MODELING OF SOLAR DOMESTIC HOT WATER SYSTEMS

by

Paul Jeffery Schaefer

A thesis submitted in partial fulfillment

of the requirements for the degree of

Master of Science

(Mechanical Engineering)

at the

UNIVERSITY OF WISCONSIN-MADISON

1991

Abstract

Solar domestic hot water (SDHW) systems use solar energy to supplement conventional, such a gas or electric, house-hold hot water heating units. SDHW systems differ from other house-hold appliances in that the performance of SDHW systems is location dependent. Extrapolating upon short-term performance results and mathematical simulations are two techniques used to predict system long-term performance. The test and extrapolate techniques yield accurate results, but suffer from the time expenses associated with performing short-term tests. The accuracy of detailed computer simulations is strongly dependent upon the accuracy of the input system parameters. Computer simulations are less expensive to perform than short-term tests, but performing multiple detailed computer simulations may take a considerable amount of time.

"Compressed" weather is one method to decrease detailed simulation computational times. Compressed weather means representing a month with N statistically created days where N is less than the actual number of days in the month. TRNSYS [Klein et al., 1990] SDHW system simulations were performed over a range of system parameters, locations, and hot water load profiles. Yearly simulation results obtained using numerous compressed weather profiles as input are compared to results obtained using TRNSYS Type 54 weather input [Knight, 1988]. TRNSYS Type 54 statistically generates 365 days of weather consisting of a deterministic and random component. The compressed weather profiles consist of deterministic days. The possible daily combinations for one to four day series are explored.

Using one mean day to represent months with low to moderate monthly average clearness indexes and moderate to high critical radiation levels is found to be not appropriate, with relative errors being on the order of 10%. Increasing the number of days per month from one to two reduces errors which were on the order of 10% to 3%. Further increasing the number of days per month from two to three is typically found to additionally decrease errors by 0.5 to 1.5%, depending upon how the three days are ordered. However, the trend in improved performance with series length ends at three days, with comparisons indicating some four day series tending to perform worse than the three day ones.

Overall, a "1-3-2-4" ordered four day series, where 1 is the cloudiest day and 4 the sunniest day, is generally found to yield results closest to the Type 54 values as compared to the other compressed profiles investigated. The relative errors associated with using the four day series generally fall within ±1%. Larger errors are found to occur for systems having a high critical radiation ratio or very small solar tank volume. The "1-3-2-4" four day relative errors in delivered solar energy for these two cases are on the order of 3%. Initial condition effects on performance are found to be small. A savings in computational time by a factor of 10 is associated with using the compressed weather profile.

Acknowledgements

I thank my parents for their support and encouragement.

Table of Contents

List of Figures

List of Tables

Nomenclature

Roman Symbols

A	Collector area
A	Amplitude in Equation (5.9
b	Ordinate axis intercept
b_o	Incidence angle modifier constant
c_p	Specific heat
$c_{pcollector}$	Collector fluid specific heat
c_{ptank}	Tank fluid specific heat
$Diam_{pipe}$	Pipe inside diameter
F	Cumulative frequency
F_C	Radiation cumulative frequency associated with the critical radiation level
FL	Volumetric flow rate
F_rU_L	Collector loss coefficient
$F_r(\tau\alpha)_n$	Collector gain coefficient at normal incidence
$F_r\overline{(\tau\alpha)}$	Monthly average collector gain coefficient
H	Daily radiation on the horizontal surface
H_d	Daily diffuse radiation on the horizontal surface
H_o	Daily extraterrestrial radiation
$\overline{H}$	Monthly average daily radiation on the horizontal surface

H_T	Daily radiation on the tilted surface
$\overline{H}_T$	Monthly average daily radiation on the tilted surface
$H_{T,on}$	Total ASHRAE-95/SRCC test day radiation on the tilted surface during pump operation
ht_{aux}	Auxiliary tank inside height
ht_{solar}	Solar tank inside height
I	Hourly radiation on the horizontal surface
I_b	Hourly beam radiation on the horizontal surface
I_d	Hourly diffuse radiation on the horizontal surface
I_o	Hourly extraterrestrial radiation
I_{max}	Maximum radiation level
$\bar{I}$	Monthly average hourly radiation on the horizontal surface
I_T	Hourly radiation on the tilted surface
$\bar{I}_T$	Monthly average hourly radiation on the tilted surface
I_{TC}	Hourly critical radiation level
$\bar{I}_{TC}$	Monthly average hourly critical radiation level
K_s	Stratification constant
k_t	Hourly clearness index
k_{tm}	Monthly average hourly clearness index
K_t	Daily clearness index
$\overline{K}_t$	Monthly average daily clearness index
$k_{\tau\alpha}$	Incidence angle modifier
$\bar{k}_{\tau\alpha}$	Monthly average incidence angle modifier
$L_{pipe,\ in}$	Length of pipe leading from solar tank (or collector-tank heat exchanger) to collector

$L_{pipe,\ out}$	Length of pipe leading from collector to solar tank (or collector-tank heat exchanger)
m	Slope of a line
M	Mass
M_d	Mass of water withdrawn from system
$\dot{m}$	Mass flow rate
$\dot{m}_{collector}$	Collector mass flow rate
$\dot{m}_{heat}$	Solar side mass flow rate across storage tank
$\dot{m}_L$	Load mass flow rate
$\dot{m}_{tank}$	Collector side tank mass flow rate ($\dot{m}_{tank} = \dot{m}_{collector}$ if a collector-tank heat exchanger is *not* present)
n	Number of time steps
N	Number of hours in a month
N	Number of days in the month in Figure 5.5
P	Statistic defined by Equation (4.4)
q	Energy flow rate
Q	Energy flow
Q_{aux}	Auxiliary energy input
Q_{del}	Total (i.e., solar plus auxiliary) energy delivered by system
$\overline{Q}_{del}$	Monthly average daily total energy delivered by system
Q_{end}	Energy present in system at end of CSTG test day, but before load draw has occurred
Q_{in}	Collector (or collector-tank heat exchanger) loop energy gain across solar storage tank
Q_{net}	$Q_s - Q_{par}$
Q_{par}	Parasitic (i.e., pump) power
Q_s	Load draw energy gain across solar tank

Q_{start}	Energy present in system at start of CSTG test day
Q_{loss}	Tank heat loss
$Q_{loss_{aux}}$	Auxiliary tank heat loss
$Q_{loss_{solar}}$	Solar tank heat loss
Q_u	Energy gain across solar collector
$\tilde{Q}_u$	Monthly average hourly energy gain across solar collector
$\overline{Q}_u$	Monthly average daily energy gain across solar collector
$\overline{R}$	Ratio of the monthly tilted surface to horizontal surface radiation
r_d	Ratio of hourly diffuse to daily diffuse radiation on the horizontal surface
r_t	Ratio of the total hourly to total daily radiation on the horizontal surface
sf	Solar fraction
sf_o	sf axis intercept on a Klein & Fanney sf-ϕ*Y curve
sf_o'	Modified solar fraction as defined by Equation (2.8)
T	Temperature
ΔT	Temperature difference
T_{begin}	System temperature at beginning of CSTG test day
$T_{boil,\ tank}$	Boiling temperature of tank fluid
TC	Temperature measurement
$T_{c_{in}}$	Collector inlet temperature
$T_{c_{out}}$	Collector exit temperature
T_{c_∞}	Collector ambient temperature
$\tilde{T}_{c_\infty}$	Daytime average collector ambient temperature
$\overline{T}_{c_\infty}$	Monthly average daytime collector ambient temperature
T_d	Delivered (i.e., solar heated) water temperature

