
10.0	87.15	5.27E+07	5.1	81.0	1.93E+08	6.07E+06
11.0	87.51	5.25E+07	5.2	81.0	1.92E+08	6.07E+06
12.0	87.84	5.24E+07	5.2	81.0	1.90E+08	6.07E+06
13.0	88.13	5.22E+07	5.2	81.0	1.88E+08	6.07E+06
14.0	88.39	5.20E+07	5.2	81.0	1.86E+08	6.07E+06
15.0	88.63	5.19E+07	5.2	81.0	1.84E+08	6.07E+06
16.0	88.84	5.17E+07	5.2	81.0	1.82E+08	6.07E+06
17.0	89.02	5.16E+07	5.3	81.0	1.81E+08	6.07E+06
18.0	89.18	5.14E+07	5.3	81.0	1.80E+08	6.07E+06
19.0	89.33	5.12E+07	5.3	81.0	1.79E+08	6.07E+06
20.0	89.45	5.11E+07	5.3	81.0	1.78E+08	6.07E+06
21.0	89.55	5.09E+07	5.3	81.0	1.66E+08	6.07E+06
22.0	89.62	5.08E+07	5.3	81.0	1.55E+08	6.07E+06
23.0	89.65	5.06E+07	5.4	81.0	1.43E+08	6.07E+06
24.0	89.64	5.05E+07	5.4	81.0	1.31E+08	6.07E+06
25.0	89.60	5.04E+07	5.4	81.0	1.30E+08	5.80E+06
26.0	89.55	5.02E+07	5.4	81.0	1.28E+08	5.80E+06
27.0	89.49	5.01E+07	5.4	81.0	1.27E+08	5.80E+06
28.0	89.43	5.00E+07	5.4	81.0	1.25E+08	5.80E+06
29.0	89.38	4.99E+07	5.4	81.0	1.24E+08	5.80E+06
30.0	89.32	4.97E+07	5.4	81.0	1.22E+08	5.80E+06
31.0	89.27	4.96E+07	5.5	81.0	1.21E+08	5.80E+06
32.0	89.21	4.95E+07	5.5	81.0	1.19E+08	5.80E+06
33.0	89.16	4.94E+07	5.5	81.0	1.18E+08	5.80E+06
34.0	89.10	4.93E+07	5.5	81.0	1.16E+08	5.80E+06
35.0	89.04	4.92E+07	5.5	81.0	1.15E+08	5.80E+06
36.0	88.99	4.91E+07	5.5	81.0	1.13E+08	5.80E+06
37.0	88.93	4.89E+07	5.5	81.0	1.12E+08	5.80E+06
38.0	88.87	4.88E+07	5.5	81.0	1.10E+08	5.80E+06
39.0	88.81	4.87E+07	5.6	81.0	1.09E+08	5.80E+06
40.0	88.76	4.86E+07	5.6	81.0	1.07E+08	5.80E+06
41.0	88.70	4.85E+07	5.6	81.0	1.06E+08	5.80E+06
42.0	88.64	4.84E+07	5.6	81.0	1.04E+08	5.80E+06
43.0	88.58	4.83E+07	5.6	81.0	1.03E+08	5.80E+06
44.0	88.52	4.82E+07	5.6	81.0	1.01E+08	5.80E+06
45.0	88.47	4.81E+07	5.6	81.0	9.99E+07	5.80E+06
46.0	88.42	4.80E+07	5.6	81.0	9.84E+07	5.80E+06
47.0	88.36	4.79E+07	5.7	81.0	9.69E+07	5.80E+06
48.0	88.30	4.78E+07	5.7	81.0	9.54E+07	5.80E+06

hours. Also, the effects of initial basin temperature and nominal ambient wet-bulb temperature can be studied by running several cases within the reasonable range.

Figure 4.16 provides a comparison of the temperature histories predicted by the full tower/basin simulation (UHSSIM) and the simplified model. As may be readily observed, the approximation as a first order system is a good one.

4.4 Program NWS -- Data Scanning Procedure

The full period of meteorological data available from the National Weather Service for the nearest reporting station is analyzed using program NWS to obtain the worst-case meteorological conditions for the area. The input variables for program NWS are set using parameter statements within the code. The variable names, descriptions, and units are given in Table 4.7.

Program NWS allows the window size, the number of windows to be saved simultaneously, and the number of records to be read in each run to be selected by the user. The term "window" refers to a segment of consecutive records from the meteorological data base. In the data scanning procedure, the window is moved chronologically through the data base by adding the next record from the meteorological data base to the bottom of the window and removing the oldest record in the window from the top of the window. Program NWS is set up to save the windows which have the highest running wet-bulb temperatures. Rather than saving just the single window with the highest value seen thus far in scanning the data base, program NWS allows the user to specify the number of windows

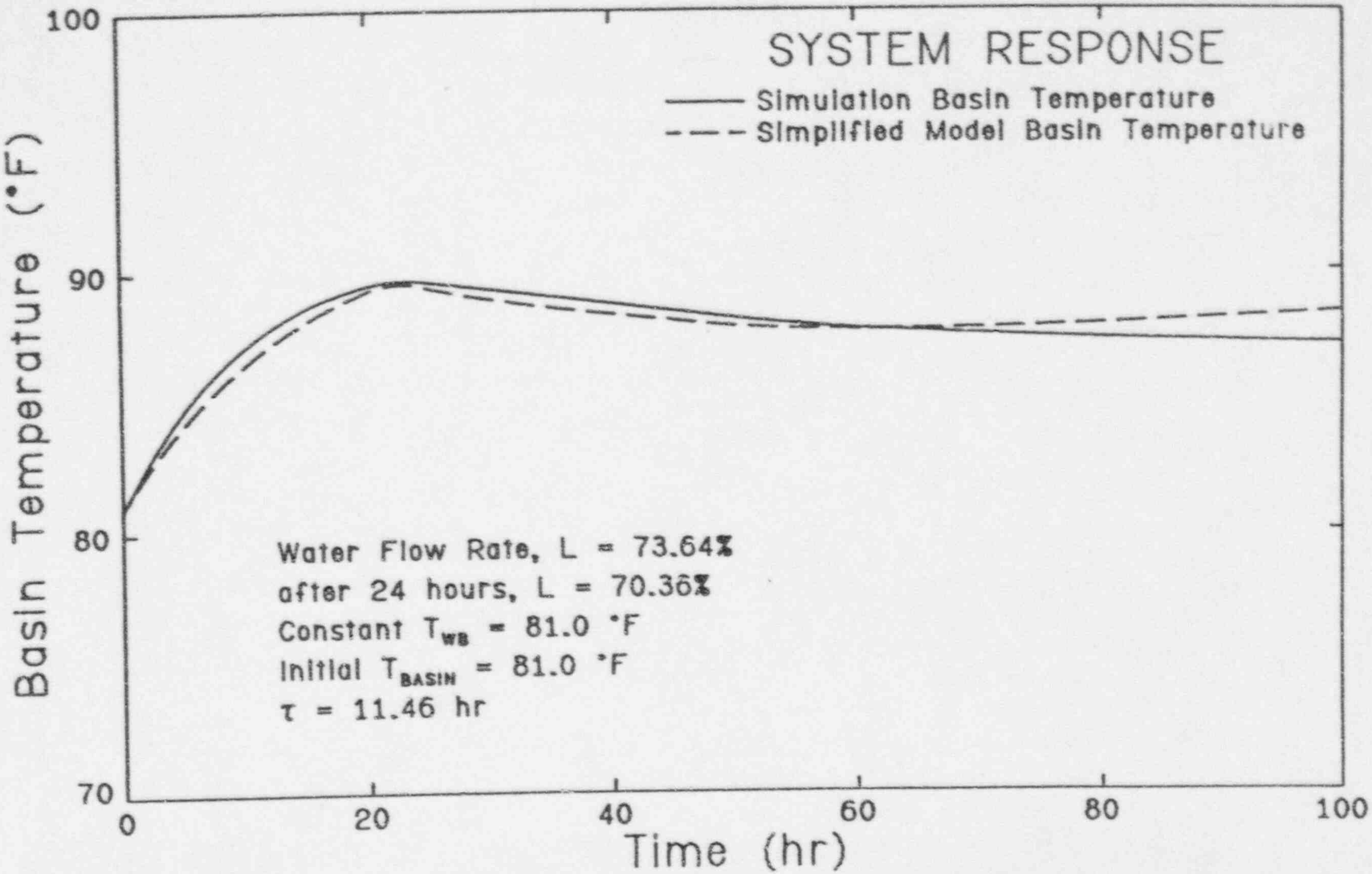


Figure 4.16 Comparison of Temperature Histories Predicted by Full Tower/Basin Simulation (UHSSIM) and Simplified Model for Constant Wet-Bulb Temperature.

Table 4.7 Description of Input Variables for Program NWS

VARIABLE NAME	DESCRIPTION AND UNITS
HEADER	40-Column Header Used to Identify Output File
TAU	Time Constant of Tower System (hr)
DELTAT	Time Between Successive Records on Meteorological Data Tape (hr)
NRECS	Number of Records to be Processed in Run
LENGTH	Number of Data Records in Each Window
IPEAK	Desired Location of Peak Running Wet-Bulb Temperature within Window
NSAVE	Number of Distinct Windows to be Saved

to be saved at any time. Saving more than just the single highest window can prevent the misleading results which ma_ be caused by sections of the database with high fractions of missing data.

In addition, program NWS is set up to be executed in several runs rather than just a single run. This device serves two purposes. First, by analyzing the data base in chunks, a lower investment in computational resources is made in each run. Thus, if a user-produced or computer-related problem should develop, only a small amount of computer resources will have been wasted. The number of records may be selected for user convenience, although 35,064 is a particularly attractive choice since it corresponds to exactly 4 years of data (including leap year). The second purpose in allowing smaller chunks of data to be analyzed in separate computer runs stems from the opportunity afforded the user to review the saved windows after each run and thus better track the data scanning procedure.

For the test case, thirty-four-and-a-half years of meteorological data are assumed to be available covering the period from 06:00 hours on July 1, 1948, through 23:00 hours on December 31, 1982. The meteorological data base contains hourly records except for a few stretches of missing data and a period of several years for which only tri-hourly observations were made. These data were scanned using the time constant of 11.46 hours discussed earlier in 4-year (35,064 records) blocks. The number of windows to be saved concurrently was set at 3 and the window length was set at 200 and the time of maximum running wet-bulb temperature was set to be record 101.

The periods of worst-case meteorological conditions identified by our procedure are, in order of decreasing severity:

a) 14:00 hours on August 13, 1981, through 21:00 hours on August 21, wherein the running wet-bulb temperature reached a maximum of 81.27°F at 18:00 hours on August 17, 1981.

b) 19:00 hours on August 6, 1969, through 2:00 hours on August 15, 1969, wherein the running wet-bulb temperature reached a maximum of 80.70°F at 23:00 hours on August 10, 1969.

c) 13:00 hours on June 23, 1980, through 20:00 hours on July 1, 1980, wherein the running wet-bulb temperature reached a maximum of 80.68°F at 17:00 hours on June 27, 1980.

Figure 4.17 gives an abbreviated listing of a file containing the meteorological data windows. The header information gives the alphanumeric identifier for the case (TEST DATA 073084), the values of the time constant (11.46 hours) and time increment (1 hour), the number of records processed in this run (35,064), the window length (200), the maximum number of windows to be saved (3), the actual number of windows saved thus far (3), the number of window save requests processed (24), and the total number of records scanned in all runs made thus far (302,418).

The header information is followed by the three windows. The first line of each window gives the maximum running wet-bulb temperature for the window (e.g., 81.27017072031) and the position number within the window at which the maximum occurs (e.g., 101). The data for each window are given in three columns corresponding to (i) the year, month,

Figure 4.17 Output Listing from Program NWS - Three Window Format.

```
TEST DATA 073084
 11.46 1. 35064 200 3
 3 24 302418
 81.27017072031 101
       81081314      76.81      79.00
       81081315      77.05      80.00
       81081316      77.19      79.00
           .           .          .
           .           .          .

       81081717      81.21      82.00
       81081718      81.27      82.00
       81081719      81.25      81.00
           .           .          .
           .           .          .

       81082119      74.31      74.00
       81082120      74.21      73.00
       81082121      74.12      73.00
 80.69924231812 101
       69080619      74.67     999.00
       69080620      74.62     999.00
       69080621      74.50      73.00
           .           .          .
           .           .          .

       69081022      80.68     999.00
       69081023      80.70     999.00
       69081100      80.65      80.00
           .           .          .
           .           .          .

       69081500      74.87      73.00
       69081501      74.73     999.00
       69081502      74.61     999.00
 80.67538029297 101
       80062313      73.55     999.00
       80062314      73.58     999.00
       80062315      73.83      77.00
           .           .          .
           .           .          .

       80062716      80.49     999.00
       80062717      80.68     999.00
       80062718      80.63      80.00
           .           .          .
           .           .          .

       80070118      78.58      79.00
       80070119      78.61     999.00
       80070120      78.64     999.00
```

day, and hour (YYMMDDhh format), (ii) the running wet-bulb temperature, and (iii) the actual wet-bulb temperature for that hour. The value of "999.0" in the wet-bulb temperature column indicates a hour for which no observation is available. As can be observed from Figure 4.17, both the second and third windows fall in periods during which only tri-hourly observations are available.

4.5 Numerical Simulation of UHS Cooling Tower Performance

The final step in the UHS cooling tower performance analysis is the accident simulation. This step is carried out using program UHSSIM which has already been discussed in Section 4.3 in the context of the time-constant evaluation. For that purpose, the program was run with a user specified initial basin temperature and a user specified constant wet-bulb temperature. In contrast, the time-varying wet-bulb temperature windows selected by the data scanning procedure are used for the actual accident simulation.

For a nonconstant wet-bulb temperature case, the user must set the control variable CONWB false and add Section 6 to the end of the input data file wherein the window number and starting time of the accident are specified. Normally window number 1, i.e., the window with the greatest running wet-bulb temperature, is used and the starting time is adjusted so that the peak running wet-bulb temperature and the peak accident-heat-load increment coincide. However, the freedom to examine alternative windows and to alter the starting time of the accident allow the sensitivity of predictions to be studied. A low sensitivity to the manner in which the analysis is carried out enhances the significance of the end result thus obtained.

For the test case, a simulation was made for 100 hours after start of the accident with results printed at the end of each hour. The input file is the same as that shown in Figure 4.14 except that the control variable CONWB is set false and the window number (NWINDW) and starting time of the accident within the window (NSTART) are added as the last line of the file. NWINDW is initially set to 1 to force use of the primary window obtained by the data scanning procedure. NSTART is initially set to 78. This value is the difference between the location of the maximum running wet-bulb temperature within the window (101) and the location of the maximum heat-load increment (23) determined from the results of the constant-wet-bulb-temperature simulation presented in Figure 4.17 and Table 4.6. For this choice, the maximum basin temperature does indeed reach a maximum of 89.90 °F, 23 hours after initiation of the accident as shown in Table 4.8.

4.6 Sensitivity Study of Meteorological Data Scanning Procedure

Figure 4.18 illustrates the sensivity of the maximum basin temperature predicted by program UHSSIM to the starting time of the accident. Here, maximum predicted basin temperature is plotted versus the time displacement between the start of the accident and the location of the maximum running wet-hulb temperature within the primary data window. Thus, an offset of 0 hours means that the accident was initiated at hour 101, i. e., the time of maximum running wet-bulb temperature within the window. Similarly, an offset of 60 hours means that the accident was started at hour 41, sixty hours before the time of maximum running wet-bulb temperature. Notably, each hour on this plot represents a separate run of program UHSSIM.

Table 4.8 Variation of Important Meteorological, Heat Load, and Basin Variables with Time During LOCA.

TIME (HR)	TBASIN (F)	BMASS (LBM)	S (PPT)	TWB (F)	Q (BTU/HR)	L (LBM/HR)
0.0	81.04	5.42E+07	5.0	81.0	6.62E+07	6.07E+06
1.0	81.83	5.41E+07	5.0	81.0	1.93E+08	6.07E+06
2.0	82.66	5.39E+07	5.0	81.0	1.93E+08	6.07E+06
3.0	83.40	5.38E+07	5.0	80.0	1.93E+08	6.07E+06
4.0	84.04	5.37E+07	5.1	80.0	1.93E+08	6.07E+06
5.0	84.62	5.35E+07	5.1	80.0	1.93E+08	6.07E+06
6.0	85.15	5.33E+07	5.1	80.0	1.93E+08	6.07E+06
7.0	85.60	5.32E+07	5.1	79.0	1.93E+08	6.07E+06
8.0	86.00	5.30E+07	5.1	79.0	1.93E+08	6.07E+06
9.0	86.36	5.29E+07	5.1	79.0	1.93E+08	6.07E+06
10.0	86.68	5.27E+07	5.1	79.0	1.93E+08	6.07E+06
11.0	86.97	5.25E+07	5.2	78.0	1.92E+08	6.07E+06
12.0	87.23	5.23E+07	5.2	80.0	1.90E+08	6.07E+06
13.0	87.53	5.22E+07	5.2	80.0	1.88E+08	6.07E+06
14.0	87.79	5.20E+07	5.2	80.0	1.86E+08	6.07E+06
15.0	88.07	5.19E+07	5.2	82.0	1.84E+08	6.07E+06
16.0	88.40	5.17E+07	5.2	82.0	1.82E+08	6.07E+06
17.0	88.67	5.15E+07	5.3	82.0	1.81E+08	6.07E+06
18.0	88.96	5.14E+07	5.3	83.0	1.80E+08	6.07E+06
19.0	89.25	5.12E+07	5.3	83.0	1.79E+08	6.07E+06
20.0	89.50	5.11E+07	5.3	83.0	1.78E+08	6.07E+06
21.0	89.71	5.09E+07	5.3	83.0	1.66E+08	6.07E+06
22.0	89.85	5.08E+07	5.3	82.0	1.55E+08	6.07E+06
23.0	89.90	5.06E+07	5.4	82.0	1.43E+08	6.07E+06
24.0	89.89	5.05E+07	5.4	81.0	1.31E+08	6.07E+06
25.0	89.84	5.04E+07	5.4	81.0	1.30E+08	5.80E+06
26.0	89.74	5.02E+07	5.4	80.0	1.28E+08	5.80E+06
27.0	89.55	5.01E+07	5.4	78.0	1.27E+08	5.80E+06
28.0	89.26	5.00E+07	5.4	76.0	1.25E+08	5.80E+06
29.0	88.92	4.98E+07	5.4	76.0	1.24E+08	5.80E+06
30.0	88.65	4.97E+07	5.5	77.0	1.22E+08	5.80E+06
31.0	88.40	4.96E+07	5.5	76.0	1.21E+08	5.80E+06
32.0	88.12	4.95E+07	5.5	76.0	1.19E+08	5.80E+06
33.0	87.88	4.93E+07	5.5	76.0	1.18E+08	5.80E+06
34.0	87.65	4.92E+07	5.5	76.0	1.16E+08	5.80E+06
35.0	87.47	4.91E+07	5.5	78.0	1.15E+08	5.80E+06
36.0	87.42	4.90E+07	5.5	79.0	1.13E+08	5.80E+06
37.0	87.38	4.89E+07	5.5	79.0	1.12E+08	5.80E+06
38.0	87.40	4.88E+07	5.6	81.0	1.10E+08	5.80E+06
39.0	87.51	4.87E+07	5.6	82.0	1.09E+08	5.80E+06
40.0	87.64	4.86E+07	5.6	82.0	1.07E+08	5.80E+06
41.0	87.78	4.85E+07	5.6	83.0	1.06E+08	5.80E+06
42.0	87.87	4.84E+07	5.6	81.0	1.04E+08	5.80E+06
43.0	87.92	4.83E+07	5.6	82.0	1.03E+08	5.80E+06
44.0	87.95	4.82E+07	5.6	81.0	1.01E+08	5.80E+06
45.0	87.91	4.81E+07	5.6	80.0	9.99E+07	5.80E+06
46.0	87.86	4.80E+07	5.6	81.0	9.84E+07	5.80E+06
47.0	87.86	4.79E+07	5.7	81.0	9.69E+07	5.80E+06
48.0	87.84	4.78E+07	5.7	81.0	9.54E+07	5.80E+06

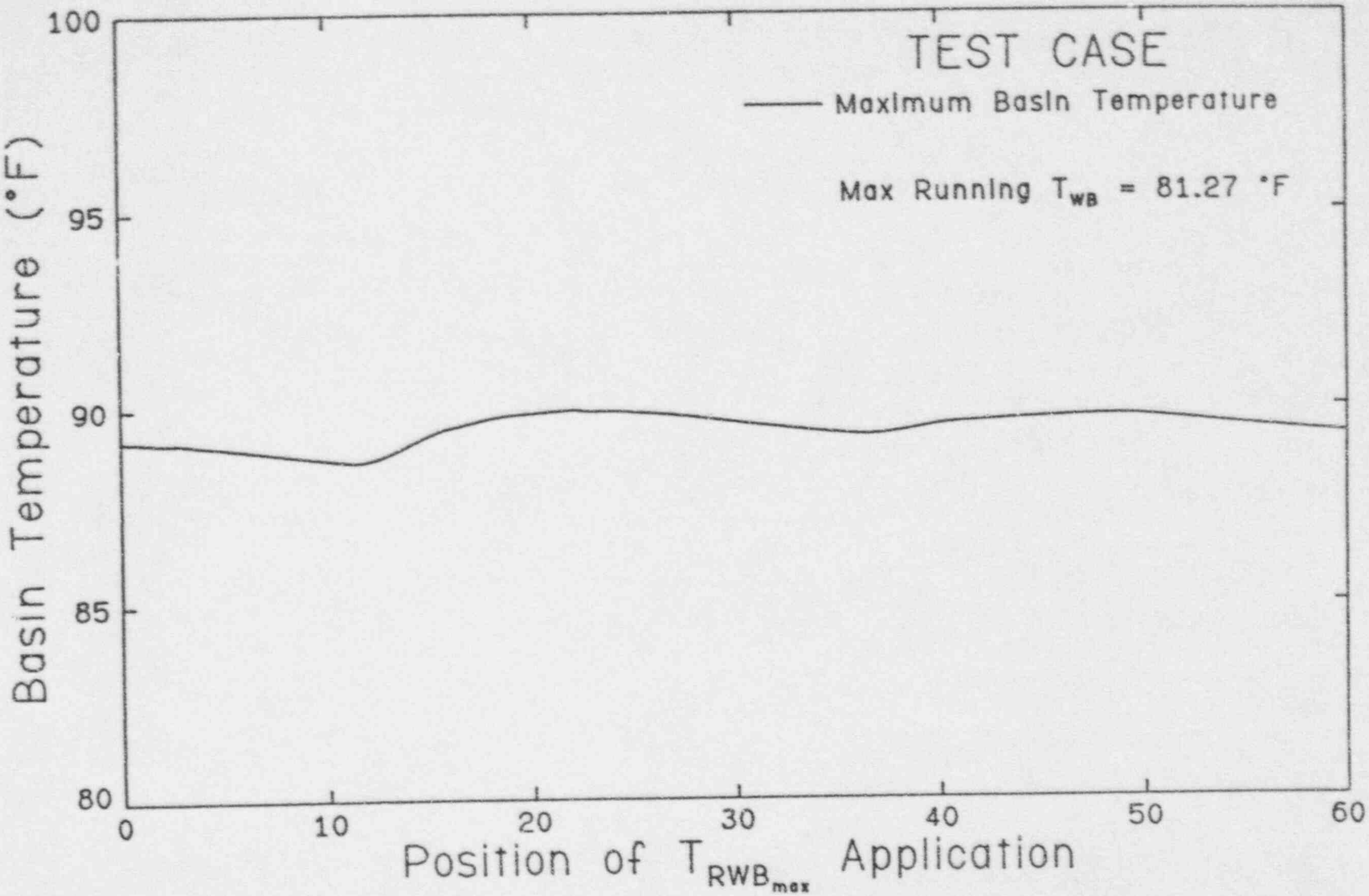


Figure 4.18 Maximum Basin Temperature as Function of Displacement of Accident Initiation from Time of Maximum Running Wet-Bulb Temperature.

Figure 4.18 shows first that the maximum predicted basin temperature is not highly sensitive to changes in accident starting time since the difference between the smallest (88.65 °F) and largest (89.94 °F) values plotted is less than one and a half degrees Fahrenheit. Second, the greatest value occurs at an offset of 22 hours, only one hour different from the initial choice based on the results of the constant-wet-bulb-temperature simulation. Varying the starting time of the accident by 3 hours on either side of the maximum yields a change in maximum predicted basin temperature of less than 0.1 °F. Finally, the plot reflects the diurnal character of the ambient wet-bulb temperature data.

The correlation between maximum predicted basin temperature and maximum running wet-bulb temperature was studied by examining five different windows that had been saved during the data scanning procedure. The results of this study are summarized in Table 4.9, wherein the time of the window, the corresponding maximum running wet-bulb temperature, the maximum predicted basin temperature, the initial basin temperature, and the heat load increment (defined as the difference between the maximum predicted basin temperature and the maximum running wet-bulb temperature) are listed. The results show that the maximum running wet-bulb temperature is a very good, although not necessarily perfect, predictor of maximum basin temperature. The heat load increment appears to decline with increasing maximum running wet-bulb temperature, although the underlying physical basis of this trend is not clear.

Figure 4.19 shows the basin temperature response under accident conditions for five different initial basin temperatures. Although a

Table 4.9 Correlation Between Maximum Predicted Basin Temperature and Maximum Running Wet-Bulb Temperature for Five Meteorological Windows.

Window Date	Maximum Running Wet-Bulb Temperature(F)	Maximum Basin Temperature(F)	Initial Basin Temperature(F)	Heat Load Increment(F), $T_{B_{max}} - T_{RWB_{max}}$
14:00 hours, 8/13/81 to 21:00 hours, 8/21/81	81.27	89.94	81.04	8.67
19:00 hours, 8/06/69 to 02:00 hours, 8/15/69	80.70	89.57	79.38	8.87
10:00 hours, 8/04/59 to 17:00 hours, 8/12/59	79.74	88.88	78.75	9.14
19:00 hours, 7/01/57 to 02:00 hours, 7/10/57	79.56	88.74	78.84	9.18
14:00 hours, 8/27/56 to 21:00 hours, 9/04/56	79.32	88.72	78.59	9.40

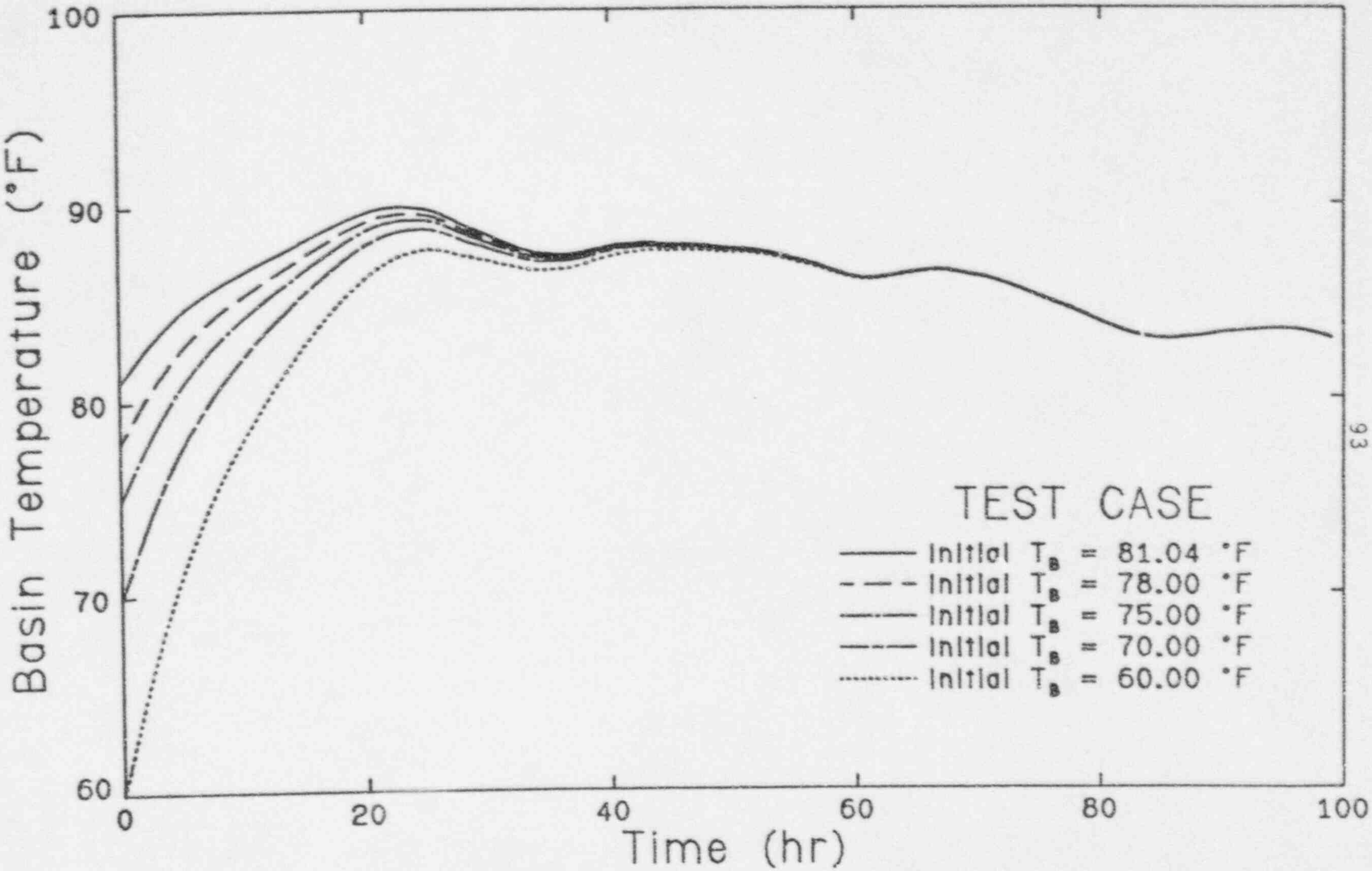


Figure 4.19 Basin Temperature During LOCA as Function of Time from Beginning of Accident for Variable Initial Basin Temperature.

higher initial basin temperature does consistently lead to a higher basin temperature at any given time, the effect of the initial condition dies away roughly exponentially in time with a time constant equal to that of the tower/basin system. This behavior may be expected due to the approximate first-order response of the system as discussed earlier.

Figure 4.20 compares the basin temperature history predicted by the simulation program UHSSIM with that given using the simplified model for the primary window of time-varying wet-bulb temperatures selected by program NWS. Using this set of meteorological conditions, program TMCON determined a time constant of 11.73 hours which is only negligibly different from the value of 11.46 hours found using a constant wet-bulb temperature of 81 °F. This comparison provides an important test of the validity of the time constant as well as on the validity of the first-order approximation itself.

4.7 Final Simulation Results

The results for the primary set of meteorological conditions are summarized in Table 4.10 and Figures 4.21 through 4.23. The basin temperature (assumed equal to the cold water return temperature), plotted in Figure 4.21, reaches a maximum of 89.94 °F 23 hours after initiation of the hypothetical accident, as noted earlier. The variation of basin mass with time is shown in Figure 4.22. As can be seen here, basin mass declines slowly due to evaporative losses. Figure 4.23 shows the variation of heat load to the tower/basin system.

Since the basin temperature maximum of 89.94°F is well below the design criterion of 93°F, the system is judged to meet the maximum tem-

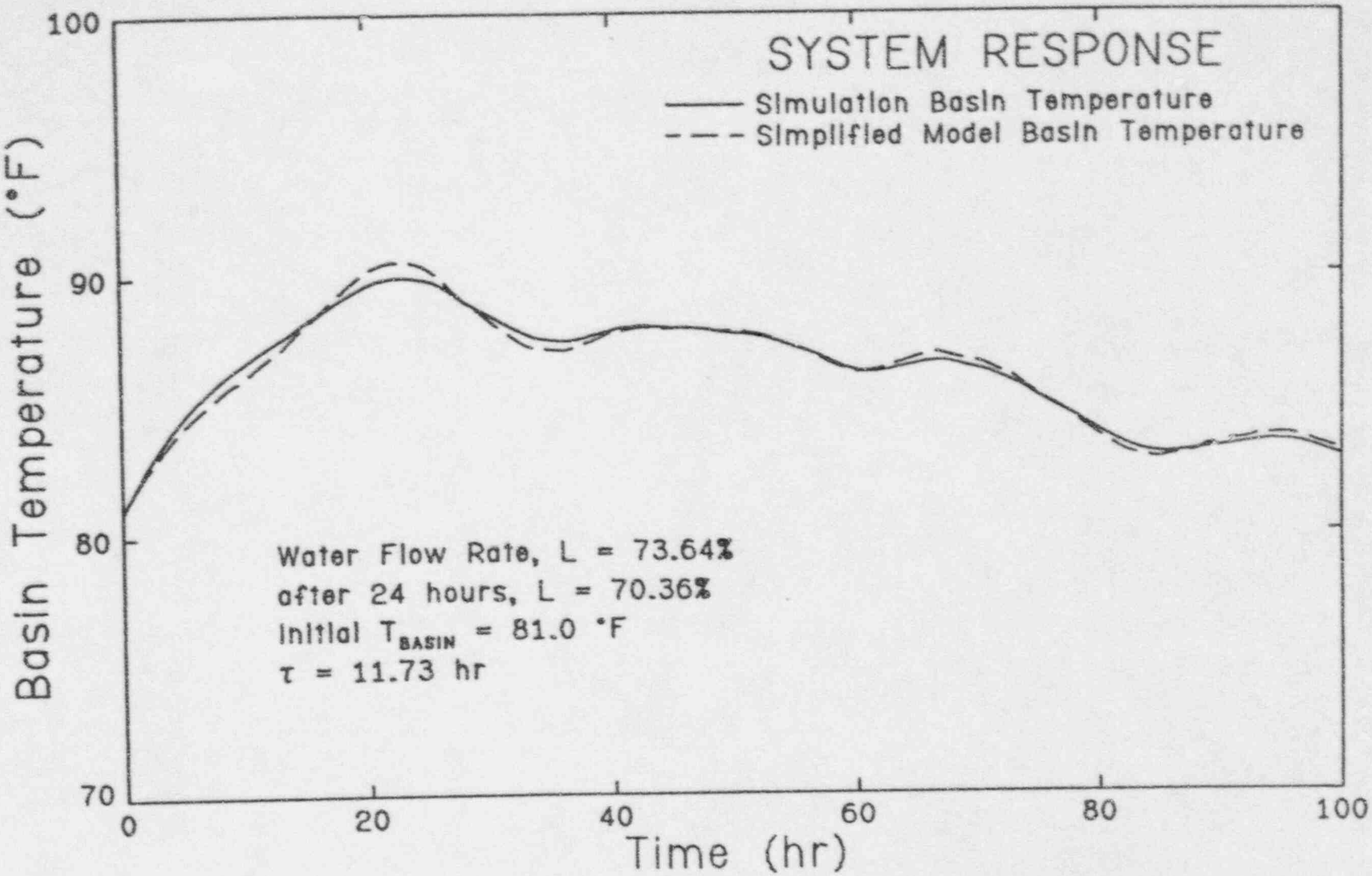


Figure 4.20 Comparison of Temperature Histories Predicted by Full Tower/Basin Simulation (UHSSIM) and Simplified Model for Primary Meteorological Window.

Table 4.10 Variation of Important Meteorological, Heat Load, and Basin Variables with Time During LOCA.

TIME (HR)	TBASIN (F)	BMASS (LBM)	S (PPT)	TWB (F)	Q (BTU/HR)	L (LBM/HR)
0.0	81.04	5.42E+07	5.0	81.0	6.62E+07	6.07E+06
1.0	81.84	5.41E+07	5.0	81.0	1.93E+08	6.07E+06
2.0	82.64	5.39E+07	5.0	80.0	1.93E+08	6.07E+06
3.0	83.34	5.38E+07	5.0	80.0	1.93E+08	6.07E+06
4.0	83.99	5.36E+07	5.1	80.0	1.93E+08	6.07E+06
5.0	84.58	5.35E+07	5.1	80.0	1.93E+08	6.07E+06
6.0	85.09	5.33E+07	5.1	79.0	1.93E+08	6.07E+06
7.0	85.52	5.32E+07	5.1	79.0	1.93E+08	6.07E+06
8.0	85.92	5.30E+07	5.1	79.0	1.93E+08	6.07E+06
9.0	86.29	5.28E+07	5.1	79.0	1.93E+08	6.07E+06
10.0	86.62	5.27E+07	5.1	78.0	1.93E+08	6.07E+06
11.0	86.96	5.25E+07	5.2	80.0	1.92E+08	6.07E+06
12.0	87.28	5.23E+07	5.2	80.0	1.90E+08	6.07E+06
13.0	87.57	5.22E+07	5.2	80.0	1.88E+08	6.07E+06
14.0	87.88	5.20E+07	5.2	82.0	1.86E+08	6.07E+06
15.0	88.22	5.19E+07	5.2	82.0	1.84E+08	6.07E+06
16.0	88.52	5.17E+07	5.2	82.0	1.82E+08	6.07E+06
17.0	88.82	5.15E+07	5.3	83.0	1.81E+08	6.07E+06
18.0	89.12	5.14E+07	5.3	83.0	1.80E+08	6.07E+06
19.0	89.39	5.12E+07	5.3	83.0	1.79E+08	6.07E+06
20.0	89.63	5.11E+07	5.3	83.0	1.78E+08	6.07E+06
21.0	89.81	5.09E+07	5.3	82.0	1.66E+08	6.07E+06
22.0	89.90	5.08E+07	5.3	82.0	1.55E+08	6.07E+06
23.0	89.94	5.06E+07	5.4	81.0	1.43E+08	6.07E+06
24.0	89.89	5.05E+07	5.4	81.0	1.31E+08	6.07E+06
25.0	89.81	5.04E+07	5.4	80.0	1.30E+08	5.80E+06
26.0	89.62	5.02E+07	5.4	78.0	1.28E+08	5.80E+06
27.0	89.33	5.01E+07	5.4	76.0	1.27E+08	5.80E+06
28.0	89.00	5.00E+07	5.4	76.0	1.25E+08	5.80E+06
29.0	88.72	4.98E+07	5.4	77.0	1.24E+08	5.80E+06
30.0	88.47	4.97E+07	5.5	76.0	1.22E+08	5.80E+06
31.0	88.20	4.96E+07	5.5	76.0	1.21E+08	5.80E+06
32.0	87.95	4.94E+07	5.5	76.0	1.19E+08	5.80E+06
33.0	87.72	4.93E+07	5.5	76.0	1.18E+08	5.80E+06
34.0	87.55	4.92E+07	5.5	78.0	1.16E+08	5.80E+06
35.0	87.49	4.91E+07	5.5	79.0	1.15E+08	5.80E+06
36.0	87.45	4.90E+07	5.5	79.0	1.13E+08	5.80E+06
37.0	87.47	4.89E+07	5.5	81.0	1.12E+08	5.80E+06
38.0	87.58	4.88E+07	5.6	82.0	1.10E+08	5.80E+06
39.0	87.71	4.87E+07	5.6	82.0	1.09E+08	5.80E+06
40.0	87.84	4.86E+07	5.6	83.0	1.07E+08	5.80E+06
41.0	87.94	4.85E+07	5.6	81.0	1.06E+08	5.80E+06
42.0	87.97	4.84E+07	5.6	82.0	1.04E+08	5.80E+06
43.0	88.01	4.83E+07	5.6	81.0	1.03E+08	5.80E+06
44.0	87.97	4.82E+07	5.6	80.0	1.01E+08	5.80E+06
45.0	87.93	4.81E+07	5.6	81.0	9.99E+07	5.80E+06
46.0	87.92	4.80E+07	5.6	81.0	9.84E+07	5.80E+06
47.0	87.90	4.79E+07	5.7	81.0	9.69E+07	5.80E+06
48.0	87.85	4.78E+07	5.7	80.0	9.54E+07	5.80E+06

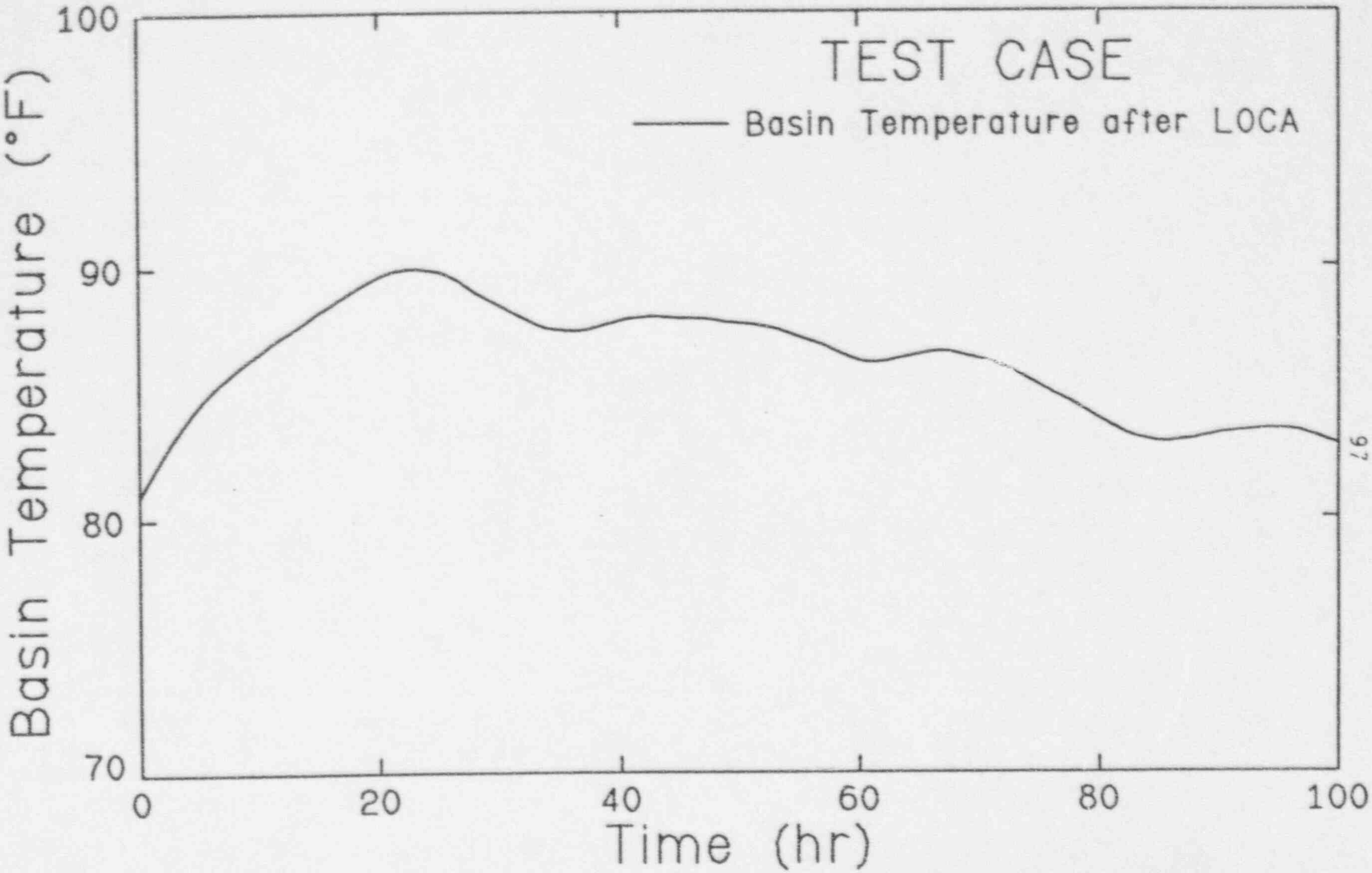


Figure 4.21 Basin Temperature During LOCA as Function of Time from Beginning of Accident.

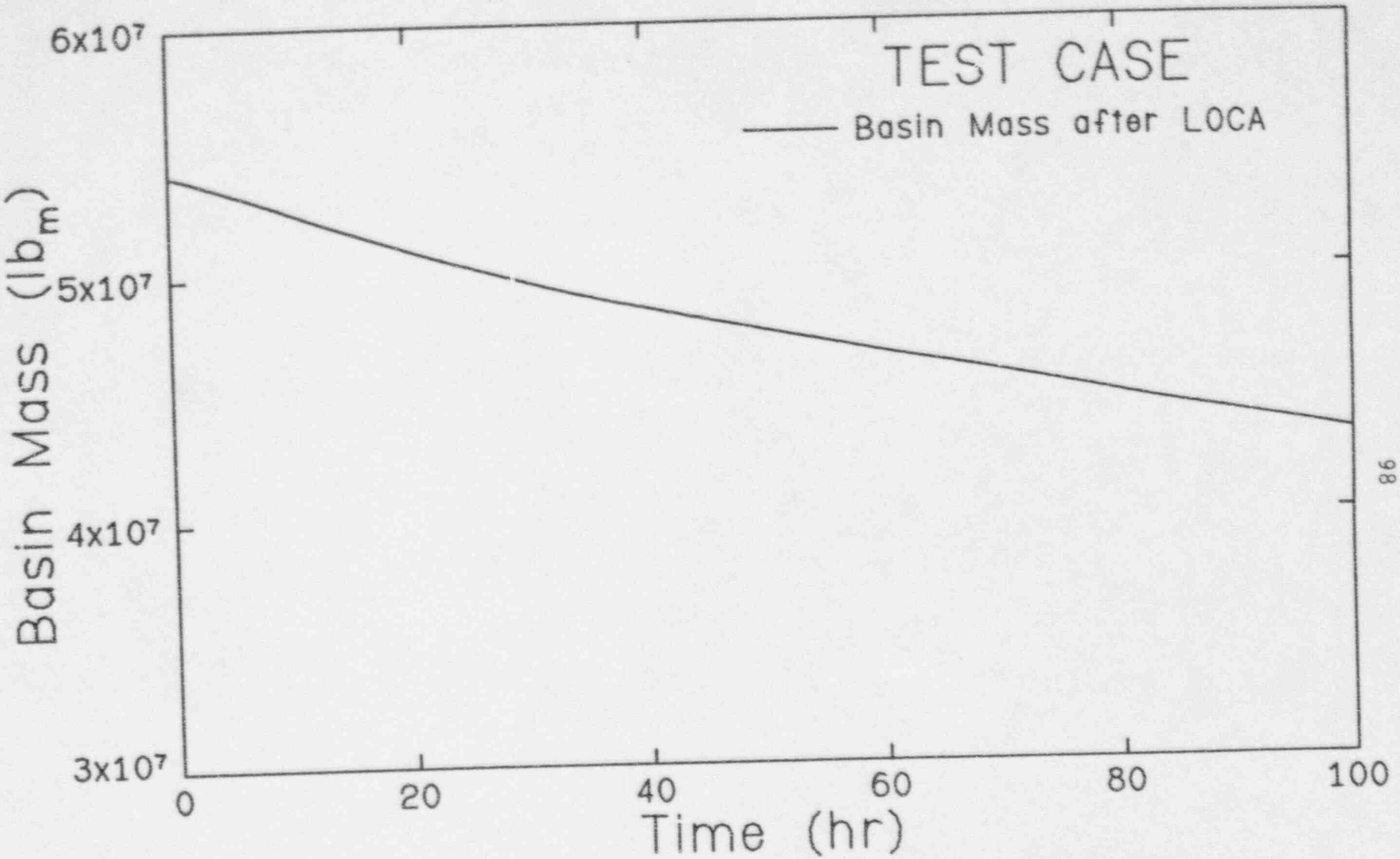


Figure 4.22 Basin Mass During LOCA as Function of Time from Beginning of Accident.