T_{final}	System temperature at end of CSTG test day after load draw has occurred
$\overline{T}_h$	Monthly average hourly ambient temperature
T_{heat}	Collector (or collector-tank heat exchanger) loop solar storage tank inlet temperature
T_i	Initial temperature
T_{in}	Inlet fluid temperature
$\overline{T}_m$	Monthly average daily ambient temperature
T_{mains}	Mains (supply) water temperature
T_{min}	Minimum useful temperature
T_{out}	Exit fluid temperature
T_{ret}	Collector (or collector-tank heat exchanger) loop solar storage tank exit temperature
$\widetilde{T}_s$	Temperature of water in the solar heated portion of the solar tank integrated over the period during which the collector pump is operating
$\overline{T}_s$	Monthly average daily system operating temperature
T_{set}	Set (i.e., solar plus auxiliary heated) water temperature
$\overline{T}_t$	Bulk average tank temperature
$\widehat{T}_t$	Time-averaged bulk-average tank temperature
T_{t_b}	Bottom node tank temperature
$\overline{T}_{t_b}$	Time averaged bottom node tank temperature
T_{t_t}	Top node tank temperature
$\overline{T}_{t_t}$	Time averaged top node tank temperature
T_{t_∞}	Tank ambient temperature
$\widehat{T}_{t_\infty}$	Nighttime average tank ambient temperature
T_o	Reference temperature

ΔU	Energy storage
U_{aux}	Auxiliary tank heat loss coefficient per unit area
U_{pipe}	Pipe heat loss coefficient per unit area
U_{solar}	Solar tank heat loss coefficient per unit area
UA	Heat loss-surface area product
V	Volume of water withdrawn from system
$V_{t_{aux}}$	Auxiliary tank volume
$V_{t_{solar}}$	Solar tank volume
w	Work
w_s	Sunset angle
x	Abscissa
X_C	Critical radiation ratio
$\overline{X}_C$	Monthly average critical radiation ratio
y	Ordinate
Y	Dimensionless variable defined by Equation (2.2)
$\overline{Y}$	Dimensionless variable defined by Equation (2.5)

Greak Symbols

$\alpha_o, \alpha_H, \alpha_T$	Regression coefficients
β	Collector slope
χ	Normally distributed variable
χ^2	Objective function defined by Equation (3.19)
ε	Heat exchanger effectiveness
ε	Randomly selected variable from a normal distribution

ϕ	Monthly average hourly utilizability
ϕ	ASHRAE-95/SRCC test day utilizability in Sections 2.2 and 3.4.2
ϕ_h	Hourly utilizability
$\bar{\phi}$	Monthly average daily utilizability
γ	Variable defined by Equation (2.12)
η	Fraction of a value ($0 \leq \eta \leq 1$)
θ	Incidence angle in Figures A.1 and A.2
θ	Time
$\Delta\theta$	Time increment
$\Delta\theta_{Night}$	Nighttime duration
$\Delta\theta_{on}$	Time of pump operation
$\Delta\theta_{tol}$	Integration period
ρ	Density
ρ_{tank}	Tank fluid density
ρ	Ground reflectance in Figures A.1, A.2, and A.3
σ	Standard deviation
σ_a	Standard deviation of the random k_t component
σ_m	Standard deviation of the monthly average daily temperature about the long-term monthly average value
σ_{yr}	Standard deviation of the monthly average daily temperatures about the yearly average daily temperature
$\bar{\xi}$	Vector of unknown system parameters

Chapter 1

Introduction

1.1 WHAT IS A SOLAR DOMESTIC HOT WATER SYSTEM?

Solar domestic hot water (SDHW) systems use solar energy to supplement conventional, such as gas or electric, house-hold hot water heating units. A collector array and storage tank connected by pipes along with the auxiliary heating source are the essential components of a SDHW system. In addition, pressure relief valves and a freeze protection scheme are incorporated into the systems. A tempering valve used to mix cold mains water with hot tap water should the hot water be at a temperature greater than that desired is also a common feature to many systems.

SDHW systems can be categorized into active and passive groups. Passive systems, such as thermosiphon, rely upon the density difference between heated and unheated fluid to cause fluid flow. In contrast, active hot water systems rely upon pumps to force fluid flow. Active solar hot water systems can be further grouped into three broad categories depending upon the freeze protection scheme employed by the system. "Antifreeze" systems circulate a glycol solution through the collector loop. A heat exchanger is used to transfer energy from the antifreeze to the drinkable water. Figure 1.1 is a schematic of how an antifreeze system may look. The use of antifreeze ensures against freezing, but also reduces system performance because of heat exchanger inefficiencies. Antifreeze systems are typically used at locations where freezing is frequent.

"Recirculation" systems circulate warm tank water through the collector array should the ambient temperature drop below freezing. However, recirculation of the warm fluid increases thermal losses from the system. Also, the system would be left unprotected from freezing if a power outage should occur, rendering the pumps inactive. Recirculation systems are usually used at locations where freezing is infrequent. Figure 1.2 is a schematic of how a recirculation system may look.

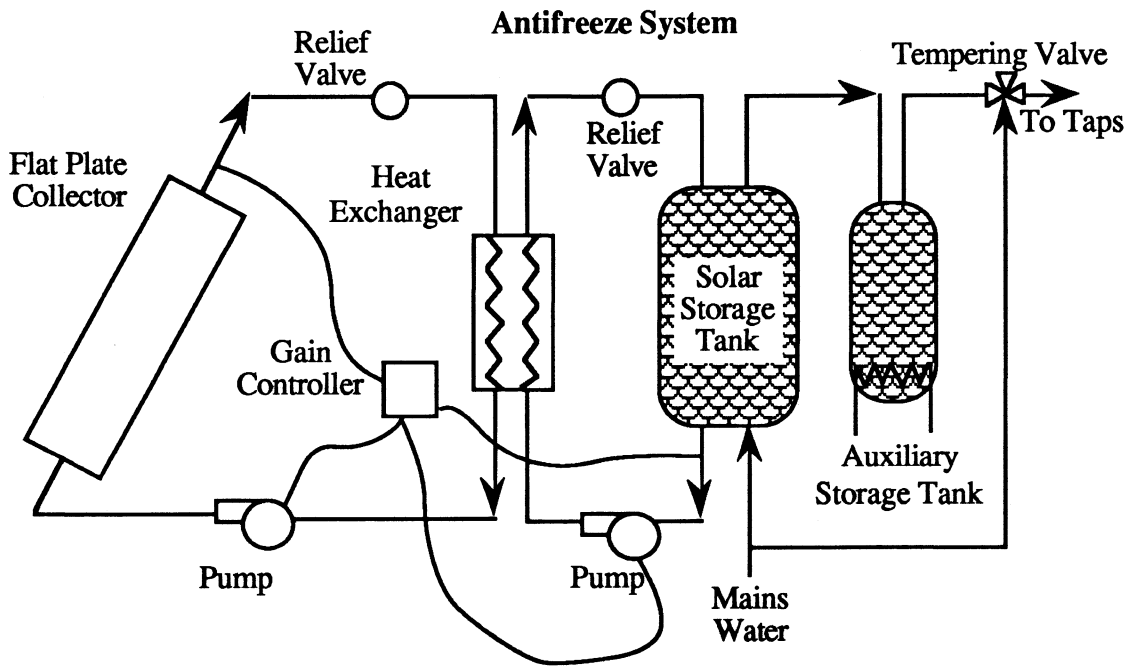


Figure 1.1 Typical Antifreeze SDHW System

"Drain-back" systems operate by allowing the fluid in the collector to drain into an indoor storage tank when the ambient drops below freezing conditions. Drainage is usually caused by a release of system pressure and gravity. Care must be taken when installing drain-back systems to ensure all of the fluid will drain out of the collector array. Also, attention must be paid to the pressure release valves such that they to not stick or freeze

shut. Drain-back systems are usually used at locations where freezing is infrequent. Figure 1.3 is a schematic of how a drain-back system may look.