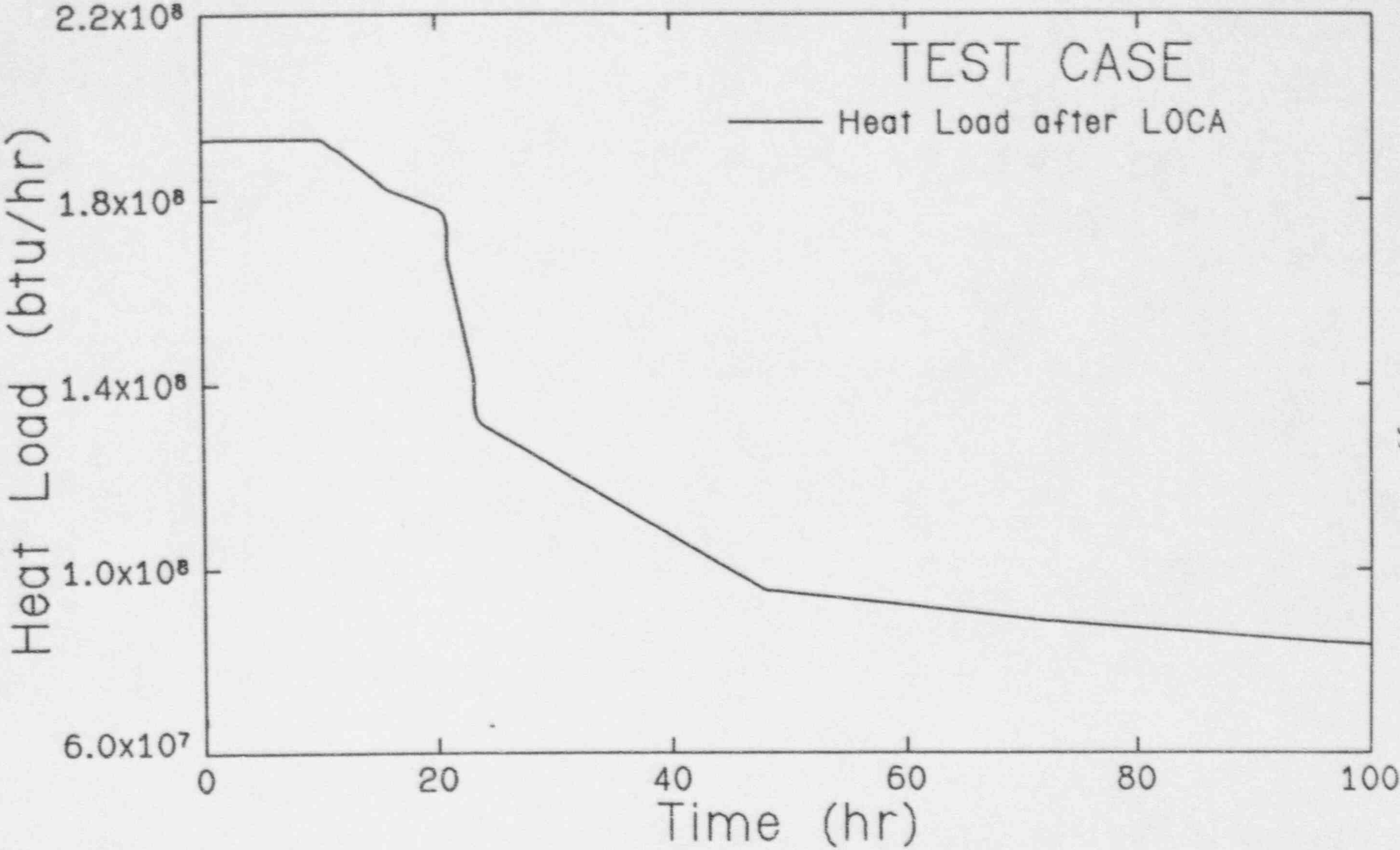


Figure 4.23 Heat Load to UHS System During LOCA as Function of Time from Beginning of Accident.

perature limit imposed. And since the basin volume is roughly 443,000 gallons of water after 30 days, the system meets the requirements of 30 days of operation without make-up water supply. The salinity content of the basin after the 30-day period was found to be 76.8 ppt as shown in Figure 4.24, indicating that the effects of dissolved solids was negligibly small for this case.

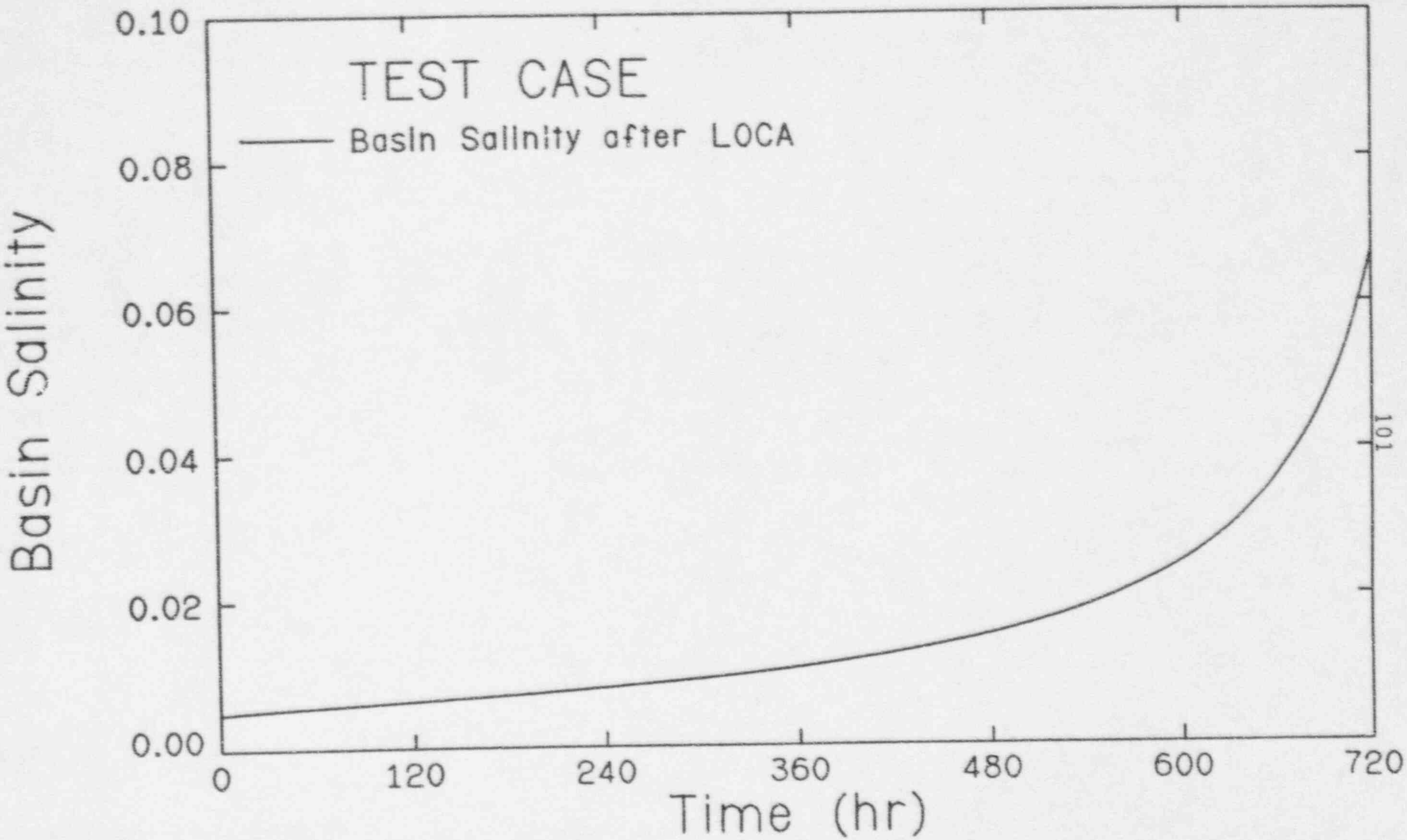


Figure 4.24 Basin Salinity During LOCA as Function of Time from Beginning of Accident for 30 days.

5. CONCLUSIONS AND RECOMMENDATIONS

The analysis of ultimate heat sink cooling tower performance during a design-basis accident is of importance in determining whether the proposed system is adequate to allow safe shutdown and cooldown of the nuclear reactor(s). The U. S. Nuclear Regulatory Commission has previously published methods of analysis for UHS cooling ponds and spray ponds. In the application of these methods, parameters are selected to provide the worst-case performance which may be reasonably expected for the proposed design and given the historical meteorological data for the site. Thus, a high degree of conservatism is built into the analysis owing to the severe consequences associated with the failure of the UHS cooling system.

To extend this type of analysis to UHS cooling towers, we have developed a set of computer codes which predict the thermal performance of the UHS cooling tower. Like the methods developed for the UHS cooling pond and the UHS spray pond, this method utilizes the full period of meteorological data recorded by the National Weather Service for the station most representative of the proposed plant. The program NWS was developed to easily handle the large amounts of data on the National Weather Service tapes and to identify the periods of worst-case system performance. Through example, the validity of the NWS methodology is demonstrated.

Based on a theoretical tower/heat exchanger analysis, a method is developed for calculating the tower performance characteristic values. Program PDAP, which calculates the tower performance characteristic

values and the predicted tower cold water temperature, allows convenient use of manufacturer's performance data as input. The prediction of tower cold water temperature by program PDAP allows critical evaluation of the manufacturer's design curves, as shown by the comparisons given in Chapter 4.

Once a reliable simulation of cooling tower performance is obtained, a simplified model of the tower/basin system is constructed assuming a simple first-order (exponential relaxation) response of basin temperature with time. The time constant of this simplified model is determined using a wet-bulb temperature representative of the accident scenario. This time constant is then used in the data scanning procedure (in program NWS) wherein a "running average wet-bulb temperature" is calculated for each hour for which a meteorological observation is available. The period of worst-case thermal performance then corresponds to the period of maximum running wet-bulb temperature. The details of this method as well as the underlying theoretical foundation is fully discussed in Chapter 3. The validity of the approach is further demonstrated by way of example in Chapter 4.

Using the worst-case meteorological conditions determined by program NWS, the design-basis accident is simulated and the basin conditions of temperature, mass, and salinity are determined as functions of time for up to 30 days following the accident. The system can then be judged as to whether the design is sufficient to safely shutdown and cooldown the nuclear reactor(s) as required by U. S. Nuclear Regulatory Commision Regulatory Guide 1.27. In summary, the procedure presented in

this work provides an objective means of evaluating the adequacy of the UHS cooling tower system.

To completely insure the reliability of the predictions of UHS system performance made using information supplied in the Final Safety Analysis Report, we recommend that all cases be re-evaluated using the cooling tower acceptance data once it is available.

REFERENCES

Beckwith, T. G., and N. L. Buck, 1973: Mechanical Measurements (Massachusetts: Addison-Wesley Publishing Company).

Brady, D. K., W. L. Graves, and J. C. Geyer, 1969: "Surface Heat Exchange at Plant Cooling Lakes," Report No. 5, Edison Electric Institute, EEI Publication 69-401.

Codell, R., and W. K. Nuttle, 1980: "Analysis of Ultimate Heat Sink Cooling Ponds," U. S. NRC Report NUREG-0693.

Codell, R., 1981: "Analysis of Ultimate-Heat-Sink Spray Ponds," U. S. NRC Report NUREG-0733.

Codell, R., 1982: "Comparison Between Field Data and Ultimate Heat Sink Cooling Pond and Spray Pond Models," U. S. NRC Report NUREG-0858.

Dunn, W. E., 1981: "Plotlib Subroutine Library User's Manual," Department of Mechanical and Industrial Engineering, University of Illinois, Urbana, IL.

Dunn, W. E., 1984: Personal Communication, University of Illinois at Urbana-Champaign.

Electric Power Research Institute, Special Report CS-3212-SR, 1983: "TEFERI, Numerical Model for Calculating the Performance of an Evaporative Cooling Tower," Electric Power Research Institute, Palo Alto, CA.

Hadlock, R. K., and O. B. Abbey, 1978: "Thermal Performance Measurements on Ultimate Heat Sinks - Cooling Ponds," U. S. NRC Report NUREG/CR-0008, PNL-2463.

Hadlock, R. K., and O. B. Abbey, 1981: "Thermal Performance and Water Utilization Measurements on Ultimate Heat Sinks - Cooling Ponds and Spray Ponds," U. S. NRC Report NUREG/CR-1886, PNL-3689.

Holl, J. H., 1951: "Heat and Mass Transfer in Evaporative Cooling," Masters Thesis, University of Illinois at Urbana-Champaign.

Majumdar, A. K., A. K. Singhal, and D. B. Spalding, 1983: "Numerical Modeling of Wet Cooling Towers - Part 1: Mathematical and Physical Models," J. Heat Transfer, 105, 728-735.

Majumdar, A. K., A. K. Singhal, H. E. Reilly, and J. A. Bartz, 1983: "Numerical Modeling of Wet Cooling Towers - Part 2: Application to Natural Draft Towers," J. Heat Transfer, 105, 736-743.

Majumdar, A. K., A. K. Singhal, and D. B. Spalding, 1983: "VERA2D - A Computer Program for Two-Dimensional Analysis of Flow, Heat, and Mass Transfer in Evaporative Cooling Towers: Vol. 1 - Mathematical, Formulation, Solution Procedure, and Application," EPRI Report No. CS-2923.

Majumdar, A. K., and A. K. Singhal, 1983: "VERA2D - A Computer Program for Two-Dimensional Analysis of Flow, Heat and Mass Transfer in Evaporative Cooling Towers: Vol. 2 - User's Manual," EPRI Report No. CS-2 '23.

Merkel, F., 1925: "Verdunslungshühlung," VDI-Forschungsarbeiten, 275.

Nahavandi, A. N., R. M. Kershah, and B. J. Serico, 1975: "The Effects of Evaporative Losses in the Analysis of Counterflow Cooling Towers," Nuc. Eng. Des., 32, 29-36.

Nahavandi, A. N., and J. J. Oellinger, 19/7: "An Improved Model for the Analysis of Evapcrative Counterflow Cooling Towers," Nuc. Eng. Des., 40, 327-336.

Poppe, M., 1973: "Wärme-und Sloffubertragung bei der Verdumslungskiih-lung in Gergenund Kreuzstrom," VDI-Forschungsheft, 560, 39-44.

Reynclds, W. C., and H. C. Perkins, 1977: Thermodynamics (New York: McGraw-Hill Book Company).

Skelland, A. H. P., 1974: Diffusional Mass Transfer (New York: John Wiley and Sons).

Stein, H. L., 1982: "Cooling Towers," Specifying Engineer, 47, 121-126.

Sullivan, S. M., and W. E. Dunn, 1984: "User's Manual for UHS Thermal Performance Analysis Codes," Department of Mechanical and Industrial Engineering, University of Illinois, Urbana, IL.

Sutherland, J. W., 1983: "Analysis of Mechanical-Draught Counterflow Air/Water Cooling Towers," J. Heat Transfer, 105, 576-586.

U. S. Nuclear Regulatory Commission, Regulatory Guide 1.27, 1976: "Ultimate Heat Sink for Nuclear Power Plants."

Yadigaroglu, G., and E. J. Pastor, 1974: "An Investigation of the Accuracy of the Merkel Equation for Evaporative Cooling Tower Calculations," ASME Paper 74-HT-59.

APPENDIX A SIMULATION COMPUTER CODES

```
      PROGRAM PDAP (TAPE5,TAPE6,TAPE7,OUTPUT)
C
C...  THIS PROGRAM CALCULATES, GIVEN THE INLET WET-BULB
C...  TEMPERATURE, THE RANGE, AND THE COLD WATER TEMPERATURE
C...  FROM PERFORMANCE CURVES, A BEST FIT LEAST SQUARES VALUE
C...  OF THE TOWER CHARACTERISTIC, KAV/L, FOR A UHS COOLING TOWER.
C...  THIS PROGRAM ALSO CALCULATES, USING THIS PREDICTED KAV/L, A
C...  COLD WATER TEMPERATURE OF THE TOWER.
C
C...  THE DATA IS ENTERED FROM LOWEST TO HIGHEST RANGE VALUES
C...  IN ORDER OF INCREASING WET-BULB TEMPERATURE WITHIN EACH
C...  RANGE. ENTERING DATA IN THIS MANNER PERMITS THE OUTPUT
C...  TO BE USED DIRECTLY FOR PLOTTING IN PROGRAM PDAPLT.
C
      CHARACTER*80 CASE,FLOW,RLABEL(6)
      PARAMETER (MAXDAT=50)
      EXTERNAL ST2
      REAL RAN(MAXDAT),NWB(MAXDAT)
      COMMON WBD(MAXDAT),RANGE(MAXDAT),TCOLD(MAXDAT),TPRED(MAXDAT)
      COMMON DIFF(MAXDAT),NPT
      COMMON /TOWERC/ S,LOVERG
      REAL KAVL,KLOW,KHGH,LOVERG
      DATA KLOW/0.5/,KHGH/3.0/,TOL/0.01/
C
C...  READ IN PERFORMANCE DATA
C
C          S          - SALINITY
C          LOVERG     - LIQUID TO GAS FLOW RATE RATIO
C          PERC       - LIQUID FLOW RATE PERCENTAGE BEING EVALUATED
C          NWB        - ARRAY CONTAINING THE NUMBER OF WET-BULB TEMPERATURES
C                       WITHIN EACH RANGE VALUE
C          RAN        - ARRAY CONTAINING THE RANGE VALUES
C          NRAN       - NUMBER OF RANGE VALUES WITHIN A FLOW RATE
C          CASE       - 80-CHARACTER LABEL USED TO IDENTIFY OUTPUT
C          FLOW       - 80-CHARACTER LABEL USED TO IDENTIFY LIQUID FLOW
C                       RATES WITHIN THE OUTPUT
C          RLABEL     - 80-CHARACTER LABEL USED TO IDENTIFY TEMPERATURE
C                       VAULUES READ WITHIN SPECIFIC RANGE VALUES
C
      READ (5,100,END=99) CASE
      WRITE (6,100) CASE
    5 NDATA=1
      NPT=0
      READ (5,100,END=99) FLOW
      READ (5,*) LOVERG,PERC,S
      READ (5,*) NRAN
      DO 1 J=1,NRAN
        READ (5,100) RLABEL(J)
        READ (5,*) RAN(J),NWB(J)
        DO 2 I=NDATA,NDATA+NWB(J)-1
          READ (5,*) WBD(I),TCOLD(I)
          RANGE(I)=RAN(J)
          NPT=NPT+1
    2   CONTINUE
      NDATA=NDATA+NWB(J)
    1 CONTINUE
C
C...  FIND LEAST SQUARES BEST FIT VALUE OF KAV/L
C
      KAVL=FMIN (KLOW,KHGH,ST2,TOL)
C
C...  WRITE RESULTS TO TAPE 6 FOR PLOTTING IN PROGRAM PDAPLT;
C...  AND TO TAPE 7 FOR PLOTTING IN PROGRAM KAVL.
C
      WRITE (6,100) FLOW
      WRITE (6,101) KAVL,LOVERG,PERC,S
      WRITE (6,105) NRAN
```

Figure A.1 Listing of Program PDAP

```
      NDATA=1
      DO 4 J=1,NRAN
      WRITE (6,100) RLABEL(J)
      WRITE (6,102) RAN(J),NWB(J)
      DO 3 I=NDATA,NDATA+NWB(J)-1
      WRITE (6,103) WBD(I),TCOLD(I),TPRED(I),DIFF(I)
    3 CONTINUE
      NDATA=NDATA+NWB(J)
    4 CONTINUE
      WRITE (7,104) KAVL,PERC
      GO TO 5
   99 STOP
C
C...  FORMAT STATEMENTS
C
  100 FORMAT (A)
  101 FORMAT (2F8.4,F8.1,F8.4)
  102 FORMAT (2F10.1)
  103 FORMAT (4F10.4)
  104 FORMAT (2F10.4)
  105 FORMAT (I1)
      END
      REAL FUNCTION ST2 (KAVL)
C
C...  THIS FUNCTION IS USED TO CALCULATE THE SUM OF THE SQUARES
C...  OF THE DIFFERENCE OF THE PERFORMANCE CURVE TOUT AND THE
C...  TOWER PREDICTED TOUT.
C
      REAL KAVL
      PARAMETER (MAXDAT=50)
      COMMON WBD(MAXDAT),RANGE(MAXDAT),TCOLD(MAXDAT),TPRED(MAXDAT)
      COMMON DIFF(MAXDAT),NPT
C
C...  LOOP TO CALCULATE SUM OF SQUARES
C
      ST2=0.0
      DO 1 I=1,NPT
C
C...  CALL SUBROUTINE TOWER USED TO CALCULATE TOUT OF THE TOWER.
C
      CALL TOWER (TCOLD(I)+RANGE(I),WBD(I),KAVL,TPRED(I))
      ST2=ST2+(TPRED(I)-TCOLD(I))**2
      DIFF(I)=TPRED(I)-TCOLD(I)
    1 CONTINUE
      RETURN
      END
      SUBROUTINE TOWER (TIN,TWB,KAVL,TOUT)
      SAVE TOLD,A,B,C,D,T,HHAT
      REAL LOVERG,KAVL
      COMMON /TOWERC/ S,LOVERG
      DIMENSION A(10),B(10),C(10),D(10),T(12),HHAT(12),WORK(20)
      DATA A/0.,9*1./,B/10*-2./,C/10*1./,D/10*0./
      DATA N/10/,NP1/11/,NP2/12/,AN/0.05/
      DATA A0/1.0/,A1/-0.66177/,A2/0.36508/,A3/-5.25926/
      DATA C0/13.3185/,C1/1.976/,C2/0.6445/,C3/0.1299/
      DATA TT1/212./,TT2/459.67/,WRATIO/0.622/
      DATA HV1/0.427555/,HV2/1062.3/
      DATA HA1/0.2403/
      DATA SMALL/0.01/,TOLD/0.0/
      E(X)=(((C3*X-C2)*X+C1)*X+C0)*X
      F(X)=A0+X*(A1+X*(A2+X*A3))
      T(NP1)=TIN
      T(NP2)=TWB
      AK=AN*KAVL
    1 D(1)=(B(1)+C(1))*TWB
      D(N)=-C(N)*TIN
      CALL TDMS (A,B,C,T,D,N,IER,WORK)
      IF (ABS(TOLD-T(1)).LT.SMALL) GO TO 4
```

Figure A.1 Continued

```
      TOLD=T(1)
      DO 2 I=1,NP2
      TT=(TT1-T(I))/(T(I)+TT2)
      PP=EXP(-E(TT))*P(S)
      W=WRATIO*PP/(1.0-PP)
      HV=HV1*T(I)+HV2
      HA=HA1*T(I)
    2 HHAT(I)=HA+W*HV
      DO 3 I=2,N
      DH=(HHAT(I+1)-HHAT(I-1))/(T(I+1)-T(I-1))
      Q=AK*(DH-LOVERG)
      A(I)=1.0+Q
    3 C(I)=1.0-Q
      B(1)=-1.0-AK*((HHAT(1)-HHAT(NP2))/(T(1)-T(NP2))+LOVERG)
      C(1)=1.0-AK*((HHAT(2)-HHAT(NP2))/(T(2)-T(NP2))-LOVERG)
      GO TO 1
    4 TOUT=T(1)
      TOLD=T(1)
      RETURN
      END
      REAL FUNCTION FMIN (AX,BX,F,TOL)
      REAL AX,BX,F,TOL
C
C       AN APPROXIMATION  X  TO THE POINT WHERE  F  ATTAINS A MINIMUM  ON
C   THE INTERVAL  (AX,BX)  IS DETERMINED.
C
C
C   INPUT..
C
C   AX     LEFT ENDPOINT OF INITIAL INTERVAL
C   BX     RIGHT ENDPOINT OF INITIAL INTERVAL
C   F      FUNCTION SUBPROGRAM WHICH EVALUATES  F(X)  FOR ANY  X
C          IN THE INTERVAL  (AX,BX)
C   TOL    DESIRED LENGTH OF THE INTERVAL OF UNCERTAINTY OF THE FINAL
C          RESULT ( .GE. 0.0D0)
C
C
C   OUTPUT..
C
C   FMIN  ABCISSA APPROXIMATING THE POINT WHERE  F  ATTAINS A MINIMUM
C
C
C       THE METHOD USED IS A COMBINATION OF  GOLDEN  SECTION  SEARCH  AND
C   SUCCESSIVE PARABOLIC INTERPOLATION.  CONVERGENCE IS NEVER MUCH SLOWER
C   THAN  THAT  FOR  A  FIBONACCI SEARCH.  IF  F  HAS A CONTINUOUS SECOND
C   DERIVATIVE WHICH IS POSITIVE AT THE MINIMUM (WHICH IS NOT  AT  AX  OR
C   BX),  THEN  CONVERGENCE  IS  SUPERLINEAR, AND USUALLY OF THE ORDER OF
C   ABOUT  1.324....
C       THE FUNCTION  F  IS NEVER EVALUATED AT TWO POINTS CLOSER TOGETHER
C   THAN  EPS*ABS(FMIN) + (TOL/3), WHERE EPS IS  APPROXIMATELY THE SQUARE
C   ROOT  OF  THE  RELATIVE  MACHINE  PRECISION.   IF  F  IS A UNIMODAL
C   FUNCTION AND THE COMPUTED VALUES OF  F  ARE  ALWAYS  UNIMODAL  WHEN
C   SEPARATED BY AT LEAST  EPS*ABS(X) + (TOL/3), THEN  FMIN  APPROXIMATES
C   THE ABCISSA OF THE GLOBAL MINIMUM OF  F  ON THE INTERVAL  AX,BX  WITH
C   AN ERROR LESS THAN  3*EPS*ABS(FMIN) + TOL.  IF  F  IS NOT UNIMODAL,
C   THEN FMIN MAY APPROXIMATE A LOCAL, BUT PERHAPS NON-GLOBAL, MINIMUM TO
C   THE SAME ACCURACY.
C       THIS FUNCTION SUBPROGRAM IS A SLIGHTLY MODIFIED  VERSION  OF  THE
C   ALGOL  60 PROCEDURE  LOCALMIN  GIVEN IN RICHARD BRENT, ALGORITHMS FOR
C   MINIMIZATION WITHOUT DERIVATIVES, PRENTICE - HALL, INC. (1973).
C
C
      REAL A,B,C,D,E,EPS,XM,P,Q,R,TOL1,TOL2,U,V,W
      REAL FU,FV,FW,FX,X
C
C  C IS THE SQUARED INVERSE OF THE GOLDEN RATIO
C
      C=0.5*(3.-SQRT(5.0))
```

Figure A.1 Continued

```
C
C  EPS IS APPROXIMATELY THE SQUARE ROOT OF THE RELATIVE MACHINE
C  PRECISION.
C
      EPS=1.0
   10 EPS=EPS/2.0
      TOL1=1.0+EPS
      IF (TOL1.GT.1.0) GO TO 10
      EPS=SQRT(EPS)
C
C  INITIALIZATION
C
      A=AX
      B=BX
      V=A+C*(B-A)
      W=V
      X=V
      E=0.0
      FX=F(X)
      FV=FX
      FW=FX
C
C  MAIN LOOP STARTS HERE
C
   20 XM=0.5*(A+B)
      TOL1=EPS*ABS(X)+TOL/3.0
      TOL2=2.0*TOL1
C
C  CHECK STOPPING CRITERION
C
      IF (ABS(X-XM).LE.(TOL2-0.5*(B-A))) GO TO 90
C
C IS GOLDEN-SECTION NECESSARY
C
      IF (ABS(E).LE.TOL1) GO TO 40
C
C  FIT PARABOLLLLA
C
      R=(X-W)*(FX-FV)
      Q=(X-V)*(FX-FW)
      P=(X-V)*Q-(X-W)*R
      Q=2.0*(Q-R)
      IF (Q.GT.0.0) P=-P
      Q=ABS(Q)
      R=E
      E=D
C
C  IS PARABOLA ACCEPTABLE
C
      IF (ABS(P).GE.ABS(0.5*Q*R)) GO TO 40
      IF (P.LE.Q*(A-X)) GO TO 40
      IF (P.GE.Q*(B-X)) GO TO 40
C
C  A PARABOLIC INTERPOLATION STEP
C
      D=P/Q
      U=X+D
C
C  F MUST NOT BE EVALUATED TOO CLOSE TO AX OR BX
C
      IF ((U-A).LT.TOL2) D=SIGN(TOL1,XM-X)
      IF ((B-U).LT.TOL2) D=SIGN(TOL1,XM-X)
      GO TO 50
C
C  A GOLDEN-SECTION STEP
C
   40 IF (X.GE.XM) E=A-X
```

Figure A.1 Continued

```
      IF (X.LT.XM) E=B-X
      D=C*E
C
C  F MUST NOT BE EVALUATED TOO CLOSE TO X
C
   50 IF (ABS(D).GE.TOL1) U=X+D
      IF (ABS(D).LT.TOL1) U=X+SIGN(TOL1,D)
      FU=F(U)
C
C  UPDATE  A, B, V, W, AND X
C
      IF (FU.GT.FX) GO TO 60
      IF (U.GE.X) A=X
      IF (U.LT.X) B=X
      V=W
      FV=FW
      W=X
      FW=FX
      X=U
      FX=FU
      GO TO 20
   60 IF (U.LT.X) A=U
      IF (U.GE.X) B=U
      IF (FU.LE.FW) GO TO 70
      IF (W.EQ.X) GO TO 70
      IF (FU.LE.FV) GO TO 80
      IF (V.EQ.X) GO TO 80
      IF (V.EQ.W) GO TO 80
      GO TO 20
   70 V=W
      FV=FW
      W=U
      FW=FU
      GO TO 20
   80 V=U
      FV=FU
      GO TO 20
C
C  END OF MAIN LOOP
C
   90 FMIN=X
      RETURN
      END
      SUBROUTINE TDMS (A,B,C,X,D,N,IER,WORK)
      DIMENSION A(N),B(N),C(N),X(N),D(N),WORK(1)
C
C     TDMS SOLVES A SET OF ALGEBRAIC EQUATIONS OF THE FORM:
C
C       A(I)*X(I-1)+B(I)*X(I)+C(I)*X(I+1)=D(I)
C
C     FOR I=1,2,....,N, USING THE TRIDIAGONAL MATRIX SOLUTION.
C
C     OTHER VARIABLES:
C
C     IER      = 0  -- NORMAL RETURN
C              = 1  -- ERROR BECAUSE N IS LESS THAN 1
C              = 2  -- EQUATIONS ARE SINGULAR
C
C     WORK    A WORK ARRAY DIMENSIONED AT LEAST 2*(N-1)
C
      IF (N.LE.1) GO TO 4
      IER=2
      IF (B(1).EQ.0.0) RETURN
      NM1=N-1
      BOLD=C(1)/B(1)
      GOLD=D(1)/B(1)
      WORK(1)=GOLD
```

Figure A.1 Continued

```
      WORK(1+NM1)=BOLD
      IF (N.EQ.2) GO TO 2
      DO 1 I=2,NM1
      DIV=B(I)-A(I)*BOLD
      IF (DIV.EQ.0.0) RETURN
      GOLD=(D(I)-A(I)*GOLD)/DIV
      WORK(I)=GOLD
      BOLD=C(I)/DIV
    1 WORK(I+NM1)=BOLD
    2 DIV=B(N)-A(N)*BOLD
      IF (DIV.EQ.0.0) RETURN
      GOLD=(D(N)-A(N)*GOLD)/DIV
      X(N)=GOLD
      L=N
      DO 3 I=2,N
      L=L-1
      GOLD=WORK(L)-WORK(L+NM1)*GOLD
    3 X(L)=GOLD
      IER=0
      RETURN
    4 IER=1
      IF (N.LE.0) RETURN
      IER=2
      IF (B(1).EQ.0.0) RETURN
      X(1)=D(1)/B(1)
      IER=0
      RETURN
      END
```

Figure A.1 Continued

```
      PROGRAM KAVL(TAPE7,TAPE8,OUTPUT)
C
C...   THIS PROGRAM USES HERMITE CUBIC SPLINE CUVRE
C...   FITTING TO EVALUATE KAV/L VALUES AT DESIRED FLOW
C...   RATES ALONG THE CURVE GENERATED BY PERFORMANCE
C...   KAV/L VALUES AND THEIR RESPECTIVE LIQUID FLOW RATES.
C
C...   INPUT TAPE 7 IS ODTAINED DIRECTLY FROM PROGRAM PDAP OUTPUT.
C
      PARAMETER (NPTS=20)
      LOGICAL HERMIT
      REAL KAVLS(NPTS),KAVLS8
      REAL PERC(NPTS),WORK(100)
      DIMENSION A(NPTS),B(NPTS),C(NPTS)
      DATA HERMIT/.TRUE./
C
C...   READ CALCULATED VALUES OF KAV/L AND THEIR RESPECTIVE
C...   LIQUID FLOW RATE PERCENTAGES
C
      DO 1 I=1,NPTS
        READ (7,100,END=99) KAVLS(I),PERC(I)
        NPT=I
    1 CONTINUE
C
C...   EVALUATE FLOW RATES POSTULATED IN THE DESIGN ACCIDENT FOR
C...   WHICH NO MANUFACTURER DATA IS AVAILABLE
C
   99 PRINT 600
      READ *,EVAL
      IF(EVAL.EQ.0) STOP
      IF(EVAL.EQ.1) THEN
        PRINT *,'ENTER PERCENTAGE'
        READ *,PERCIN
        CALL SPLFIT(PERC,KAVLS,NPT,A,IER)
        KAVLS8=SPLFCN(PERCIN,PERC,KAVLS,NPT,A,IER)
        PRINT *,'AT ',PERCIN,' KAV/L=',KAVLS8
        WRITE(8,102)PERCIN,KAVLS8
      ENDIF
      GO TO 99
C
C...   FORMAT STATEMENTS
C
  100 FORMAT (2F10.4)
  102 FORMAT (5X,'AT % ',F6.2,5X,'  KAV/L = ',F6.4,/)
  600 FORMAT ('IF KAV/L EVALUATION IS DESIRED ALONG THE CURVE',
     1        ' OF VALUES, ENTER 1;',/,
     2        'IF NOT,                                          ',
     3        '    ENTER 0')
      END
      SUBROUTINE SPLFIT (X,Y,N,HERMITE,A,B,C,INTER,IER)
      LOGICAL HERMITE
      DIMENSION X(N), Y(N), A(N), B(N), C(N)
C
C  SUBROUTINE SPLFIT CALCULATES THE COEFFICIENTS OF AN INTERPOLATING
C  NATURAL OR HERMITE CUBIC SPLINE.  SPLFCN CAN BE USED TO EVAULATE
C  THE INTERPOLATING POLYNOMIAL.
C
C  THE 1-D ARRAYS X AND Y CONTAIN N DATA PAIRS WHICH ARE TO
C  BE THE KNOTS OF THE INTERPOLATION.  THE INTERPOLATING FUNCTION
C  IS DEFINED BY
C
C             Y=Y(I)+((C(I)*T+B(I))*T+A(I))*T
C
C  WHERE      T=X-X(I)
C             X(I).LE.X.LT.X(I+1)
C  AND        A, B, AND C ARE ARRAYS OF LENGTH N CONTAINING THE
C             INTERPOLATING COEFFICIENTS.
```

Figure A.2 Listing of Program KAVL

```
C
C   HERMITE IS A LOGICAL VARIABLE (.TRUE. OR .FALSE.) WHICH
C   DETERMINES WHETHER A HERMITE OR NATURAL SPLINE IS USED
C   AS FOLLOWS.
C
C           HERMITE=.TRUE.   ==> HERMITE SPLINE USED IF N > 3
C           HERMITE=.FALSE.  ==> NATURAL SPLINE USED IF N > 3
C           N=2 OR 3         ==> POLYNOMIAL INTERPOLATION USED
C
C   INTER IS AN INTEGER VARIABLE DESIGNATING THE INTERVAL OF INTER-
C   POLATION.  SPLFIT INITALIZES INTER TO 1
C
C   IER IS AN INTEGER ERROR FLAG DEFINED AS FOLLOWS.
C
C           IER=0            ==> NO ERROR -- NORMAL RETURN
C           IER=1            ==> N .LE. 1
C           IER=2            ==> X ARRAY NOT IN STRICTLY ASCENDING
C                                OR DESCENDING ORDER
C
C.....INITIALIZE INTER
      INTER=1
C
C.....CHECK INPUT DATA
      IER=1
      IF (N.LE.1) RETURN
      IER=2
      IF (X(2).EQ.X(1)) RETURN
      IF (N.GT.2) GO TO 1
C.....FIT A STRAIGHT LINE TO DATA
      A(1)=(Y(2)-Y(1))/(X(2)-X(1))
      B(1)=0.0
      C(1)=0.0
      RETURN
    1 IF (X(2).LT.X(1)) GO TO 3
      DO 2 I=3,N
      IF (X(I).LE.X(I-1)) RETURN
    2 CONTINUE
      GO TO 5
    3 DO 4 I=3,N
      IF (X(I).GE.X(I-1)) RETURN
    4 CONTINUE
    5 IER=0
      IF (N.GT.3) GO TO 6
C.....FIT A PARABOLA TO DATA
      X2X1=X(2)-X(1)
      X3X1=X(3)-X(1)
      Y2Y1=Y(2)-Y(1)
      Y3Y1=Y(3)-Y(1)
      D=X2X1*X3X1
      D=D*X2X1-D*X3X1
      BB=(Y2Y1*X3X1-Y3Y1*X2X1)/D
      AA=(X2X1*X2X1*Y3Y1-X3X1*X3X1*Y2Y1)/D
      A(1)=AA
      B(1)=BB
      C(1)=0.0
      B(2)=BB
      A(2)=AA+2.*BB*X2X1
      C(2)=0.0
      RETURN
    6 IF (HERMITE) GO TO 11
C.....NATURAL SPLINE INTERPOLATION
      M=N-1
      CI=X(2)-X(1)
      C(1)=CI
      BI=(Y(2)-Y(1))/CI
      B(2)=BI
      DO 7 I=2,M
      CC=X(I+1)-X(I)
```

Figure A.2 Continued

```
      C(I)=CC
      BB=(Y(I+1)-Y(I))/CC
      B(I)=BB-BI
      BI=BB
      AA=CC+CI
      A(I)=AA+AA
    7 CI=CC
      DO 8 I=3,M
      AA=C(I-1)/A(I-1)
      A(I)=A(I)-AA*C(I-1)
    8 B(I)=B(I)-AA*B(I-1)
      B(1)=0.0
      B(N)=0.0
      BB=B(M)/A(M)
      B(M)=BB
      JJ=M
      DO 9 I=3,M
      JJ=JJ-1
      BB=(B(JJ)-C(JJ)*BB)/A(JJ)
    9 B(JJ)=BB
      BI=0.0
      JJ=N
      DO 10 I=1,M
      JJ=JJ-1
      BB=B(JJ)
      CC=C(JJ)
      A(JJ)=(Y(JJ+1)-Y(JJ))/CC-CC*(BI+BB+BB)
      C(JJ)=(BI-BB)/CC
      B(JJ)=BB+BB+BB
   10 BI=BB
      RETURN
C.....HERMITE SPLINE
   11 RM3=(Y(2)-Y(1))/(X(2)-X(1))
      T1=RM3-(Y(2)-Y(3))/(X(2)-X(3))
      RM2=RM3+T1
      RM1=RM2+T1
      NO=N-2
      DO 12 I=1,N
      RM4=RM3-RM2+RM3
      IF (I.LE.NO) RM4=(Y(I+2)-Y(I+1))/(X(I+2)-X(I+1))
      T1=ABS(RM4-RM3)
      T2=ABS(RM2-RM1)
      BB=T1+T2
      A(I)=0.5*(RM2+RM3)
      IF (BB.NE.0.0) A(I)=(T1*RM2+T2*RM3)/BB
      RM1=RM2
      RM2=RM3
   12 RM3=RM4
      NO=N-1
      DO 13 I=1,NO
      T1=1.0/(X(I+1)-X(I))
      T2=(Y(I+1)-Y(I))*T1
      BB=(A(I)+A(I+1)-T2-T2)*T1
      B(I)=-BB+(T2-A(I))*T1
   13 C(I)=BB*T1
      RETURN
      END
      REAL FUNCTION SPLFCN (XIN,X,Y,N,A,B,C,INTER,IER)
      DIMENSION X(N), Y(N), A(N), B(N), C(N)
C
C  FUNCTION SPLFCN EVALUATES A SPLINE INTERPOLATION AT XIN
C
C  ARGUMENTS X,Y,N,A,B AND C ARE DEFINED AS IN SPLFIT.
C
C  INTER IS AN INTEGER VARIABLE INDICATING THE INTERVAL USED
C  IN THE INTERPOLATION (EXTRAPOLATION)
C
```

Figure A.2 Continued

```
C  IER IS AN INTEGER ERROR FLAG DEFINED AS FOLLOWS.
C
C        IER=0      ==> NO ERROR -- NORMAL RETURN
C        IER=1      ==> XIN .LT. XMIN -- EXTRAPOLATION
C        IER=2      ==> XIN .GT. XMAX -- EXTRAPOLATION
C
C  WHERE XMAX AND XMIN ARE THE MAXIMUM AND MINIMUM VALUES
C  OF THE X-ARRAY, RESPECTIVELY.
C
      IF (INTER.LE.0.OR.INTER.GT.N) INTER=1
      IER=0
      T=XIN-X(INTER)
      U=XIN-X(INTER+1)
      IF (U*T.GT.0.0) GO TO 1
      SPLFCN=Y(INTER)+((C(INTER)*T+B(INTER))*T+A(INTER))*T
      RETURN
    1 K=1
      IF (T.LT.0.0) K=-K
      IF ((T-U).LT.0.0) K=-K
    2 INTER=INTER+K
      IF ((N-INTER)*INTER.LE.0) GO TO 3
      T=XIN-X(INTER)
      U=XIN-X(INTER+1)
      IF (U*T.GT.0.0) GO TO 2
      SPLFCN=Y(INTER)+((C(INTER)*T+B(INTER))*T+A(INTER))*T
      RETURN
    3 IER=1
      IF (T.LT.0.0) IER=2
      INTER=INTER-K
      SPLFCN=Y(INTER)+((C(INTER)*T+B(INTER))*T+A(INTER))*T
      RETURN
      END
```

Figure A.2 Continued

```
      PROGRAM TMCON(TAPE7,TAPE6,TAPE9,OUTPUT)
C
C...  PROGRAM TMCON DETERMINES THE FIRST ORDER SYSTEM TIME CONSTANT OF
C     AN UHS COOLING TOWER BASIN SYSTEM FROM THE TIME-DEPENDENT BASIN
C     TEMPERATURE GIVEN THE TIME HISTORY OF HEAT INPUT AND WET-BULB
C     TEMPERATURE
C
C...  SET PARAMETERS -- SEE ALSO FUNCTION ST2
C
C     MSTEP   =  MAXIMUM NUMBER OF TIME STEPS (USED IN DIMENSIONING)
C     MTOWER  =  MAXIMUM NUMBER OF TOWERS ALLOWED (USED IN DIMENSIONING)
C
      PARAMETER (MSTEP=101,MTOWER=5)
C
C...  DECLARE VARIABLES
C
      EXTERNAL ST2
      LOGICAL NOEVAP,CONWB
      COMMON NTOWER,TSTEP,NSTEP,TIME(MSTEP),T(MSTEP),BM(MSTEP),
     1       TWB(MSTEP),Q(MSTEP,MTOWER),FLOW(MSTEP,MTOWER),
     2       HL(MSTEP),HM(MSTEP),TFIT(MSTEP)
      DATA TOL/1.0E-5/,ALOW/0.05/,AHGH/0.5/
C
C...  READ INPUT DATA
C
      READ (7,101) NTOWER,BASMAS,S,TSTEP,NSTP,NOEVAP,CONWB
      NSTEP=NSTP+1
      DO 2 I=1,NSTEP
      READ (7,102) TIME(I),T(I),BM(I),SS,TWB(I),(Q(I,J),FLOW(I,J),
     1             J=1,NTOWER)
      SUM=0.0
      SUML=0.0
      DO 1 J=1,NTOWER
      SUM=SUM+Q(I,J)
      SUML=SUML+Q(I,J)/FLOW(I,J)
    1 CONTINUE
C
C...  THE NEXT LINE IS TSTEP*QTOTAL/((BASIN MASS)*(SPECIFIC HEAT)),
C     BUT IN ENGLISH UNITS THE SPECIFIC HEAT OF WATER IS APPROXIMATELY
C     1 BTU/LBM-F
C
      HM(I)=TSTEP*SUM/BM(I)
      HL(I)=SUML/FLOAT(NTOWER)
    2 CONTINUE
C
C...  DETERMINE THE VALUE OF 'TAU' WHICH MINIMIZES THE SUM OF THE
C     SQUARES OF THE DIFFERENCES BETWEEN THE FIRST ORDER APPROXIMATION
C     AND THE PREDICTIONS OF UHSSIM
C
      A=FMIN(ALOW,AHGH,ST2,TOL)
      TAU=TSTEP/A
C
C...  WRITE OUT THE RESULTS
C
      WRITE (6,103) NTOWER,BASMAS,S,TSTEP,NSTP,NOEVAP,CONWB,TAU
      WRITE (6,104)
      WRITE (6,105) (TIME(I),T(I),TFIT(I),TWB(I),I=1,NSTEP)
      WRITE (9,106) NSTEP,TAU
      WRITE (9,107) (TIME(I),T(I),TFIT(I),TWB(I),I=1,NSTEP)
      STOP
C
  101 FORMAT (I10,3E10.3,I10,2L10)
  102 FORMAT (8E10.3)
  103 FORMAT ('1 TIME CONSTANT ANALYSIS PROGRAM',//,'  INPUT DATA',//,
     1        '  NUMBER OF TOWERS                    ',I10,/,
     2        '  BASIN MASS (LBM)                    ',1PE10.3,/,
     3        '  SALINITY OF BASIN (PPM)             ',3PF10.2,/,
     4        '  TIME STEP SIZE (HR)                 ',0PF10.2,/,
```