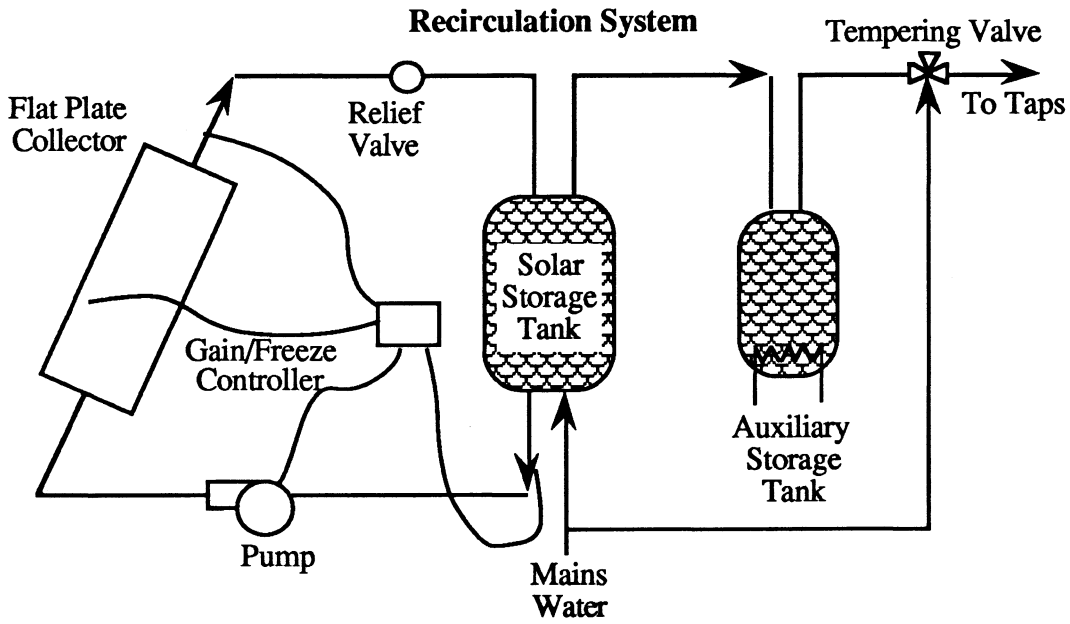


Figure 1.2 Typical Recirculation SDHW System

All of the SDHW systems, regardless of configuration, operate most efficiently when the fluid entering the collector is at a low temperature. Two tank systems add the auxiliary energy into a tank separate from that in which the solar heated water is stored. Adding the auxiliary into a separate tank insures none of the auxiliary energy will act to "pre-heat" water which is to be solar heated. However, two tank systems have the extra cost burden of an additional tank and additional piping. All of the SDHW systems shown in Figures 1.1 through 1.3 are two tank systems. Single tank systems usually add the auxiliary energy directly into the solar storage tank. The auxiliary energy in single tank systems is added towards the top of the tank in order to maintain a cool lower tank temperature.

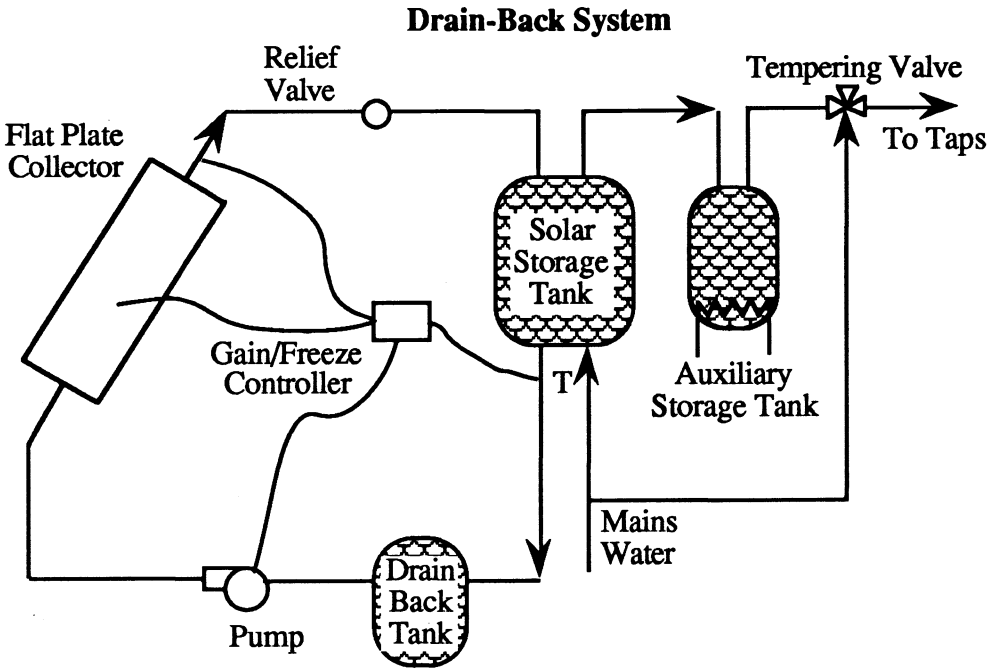


Figure 1.3 Typical Drain-Back SDHW System

1.2 CURRENT RATING METHOD

Currently SDHW systems are rated via the ASHRAE-95/SRCC short-term test guide lines [Wood, 1989]. The ASHRAE-95/SRCC test requires that a system be assembled and subjected to the 24 hour periodic radiation and incidence angle profiles shown in Figure 1.4. The collector and tank ambient temperatures are both constant at 22±2 °C at all times. A load draw occurs at 8:00 A.M., 12:00 noon and 5:00 P.M. each day. A draw continues until the following criterion is met:

$$\int_{time} \dot{m}_L * c_p \, (T_{set} - T_{mains}) \, d\theta = 14100 \text{ kJ} \qquad (1.1)$$

where

$\dot{m}_L$ = 0.20 kg/sec

T_{mains} = 22±1 °C

$T_{set} \geq 48.9\,°C$

The test continues until a 24 hour periodic steady-state condition has been reached, or four days have elapsed. A solar fraction (i.e., percentage of delivered energy which is not provided by the auxiliary energy source) is calculated and assigned to the system.

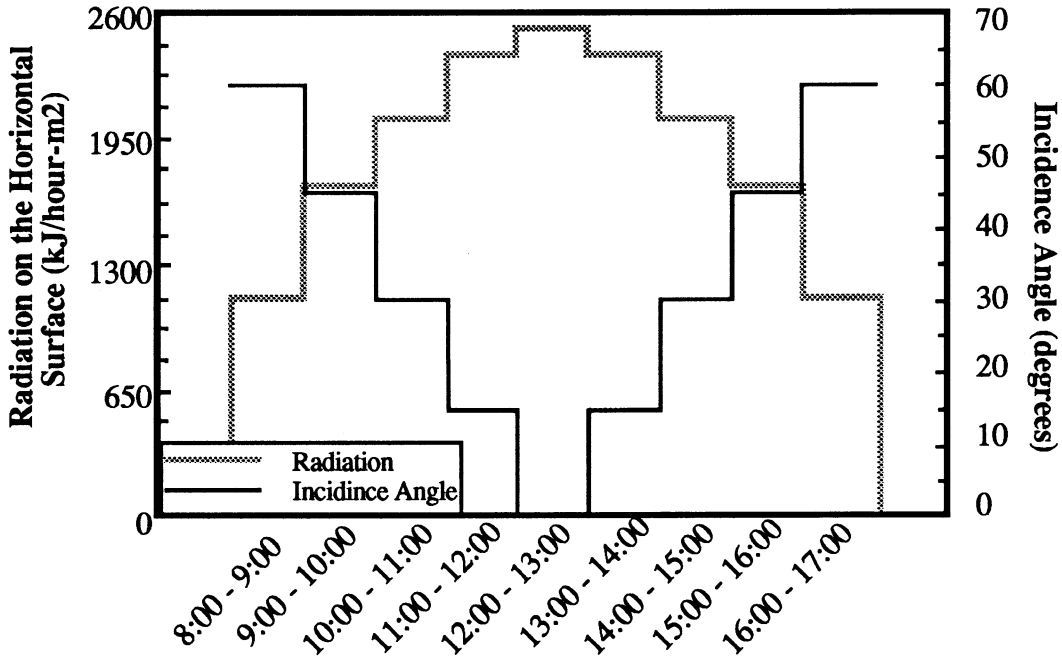


Figure 1.4 ASHRAE-95/SRCC Irradiance Profile

The ASHRAE-95/SRCC test procedure has some limitations. First of all, the performing of short term tests is expensive in terms of both monetary and time considerations. Secondly, the ASHRAE-95/SRCC short-term test results are not necessarily indicative of how the system will perform when subjected to actual climatic and load conditions. Furthermore, the short-term test results cannot be used to compare the relative merits of different systems because of differences between the ASHRAE-95/SRCC short-term and actual long-term system performances.