Figure A.3 Listing of Program TMCON

```
     5         '  NUMBER OF STEPS TO BE TAKEN       ',I10,/,
     6         '  NO EVAPORATION                    ',L10,/,
     7         '  CONSTANT WET-BULB TEMPERATURE     ',L10,//,
     8         '  TIME CONSTANT (HR)                ',F10.2,//)
  104 FORMAT (6X,'TIME',4X,'TBASIN',6X,'TFIT',7X,'TWB',/,
     1        3X,'(HOURS)',7X,'(F)',7X,'(F)',7X,'(F)',/)
  105 FORMAT (4F10.1)
  106 FORMAT (I10,1PE10.3)
  107 FORMAT (1P4E10.3)
      END
      FUNCTION ST2(A)
C
C...  SET PARAMETERS -- SEE ALSO PROGRAM TMCON
C
C     MSTEP   =  MAXIMUM NUMBER OF TIME STEPS (USED IN DIMENSIONING)
C     MTOWER  =  MAXIMUM NUMBER OF TOWERS ALLOWED (USED IN DIMENSIONING)
C
      PARAMETER (MSTEP=101,MTOWER=5)

 ...  DECLARE VARIABLES
C
      COMMON NTOWER,TSTEP,NSTEP,TIME(MSTEP),T(MSTEP),BM(MSTEP),
     1       TWB(MSTEP),Q(MSTEP,MTOWER),FLOW(MSTEP,MTOWER),
     2       HL(MSTEP),HM(MSTEP),TFIT(MSTEP)
C
C...  DETERMINE THE FIRST ORDER RESPONSE FOR 'A' WHERE A = TSTEP/TAU
C
      TFIT(1)=T(1)
      DO 1 I=2,NSTEP
      TFIT(I)=(TFIT(I-1)+A*(TWB(I)-HL(I))+HM(I))/(1.0+A)
    1 CONTINUE
C
C...  CALCULATE SUM OF SQUARE OF DIFFERENCES
C
      ST2=0.0
      DO 2 I=2,NSTEP
      ST2=ST2+(T(I)-TFIT(I))**2
    2 CONTINUE
      RETURN
      END
      REAL FUNCTION FMIN (AX,BX,F,TOL)
      REAL AX,BX,F,TOL
      EXTERNAL F
C
C       AN APPROXIMATION  X  TO THE POINT WHERE  F  ATTAINS A MINIMUM  ON
C   THE INTERVAL  (AX,BX)  IS DETERMINED.
C
C
C   INPUT..
C
C   AX     LEFT ENDPOINT OF INITIAL INTERVAL
C   BX     RIGHT ENDPOINT OF INITIAL INTERVAL
C   F      FUNCTION SUBPROGRAM WHICH EVALUATES  F(X)  FOR ANY  X
C          IN THE INTERVAL  (AX,BX)
C   TOL    DESIRED LENGTH OF THE INTERVAL OF UNCERTAINTY OF THE FINAL
C          RESULT ( .GE. 0.0D0)
C
C
C   OUTPUT..
C
C   FMIN  ABCISSA APPROXIMATING THE POINT WHERE  F  ATTAINS A MINIMUM
C
C
C       THE METHOD USED IS A COMBINATIO   GOLDEN  SECTION  SEARCH  AND
C   SUCCESSIVE PARABOLIC INTERPOLATION.  ONVERGENCE IS NEVER MUCH SLOWER
C   THAN  THAT  FOR  A  FIBONACCI SEARCH.  IF  F  HAS A CONTINUOUS SECOND
C   DERIVATIVE WHICH IS POSITIVE AT THE MINIMUM (WHICH IS NOT  AT  AX  OR
C   BX),  THEN  CONVERGENCE  IS  SUPERLINEAR, AND USUALLY OF THE ORDER OF
```

Figure A.3 Continued

```
C  ABOUT  1.324....
C      THE FUNCTION  F  IS NEVER EVALUATED AT TWO POINTS CLOSER TOGETHER
C  THAN  EPS*ABS(FMIN) + (TOL/3), WHERE EPS IS  APPROXIMATELY THE SQUARE
C  ROOT  OF  THE  RELATIVE  MACHINE  PRECISION.  IF  F  IS A UNIMODAL
C  FUNCTION AND THE COMPUTED VALUES OF   F  ARE  ALWAYS  UNIMODAL  WHEN
C  SEPARATED BY AT LEAST  EPS*ABS(X) + (TOL/3), THEN  FMIN  APPROXIMATES
C  THE ABCISSA OF THE GLOBAL MINIMUM OF  F  ON THE INTERVAL  AX,BX  WITH
C  AN ERROR LESS THAN  3*EPS*ABS(FMIN) + TOL.  IF   F   IS NOT UNIMODAL,
C  THEN FMIN MAY APPROXIMATE A LOCAL, BUT PERHAPS NON-GLOBAL, MINIMUM TO
C  THE SAME ACCURACY.
C      THIS FUNCTION SUBPROGRAM IS A SLIGHTLY MODIFIED  VERSION  OF  THE
C  ALGOL  60 PROCEDURE  LOCALMIN  GIVEN IN RICHARD BRENT, ALGORITHMS FOR
C  MINIMIZATION WITHOUT DERIVATIVES, PRENTICE - HALL, INC. (1973).
C
C
      REAL A,B,C,D,E,EPS,XM,P,Q,R,TOL1,TOL2,U,V,W
      REAL FU,FV,FW,FX,X
C
C  C IS THE SQUARED INVERSE OF THE GOLDEN RATIO
C
      C=0.5*(3.-SQRT(5.0))
C
C  EPS IS APPROXIMATELY THE SQUARE ROOT OF THE RELATIVE MACHINE
C  PRECISION.
C
      EPS=1.0
   10 EPS=EPS/2.0
      TOL1=1.0+EPS
      IF (TOL1.GT.1.0) GO TO 10
      EPS=SQRT(EPS)
C
C  INITIALIZATION
C
      A=AX
      B=BX
      V=A+C*(B-A)
      W=V
      X=V
      E=0.0
      FX=F(X)
      FV=FX
      FW=FX
C
C  MAIN LOOP STARTS HERE
C
   20 XM=0.5*(A+B)
      TOL1=EPS*ABS(X)+TOL/3.0
      TOL2=2.0*TOL1
C
C  CHECK STOPPING CRITERION
C
      IF (ABS(X-XM).LE.(TOL2-0.5*(B-A))) GO TO 90
C
C IS GOLDEN-SECTION NECESSARY
C
      IF (ABS(E).LE.TOL1) GO TO 40
C
C  FIT PARABOLA
C
      R=(X-W)*(FX-FV)
      Q=(X-V)*(FX-FW)
      P=(X-V)*Q-(X-W)*R
      Q=2.0*(Q-R)
      IF (Q.GT.0.0) P=-P
      Q=ABS(Q)
      R=E
      E=D
C
```

Figure A.3 Continued

```
C  IS PARABOLA ACCEPTABLE
C
      IF (ABS(P).GE.ABS(0.5*Q*R)) GO TO 40
      IF (P.LE.Q*(A-X)) GO TO 40
      IF (P.GE.Q*(B-X)) GO TO 40
C
C  A PARABOLIC INTERPOLATION STEP
C
      D=P/Q
      U=X+D
C
C  F MUST NOT BE EVALUATED TOO CLOSE TO AX OR BX
C
      IF ((U-A).LT.TOL2) D=SIGN(TOL1,XM-X)
      IF ((B-U).LT.TOL2) D=SIGN(TOL1,XM-X)
      GO TO 50
C
C  A GOLDEN-SECTION STEP
C
   40 IF (X.GE.XM) E=A-X
      IF (X.LT.XM) E=B-X
      D=C*E
C
C  F MUST NOT BE EVALUATED TOO CLOSE TO X
C
   50 IF (ABS(D).GE.TOL1) U=X+D
      IF (ABS(D).LT.TOL1) U=X+SIGN(TOL1,D)
      FU=F(U)
C
C  UPDATE  A, B, V, W, AND X
C
      IF (FU.GT.FX) GO TO 60
      IF (U.GE.X) A=X
      IF (U.LT.X) B=X
      V=W
      FV=FW
      W=X
      FW=FX
      X=U
      FX=FU
      GO TO 20
   60 IF (U.LT.X) A=U
      IF (U.GE.X) B=U
      IF (FU.LE.FW) GO TO 70
      IF (W.EQ.X) GO TO 70
      IF (FU.LE.FV) GO TO 80
      IF (V.EQ.X) GO TO 80
      IF (V.EQ.W) GO TO 80
      GO TO 20
   70 V=W
      FV=FW
      W=U
      FW=FU
      GO TO 20
   80 V=U
      FV=FU
      GO TO 20
C
C  END OF MAIN LOOP
C
   90 FMIN=X
      RETURN
      END
```

Figure A.3 Continued

```
      PROGRAM UHSSIM(TAPE5,TAPE8,TAPE6,TAPE7,OUTPUT)
C
C...  PROGRAM UHSSIM (ULTIMATE HEAT SINK SIMULATION) SIMULATES
C...  THE PERFORMANCE OF AN ULTIMATE HEAT SINK TOWER/BASIN
C...  SYSTEM. THE BASIN TEMPERATURE, BASIN MASS, AND BASIN
C...  SALINITY ARE CALCULATED WITH RESPECT TO TIME FOLLOWING
C...  A LOCA (LOSS OF COOLANT ACCIDENT). WORST-CASE METEOROLOGICAL
C...  CONDITIONS OF WET-BULB TEMPERATURE ALONG WITH DESIGN-BASIS-
C...  ACCIDENT FLOW RATE VALUES ARE USED IN THE SIMULATION.
C
C...  CODE NWS SUPPLIES THE WORST-CASE WET-BULB  TEMPERATURES.
C
C
C...  DECLARE EXTERNALS
C
C       BASIN  - A SUBROUTINE USED TO COMPUTE DERIVATIVES OF
C                BASIN TEMPERATURE AND MASS (NAME PASSED TO
C                SUBROUTINE ODE AS AN ARGUMENT)
C
      EXTERNAL BASIN
C
C...  DECLARE MISCELLANEOUS VARIABLE TYPES
C
C       ITOWER  - MAXIMUM NUMBER OF LIQUID FLOW RATE, KAV/L,
C                 AND L/G VALUES FOR EACH TOWER
C       JTOWER  - MAXIMUM NUMBER OF TOWERS
C       IHEAT   - MAXIMUM NUMBER OF HEAT LOAD VALUES FOR EACH
C                 TOWER
C       IWNDL   - MAXIMUM LENGTH OF WET-BULB TEMPERATURE WINDOW
C
      PARAMETER (ITOWER=50,JTOWER=5,IHEAT=50,IWNDL=200)
      LOGICAL HERMIT,NOEVAP,CONWB
      CHARACTER*40 HEADER
      CHARACTER*8 YMDHH,YMDH(IWNDL)
C
C...  DECLARE REAL ARRAYS
C
C       LOVERG       - LIQUID OVER GAS FLOW RATE RATIO
C       L            - LIQUID FLOW RATE
C       KAVL         - TOWER CHARACTERISTIC
C       LOVRGS       - ARRAY CONTAINING TIME-VARYING L/G VALUES
C       KAVLS        - ARRAY CONTAINING TIME-VARYING KAV/L VALUES
C       LS           - ARRAY CONTAINING TIME-VARYING L VALUES
C       LOLD         - DUMMY VARIABLE USED FOR L VALUE COMPARISONS
C
      REAL LOVERG,L,KAVL,LOLD(JTOWER)
      REAL LOVRGS(ITOWER,JTOWER),KAVLS(ITOWER,JTOWER),LS(ITOWER,JTOWER)
C
C...  DECLARE ARRAYS
C
C       Y     - ARRAY STORING INTEGRATION VARIABLES (T AND M)
C       YP    - ARRAY STORING DERIVATIVES
C       WORK  - WORK ARRAY FOR SUBROUTINE ODE
C
      DIMENSION Y(2),YP(2),WORK(426)
C
C...  COMMON BLOCKS
C
      COMMON /TOWERC/ LOVERG(JTOWER),L(JTOWER),KAVL(JTOWER),NH(JTOWER),
     1                NL(JTOWER),NTOWER,SALT
      COMMON /TOWERC2/ TE(ITOWER,JTOWER),ALG(ITOWER,JTOWER),
     1                AL(ITOWER,JTOWER),AK(ITOWER,JTOWER)
      COMMON /HEATC/ Q(IHEAT,JTOWER),TME(IHEAT,JTOWER),AH(IHEAT,JTOWER)
      COMMON /AIRC/ WB(IWNDL),TM(IWNDL),AWB(IWNDL),BWB(IWNDL),
     1              CWB(IWNDL),NAMB,RWB(IWNDL)
      COMMON /OUT/ QQ(JTOWER),TWB,S
      COMMON /EVAPC/ NOEVAP
C
```

Figure A.5 Listing of Program UHSSIM

```
C...  SET MISCELLANEOUS DATA
C
C       NEQU    - NUMBER OF EQUATIONS TO BE INTEGRATED
C       RELERR  - RELATIVE ERROR CRITERION FOR INTEGRATION ROUTINE
C       ASBERR  - ABSOLUTE ERROR CRITERION FOR INTEGRATION ROUTINE
C       HERMIT  - LOGICAL VARIABLE USED TO SPECIFY HERMITE SPLINE
C                 INTERPOLATION
C       DINTER  - DATA INTERVAL
C       START   - INITIAL STARTING LOCATION
C
      DATA NEQN/2/,RELERR/1.E-4/,ABSERR/1.E-4/,HERMIT/.TRUE./
      DATA DINTER/1.0/,START/0.0/
C
C...  READ INITIAL TOWER INFORMATION
C
C       TSTEP   - TIME STEP
C       NSTEP   - NUMBER OF TIME STEPS TO BE TAKEN
C       BASMAS  - INITIAL BASIN MASS
C       S       - SALINITY
C       NTOWER  - NUMBER OF TOWERS
C       NOEVAP  - LOGICAL VARIABLE SET .TRUE. FOR NO EVAPORATION
C                 .FALSE. OTHERWISE
C       CONWB   - LOGICAL VARIABLE SET .TRUE. FOR CONSTANT WET-BULB
C                 TEMPERATURE .FALSE. OTHERWISE
C       BASTMP  - INITIAL BASIN TEMPERATURE
C
      READ (5,*) NTOWER,BASMAS,S
      READ (5,*) TSTEP,NSTEP,NOEVAP,CONWB
C
C...  WRITE INITIAL VALUES TO TAPE 6 AND 7
C
      WRITE (6,203) NTOWER,BASMAS,S,TSTEP,NSTEP,NOEVAP,CONWB
      WRITE (7,103) NTOWER,BASMAS,S,TSTEP,NSTEP,NOEVAP,CONWB
      WRITE (6,202)
C
C...  READ TIME-VARYING LIQUID FLOW RATE, L, KAV/L, AND L/G VALUES
C
C       NL     - NUMBER OF FLOW RATE VALUES PER TOWER
C
      READ (5,*) (NL(I),I=1,NTOWER)
      DO 1 I=1,NTOWER
      NP=NL(I)
      IF(NP.GT.0) READ (5,*)(TE(J,I),LOVRGS(J,I),KAVLS(J,I),LS(J,I),
     1                       J=1,NP)
    1 CONTINUE
C
C...  READ TIME-VARYING HEAT LOAD VALUES FOR EACH TOWER
C
C       TME   - TIME OF CORESPONDING HEAT LOAD
C       Q     - HEAT LOAD VALUES
C       NH    - NUMBER OF HEAT LOAD VALUES PER TOWER
C
      READ (5,*) (NH(I),I=1,NTOWER)
      DO 2 I=1,NTOWER
      NN=NH(I)
      IF (NN.GT.0) READ (5,*) (TME(J,I),Q(J,I),J=1,NN)
    2 CONTINUE
C
C...  IF WET-BULB TEMPERATURE IS CONSTANT, READ WET-BULB
C...  TEMPERATURE AND INITIAL BASIN TEMPERATURE
C
      IF (CONWB) THEN
        NAMB=1
        READ (5,*) WB(1),BASTMP
        GO TO 21
      ELSE
C
C...  READ WINDOW TO BE READ FROM THE WORST-CASE METEOROLOGICAL
```

Figure A.5 Continued

```
C...  WINDOWS DETERMINED BY PROGRAM NWS AND THE NSTART POSITION;
C...  READ HEADER INFORMATION ABOUT WINDOWS LOCATED AT THE BEGINNING
C...  OF THE TAPE. ADVANCE TO WINDOW SELECTED USING DUMMING
C...  VARIABLES YMDHH, RWBB, WBB. VARY THE ACCIDENT STARTING
C...  HOUR USING DUMMY VARIABLES YMDHH, RWBB, WBB.
C
C       NAMB     - NUMBER OF AMBIENT WET-BULB TEMPERATURES
C       TAPE 8   - TAPE OF WINDOWS DETERMINED FROM CODE NWS
C       NWINDW   - INTEGER NUMBER COORESPONDING TO METEOROLOGICAL
C                  WINDOW TO BE READ
C       NSTART   - POSITION AT WHICH TO START READING TWB;
C                  SET = 0 IMPLIES CONSTANT TWB
C
        READ (5,*) WBB,BAST
        READ (5,*) NWINDW,NSTART
        READ (8,105) HEADER
        READ (8,*) TAU,DELTAT,NRECS,LENGTH,NSAVE
        READ (8,*) MSAVE,NPASS,MREAD
C
        DO 20 I=NSAVE,2,-1
          IF(NWINDW.NE.I) GO TO 20
          N=(FLOAT(I)-1)*LENGTH+(FLOAT(I)-1)
          READ (8,100)(YMDHH,RWBB,WBB,J=1,N)
          GO TO 3
   20   CONTINUE
C
    3   READ (8,*) TMX,I1
        READ (8,100,END=14) (YMDHH,RWBB,WBB,I=2,NSTART)
        NAMB=FLOAT(NSTEP)*TSTEP/DINTER+1.0
        BASTMP=RWBB
        READ (8,100,END=14) (YMDH(I),RWB(I),WB(I),I=1,NAMB)
      ENDIF
C
C...   INITIALIZE VARIABLES FOR INTEGRATION
C
C        TIME  - INTEGRATION TIME (HR)
C        IFLAG - CONTROL FLAG
C        TOUT  - CONTINUOUS TIME (HR)
C        Y(1)  - INITIAL BASIN TEMPERATURE (F)
C        Y(2)  - INITIAL BASIN MASS (LBM)
C        SALT  - INITIAL MASS OF SALT IN BASIN
C
   21 KOUNT=-1
      IFLAG=1
      TIME=START
      TOUT=TIME
      Y(1)=BASTMP
      Y(2)=BASMAS
      SALT=Y(2)*S
      XXX=START
C
C...   IF THE WET-BULB TEMPERATURE IS NOT CONSTANT,
C...   FIT HERMITE CUBIC SPLINE TO TIME VS. TEMPERATURE
C
C...   CHECK FOR 'BAD' WB VALUES WITHIN WINDOW
C
       IF (NAMB.GT.1) THEN
         YYY=999.0
         DO 4 I=1,NAMB
         IF(WB(I).GT.998.0) GO TO 4
         YYY=WB(I)
         GO TO 5
    4    CONTINUE
         WRITE (6,*) 'NO VALUE WB FOUND IN REQUESTED WINDOW'
         STOP
    5    DO 6 I=1,NAMB
         IF(WB(I).GT.998.0) WB(I)=YYY
         YYY=WB(I)
```

Figure A.5 Continued

```
        TM(I)=XXX
    6   XXX=XXX+DINTER
        CALL SPLFIT (TM,WB,NAMB,HERMIT,AWB,BWB,CWB,INTER,IER)
        IF (IER.NE.0) GO TO 15
      ENDIF
C
C...  IF THE HEAT LOAD IS TIME DEPENDENT, FIT
C...  DATA OF TIME VS. HEAT LOAD BY LINEAR INTERPOLATION
C
      DO 7 I=1,NTOWER
      IF (NH(I).LE.1) GO TO 7
      CALL LFIT (TME(1,I),Q(1,I),NH(I),AH(1,I),IER)
      IF (IER.NE.0) GO TO 16
    7 CONTINUE
C
C...  IF THE LIQUID FLOW RATE IS TIME DEPENDENT, FIT DATA OF
C...  TIME VS. L, TIME VS. L/G, AND TIME VS. KAV/L
C...  USING LINEAR INTERPOLATION
C
      DO 8 I=1,NTOWER
       IF(NL(I).LE.1) GO TO 8
       CALL LFIT (TE(1,I),LOVRGS(1,I),NL(I),ALG(1,I),IER)
        IF(IER.NE.0) GO TO 17
       CALL LFIT (TE(1,I),KAVLS(1,I),NL(I),AK(1,I),IER)
        IF(IER.NE.0) GO TO 18
       CALL LFIT (TE(1,I),LS(1,I),NL(I),AL(1,I),IER)
        IF(IER.NE.0) GO TO 19
    8 CONTINUE
C
C...  BEGIN ACCIDENT ANALYSIS
C
C...  SET L/G, KAV/L, L AT EACH TIME STEP
C
   12 DO 9 I=1,NTOWER
       IF(NL(I).GT.1) THEN
         LOVERG(I)=SLFCN (TIME,TE(1,I),LOVRGS(1,I),NL(I),ALG(1,I),IER)
         KAVL(I)=SLFCN (TIME,TE(1,I),KAVLS(1,I),NL(I),AK(1,I),IER)
         L(I)=SLFCN (TIME,TE(1,I),LS(1,I),NL(I),AL(1,I),IER)
       ELSEIF(NL(I).EQ.1) THEN
         LOVERG(I)=LOVRGS(1,I)
         KAVL(I)=KAVLS(1,I)
         L(I)=LS(1,I)
       ENDIF
    9 CONTINUE
C
C...  IF LIQUID FLOW RATE CHANGES, REINITIALIZE ODE
C
      IF (TIME.GT.1) THEN
        DO 10 I=1,NTOWER
        IF (L(I).NE.LOLD(I)) IFLAG=1
   10   CONTINUE
      ENDIF
C
      KOUNT=KOUNT+1
      IF (NAMB.EQ.1) THEN
        TWB=WB(1)
      ENDIF
      CALL BASIN (TIME,Y,YP)
C
C...  WRITE RESULTS TO TAPE 6
C
      WRITE (6,102) TIME,Y,S,TWB,(QQ(I),L(I),I=1,NTOWER)
C
C...  WRITE RESULTS FOR PLOTTING TO TAPE 7
C
      WRITE (7,104) TIME,Y,S,TWB,(QQ(I),L(I),I=1,NTOWER)
      IF (KOUNT.GE.NSTEP) STOP
C
```

Figure A.5 Continued

```
C...  BEGIN INTEGRATION LOOP
C
      TOUT=TOUT+TSTEP
   13 CALL ODE (BASIN,NEQN,Y,TIME,TOUT,RELERR,ABSERR,IFLAG,WORK)
      IF (IFLAG.EQ.2) THEN
        DO 11 I=1,NTOWER
   11   LOLD(I)=L(I)
        GO TO 12
      ENDIF
      IF (IFLAG.EQ.3) GO TO 13
      WRITE (6,101) IFLAG,Y,RELERR,ABSERR
      STOP
C
C...  FORMAT STATEMENTS
C
  100 FORMAT (5X,A,2F10.2)
  101 FORMAT (/,' ***ERROR***  ERROR NUMBER ',I5,
     1 ' IN BASIN INTEGRATION ',/,14X,'DTDT=',1PE15.3,' F/HR',/,
     2 14X,'DMDT=',1PE15.3,' LBM/HR',/,14X,'RELERR=',1PE14.3,/,
     3 14X,'ABSERR=',1PE14.3,//)
  102 FORMAT (F10.1,F10.2,1PE10.2,3PF10.1,0PF10.1,1P2E10.2,:,/,(50X,
     1 1P2E10.2))
  103 FORMAT (I10,1PE10.3,3PF10.2,0PF10.2,I10,2L10)
  104 FORMAT (1P8E10.3)
  105 FORMAT (A)
  201 FORMAT (1P2E10.3)
  202 FORMAT (6X,'TIME',4X,'TBASIN',3X,'BMASS',10X,'S',7X,'TWB',
     1         6X,'Q',9X,'L',/,6X,'(HR)',6X,'(F)',4X,'(LBM)',
     2         8X,'(PPT)',5X,'(F)',3X,'(BTU/HR)',2X,'(LBM/HR)',/,
     3         6X,'----',4X,'------',3X,'-----',8X,'-----',5X,'---',
     4         3X,'--------',2X,'--------')
  203 FORMAT (//,'NUMBER OF TOWERS                    ',I10,/,
     1            'BASIN MASS (LBM)                    ',1PE10.3,/,
     2            'SALINITY OF BASIN (PPM)             ',3PF10.2,/,
     3            'TIME STEP SIZE (HR)                 ',0PF10.2,/,
     4            'NUMBER OF STEPS TO BE TAKEN         ',I10,/,
     5            'NO EVAPORATION                      ',L10,/,
     6            'CONSTANT WET-BULB TEMPERATURE       ',L10,//)
   14 STOP ' INSUFFICIENT AMBIENT DATA'
   15 STOP ' ERROR IN WB'
   16 STOP ' ERROR IN HEAT'
   17 STOP ' ERROR IN L/G'
   18 STOP ' ERROR IN KAV/L'
   19 STOP ' ERROR IN L'
      END
      SUBROUTINE BASIN (TIME,Y,YP)
C
C...  THIS SUBROUTINE IS USED TO EVALUATE DERIVATIVES OF THE
C...  FORM YP(I)=DY(I)/DT. BASIN HAS TWO EQUATIONS; BASIN
C...  TEMPERATURE AND BASIN MASS WRT TIME.
C
      PARAMETER (ITOWER=50,JTOWER=5,IHEAT=50,IWNDL=200)
      REAL LOVERG,L,KAVL
      DIMENSION Y(2),YP(2)
      COMMON /TOWERC/ LOVERG(JTOWER),L(JTOWER),KAVL(JTOWER),NH(JTOWER),
     1                NL(JTOWER),NTOWER,SALT
      COMMON /TOWERC2/ TE(ITOWER,JTOWER),ALG(ITOWER,JTOWER),
     1                 AL(ITOWER,JTOWER),AK(ITOWER,JTOWER)
      COMMON /HEATC/ Q(IHEAT,JTOWER),TME(IHEAT,JTOWER),AH(IHEAT,JTOWER)
      COMMON /AIRC/ WB(IWNDL),TM(IWNDL),AWB(IWNDL),BWB(IWNDL),
     1              CWB(IWNDL),NAMB,RWB(IWNDL)
      COMMON /OUT/ QQ(JTOWER),TWB,S
C
C...  SET WET-BULB TEMPERATURE AT EACH TIME STEP
C
      IF (NAMB.GT.1) THEN
        TWB=SPLFCN (TIME,TM,WB,NAMB,AWB,BWB,CWB,INTER,IER)
      ENDIF
```

Figure A.5 Continued

```
C
C...  CALCULATE SALINITY
C
      S=SALT/Y(2)
C
C...  BEGIN LOOP OVER TOWERS
C
      QQQ=0.0
      DTDT=0.0
      DMDT=0.0
C
C...  SET HEAT LOAD AT EACH TIME STEP
C
      DO 1 I=1,NTOWER
      IF(NH(I).GT.1) THEN
        QQQ=SLFCN (TIME,TME(1,I),Q(1,I),NH(I),AH(1,I),IER)
      ELSEIF (NH(I).EQ.1) THEN
        QQQ=Q(1,I)
      ENDIF
      QQ(I)=QQQ
      TIN=Y(1)+QQQ/L(I)
C
C...  CALCULATE OUTLET TEMPERATURE OF TOWER (TOUT) AND THE
C...  FRACTION OF WATER EVAPORATED, GIVEN
C       TIN   - TOWER INLET TEMPERATURE, AND
C       S, TWB, KAV/L, AND L/G .
C
      CALL TOWER (TIN,S,TWB,KAVL(I),LOVERG(I),TOUT,F)
      DTDT=DTDT+L(I)*(1.0-F)*(TOUT-Y(1))/(Y(2)-F*L(I))
      DMDT=DMDT-F*L(I)
    1 CONTINUE
      YP(1)=DTDT
      YP(2)=DMDT
      RETURN
      END
      SUBROUTINE TOWER (TIN,S,TWB,KAVL,LOVERG,TOUT,FEVAP)
C
C...  THIS SUBROUTINE IS USED TO CALCULATE TOUT-THE OUTLET
C...  TEMPERATURE OR COLD WATER TEMPERATURE-OF THE TOWER,
C...  AND THE FRACTION OF WATER EVAPORATED WITHIN EACH TOWER.
C
C
      SAVE TOLD,A,B,C,D,T,HHAT
      REAL LOVERG,KAVL
      LOGICAL NOEVAP
      COMMON /EVAPC/ NOEVAP
      DIMENSION A(10),B(10),C(10),D(10),T(12),HHAT(12),WORK(20)
C
C...  SET MISCELLANEOUS DATA
C
      DATA A/0.,9*1./,B/10*-2./,C/10*1./,D/10*0./
      DATA N/10/,NP1/11/,NP2/12/,AN/0.05/
      DATA A0/1.0/,A1/-0.66177/,A2/0.36508/,A3/-5.25926/
      DATA C0/13.3185/,C1/1.976/,C2/0.6445/,C3/0.1299/
      DATA TT1/212./,TT2/459.67/,WRATIO/0.622/
      DATA HV1/0.427555/,HV2/1062.3/
      DATA HA1/0.2403/
      DATA HFG/1048.0/,LHF/1.0/
      DATA SMALL/0.01/,TOLD/0.0/
C
      E(X)=(((C3*X-C2)*X+C1)*X+C0)*X
      F(X)=A0+X*(A1+X*(A2+X*A3))
      T(NP1)=TIN
      T(NP2)=TWB
C
C...  AN=1/2*1/N WHERE 1/N=DELTA A*
C
      AK1=AN*KAVL
```

Figure A.5 Continued

```
C
C...  SET-UP COEFFICIENTS FOR THE TRIDIAGONAL MATRIX SOLUTION
C
    1 D(1)=(B(1)+C(1))*TWB
      D(N)=-C(N)*TIN
      CALL TDMS (A,B,C,T,D,N,IER,WORK)
      IF (ABS(TOLD-T(1)).LT.SMALL) GO TO 4
      TOLD=T(1)
C
C...  CALCULATE THE TOTAL ENTHALPY, HHAT, FOR EACH CELL BY
C...  CALCULATING THE RELATIVE HUMIDITY, W, AT EACH TEMPERATURE,
C...  MULTIPLYING THE ENTHALPY DUE TO SATURATED VAPOR BY W,
C...  AND ADDING THIS TO THE ENTHAPLY DUE TO SATURATED AIR
C...  ALL FOUND AT THE CELL TEMPERATURE.
C
      DO 2 I=1,NP2
      TT=(TT1-T(I))/(T(I)+TT2)
      PP=EXP(-E(TT))*AMAX1(0.775,F(S))
      W=WRATIO*PP/(1.0-PP)
      HV=HV1*T(I)+HV2
      HA=HA1*T(I)
    2 HHAT(I)=HA+W*HV
C
C...  CALCULATE DH/DT FOR EACH CALL TO CALCULATE MATRIX COEFFICIENTS
C
      DO 3 I=2,N
      DH=(HHAT(I+1)-HHAT(I-1))/(T(I+1)-T(I-1))
      Q=AK1*(DH-LOVERG)
      A(I)=1.0+Q
    3 C(I)=1.0-Q
      B(1)=-1.0-AK1*((HHAT(1)-HHAT(NP2))/(T(1)-T(NP2))+LOVERG)
      C(1)=1.0-AK1*((HHAT(2)-HHAT(NP2))/(T(2)-T(NP2))-LOVERG)
      GO TO 1
    4 TOUT=T(1)
      TOLD=T(1)
      IF (NOEVAP) THEN
        FEVAP=0.0
C
C...  CALCULATE THE EFFECTS OF EVAPORATION
C
      ELSE
        FEVAP=(TIN-TOUT)/HFG*LHF
      ENDIF
      RETURN
      END
      SUBROUTINE SPLFIT (X,Y,N,HERMITE,A,B,C,INTER,IER)
      LOGICAL HERMITE
      DIMENSION X(N), Y(N), A(N), B(N), C(N)
C
C  SUBROUTINE SPLFIT CALCULATES THE COEFFICIENTS OF AN INTERPOLATING
C  NATURAL OR HERMITE CUBIC SPLINE.  SPLFCN CAN BE USED TO EVAULATE
C  THE INTERPOLATING POLYNOMIAL.
C
C  THE 1-D ARRAYS X AND Y CONTAIN N DATA PAIRS WHICH ARE TO
C  BE THE KNOTS OF THE INTERPOLATION.  THE INTERPOLATING FUNCTION
C  IS DEFINED BY
C
C              Y=Y(I)+((C(I)*T+B(I))*T+A(I))*T
C
C  WHERE       T=X-X(I)
C              X(I).LE.X.LT.X(I+1)
C  AND         A, B, AND C ARE ARRAYS OF LENGTH N CONTAINING THE
C              INTERPOLATING COEFFICIENTS.
C
C  HERMITE IS A LOGICAL VARIABLE (.TRUE. OR .FALSE.) WHICH
C  DETERMINES WHETHER A HERMITE OR NATURAL SPLINE IS USED
C  AS FOLLOWS.
C
C         HERMITE=.TRUE.   ==> HERMITE SPLINE USED IF N > 3
C         HERMITE=.FALSE.  ==> NATURAL SPLINE USED IF N > 3
```

Figure A.5 Continued

```
C          N=2 OR 3          ==> POLYNOMIAL INTERPOLATION USED
C
C  INTER IS AN INTEGER VARIABLE DESIGNATING THE INTERVAL OF INTER-
C  POLATION.  SPLFIT INITALIZES INTER TO 1
C
C  IER IS AN INTEGER ERROR FLAG DEFINED AS FOLLOWS.
C
C          IER=0             ==> NO ERROR -- NORMAL RETURN
C          IER=1             ==> N .LE. 1
C          IER=2             ==> X ARRAY NOT IN STRICTLY ASCENDING
C                                OR DESCENDING ORDER
C
C.....INITIALIZE INTER
      INTER=1
C
C.....CHECK INPUT DATA
      IER=1
      IF (N.LE.1) RETURN
      IER=2
      IF (X(2).EQ.X(1)) RETURN
      IF (N.GT.2) GO TO 1
C.....FIT A STRAIGHT LINE TO DATA
      A(1)=(Y(2)-Y(1))/(X(2)-X(1))
      B(1)=0.0
      C(1)=0.0
      IER=0
      RETURN
    1 IF (X(2).LT.X(1)) GO TO 3
      DO 2 I=3,N
      IF (X(I).LE.X(I-1)) RETURN
    2 CONTINUE
      GO TO 5
    3 DO 4 I=3,N
      IF (X(I).GE.X(I-1)) RETURN
    4 CONTINUE
    5 IER=0
      IF (N.GT.3) GO TO 6
C.....FIT A PARABOLA TO DATA
      X2X1=X(2)-X(1)
      X3X1=X(3)-X(1)
      Y2Y1=Y(2)-Y(1)
      Y3Y1=Y(3)-Y(1)
      D=X2X1*X3X1
      D=D*X2X1-D*X3X1
      BB=(Y2Y1*X3X1-Y3Y1*X2X1)/D
      AA=(X2X1*X2X1*Y3Y1-X3X1*X3X1*Y2Y1)/D
      A(1)=AA
      B(1)=BB
      C(1)=0.0
      B(2)=BB
      A(2)=AA+2.*BB*X2X1
      C(2)=0.0
      RETURN
    6 IF (HERMITE) GO TO 11
C.....NATURAL SPLINE INTERPOLATION
      M=N-1
      CI=X(2)-X(1)
      C(1)=CI
      BI=(Y(2)-Y(1))/CI
      B(2)=BI
      DO 7 I=2,M
      CC=X(I+1)-X(I)
      C(I)=CC
      BB=(Y(I+1)-Y(I))/CC
      B(I)=BB-BI
      BI=BB
      AA=CC+CI
      A(I)=AA+AA
```

Figure A.5 Continued

```
    7 CI=CC
      DO 8 I=3,M
      AA=C(I-1)/A(I-1)
      A(I)=A(I)-AA*C(I-1)
    8 B(I)=B(I)-AA*B(I-1)
      B(1)=0.0
      B(N)=0.0
      BB=B(M)/A(M)
      B(M)=BB
      JJ=M
      DO 9 I=3,M
      JJ=JJ-1
      BB=(B(JJ)-C(JJ)*BB)/A(JJ)
    9 B(JJ)=BB
      BI=0.0
      JJ=N
      DO 10 I=1,M
      JJ=JJ-1
      BB=B(JJ)
      CC=C(JJ)
      A(JJ)=(Y(JJ+1)-Y(JJ))/CC-CC*(BI+BB+BB)
      C(JJ)=(BI-BB)/CC
      B(JJ)=BB+BB+BB
   10 BI=BB
      RETURN
C.....HERMITE SPLINE
   11 RM3=(Y(2)-Y(1))/(X(2)-X(1))
      T1=RM3-(Y(2)-Y(3))/(X(2)-X(3))
      RM2=RM3+T1
      RM1=RM2+T1
      NO=N-2
      DO 12 I=1,N
      RM4=RM3-RM2+RM3
      IF (I.LE.NO) RM4=(Y(I+2)-Y(I+1))/(X(I+2)-X(I+1))
      T1=ABS(RM4-RM3)
      T2=ABS(RM2-RM1)
      BB=T1+T2
      A(I)=0.5*(RM2+RM3)
      IF (BB.NE.0.0) A(I)=(T1*RM2+T2*RM3)/BB
      RM1=RM2
      RM2=RM3
   12 RM3=RM4
      NO=N-1
      DO 13 I=1,NO
      T1=1.0/(X(I+1)-X(I))
      T2=(Y(I+1)-Y(I))*T1
      BB=(A(I)+A(I+1)-T2-T2)*T1
      B(I)=-BB+(T2-A(I))*T1
   13 C(I)=BB*T1
      RETURN
      END
      REAL FUNCTION SPLFCN (XIN,X,Y,N,A,B,C,INTER,IER)
      DIMENSION X(N), Y(N), A(N), B(N), C(N)
C
C   FUNCTION SPLFCN EVALUATES A SPLINE INTERPOLATION AT XIN
C
C   ARGUMENTS X,Y,N,A,B AND C ARE DEFINED AS IN SPLFIT.
C
C   INTER IS AN INTEGER VARIABLE INDICATING THE INTERVAL USED
C   IN THE INTERPOLATION (EXTRAPOLATION)
C
C   IER IS AN INTEGER ERROR FLAG DEFINED AS FOLLOWS.
C
C          IER=0       ==> NO ERROR -- NORMAL RETURN
C          IER=1       ==> XIN .LT. XMIN -- EXTRAPOLATION
C          IER=2       ==> XIN .GT. XMAX -- EXTRAPOLATION
C
```

Figure A.5 Continued

```
C  WHERE XMAX AND XMIN ARE THE MAXIMUM AND MINIMUM VALUES
C  OF THE X-ARRAY, RESPECTIVELY.
C
      IF (INTER.LE.0.OR.INTER.GT.N) INTER=1
      IER=0
      T=XIN-X(INTER)
      U=XIN-X(INTER+1)
      IF (U*T.GT.0.0) GO TO 1
      SPLFCN=Y(INTER)+((C(INTER)*T+B(INTER))*T+A(INTER))*T
      RETURN
    1 K=1
      IF (T.LT.0.0) K=-K
      IF ((T-U).LT.0.0) K=-K
    2 INTER=INTER+K
      IF ((N-INTER)*INTER.LE.0) GO TO 3
      T=XIN-X(INTER)
      U=XIN-X(INTER+1)
      IF (U*T.GT.0.0) GO TO 2
      SPLFCN=Y(INTER)+((C(INTER)*T+B(INTER))*T+A(INTER))*T
      RETURN
    3 IER=1
      IF (T.LT.0.0) IER=2
      INTER=INTER-K
      SPLFCN=Y(INTER)+((C(INTER)*T+B(INTER))*T+A(INTER))*T
      RETURN
      END
      SUBROUTINE LFIT (XTABLE,YTABLE,NDATA,AAA,IER)
      DIMENSION XTABLE(NDATA), YTABLE(NDATA), AAA(NDATA)
C
C...  SUBROUTINE LFIT CALCULATES THE SLOPE FOR LINEAR
C...  INTERPOLATION OVER AN INTERVAL.
C...  SLFCN CAN BE USED TO EVALUATE A LINEAR FUNCTION
C...  WITHIN AN INTERVAL.
C
C...  XTABLE AND YTABLE ARE 1-D ARRAYS CONTAINING NDATA
C...  POINTS WHICH FORM NDATA-1 INTERVALS FOR INTERPOLATION.
C
C...  THE LINEAR FUNCTION IS
C
C             Y=YTABLE(I)+(X-XTABLE(I))*((YTABLE(I+1)-YTABLE(I))
C                                         -----------------------
C                                         (XTABLE(I+1)-XTABLE(I)))
C
C
C
C         WHERE      X LIES BETWEEN XTABLE(I) AND XTABLE(I+1)
C
C         AAA IS AN ARRAY CONTAINING THE SLOPE VALUES FOR EACH INTERVAL
C
C...  IER IS AN ERROR FLAG DEFINED AS FOLLOWS
C
C             IER=0             ==> NO ERROR
C             IER=1             ==> NDATA .LE. 1
C             IER=2             ==> X ARRAY IS NOT IN STRICTLY
C                                   ASCENDING OR DESCENDING ORDER
C
C
C
C...  CHECK INPUT
C
      IER=1
      IF(NDATA.LE.1) RETURN
      IER=0
      IF(NDATA.EQ.2) GO TO 5
      IER=2
C
C...  CHECK TO MAKE SURE DATA IS IN STRICTLY ASCENDING OR
C...  DESCENDING X ORDER
C
      IF(XTABLE(2).LT.XTABLE(1)) GO TO 2
```

Figure A.5 Continued

```
      DO 1 I=3,NDATA
        IF(XTABLE(NDATA).LT.XTABLE(NDATA-1)) RETURN
    1 CONTINUE
      GO TO 4
    2 DO 3 I=3,NDATA
        IF (XTABLE(NDATA).GT.XTABLE(NDATA-1)) RETURN
    3 CONTINUE
    4 IER=0
C
C...  CALCULATE SLOPE FOR EACH INTERVAL
C
    5 DO 6 I=1,NDATA-1
      AA=XTABLE(I+1)-XTABLE(I)
      IF(AA.EQ.0.0) THEN
        AAA(I)=0.0
        GO TO 6
      ENDIF
      AAA(I)=(YTABLE(I+1)-YTABLE(I))/AA
    6 CONTINUE
      RETURN
      END
      REAL FUNCTION SLFCN (XIN,XTABLE,YTABLE,NDATA,AAA,IER)
      DIMENSION XTABLE(NDATA), YTABLE(NDATA), AAA(NDATA)
      SAVE INTER,K
C
C...  THIS FUNCTION EVALUATES A LINEAR INTERPOLATION AT XIN.
C
C
C...  ARGUMENTS XTABLE, YTABLE, NDATA, AND AAA ARE AS DEFINED
C               IN LFIT.
C
C...  IER IS AN INTERGER ERROR FLAG DEFINED AS FOLLOWS.
C
C              IER=0        ==> NO ERROR
C              IER=1        ==> XIN .LT. MINIMUM VALUE OF X-ARRAY
C              IER=2        ==> XIN .GT. MAXIMUM VALUE OF X-ARRAY
C
C
C...  THE INTER (INTERVAL) VALUE IS SAVED EACH TIME SLFCN
C...  IS CALL TO PREVENT REPEATATIVE SEARCHING.
C
      DATA INTER/1/
      IER=0
      INTER=MIN0(INTER,NDATA-1)
      XXI=XIN-XTABLE(INTER)
      XXIP1=XIN-XTABLE(INTER+1)
      IF(XXI*XXIP1.GT.0.0) GO TO 1
      SLFCN=YTABLE(INTER)+XXI*AAA(INTER)
      RETURN
C
C...  SET K = 1,-1 DEPENDING ON PRESENT X LOCATION, TO FIND
C...  THE INTERVAL CONTAINING X
C
    1 K=1
      IF(XXI.LT.0.0) K=-K
      IF((XXI-XXIP1).LT.0.0) K=-K
    2 INTER=INTER+K
      IF((NDATA-INTER)*INTER.LE.0) GO TO 3
      XXI=XIN-XTABLE(INTER)
      XXIP1=XIN-XTABLE(INTER+1)
      IF(XXI*XXIP1.GT.0.0) GO TO 2
      SLFCN=YTABLE(INTER)+XXI*AAA(INTER)
      RETURN
C
C...  EXTRAPOLATION
C
    3 IER=1
      IF(XXI.LT.0.0) IER=2
```