1.3 PURPOSE OF THIS STUDY

The purpose of this study is to recommend a method for evaluating the long-term performance of SDHW systems subjected to typical climatic and load conditions. Solar systems differ from other house hold appliances (such as refrigerators, freezers, ovens, etc...) in that the performance of a solar system is location dependent. The ability to accurately show how the system performance will vary with location within reasonable time and monetary constraints is the basic criteria for a suitable performance prediction method. Figure 1.5 shows two possible "paths" which may be taken to arrive at the long-term performance of a SDHW system. The "testing" path extrapolates upon short-term system performance results to predict the long-term system performance. The Klein and Fanney method [1983] and The European Collector and System Testing Group (CSTG) method [Bourges et al., 1990] are two "test and extrapolate" long-term performance prediction procedures. The test and extrapolated methods are reasonably accurate, but suffer from the time and monetary expenses associated with performing short-term tests.

The other possibility for obtaining the long-term performance results involves following the "simulation" path. Obtaining the system parameters and then using these parameters as input to a simulation routine are the two steps encountered along the simulation path. The system parameters may be obtained via individual component knowledge, or the Minnerly [1989], Buckles [1983], or Spirkl [1990]methods. Possible simulation methods are performing detailed (such as TRNSYS [Klein et al., 1990]) simulations, using monthly average hourly utilizability, f-Chart [Klein, 1976], or $\bar{\phi}$, f-Chart [Beckman et al., 1976]. Simulations can be accurate and do not involve the monetary and time expense associated with short-term tests. The focus of this thesis pertains to the simulation path.

Chapter 2 briefly discusses the Klein and Fanney and the European CSTG test and extrapolate methods. Chapter 3 discusses the methods for obtaining system parameters and simulating listed in Figure 1.5. Chapter 4 presents comparisons between experimental and TRNSYS simulated ASHRAE-95/SRCC short-term tests. Chapter 5 presents a simulation method based upon detailed simulation techniques which meets the criteria for a suitable long-term performance prediction method. Chapter 6 presents the conclusions of this thesis study, and outlines possible future work.

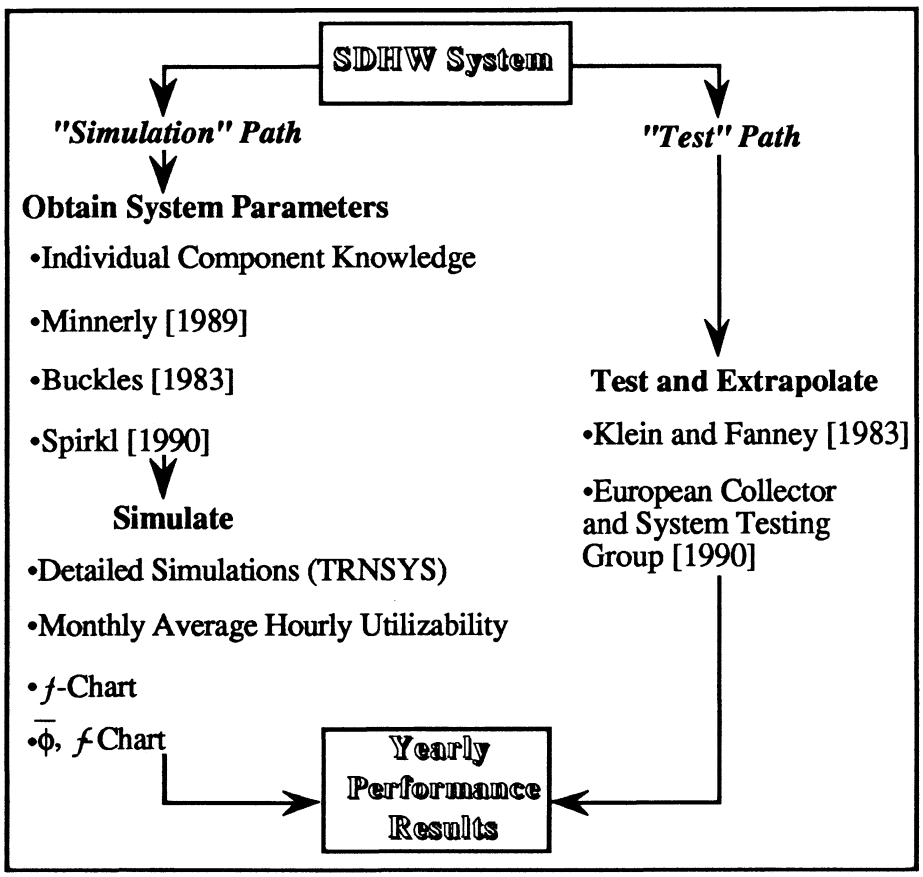


Figure 1.5 Possible Means of Obtaining SDHW System Long-Term Performance Results

Chapter 2

Review of Two Test and Extrapolate Long-Term Performance Prediction Methods

2.1 INTRODUCTION

Long-term performance prediction methods may be categorized into two groups. One group of methods uses information about the system parameters along with computer simulations to predict the long-term performance. The other group of methods extrapolates upon short-term performance results to predict the long-term performance. The Klein and Fanney [1983] and the European Collector and System Testing Group (CSTG) [Bourges et al., 1990] methods are two procedures which fall under the later category. A brief explanation of the Klein and Fanney and the European CSTG methods is presented in this chapter.

2.2 KLEIN & FANNEY

The Klein and Fanney performance prediction method utilizes data from possibly two, but preferably three or more, ASHRAE-95/SRCC short-term tests to determine a system operating curve. The system long-term performance is subsequently calculated from the operating curve.

Klein and Fanney observed a linear relationship between the ASHRAE-95/SRCC test day solar fraction, sf, and the ϕ*Y product where ϕ is the test-day utilizability and Y is a dimensionless number. In equation form:

$$sf = 1 - \frac{Q_{aux} + Q_{loss}}{Q_{del}} \tag{2.1}$$

where

Q_{aux} = Required auxiliary energy integrated over the test day
Q_{loss} = Solar storage tank heat loss integrated over the test day
Q_{del} = Hot water load integrated over the test day

Y is defined as:

$$Y = \frac{A\, F_r(\tau\alpha)_n \sum^{Day} \left[I_T\, k_{\tau\alpha}\, \Delta\theta\right]}{Q_{del}} \tag{2.2}$$

where

A = Collector area
$F_r(\tau\alpha)_n$ = Collector gain coefficient at normal incidence
I_T = Tilted surface radiation
$k_{\tau\alpha}$ = Incidence angle modifier

The third parameter, ϕ, is calculated via:

$$\phi = \frac{\sum^{Day} \left[I_T - I_{TC}\right]}{\sum^{Day} I_T} \tag{2.3}$$

The variable I_{TC} in Equation (2.3) is the critical radiation level and defined as:

$$I_{TC} = \frac{F_r U_L \left[\tilde{T}_s - \tilde{T}_{c_\infty}\right]}{F_r(\tau\alpha)_n} \qquad (2.4)$$

where

F_rU_L = Collector loss coefficient

$\tilde{T}_s$ = Average temperature of water in the solar heated portion of the solar tank integrated over the period during which the collector pump is operating

$\tilde{T}_{c_\infty}$ = Daytime average collector temperature

A hypothetical sf - ϕ*Y curve is shown in Figure 2.1.

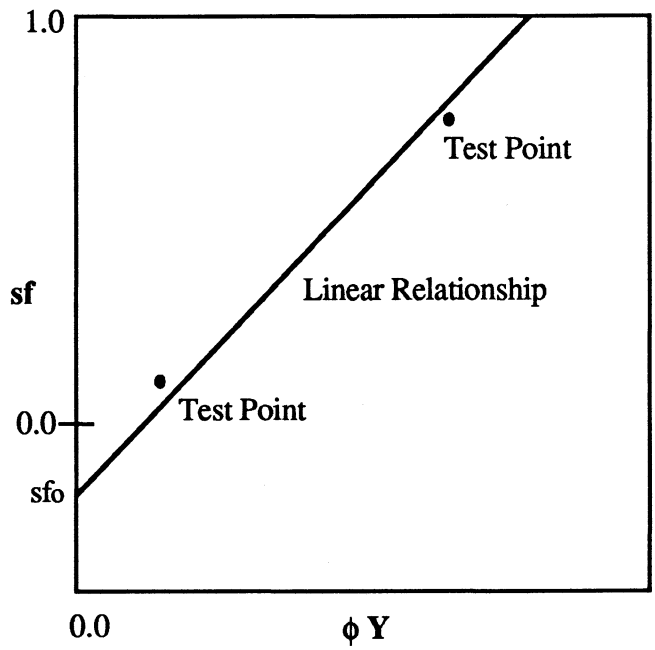


Figure 2.1 Hypothetical sf - ϕ*Y Curve

Klein and Fanney observed the monthly solar fraction is approximately equal to the test day solar fraction obtained from the performance curve, if ϕ and Y are replaced by their

monthly average daily values, $\bar{\phi}$ and $\bar{Y}$, and the monthly average daily hot water load, $\bar{Q}_{del}$, equals to the test day load. Modifications to the test day solar fraction as outlined by Klein and Fanney are required if the monthly average daily hot water load does not equal the test day load.

$\bar{Y}$ is defined as:

$$\bar{Y} = \frac{A\, F_r(\tau\alpha)_n\, \bar{H}\, \bar{R}\, \bar{k}_{\tau\alpha}}{\bar{Q}_{del}} \tag{2.5}$$

where

$\bar{H}$ = Monthly average daily radiation per unit area on the horizontal surface

$\bar{R}$ = Ratio of the monthly integrated tilted surface to horizontal surface radiation

$\bar{k}_{\tau\alpha}$ = Monthly average incidence angle modifier

Methods are outlined in Duffie and Beckman [1980] for determining $\bar{\phi}$ if the monthly average daily critical radiation level, $\bar{I}_{TC}$, is known. $\bar{I}_{TC}$ is defined as:

$$\bar{I}_{TC} = \frac{F_r U_L \left[\bar{T}_s - \bar{T}_{c_\infty}\right]}{F_r(\tau\alpha)_n} \tag{2.6}$$

where

$\bar{T}_s$ = Monthly average daily system operating temperature

$\bar{T}_{c_\infty}$ = Monthly average daytime collector ambient temperature

Klein and Fanney related $\bar{T}_s$ to sf via the following empirical correlation:

$$\frac{\bar{T}_s - T_{mains}}{T_{set} - T_{mains}} = 0.688\left[sf - sf'_o\right] + 0.201\left[sf - sf'_o\right]^2 \qquad (2.7)$$

where

$$sf'_o = sf_o \frac{Q_{del}}{\bar{Q}_{del}} \qquad (2.8)$$

sf_o = sf axis intercept of the sf - ϕ*Y curve

The above equations are solved by an iterative process for the monthly solar fraction.

Klein and Fanney compared the predicted performance of an active SDHW system as calculated by the algorithm outlined above to the actual long-term system performance. The actual system performance was monitored and evaluated by the National Bureau of Standards. An absolute error of 2.2% (4.9% relative error) was observed between the predicted and actual yearly solar fractions.

2.3 CSTG

The rating method discussed below is designed to coincide with the SDHW short-term test specified by The European Collector and System Testing Group (CSTG). A linear regression analysis is performed on data obtained from several CSTG short-term tests. The system parameters obtained from the regression analysis are subsequently used in a recursive relationship from which the long-term system performance is calculated.

The CSTG short-term test begins by circulating fluid at a constant temperature, T_{mains}, throughout the system until a uniform temperature is reached. The system is allowed to function normally from 6:00 A.M. to 6:00 P.M. solar time with the daily integrated solar radiation on the collector, H_T, being recorded. Three tank volumes of water are withdrawn at a constant flow rate of 0.6 m^3/hr at 6:00 P.M., with the tank being replenished with mains water at a temperature T_{mains}. The draw temperature is monitored

as a function of the volume withdrawn. A pictorial representation of the testing steps and a typical draw profile are shown in Figures 2.2 and 2.3, respectively.

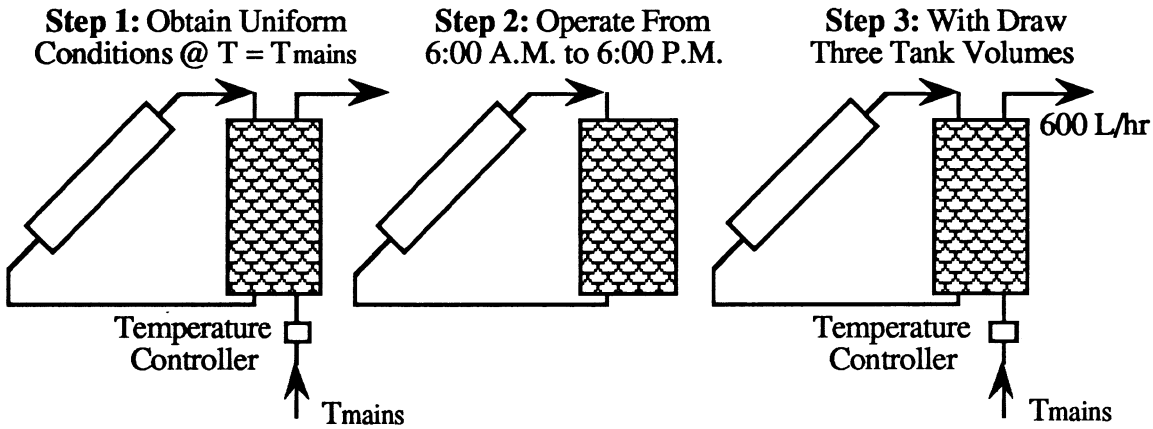


Figure 2.2 European CSTG Daily SDHW System Test Procedure

Six, but preferably nine or more, tests are performed. One pair (or preferably a trio) of tests are subjected to a different integrated radiation value as compared to the others, with all else being the same. Similarly, the average collector ambient temperature is different for a second pair (or preferably trio) of tests as compared to the others, with all else being the same. The following model is subsequently fitted to the test results via a linear regression analysis:

$$Q_u = \left[\alpha_o + \alpha_H H_T + \alpha_T \left(\tilde{T}_{c_\infty} - T_{begin}\right)\right] \tag{2.9}$$

where

$\alpha o, \alpha_H, \alpha_T$ = Regression coefficients

H_T = Daily integrated radiation on the tilted surface

T_{begin} = System temperature at the beginning of the test day

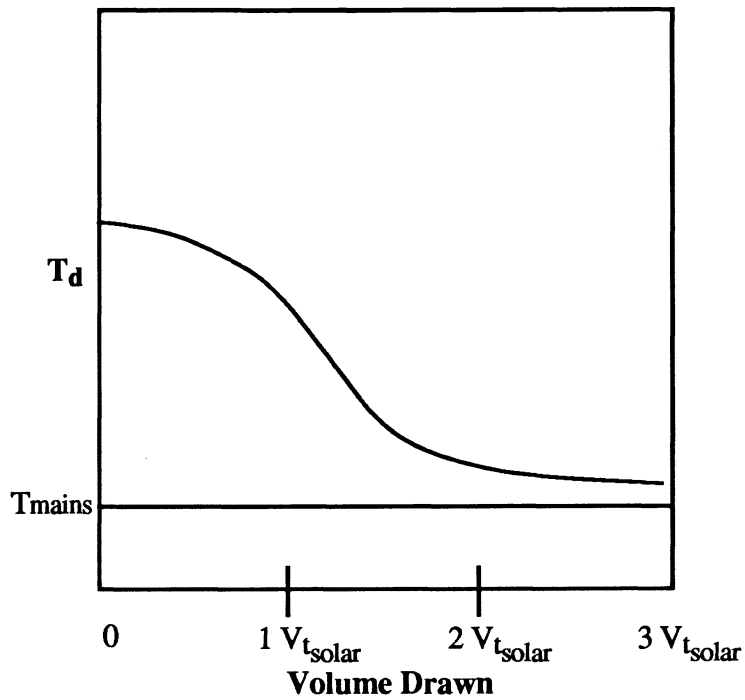


Figure 2.3 Hypothetical Load Draw Temperature Profile

The CSTG have developed a recursive relationship for the daily useful energy gain based upon the following rationale. The energy in the tank at the beginning of day i is:

$$Q_{start_i} = V_{t_{solar}} \rho c_p [T_{begin_i} - T_{mains_i}] \qquad (2.10)$$

where

$V_{t_{solar}}$ = Tank volume
ρ = Density of water
c_p = Specific heat of water

The energy in the tank at the end of the day is the sum of the energy present at the start of the day, Equation (2.10), and the useful energy collected during the day, Equation (2.9):

$$Q_{end_i} = V_{t_{solar}} \rho c_p [T_{begin_i} - T_{mains_i}] + \left[\alpha_o + \alpha_H H_{T_i} + \alpha_T \left(\tilde{T}_{c_{\infty,i}} - T_{begin_i}\right)\right] \quad (2.11)$$