Figure A.5 Continued

```
      INTER=INTER-K
      SLFCN=YTABLE(INTER)+XXI*AAA(INTER)
      RETURN
      END
      SUBROUTINE TDMS (A,B,C,X,D,N,IER,WORK)
      DIMENSION A(N),B(N),C(N),X(N),D(N),WORK(1)
C
C...  TDMS SOLVES A SET OF ALGEBRAIC EQUATIONS OF THE FORM:
C
C       A(I)*X(I-1)+B(I)*X(I)+C(I)*X(I+1)=D(I)
C
C...  FOR I=1,2,....,N, USING THE TRIDIAGONAL MATRIX SOLUTION.
C
C...  OTHER VARIABLES:
C
C     IER      = 0  -- NORMAL RETURN
C              = 1  -- ERROR BECAUSE N IS LESS THAN 1
C              = 2  -- EQUATIONS ARE SINGULAR
C
C...  WORK     A WORK ARRAY DIMENSIONED AT LEAST 2*(N-1)
C
      IF (N.LE.1) GO TO 4
      IER=2
      IF (B(1).EQ.0.0) RETURN
      NM1=N-1
      BOLD=C(1)/B(1)
      GOLD=D(1)/B(1)
      WORK(1)=GOLD
      WORK(1+NM1)=BOLD
      IF (N.EQ.2) GO TO 2
      DO 1 I=2,NM1
      DIV=B(I)-A(I)*BOLD
      IF (DIV.EQ.0.0) RETURN
      GOLD=(D(I)-A(I)*GOLD)/DIV
      WORK(I)=GOLD
      BOLD=C(I)/DIV
    1 WORK(I+NM1)=BOLD
    2 DIV=B(N)-A(N)*BOLD
      IF (DIV.EQ.0.0) RETURN
      GOLD=(D(N)-A(N)*GOLD)/DIV
      X(N)=GOLD
      L=N
      DO 3 I=2,N
      L=L-1
      GOLD=WORK(L)-WORK(L+NM1)*GOLD
    3 X(L)=GOLD
      IER=0
      RETURN
    4 IER=1
      IF (N.LE.0) RETURN
      IER=2
      IF (B(1).EQ.0.0) RETURN
      X(1)=D(1)/B(1)
      IER=0
      RETURN
      END
      SUBROUTINE ODE(F,NEQN,Y,T,TOUT,RELERR,ABSERR,IFLAG,WORK)
C
C    SUBROUTINE  ODE  INTEGRATES A SYSTEM OF  NEQN  FIRST ORDER
C    ORDINARY DIFFERENTIAL EQUATIONS OF THE FORM
C              DY(I)/DT = F(T,Y(1),Y(2),...,Y(NEQN))
C              Y(I) GIVEN AT  T .
C    THE SUBROUTINE INTEGRATES FROM  T  TO  TOUT .  ON RETURN THE
C    PARAMETERS IN THE CALL LIST ARE SET FOR CONTINUING THE INTEGRATION.
C    THE USER HAS ONLY TO DEFINE A NEW VALUE  TOUT  AND CALL  ODE  AGAIN.
C
C    THE DIFFERENTIAL EQUATIONS ARE ACTUALLY SOLVED BY A SUITE OF CODES
C    DE ,  STEP , AND  INTRP .  ODE  ALLOCATES VIRTUAL STORAGE IN THE
```

Figure A.5 Continued

```
C    ARRAYS  WORK  AND  IWORK  AND CALLS  DE .  DE  IS A SUPERVISOR WHICH
C    DIRECTS THE SOLUTION.  IT CALLS ON THE ROUTINES  STEP  AND  INTRP
C    TO ADVANCE THE INTEGRATION AND TO INTERPOLATE AT OUTPUT POINTS.
C    STEP  USES A MODIFIED DIVIDED DIFFERENCE FORM OF THE ADAMS PECE
C    FORMULAS AND LOCAL EXTRAPOLATION.  IT ADJUSTS THE ORDER AND STEP
C    SIZE TO CONTROL THE LOCAL ERROR PER UNIT STEP IN A GENERALIZED
C    SENSE.  NORMALLY EACH CALL TO  STEP  ADVANCES THE SOLUTION ONE STEP
C    IN THE DIRECTION OF  TOUT .  FOR REASONS OF EFFICIENCY  DE
C    INTEGRATES BEYOND  TOUT  INTERNALLY, THOUGH NEVER BEYOND
C    T+10*(TOUT-T), AND CALLS  INTRP  TO INTERPOLATE THE SOLUTION AT
C    TOUT .  AN OPTION IS PROVIDED TO STOP THE INTEGRATION AT  TOUT  BUT
C    IT SHOULD BE USED ONLY IF IT IS IMPOSSIBLE TO CONTINUE THE
C    INTEGRATION BEYOND  TOUT .
C
C    THIS CODE IS COMPLETELY EXPLAINED AND DOCUMENTED IN THE TEXT,
C    COMPUTER SOLUTION OF ORDINARY DIFFERENTIAL EQUATIONS:  THE INITIAL
C    VALUE PROBLEM  BY L. F. SHAMPINE AND M. K. GORDON.
C
C    THE PARAMETERS REPRESENT:
C       F -- SUBROUTINE F(T,Y,YP) TO EVALUATE DERIVATIVES YP(I)=DY(I)/DT
C       NEQN -- NUMBER OF EQUATIONS TO BE INTEGRATED
C       Y(*) -- SOLUTION VECTOR AT  T
C       T -- INDEPENDENT VARIABLE
C       TOUT -- POINT AT WHICH SOLUTION IS DESIRED
C       RELERR,ABSERR -- RELATIVE AND ABSOLUTE ERROR TOLERANCES FOR LOCAL
C            ERROR TEST.  AT EACH STEP THE CODE REQUIRES
C              ABS(LOCAL ERROR) .LE. ABS(Y)*RELERR + ABSERR
C            FOR EACH COMPONENT OF THE LOCAL ERROR AND SOLUTION VECTORS
C       IFLAG -- INDICATES STATUS OF INTEGRATION
C       WORK(*) -- ARRAY TO HOLD INFORMATION INTERNAL TO CODE
C            WHICH IS NECESSARY FOR SUBSEQUENT CALLS
C
C    FIRST CALL TO ODE --
C
C    THE USER MUST PROVIDE STORAGE IN HIS CALLING PROGRAM FOR THE ARRAYS
C    IN THE CALL LIST,
C       Y(NEQN), WORK(100+21*NEQN),
C    DECLARE  F  IN AN EXTERNAL STATEMENT, SUPPLY THE SUBROUTINE
C    F(T,Y,YP) TO EVALUATE
C       DY(I)/DT = YP(I) = F(T,Y(1),Y(2),...,Y(NEQN))
C    AND INITIALIZE THE PARAMETERS:
C       NEQN -- NUMBER OF EQUATIONS TO BE INTEGRATED
C       Y(*) -- VECTOR OF INITIAL CONDITIONS
C       T -- STARTING POINT OF INTEGRATION
C       TOUT -- POINT AT WHICH SOLUTION IS DESIRED
C       RELERR,ABSERR -- RELATIVE AND ABSOLUTE LOCAL ERROR TOLERANCES
C       IFLAG -- +1,-1.  INDICATOR TO INITIALIZE THE CODE.  NORMAL INPUT
C            IS +1.  THE USER SHOULD SET IFLAG=-1 ONLY IF IT IS
C            IMPOSSIBLE TO CONTINUE THE INTEGRATION BEYOND  TOUT .
C    ALL PARAMETERS EXCEPT  F ,  NEQN  AND  TOUT  MAY BE ALTERED BY THE
C    CODE ON OUTPUT SO MUST BE VARIABLES IN THE CALLING PROGRAM.
C
C    OUTPUT FROM  ODE  --
C
C       NEQN -- UNCHANGED
C       Y(*) -- SOLUTION AT  T
C       T -- LAST POINT REACHED IN INTEGRATION.  NORMAL RETURN HAS
C            T = TOUT .
C       TOUT -- UNCHANGED
C       RELERR,ABSERR -- NORMAL RETURN HAS TOLERANCES UNCHANGED.  IFLAG=3
C            SIGNALS TOLERANCES INCREASED
C       IFLAG = 2 -- NORMAL RETURN.  INTEGRATION REACHED  TOUT
C             = 3 -- INTEGRATION DID NOT REACH  TOUT  BECAUSE ERROR
C                    TOLERANCES TOO SMALL.  RELERR ,  ABSERR  INCREASED
C                    APPROPRIATELY FOR CONTINUING
C             = 4 -- INTEGRATION DID NOT REACH  TOUT  BECAUSE MORE THAN
C                    500 STEPS NEEDED
C             = 5 -- INTEGRATION DID NOT REACH  TOUT  BECAUSE EQUATIONS
```

Figure A.5 Continued

```
C                 APPEAR TO BE STIFF
C              = 6 -- INVALID INPUT PARAMETERS (FATAL ERROR)
C             THE VALUE OF  IFLAG  IS RETURNED NEGATIVE WHEN THE INPUT
C             VALUE IS NEGATIVE AND THE INTEGRATION DOES NOT REACH  TOUT ,
C             I.E., -3, -4, -5.
C        WORK(*),IWORK(*) -- INFORMATION GENERALLY OF NO INTEREST TO THE
C             USER BUT NECESSARY FOR SUBSEQUENT CALLS.
C
C   SUBSEQUENT CALLS TO  ODE --
C
C   SUBROUTINE  ODE  RETURNS WITH ALL INFORMATION NEEDED TO CONTINUE
C   THE INTEGRATION.  IF THE INTEGRATION REACHED  TOUT , THE USER NEED
C   ONLY DEFINE A NEW  TOUT  AND CALL AGAIN.  IF THE INTEGRATION DID NOT
C   REACH  TOUT  AND THE USER WANTS TO CONTINUE, HE JUST CALLS AGAIN.
C   THE OUTPUT VALUE OF  IFLAG  IS THE APPROPRIATE INPUT VALUE FOR
C   SUBSEQUENT CALLS.  THE ONLY SITUATION IN WHICH IT SHOULD BE ALTERED
C   IS TO STOP THE INTEGRATION INTERNALLY AT THE NEW  TOUT , I.E.,
C   CHANGE OUTPUT  IFLAG=2  TO INPUT  IFLAG=-2 .  ERROR TOLERANCES MAY
C   BE CHANGED BY THE USER BEFORE CONTINUING.  ALL OTHER PARAMETERS MUST
C   REMAIN UNCHANGED.
C
C***********************************************************************
C*  SUBROUTINES  DE  AND  STEP  CONTAIN MACHINE DEPENDENT CONSTANTS.   *
C*  BE SURE THEY ARE SET BEFORE USING  ODE .                           *
C***********************************************************************
C
      LOGICAL START,PHASE1,NORND
      DIMENSION Y(NEQN),WORK(1),IWORK(5)
      DATA IALPHA,IBETA,ISIG,IV,IW,IG,IPHASE,IPSI,IX,IH,IHOLD,ISTART,
     1  ITOLD,IDELSN/1,13,25,38,50,62,75,76,88,89,90,91,92,93/
      IYY = 100
      IWT = IYY + NEQN
      IP = IWT + NEQN
      IYP = IP + NEQN
      IYPOUT = IYP + NEQN
      IPHI = IYPOUT + NEQN
      IF(IABS(IFLAG) .EQ. 1) GO TO 1
      START = WORK(ISTART) .GT. 0.0
      PHASE1 = WORK(IPHASE) .GT. 0.0
      NORND = IWORK(2) .NE. -1
    1 CALL DE(F,NEQN,Y,T,TOUT,RELERR,ABSERR,IFLAG,WORK(IYY),
     1  WORK(IWT),WORK(IP),WORK(IYP),WORK(IYPOUT),WORK(IPHI),
     2  WORK(IALPHA),WORK(IBETA),WORK(ISIG),WORK(IV),WORK(IW),WORK(IG),
     3  PHASE1,WORK(IPSI),WORK(IX),WORK(IH),WORK(IHOLD),START,
     4  WORK(ITOLD),WORK(IDELSN),IWORK(1),NORND,IWORK(3),IWORK(4),
     5  IWORK(5))
      WORK(ISTART) = -1.0
      IF(START) WORK(ISTART) = 1.0
      WORK(IPHASE) = -1.0
      IF(PHASE1) WORK(IPHASE) = 1.0
      IWORK(2) = -1
      IF(NORND) IWORK(2) = 1
      RETURN
      END
      SUBROUTINE DE(F,NEQN,Y,T,TOUT,RELERR,ABSERR,IFLAG,
     1  YY,WT,P,YP,YPOUT,PHI,ALPHA,BETA,SIG,V,W,G,PHASE1,PSI,X,H,HOLD,
     2  START,TOLD,DELSGN,NS,NORND,K,KOLD,ISNOLD)
C
C   ODE  MERELY ALLOCATES STORAGE FOR  DE  TO RELIEVE THE USER OF THE
C   INCONVENIENCE OF A LONG CALL LIST.  CONSEQUENTLY  DE  IS USED AS
C   DESCRIBED IN THE COMMENTS FOR  ODE .
C
C   THIS CODE IS COMPLETELY EXPLAINED AND DOCUMENTED IN THE TEXT,
C   COMPUTER SOLUTION OF ORDINARY DIFFERENTIAL EQUATIONS:  THE INITIAL
C   VALUE PROBLEM  BY L. F. SHAMPINE AND M. K. GORDON.
C
      LOGICAL STIFF,CRASH,START,PHASE1,NORND
      DIMENSION Y(NEQN),YY(NEQN),WT(NEQN),PHI(NEQN,16),P(NEQN),YP(NEQN),
```

Figure A.5 Continued

```
      1  YPOUT(NEQN),PSI(12),ALPHA(12),BETA(12),SIG(13),V(12),W(12),G(13)
         EXTERNAL F
C
C***********************************************************************
C*  THE ONLY MACHINE DEPENDENT CONSTANT IS BASED ON THE MACHINE UNIT   *
C*  ROUNDOFF ERROR  U  WHICH IS THE SMALLEST POSITIVE NUMBER SUCH THAT *
C*  1.0+U .GT. 1.0 .  U  MUST BE CALCULATED AND  FOURU=4.0*U  INSERTED *
C*  IN THE FOLLOWING DATA STATEMENT BEFORE USING  DE .  THE ROUTINE    *
C*  MACHIN  CALCULATES  U .  FOURU  AND  TWOU=2.0*U  MUST ALSO BE      *
C*  INSERTED IN SUBROUTINE  STEP  BEFORE CALLING  DE .                 *
         DATA FOURU/28.E-15/
C***********************************************************************
C
C   THE CONSTANT  MAXNUM  IS THE MAXIMUM NUMBER OF STEPS ALLOWED IN ONE
C   CALL TO  DE .  THE USER MAY CHANGE THIS LIMIT BY ALTERING THE
C   FOLLOWING STATEMENT
         DATA MAXNUM/500/
C
C             ***              ***              ***
C   TEST FOR IMPROPER PARAMETERS
C
         IF(NEQN .LT. 1) GO TO 10
         IF(T .EQ. TOUT) GO TO 10
         IF(RELERR .LT. 0.0  .OR.  ABSERR .LT. 0.0) GO TO 10
         EPS = AMAX1(RELERR,ABSERR)
         IF(EPS .LE. 0.0) GO TO 10
         IF(IFLAG .EQ. 0) GO TO 10
         ISN = ISIGN(1,IFLAG)
         IFLAG = IABS(IFLAG)
         IF(IFLAG .EQ. 1) GO TO 20
         IF(T .NE. TOLD) GO TO 10
         IF(IFLAG .GE. 2  .AND.  IFLAG .LE. 5) GO TO 20
   10    IFLAG = 6
         RETURN
C
C   ON EACH CALL SET INTERVAL OF INTEGRATION AND COUNTER FOR NUMBER OF
C   STEPS.  ADJUST INPUT ERROR TOLERANCES TO DEFINE WEIGHT VECTOR FOR
C   SUBROUTINE  STEP
C
   20 DEL = TOUT - T
         ABSDEL = ABS(DEL)
         TEND = T + 10.0*DEL
         IF(ISN .LT. 0) TEND = TOUT
         NOSTEP = 0
         KLE4 = 0
         STIFF = .FALSE.
         RELEPS = RELERR/EPS
         ABSEPS = ABSERR/EPS
         IF(IFLAG .EQ. 1) GO TO 30
         IF(ISNOLD .LT. 0) GO TO 30
         IF(DELSGN*DEL .GT. 0.0) GO TO 50
C
C   ON START AND RESTART ALSO SET WORK VARIABLES X AND YY(*), STORE THE
C   DIRECTION OF INTEGRATION AND INITIALIZE THE STEP SIZE
C
   30 START = .TRUE.
         X = T
         DO 40 L = 1,NEQN
   40    YY(L) = Y(L)
         DELSGN = SIGN(1.0,DEL)
         H = SIGN(AMAX1(ABS(TOUT-X),FOURU*ABS(X)),TOUT-X)
C
C   IF ALREADY PAST OUTPUT POINT, INTERPOLATE AND RETURN
C
   50 IF(ABS(X-T) .LT. ABSDEL) GO TO 60
         CALL INTRP(X,YY,TOUT,Y,YPOUT,NEQN,KOLD,PHI,PSI)
         IFLAG = 2
         T = TOUT
```

Figure A.5 Continued

```
      TOLD = T
      ISNOLD = ISN
      RETURN
C
C   IF CANNOT GO PAST OUTPUT POINT AND SUFFICIENTLY CLOSE,
C   EXTRAPOLATE AND RETURN
C
   60 IF(ISN .GT. 0  .OR.  ABS(TOUT-X) .GE. FOURU*ABS(X)) GO TO 80
      H = TOUT - X
      CALL F(X,YY,YP)
      DO 70 L = 1,NEQN
   70   Y(L) = YY(L) + H*YP(L)
      IFLAG = 2
      T = TOUT
      TOLD = T
      ISNOLD = ISN
      RETURN
C
C   TEST FOR TOO MANY STEPS
C
   80 IF(NOSTEP .LT. MAXNUM) GO TO 100
      IFLAG = ISN*4
      IF(STIFF) IFLAG = ISN*5
      DO 90 L = 1,NEQN
   90   Y(L) = YY(L)
      T = X
      TOLD = T
      ISNOLD = 1
      RETURN
C
C   LIMIT STEP SIZE, SET WEIGHT VECTOR AND TAKE A STEP
C
  100 H = SIGN(AMIN1(ABS(H),ABS(TEND-X)),H)
      DO 110 L = 1,NEQN
  110   WT(L) = RELEPS*ABS(YY(L)) + ABSEPS
      CALL STEP(X,YY,F,NEQN,H,EPS,WT,START,
     1  HOLD,K,KOLD,CRASH,PHI,P,YP,PSI,
     2  ALPHA,BETA,SIG,V,W,G,PHASE1,NS,NORND)
C
C   TEST FOR TOLERANCES TOO SMALL
C
      IF(.NOT.CRASH) GO TO 130
      IFLAG = ISN*3
      RELERR = EPS*RELEPS
      ABSERR = EPS*ABSEPS
      DO 120 L = 1,NEQN
  120   Y(L) = YY(L)
      T = X
      TOLD = T
      ISNOLD = 1
      RETURN
C
C   AUGMENT COUNTER ON NUMBER OF STEPS AND TEST FOR STIFFNESS
C
  130 NOSTEP = NOSTEP + 1
      KLE4 = KLE4 + 1
      IF(KOLD .GT. 4) KLE4 = 0
      IF(KLE4 .GE. 50) STIFF = .TRUE.
      GO TO 50
      END
      SUBROUTINE STEP(X,Y,F,NEQN,H,EPS,WT,START,
     1  HOLD,K,KOLD,CRASH,PHI,P,YP,PSI,
     2  ALPHA,BETA,SIG,V,W,G,PHASE1,NS,NORND)
C
C   SUBROUTINE  STEP  INTEGRATES A SYSTEM OF FIRST ORDER ORDINARY
C   DIFFERENTIAL EQUATIONS ONE STEP, NORMALLY FROM X TO X+H, USING A
C   MODIFIED DIVIDED DIFFERENCE FORM OF THE ADAMS PECE FORMULAS.  LOCAL
```

Figure A.5 Continued

```
C   EXTRAPOLATION IS USED TO IMPROVE ABSOLUTE STABILITY AND ACCURACY.
C   THE CODE ADJUSTS ITS ORDER AND STEP SIZE TO CONTROL THE LOCAL ERROR
C   PER UNIT STEP IN A GENERALIZED SENSE.  SPECIAL DEVICES ARE INCLUDED
C   TO CONTROL ROUNDOFF ERROR AND TO DETECT WHEN THE USER IS REQUESTING
C   TOO MUCH ACCURACY.
C
C   THIS CODE IS COMPLETELY EXPLAINED AND DOCUMENTED IN THE TEXT,
C   COMPUTER SOLUTION OF ORDINARY DIFFERENTIAL EQUATIONS:  THE INITIAL
C   VALUE PROBLEM  BY L. F. SHAMPINE AND M. K. GORDON.
C
C
C   THE PARAMETERS REPRESENT:
C      X -- INDEPENDENT VARIABLE
C      Y(*) -- SOLUTION VECTOR AT X
C      YP(*) -- DERIVATIVE OF SOLUTION VECTOR AT  X  AFTER SUCCESSFUL
C           STEP
C      NEQN -- NUMBER OF EQUATIONS TO BE INTEGRATED
C      H -- APPROPRIATE STEP SIZE FOR NEXT STEP.  NORMALLY DETERMINED BY
C           CODE
C      EPS -- LOCAL ERROR TOLERANCE.  MUST BE VARIABLE
C      WT(*) -- VECTOR OF WEIGHTS FOR ERROR CRITERION
C      START -- LOGICAL VARIABLE SET .TRUE. FOR FIRST STEP,  .FALSE.
C           OTHERWISE
C      HOLD -- STEP SIZE USED FOR LAST SUCCESSFUL STEP
C      K -- APPROPRIATE ORDER FOR NEXT STEP (DETERMINED BY CODE)
C      KOLD -- ORDER USED FOR LAST SUCCESSFUL STEP
C      CRASH -- LOGICAL VARIABLE SET .TRUE. WHEN NO STEP CAN BE TAKEN,
C           .FALSE. OTHERWISE.
C   THE ARRAYS  PHI, PSI  ARE REQUIRED FOR THE INTERPOLATION SUBROUTINE
C   INTRP .  THE ARRAY  P  IS INTERNAL TO THE CODE.
C
C   INPUT TO  STEP
C
C      FIRST CALL --
C
C   THE USER MUST PROVIDE STORAGE IN HIS DRIVER PROGRAM FOR ALL ARRAYS
C   IN THE CALL LIST, NAMELY
C
C     DIMENSION Y(NEQN),WT(NEQN),PHI(NEQN,16),P(NEQN),YP(NEQN),PSI(12)
C
C   THE USER MUST ALSO DECLARE  START  AND  CRASH  LOGICAL VARIABLES
C   AND  F  AN EXTERNAL SUBROUTINE, SUPPLY THE SUBROUTINE  F(X,Y,YP)
C   TO EVALUATE
C      DY(I)/DX = YP(I) = F(X,Y(1),Y(2),...,Y(NEQN))
C   AND INITIALIZE ONLY THE FOLLOWING PARAMETERS:
C      X -- INITIAL VALUE OF THE INDEPENDENT VARIABLE
C      Y(*) -- VECTOR OF INITIAL VALUES OF DEPENDENT VARIABLES
C      NEQN -- NUMBER OF EQUATIONS TO BE INTEGRATED
C      H -- NOMINAL STEP SIZE INDICATING DIRECTION OF INTEGRATION
C           AND MAXIMUM SIZE OF STEP.  MUST BE VARIABLE
C      EPS -- LOCAL ERROR TOLERANCE PER STEP.  MUST BE VARIABLE
C      WT(*) -- VECTOR OF NON-ZERO WEIGHTS FOR ERROR CRITERION
C      START -- .TRUE.
C
C   STEP  REQUIRES THE L2 NORM OF THE VECTOR WITH COMPONENTS
C   LOCAL ERROR(L)/WT(L)  BE LESS THAN  EPS  FOR A SUCCESSFUL STEP.  THE
C   ARRAY  WT  ALLOWS THE USER TO SPECIFY AN ERROR TEST APPROPRIATE
C   FOR HIS PROBLEM.  FOR EXAMPLE,
C      WT(L) = 1.0  SPECIFIES ABSOLUTE ERROR,
C            = ABS(Y(L))  ERROR RELATIVE TO THE MOST RECENT VALUE OF THE
C                 L-TH COMPONENT OF THE SOLUTION,
C            = ABS(YP(L))  ERROR RELATIVE TO THE MOST RECENT VALUE OF
C                 THE L-TH COMPONENT OF THE DERIVATIVE,
C            = AMAX1(WT(L),ABS(Y(L)))  ERROR RELATIVE TO THE LARGEST
C                 MAGNITUDE OF L-TH COMPONENT OBTAINED SO FAR,
C            = ABS(Y(L))*RELERR/EPS + ABSERR/EPS  SPECIFIES A MIXED
C                 RELATIVE-ABSOLUTE TEST WHERE  RELERR  IS RELATIVE
C                 ERROR,  ABSERR  IS ABSOLUTE ERROR AND  EPS =
```

Figure A.5 Continued

```
C                  AMAX1(RELERR,ABSERR) .
C
C       SUBSEQUENT CALLS --
C
C    SUBROUTINE  STEP  IS DESIGNED SO THAT ALL INFORMATION NEEDED TO
C    CONTINUE THE INTEGRATION, INCLUDING THE STEP SIZE  H  AND THE ORDER
C    K , IS RETURNED WITH EACH STEP.  WITH THE EXCEPTION OF THE STEP
C    SIZE, THE ERROR TOLERANCE, AND THE WEIGHTS, NONE OF THE PARAMETERS
C    SHOULD BE ALTERED.  THE ARRAY  WT  MUST BE UPDATED AFTER EACH STEP
C    TO MAINTAIN RELATIVE ERROR TESTS LIKE THOSE ABOVE.  NORMALLY THE
C    INTEGRATION IS CONTINUED JUST BEYOND THE DESIRED ENDPOINT AND THE
C    SOLUTION INTERPOLATED THERE WITH SUBROUTINE  INTRP .  IF IT IS
C    IMPOSSIBLE TO INTEGRATE BEYOND THE ENDPOINT, THE STEP SIZE MAY BE
C    REDUCED TO HIT THE ENDPOINT SINCE THE CODE WILL NOT TAKE A STEP
C    LARGER THAN THE  H  INPUT.  CHANGING THE DIRECTION OF INTEGRATION,
C    I.E., THE SIGN OF  H , REQUIRES THE USER SET  START = .TRUE. BEFORE
C    CALLING  STEP  AGAIN.  THIS IS THE ONLY SITUATION IN WHICH  START
C    SHOULD BE ALTERED.
C
C    OUTPUT FROM  STEP
C
C       SUCCESSFUL STEP --
C
C    THE SUBROUTINE RETURNS AFTER EACH SUCCESSFUL STEP WITH  START  AND
C    CRASH  SET .FALSE. .  X  REPRESENTS THE INDEPENDENT VARIABLE
C    ADVANCED ONE STEP OF LENGTH  HOLD  FROM ITS VALUE ON INPUT AND  Y
C    THE SOLUTION VECTOR AT THE NEW VALUE OF  X .  ALL OTHER PARAMETERS
C    REPRESENT INFORMATION CORRESPONDING TO THE NEW  X  NEEDED TO
C    CONTINUE THE INTEGRATION.
C
C       UNSUCCESSFUL STEP --
C
C    WHEN THE ERROR TOLERANCE IS TOO SMALL FOR THE MACHINE PRECISION,
C    THE SUBROUTINE RETURNS WITHOUT TAKING A STEP AND  CRASH = .TRUE. .
C    AN APPROPRIATE STEP SIZE AND ERROR TOLERANCE FOR CONTINUING ARE
C    ESTIMATED AND ALL OTHER INFORMATION IS RESTORED AS UPON INPUT
C    BEFORE RETURNING.  TO CONTINUE WITH THE LARGER TOLERANCE, THE USER
C    JUST CALLS THE CODE AGAIN.  A RESTART IS NEITHER REQUIRED NOR
C    DESIRABLE.
C
      EXTERNAL F
      LOGICAL START,CRASH,PHASE1,NORND
      DIMENSION Y(NEQN),WT(NEQN),PHI(NEQN,16),P(NEQN),YP(NEQN),PSI(12)
      DIMENSION ALPHA(12),BETA(12),SIG(13),W(12),V(12),G(13),
     1  GSTR(13),TWO(13)
C***********************************************************************
C*  THE ONLY MACHINE DEPENDENT CONSTANTS ARE BASED ON THE MACHINE UNIT *
C*  ROUNDOFF ERROR  U  WHICH IS THE SMALLEST POSITIVE NUMBER SUCH THAT *
C*  1.0+U .GT. 1.0  .  THE USER MUST CALCULATE  U  AND INSERT          *
C*  TWOU=2.0*U  AND  FOURU=4.0*U  IN THE DATA STATEMENT BEFORE CALLING *
C*  THE CODE.  THE ROUTINE  MACHIN  CALCULATES  U .                    *
      DATA TWOU,FOURU/14.2E-15,28.E-15/
C***********************************************************************
      DATA TWO/2.0,4.0,8.0,16.0,32.0,64.0,128.0,256.0,512.0,1024.0,
     1  2048.0,4096.0,8192.0/
      DATA GSTR/0.500,0.0833,0.0417,0.0264,0.0188,0.0143,0.0114,0.00936,
     1  0.00789,0.00679,0.00592,0.00524,0.00468/
C     DATA G(1),G(2)/1.0,0.5/,SIG(1)/1.0/
C
C
C        ***     BEGIN BLOCK 0     ***
C    CHECK IF STEP SIZE OR ERROR TOLERANCE IS TOO SMALL FOR MACHINE
C    PRECISION.  IF FIRST STEP, INITIALIZE PHI ARRAY AND ESTIMATE A
C    STARTING STEP SIZE.
C                   ***
C
C    IF STEP SIZE IS TOO SMALL, DETERMINE AN ACCEPTABLE ONE
C
```

Figure A.5 Continued

```
      CRASH = .TRUE.
      IF(ABS(H) .GE. FOURU*ABS(X)) GO TO 5
      H = SIGN(FOURU*ABS(X),H)
      RETURN
    5 P5EPS = 0.5*EPS
C
C   IF ERROR TOLERANCE IS TOO SMALL, INCREASE IT TO AN ACCEPTABLE VALUE
C
      ROUND = 0.0
      DO 10 L = 1,NEQN
   10   ROUND = ROUND + (Y(L)/WT(L))**2
      ROUND = TWOU*SQRT(ROUND)
      IF(P5EPS .GE. ROUND) GO TO 15
      EPS = 2.0*ROUND*(1.0 + FOURU)
      RETURN
   15 CRASH = .FALSE.
      G(1)=1.0
      G(2)=0.5
      SIG(1)=1.0
      IF(.NOT.START) GO TO 99
C
C   INITIALIZE.  COMPUTE APPROPRIATE STEP SIZE FOR FIRST STEP
C
      CALL F(X,Y,YP)
      SUM = 0.0
      DO 20 L = 1,NEQN
        PHI(L,1) = YP(L)
        PHI(L,2) = 0.0
   20   SUM = SUM + (YP(L)/WT(L))**2
      SUM = SQRT(SUM)
      ABSH = ABS(H)
      IF(EPS .LT. 16.0*SUM*H*H) ABSH = 0.25*SQRT(EPS/SUM)
      H = SIGN(AMAX1(ABSH,FOURU*ABS(X)),H)
      HOLD = 0.0
      K = 1
      KOLD = 0
      START = .FALSE.
      PHASE1 = .TRUE.
      NORND = .TRUE.
      IF(P5EPS .GT. 100.0*ROUND) GO TO 99
      NORND = .FALSE.
      DO 25 L = 1,NEQN
   25   PHI(L,15) = 0.0
   99 IFAIL = 0
C       ***      END BLOCK 0      ***
C
C       ***      BEGIN BLOCK 1      ***
C   COMPUTE COEFFICIENTS OF FORMULAS FOR THIS STEP.  AVOID COMPUTING
C   THOSE QUANTITIES NOT CHANGED WHEN STEP SIZE IS NOT CHANGED.
C                    ***
C
  100 KP1 = K+1
      KP2 = K+2
      KM1 = K-1
      KM2 = K-2
C
C   NS IS THE NUMBER OF STEPS TAKEN WITH SIZE H, INCLUDING THE CURRENT
C   ONE.  WHEN K.LT.NS, NO COEFFICIENTS CHANGE
C
      IF(H .NE. HOLD) NS = 0
      IF (NS .LE. KOLD) NS=NS+1
      NSP1 = NS+1
      IF (K .LT. NS) GO TO 199
C
C   COMPUTE THOSE COMPONENTS OF ALPHA(*),BETA(*),PSI(*),SIG(*) WHICH
C   ARE CHANGED
C
      BETA(NS) = 1.0
```

Figure A.5 Continued

```
      REALNS = NS
      ALPHA(NS) = 1.0/REALNS
      TEMP1 = H*REALNS
      SIG(NSP1) = 1.0
      IF(K .LT. NSP1) GO TO 110
      DO 105 I = NSP1,K
        IM1 = I-1
        TEMP2 = PSI(IM1)
        PSI(IM1) = TEMP1
        BETA(I) = BETA(IM1)*PSI(IM1)/TEMP2
        TEMP1 = TEMP2 + H
        ALPHA(I) = H/TEMP1
        REALI = I
  105   SIG(I+1) = REALI*ALPHA(I)*SIG(I)
  110 PSI(K) = TEMP1
C
C   COMPUTE COEFFICIENTS G(*)
C
C   INITIALIZE V(*) AND SET W(*).  G(2) IS SET IN DATA STATEMENT
C
      IF(NS .GT. 1) GO TO 120
      DO 115 IQ = 1,K
        TEMP3 = IQ*(IQ+1)
        V(IQ) = 1.0/TEMP3
  115   W(IQ) = V(IQ)
      GO TO 140
C
C   IF ORDER WAS RAISED, UPDATE DIAGONAL PART OF V(*)
C
  120 IF(K .LE. KOLD) GO TO 130
      TEMP4 = K*KP1
      V(K) = 1.0/TEMP4
      NSM2 = NS-2
      IF(NSM2 .LT. 1) GO TO 130
      DO 125 J = 1,NSM2
        I = K-J
  125   V(I) = V(I) - ALPHA(J+1)*V(I+1)
C
C   UPDATE V(*) AND SET W(*)
C
  130 LIMIT1 = KP1 - NS
      TEMP5 = ALPHA(NS)
      DO 135 IQ = 1,LIMIT1
        V(IQ) = V(IQ) - TEMP5*V(IQ+1)
  135   W(IQ) = V(IQ)
      G(NSP1) = W(1)
C
C   COMPUTE THE G(*) IN THE WORK VECTOR W(*)
C
  140 NSP2 = NS + 2
      IF(KP1 .LT. NSP2) GO TO 199
      DO 150 I = NSP2,KP1
        LIMIT2 = KP2 - I
        TEMP6 = ALPHA(I-1)
        DO 145 IQ = 1,LIMIT2
  145     W(IQ) = W(IQ) - TEMP6*W(IQ+1)
  150   G(I) = W(1)
  199   CONTINUE
C       ***      END BLOCK 1      ***
C
C       ***      BEGIN BLOCK 2      ***
C   PREDICT A SOLUTION P(*), EVALUATE DERIVATIVES USING PREDICTED
C   SOLUTION, ESTIMATE LOCAL ERROR AT ORDER K AND ERRORS AT ORDERS K,
C   K-1, K-2 AS IF CONSTANT STEP SIZE WERE USED.
C                    ***
C
C   CHANGE PHI TO PHI STAR
C
```

Figure A.5 Continued

```
      IF(K .LT. NSP1) GO TO 215
      DO 210 I = NSP1,K
        TEMP1 = BETA(I)
        DO 205 L = 1,NEQN
  205     PHI(L,I) = TEMP1*PHI(L,I)
  210   CONTINUE
C
C   PREDICT SOLUTION AND DIFFERENCES
C
  215 DO 220 L = 1,NEQN
        PHI(L,KP2) = PHI(L,KP1)
        PHI(L,KP1) = 0.0
  220   P(L) = 0.0
      DO 230 J = 1,K
        I = KP1 - J
        IP1 = I+1
        TEMP2 = G(I)
        DO 225 L = 1,NEQN
          P(L) = P(L) + TEMP2*PHI(L,I)
  225     PHI(L,I) = PHI(L,I) + PHI(L,IP1)
  230   CONTINUE
      IF(NORND) GO TO 240
      DO 235 L = 1,NEQN
        TAU = H*P(L) - PHI(L,15)
        P(L) = Y(L) + TAU
  235   PHI(L,16) = (P(L) - Y(L)) - TAU
      GO TO 250
  240 DO 245 L = 1,NEQN
  245   P(L) = Y(L) + H*P(L)
  250 XOLD = X
      X = X + H
      ABSH = ABS(H)
      CALL F(X,P,YP)
C
C   ESTIMATE ERRORS AT ORDERS K,K-1,K-2
C
      ERKM2 = 0.0
      ERKM1 = 0.0
      ERK = 0.0
      DO 265 L = 1,NEQN
        TEMP3 = 1.0/WT(L)
        TEMP4 = YP(L) - PHI(L,1)
        IF(KM2)265,260,255
  255   ERKM2 = ERKM2 + ((PHI(L,KM1)+TEMP4)*TEMP3)**2
  260   ERKM1 = ERKM1 + ((PHI(L,K)+TEMP4)*TEMP3)**2
  265   ERK = ERK + (TEMP4*TEMP3)**2
      IF(KM2)280,275,270
  270 ERKM2 = ABSH*SIG(KM1)*GSTR(KM2)*SQRT(ERKM2)
  275 ERKM1 = ABSH*SIG(K)*GSTR(KM1)*SQRT(ERKM1)
  280 TEMP5 = ABSH*SQRT(ERK)
      ERR = TEMP5*(G(K)-G(KP1))
      ERK = TEMP5*SIG(KP1)*GSTR(K)
      KNEW = K
C
C   TEST IF ORDER SHOULD BE LOWERED
C
      IF(KM2)299,290,285
  285 IF(AMAX1(ERKM1,ERKM2) .LE. ERK) KNEW = KM1
      GO TO 299
  290 IF(ERKM1 .LE. 0.5*ERK) KNEW = KM1
C
C   TEST IF STEP SUCCESSFUL
C
  299 IF(ERR .LE. EPS) GO TO 400
C       ***      END BLOCK 2      ***
C
```

Figure A.5 Continued

```
C        ***      BEGIN BLOCK 3      ***
C   THE STEP IS UNSUCCESSFUL.  RESTORE  X, PHI(*,*), PSI(*) .
C   IF THIRD CONSECUTIVE FAILURE, SET ORDER TO ONE.  IF STEP FAILS MORE
C   THAN THREE TIMES, CONSIDER AN OPTIMAL STEP SIZE.  DOUBLE ERROR
C   TOLERANCE AND RETURN IF ESTIMATED STEP SIZE IS TOO SMALL FOR MACHINE
C   PRECISION.
C                    ***
C
C   RESTORE X, PHI(*,*) AND PSI(*)
C
      PHASE1 = .FALSE.
      X = XOLD
      DO 310 I = 1,K
        TEMP1 = 1.0/BETA(I)
        IP1 = I+1
        DO 305 L = 1,NEQN
 305      PHI(L,I) = TEMP1*(PHI(L,I) - PHI(L,IP1))
 310    CONTINUE
      IF(K .LT. 2) GO TO 320
      DO 315 I = 2,K
 315    PSI(I-1) = PSI(I) - H
C
C   ON THIRD FAILURE, SET ORDER TO ONE.  THEREAFTER, USE OPTIMAL STEP
C   SIZE
C
 320 IFAIL = IFAIL + 1
      TEMP2 = 0.5
      IF(IFAIL - 3) 335,330,325
 325 IF(P5EPS .LT. 0.25*ERK) TEMP2 = SQRT(P5EPS/ERK)
 330 KNEW = 1
 335 H = TEMP2*H
      K = KNEW
      IF(ABS(H) .GE. FOURU*ABS(X)) GO TO 340
      CRASH = .TRUE.
      H = SIGN(FOURU*ABS(X),H)
      EPS = EPS + EPS
      RETURN
 340 GO TO 100
C        ***      END BLOCK 3      ***
C
C        ***      BEGIN BLOCK 4      ***
C   THE STEP IS SUCCESSFUL.  CORRECT THE PREDICTED SOLUTION, EVALUATE
C   THE DERIVATIVES USING THE CORRECTED SOLUTION AND UPDATE THE
C   DIFFERENCES.  DETERMINE BEST ORDER AND STEP SIZE FOR NEXT STEP.
C                    ***
 400 KOLD = K
      HOLD = H
C
C   CORRECT AND EVALUATE
C
      TEMP1 = H*G(KP1)
      IF(NORND) GO TO 410
      DO 405 L = 1,NEQN
        RHO = TEMP1*(YP(L) - PHI(L,1)) - PHI(L,16)
        Y(L) = P(L) + RHO
 405    PHI(L,15) = (Y(L) - P(L)) - RHO
      GO TO 420
 410 DO 415 L = 1,NEQN
 415    Y(L) = P(L) + TEMP1*(YP(L) - PHI(L,1))
 420 CALL F(X,Y,YP)
C
C   UPDATE DIFFERENCES FOR NEXT STEP
C
      DO 425 L = 1,NEQN
        PHI(L,KP1) = YP(L) - PHI(L,1)
 425    PHI(L,KP2) = PHI(L,KP1) - PHI(L,KP2)
      DO 435 I = 1,K
```

Figure A.5 Continued

```
      DO 430 L = 1,NEQN
 430     PHI(L,I) = PHI(L,I) + PHI(L,KP1)
 435   CONTINUE
C
C   ESTIMATE ERROR AT ORDER K+1 UNLESS:
C     IN FIRST PHASE WHEN ALWAYS RAISE ORDER,
C     ALREADY DECIDED TO LOWER ORDER,
C     STEP SIZE NOT CONSTANT SO ESTIMATE UNRELIABLE
C
      ERKP1 = 0.0
      IF(KNEW .EQ. KM1  .OR.  K .EQ. 12) PHASE1 = .FALSE.
      IF(PHASE1) GO TO 450
      IF(KNEW .EQ. KM1) GO TO 455
      IF(KP1 .GT. NS) GO TO 460
      DO 440 L = 1,NEQN
 440    ERKP1 = ERKP1 + (PHI(L,KP2)/WT(L))**2
      ERKP1 = ABSH*GSTR(KP1)*SQRT(ERKP1)
C
C   USING ESTIMATED ERROR AT ORDER K+1, DETERMINE APPROPRIATE ORDER
C   FOR NEXT STEP
C
      IF(K .GT. 1) GO TO 445
      IF(ERKP1 .GE. 0.5*ERK) GO TO 460
      GO TO 450
 445  IF(ERKM1 .LE. AMIN1(ERK,ERKP1)) GO TO 455
      IF(ERKP1 .GE. ERK  .OR.  K .EQ. 12) GO TO 460
C
C   HERE ERKP1 .LT. ERK .LT. AMAX1(ERKM1,ERKM2) ELSE ORDER WOULD HAVE
C   BEEN LOWERED IN BLOCK 2.  THUS ORDER IS TO BE RAISED
C
C   RAISE ORDER
C
 450  K = KP1
      ERK = ERKP1
      GO TO 460
C
C   LOWER ORDER
C
 455  K = KM1
      ERK = ERKM1
C
C   WITH NEW ORDER DETERMINE APPROPRIATE STEP SIZE FOR NEXT STEP
C
 460  HNEW = H + H
      IF(PHASE1) GO TO 465
      IF(P5EPS .GE. ERK*TWO(K+1)) GO TO 465
      HNEW = H
      IF(P5EPS .GE. ERK) GO TO 465
      TEMP2 = K+1
      R = (P5EPS/ERK)**(1.0/TEMP2)
      HNEW = ABSH*AMAX1(0.5,AMIN1(0.9,R))
      HNEW = SIGN(AMAX1(HNEW,FOURU*ABS(X)),H)
 465  H = HNEW
      RETURN
C       ***     END BLOCK 4     ***
      END
      SUBROUTINE INTRP(X,Y,XOUT,YOUT,YPOUT,NEQN,KOLD,PHI,PSI)
C
C   THE METHODS IN SUBROUTINE  STEP  APPROXIMATE THE SOLUTION NEAR  X
C   BY A POLYNOMIAL.  SUBROUTINE  INTRP  APPROXIMATES THE SOLUTION AT
C   XOUT  BY EVALUATING THE POLYNOMIAL THERE.  INFORMATION DEFINING THIS
C   POLYNOMIAL IS PASSED FROM  STEP  SO  INTRP  CANNOT BE USED ALONE.
C
C   THIS CODE IS COMPLETELY EXPLAINED AND DOCUMENTED IN THE TEXT,
C   COMPUTER SOLUTION OF ORDINARY DIFFERENTIAL EQUATIONS:  THE INITIAL
C   VALUE PROBLEM  BY L. F. SHAMPINE AND M. K. GORDON.
C
C   INPUT TO INTRP --
```

Figure A.5 Continued

```
C
C   THE USER PROVIDES STORAGE IN THE CALLING PROGRAM FOR THE ARRAYS IN
C   THE CALL LIST
       DIMENSION Y(NEQN),YOUT(NEQN),YPOUT(NEQN),PHI(NEQN,16),PSI(12)
C   AND DEFINES
C      XOUT -- POINT AT WHICH SOLUTION IS DESIRED.
C   THE REMAINING PARAMETERS ARE DEFINED IN  STEP  AND PASSED TO  INTRP
C   FROM THAT SUBROUTINE
C
C   OUTPUT FROM  INTRP --
C
C      YOUT(*) -- SOLUTION AT  XOUT
C      YPOUT(*) -- DERIVATIVE OF SOLUTION AT  XOUT
C   THE REMAINING PARAMETERS ARE RETURNED UNALTERED FROM THEIR INPUT
C   VALUES.  INTEGRATION WITH  STEP  MAY BE CONTINUED.
C
      DIMENSION G(13),W(13),RHO(13)
      DATA G(1)/1.0/,RHO(1)/1.0/
C
      HI = XOUT - X
      KI = KOLD + 1
      KIP1 = KI + 1
C
C   INITIALIZE W(*) FOR COMPUTING G(*)
C
      DO 5 I = 1,KI
        TEMP1 = I
    5   W(I) = 1.0/TEMP1
      TERM = 0.0
C
C   COMPUTE G(*)
C
      DO 15 J = 2,KI
        JM1 = J - 1
        PSIJM1 = PSI(JM1)
        GAMMA = (HI + TERM)/PSIJM1
        ETA = HI/PSIJM1
        LIMIT1 = KIP1 - J
        DO 10 I = 1,LIMIT1
   10     W(I) = GAMMA*W(I) - ETA*W(I+1)
        G(J) = W(1)
        RHO(J) = GAMMA*RHO(JM1)
   15   TERM = PSIJM1
C
C   INTERPOLATE
C
      DO 20 L = 1,NEQN
        YPOUT(L) = 0.0
   20   YOUT(L) = 0.0
      DO 30 J = 1,KI
        I = KIP1 - J
        TEMP2 = G(I)
        TEMP3 = RHO(I)
        DO 25 L = 1,NEQN
          YOUT(L) = YOUT(L) + TEMP2*PHI(L,I)
   25     YPOUT(L) = YPOUT(L) + TEMP3*PHI(L,I)
   30   CONTINUE
      DO 35 L = 1,NEQN
   35   YOUT(L) = Y(L) + HI*YOUT(L)
      RETURN
      END
```

Figure A.5 Continued