It is assumed the amount $[1 - \gamma_i]^* Q_{end_i}$ of energy is withdrawn from the storage tank for use at the end of day i. γ_i is defined as the maximum of the following two quantities:

$$\gamma_i = MAX\left[\left(1 - \int_0^{V_{del,i}} \left[\frac{T_{d_i} - T_{mains_i}}{T_{d_i}(V=0) - T_{mains_i}}\right] \frac{dV}{V_{t_{solar}}}\right), \left(1 - \frac{Q_{del_i}}{Q_{u_i}}\right)\right] \quad (2.12)$$

where

T_d = Delivered water temperature

Quantitatively, the first term in Equation (2.12) is the delivered solar energy when tempering has not taken place during the draw. The second term is an approximation for the delivered solar energy when tempering has occurred during the draw period. The tank temperature after the energy draw, T_{final_i}, is calculated from an energy balance to be:

$$T_{final_i} = T_{mains_i} + \frac{Q_{end_i} \gamma_i}{V_{t_{solar}} \rho c_p} \quad (2.13)$$

An energy balance on the tank during the night time is approximated as:

$$V_{t_{solar}} \rho c_p [T_{final_i} - T_{begin_{i+1}}] = UA \, \Delta\theta_{Night} \left[\left(\frac{T_{final_i} - T_{begin_{i+1}}}{2}\right) - \hat{T}_{t_{\infty,i}}\right] \quad (2.14)$$

where

$\hat{T}_{t_\infty}$ = Nighttime average tank ambient temperature

$\Delta\theta_{Night}$ = Nighttime duration

UA = Tank heat loss coefficient-surface are product and is determined by performing a "cool down" test one the tank.

Finally, the useful energy gain for day i+1 is simply:

$$Qu_{i+1} = \left[\alpha_o + \alpha_H H_{Ti+1} + \alpha_T \left(\tilde{T}_{c_{\infty}, i+1} - Tbegin_{i+1}\right)\right] \tag{2.15}$$

Combining Equations (2.11), (2.12), (2.13), (2.14) and (2.15) results in the following recursive relationship for the daily useful energy gain:

$$Qu_{i+1} = V_{tsolar}\,\rho\,c_p \left\{ r\left[T_{mains_i} + \frac{Qu_i\,\gamma_i}{V_{tsolar}\,\rho\,c_p}\right] + [1 - r]\,\hat{T}_{t_{\infty}, i} - T_{mains_i} \right\} + \tag{2.16}$$

$$\left\{ \alpha_o + \alpha_H H_{Ti+1} + \alpha_T \left[\tilde{T}_{c_{\infty}, i+1} - r\left(T_{mains_i} + \frac{Qu_i\,\gamma_i}{V_{tsolar}\,\rho\,c_p} \right) - (1 - r)\,\hat{T}_{t_{\infty}, i} \right] \right\}$$

where

$$r = \frac{V_{tsolar}\,\rho\,c_p - \dfrac{UA\,\Delta\theta_{Night}}{2}}{V_{tsolar}\,\rho\,c_p + \dfrac{UA\,\Delta\theta_{Night}}{2}}$$

Bourges et al. [1990] performed an error analysis concerning the CSTG testing and long-term performance prediction procedure. Bourges et al. concluded standard deviations for long-term performance predictions are on the order of 5% for sunny climates and 10% for cloudy climates.

2.4 CONCLUSIONS

Both the Klein and Fanney and the European CSTG test and extrapolate long-term performance prediction methods show reasonable accuracy. The test and extrapolate methods do, however, suffer from the time and monetary costs associated with performing short-term system tests.

Chapter 3

Review of Simulation Long-Term Performance Prediction Methods

3.1 INTRODUCTION

SDHW system simulations involve two steps. The first step is to gather information about the system upon which a model may be constructed. Typical information required by existing modeling packages are tank and pipe heat loss coefficients, collector $F_r(\tau\alpha)_n$ and F_rU_L, fluid flow rates, etc. The second step is to input the required information into a simulation routine. Execution of the routine will yield results indicative of the actual long-term performance.

This chapter is divided into two sections. The first half, section 3.2 through 3.3, briefly discusses some common long-term performance simulation methods. The second half of the chapter, section 3.3, reviews different methods for determining the values of the system parameters.

3.2 DETAILED SIMULATION METHODS

Detailed simulations solve system mass and energy balances at times θ_o, $\theta_o+\Delta\theta$, $\theta_o+2\Delta\theta$, $\theta_o+3\Delta\theta$, ...$\theta_o+n\Delta\theta$, ...where θ_o is the initial time and $\Delta\theta$ is the time step. The conditions at the end of a time step are used as the initial conditions for the next time step when differential equations need to be solved. The time step, $\Delta\theta$, must be sufficiently

small to ensure numerical stability. The mass of water in the solar tank divided by the tank-loop flow rate when a collector-tank heat exchanger is present, or collector flow rate when a collector-tank heat exchanger is not present, is a rough estimate as to the critical time step. Hourly weather (i.e., data spaced one hour apart in time) is used as the input to the system, although weather data at intervals less than an hour may be used if available. Interpolation of the weather data is required if the time step, $\Delta\theta$, is less than the increment between weather data values.

Two approaches exist concerning the implementation of detailed simulations. One idea involves modeling each component of the actual system with a computer subroutine. The separate subroutines are linked together by a main program to compose the complete system model. The advantage of creating separate subroutines is that over time a "library" of various models may be developed. The various models allows for the creation of a wide assortment of configurations and systems. For example, a pump model used in a SDHW system simulation can also be used in a refrigeration cycle simulation.

The other detailed simulation approach is to use one "combined" subroutine to model a particular (such as an active SDHW) configuration. The combined system incorporates the mass and energy balances for each of the system components into one subroutine. A reduction in computation effort is the advantage to using a combined subroutine over several individual component routines. Combined routines are designed for a specific system and therefore can be designed for computational speed rather than generality, resulting in reduced computational times. SDHW systems in particular can be "sped up" by using a combined system. Methods are outlined in Duffie and Beckman [1980] for modifying the collector parameters, $F_r(\tau\alpha)_n$ and F_rU_L, to account for the effects of collector-loop pipe heat losses and collector-tank heat exchanger inefficiencies.

Modifying the collector allows for the removal of the pipe and heat exchanger equations from the system of coupled governing equations, decreasing computational effort.

The collector modification scheme is limited, however, in that it does not account for pipe water capacitance effects. Simulations were performed to investigate the effects of pipe capacitance on system performance. The simulations use individual TRNSYS types [Klein et al., 1990] to model the system components and are subjected to Madison, WI TMY data. The system parameters are equivalent to the CSU test #7 system to be discussed in Chapter 4 with a few exceptions. The system is subjected to a 200 L/day RAND load [Mutch, 1974] (not the ASHRAE-95/SRCC load profile), has a collector slope of 43°, the incidence angle modifier constant is 0.1, the mains temperature is 10 °C (not 22 °C), and the parasitic power contribution is assumed to be negligible. The pipe lengths, diameter, and insulation conductivity are listed in Table 3.1. Two simulations were performed using the system. One simulation considered the effect of pipe water capacitance, where as the other neglected the pipe water capacitance effects. The simulation considering capacitance was found to require 1.95% more auxiliary energy over the year.

Pipe Parameter	Value
Length from tank to collector	11.8 m
Length from collector to tank	13.4 m
Diameter	3/4 inch {0.01905 m}
Insulation Conductivity	1.21 W/(m^2-°C)

Table 3.1 CSU SDHW System Pipe Parameters

Yearly simulations were also performed using a TRNSYS-compatible combined model and an equivalent system model formed of individual TRNSYS Types. The combined model takes into account pipe and heat exchanger effects by modifying the collector parameters, and therefore does not consider pipe capacitance effects.

A listing of the combined model subroutine is given in Appendix A. The combined model was repeatedly found to perform 3.7 times faster than the individual component TRNSYS model. It is felt the benefit of a decreased computational time out ways the disadvantages associated with neglecting pipe capacitance.

3.3 SHORT-CUT SIMULATION METHODS

The computational time and expertise associated with detailed system modeling has lead to the development of simulation "short-cut" (or design) methods. Three commonly used SDHW simulation short-cut methods are discussed in this section.

3.3.1 f-Chart

One short-cut simulation method is the *f*-Chart empirical correlations of Klein [1976]. Klein correlated the results of many detailed SDHW system simulations to two dimensionless variables. The dimensionless variables are functions of selected system parameters and monthly average daily weather data. The *f*-Chart relationships are easily programmable, and have been incorporated into the F-CHART computer program [Klein and Beckman, 1988].

Duffie and Mitchell [1983] compared *f*-Chart results to measured annual or seasonal performances and generally found absolute errors in the range of ±3%. *f*-Chart does tend to under predict the performance of highly stratified systems due to the assumption of a fully mixed solar tank used in the detailed simulations from which the *f*-Chart correlations are based. Also, as with any empirical correlation, the SDHW *f*-Chart relations are limited to the range of dimensionless values for which use was intended.

3.3.2 Utilizability

Utilizability is defined as the fraction of the total radiation received which as at a greater intensity than some critical level. In equation form, the hourly utilizability, ϕ_h, is defined as:

$$\phi_h = \frac{(I_T - I_{TC})^+}{I_T} \tag{3.1}$$

where

I_T = Tilted surface radiation
I_{TC} = Critical radiation level.

In Figure 3.1, the utilizability for the hour in question is the ratio of the cross-hatched area to the cross-hatched plus solid areas.

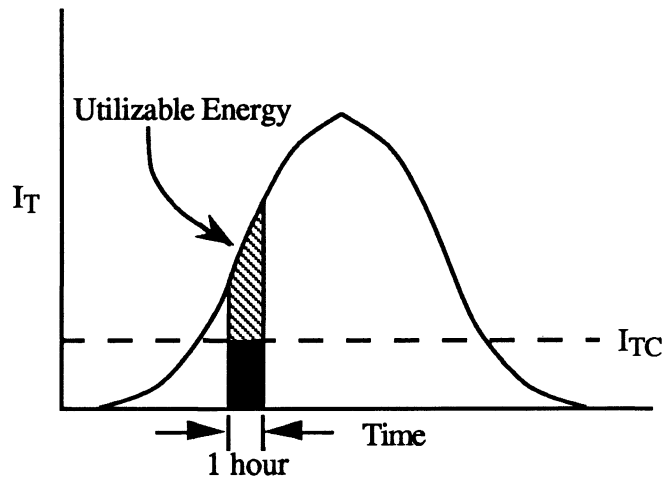


Figure 3.1 Visual Representation of Hourly Utilizability

The utilizability for a particular hour in a month (i.e., the monthly average hourly utilizability) is defined as:

$$\phi = \frac{1}{N} \sum^{N} \frac{(I_T - I_{TC})^+}{\bar{I}_T} \tag{3.2}$$

where

N = Number of hours in the month

$\bar{I}_T$ = Monthly average hourly tilted surface radiation.

In theory, the monthly average hourly utilizability, ϕ, is calculated from the cumulative frequency distribution of hourly radiation values. A hypothetical hourly-radiation cumulative frequency distribution is shown in Figure 3.2. The utilizability is calculated as:

$$\phi = \int_{F_C}^{1} \phi_h \, dF \tag{3.3}$$

where

F_C = The cumulative frequency associated with the critical radiation ratio, X_C

$$X_C = \frac{I_{TC}}{\bar{I}_T}$$

In other words, ϕ is the cross-hatched region in Figure 3.2. Performing the integration in Equation (3.3) from various lower limits (i.e., from various F_C values) results in the relationship pictured in Figure 3.3. The monthly average utilizable energy for the hour in question given a critical radiation ratio is simply $\bar{I}_T \phi$. The monthly average useful energy gain across the collector for the hour is:

$$\tilde{Q}_u = A F_r(\tau\alpha)_n k_{\tau\alpha} \bar{I}_T \phi \tag{3.4}$$

Monthly average hourly utilizability reduces the calculations required to simulate a month from individual hourly calculations to monthly average hourly calculations.

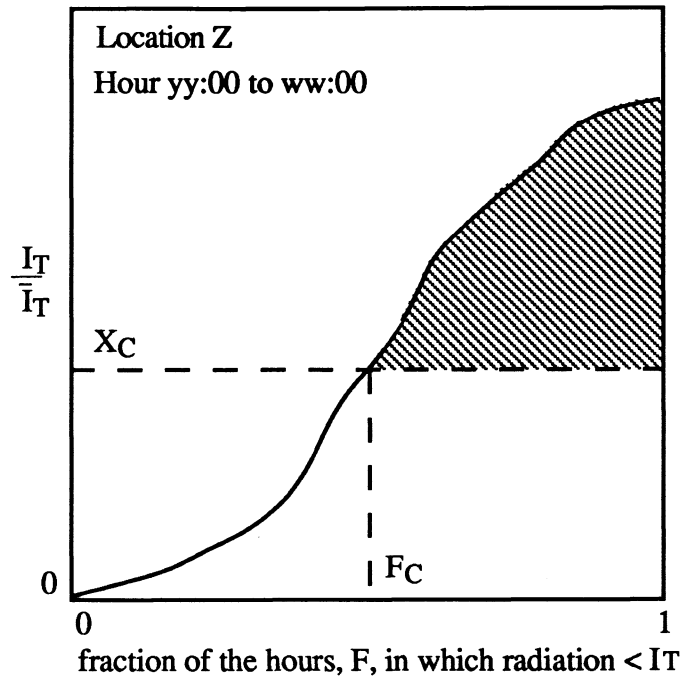


Figure 3.2 Hypothetical Hourly-Radiation Cumulative Frequency Distribution

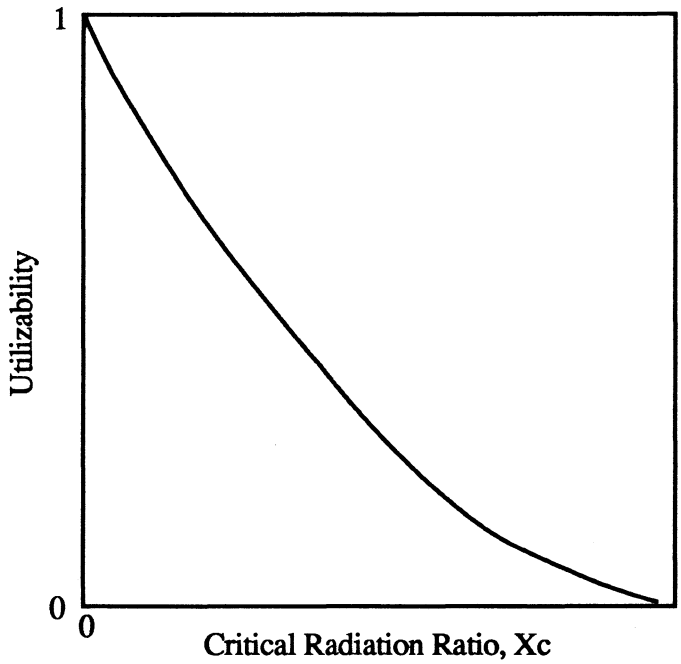


Figure 3.3 Hypothetical ϕ Curve

The development of generalized utilizability has greatly simplified the calculation process outlined above. The relationships of generalized utilizability allow for the prediction of the utilizable energy at a critical radiation level from knowledge of long-term average radiation data rather than from Equation (3.3). For example, the Clark et al. method [1983] is an algorithm for calculating the monthly average hourly utilizability from knowledge of monthly average hourly weather values. The ease in which it can be implemented into a simulation routine is the advantage of general utilizability over Equation (3.3).

The use of monthly average hourly utilizability is limited, however in that hourly utilizability assumes the critical radiation level is constant for an hour throughout the month. For example, hourly utilizability assumes the critical radiation level for the hour 10:00-11:00 A.M. on May 1 is the same as that from 10:00-11:00 A.M. on May 2, 3, 4, etc. Some variation in the critical radiation level is inherent, however, mainly because of a changing collector inlet temperature as a result of a finite storage capacity. TRNSYS simulations were performed in order to obtain an indication of the effect of storage capacity on monthly average hourly utilizability results. Figure 3.4 presents the results for two of the four yearly simulations. The black bars represent the monthly solar fractions using Madison, WI TMY radiation data. The collector ambient temperature is constant at 10 °C at all times. The solar tank volume is 300 L, and the system is subjected to a 300 L/day RAND load profile.

The gray bars are results for the same system subjected to the same conditions with one exception. The second calculations use monthly average radiation data and Equation (3.4) to calculate the useful energy gain rather than TMY radiation data and the TRNSYS Type 1 flat plate collector model. The utilizability months are repeated until a periodic steady-state condition is achieved in order to eliminate the effects of initial conditions on

monthly results. The average relative error in solar fraction between the TMY and utilizability months is -3.52%, with a maximum error of -9.08% occurring for February and a minimum of +0.11% in March.

Figure 3.5 is the analogous plot to Figure 3.4 for two simulations in which the tank and load have been reduced to 100 L and 100 L/day, respectively. In addition, the collector area has been reduced from 4.5 m^2 to 3.0 m^2 to maintain reasonable monthly solar fractions. The average relative error in solar fraction for the case of a reduced tank capacitance has increased to -6.16% with a maximum value in February of -12.07% and a minimum value in March of -2.25%. Hence, a reduction in storage capacity by a factor of three roughly doubles the monthly average hourly utilizability performance prediction error for this particular SDHW system.

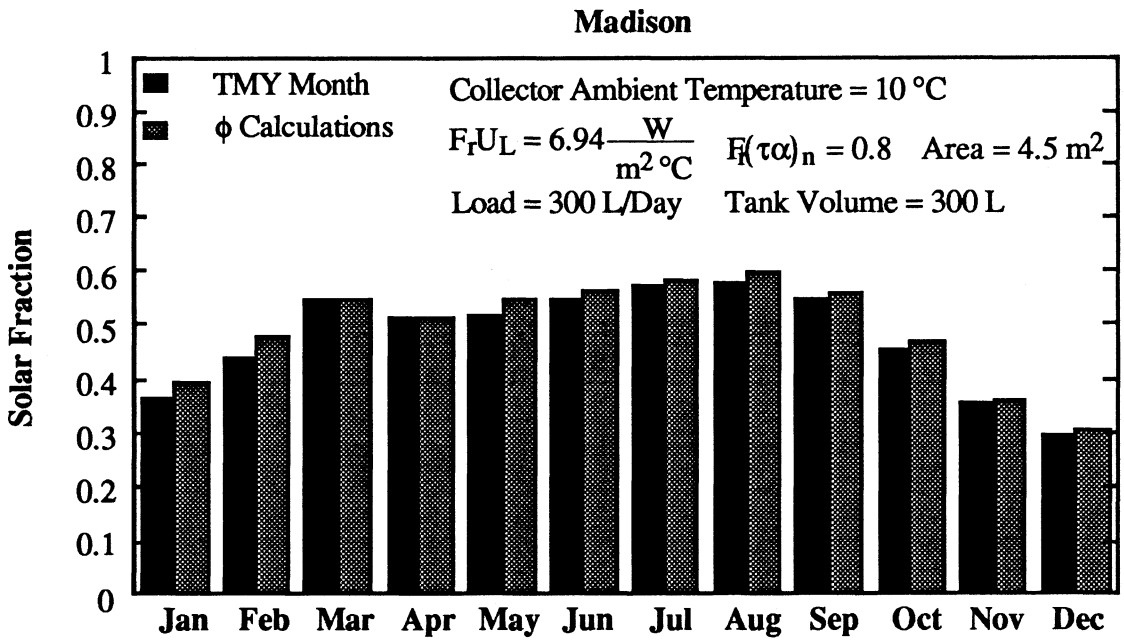


Figure 3.4 Comparison Between TMY and Monthly Average Hourly Utilizability Simulation Results

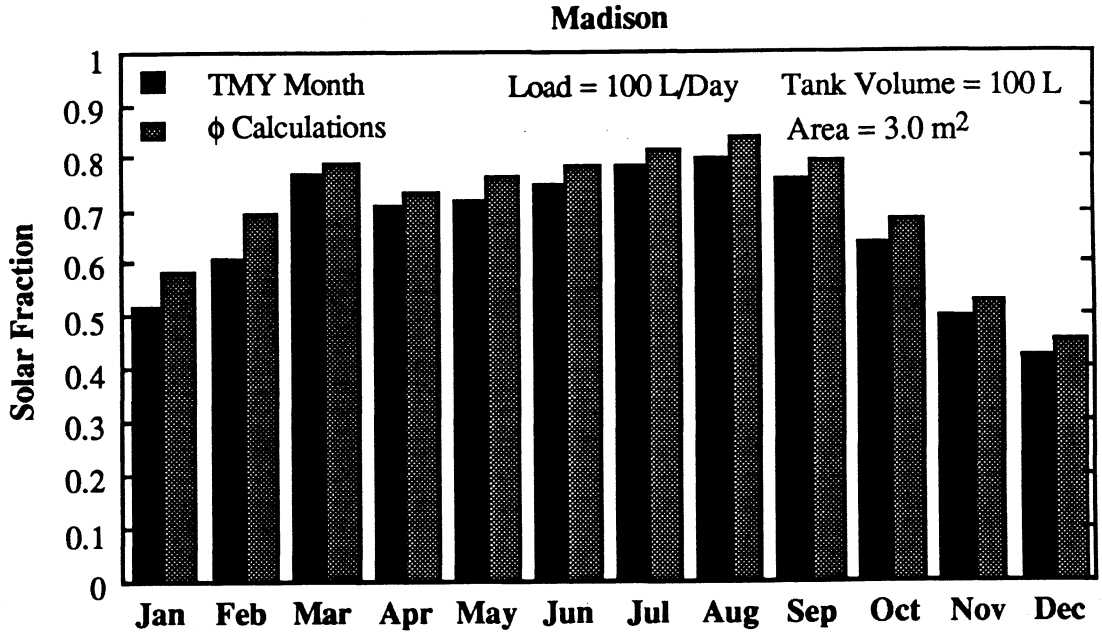


Figure 3.5 Comparison Between TMY and Monthly Average Hourly Utilizability Simulation Results

Two other short-cut simulation methods should be presented in light of the utilizability discussion presented. The first of these two methods is monthly average daily utilizability. Monthly average daily utilizability, $\bar{\phi}$, is defined as:

$$\bar{\phi} = \frac{\sum_{\text{days}} \sum_{\text{hours}} (I_T - I_{TC})^+}{\bar{H}_T N} = \frac{\sum_{\text{day}} \bar{I}_T \phi}{\sum_{\text{day}} \bar{I}_T} \tag{3.5}$$

where

N = Number of days in the month

$\bar{H}_T$ = Monthly average daily radiation on the tilted surface.

$\bar{\phi}$ is the ratio of the cross-hatched to cross-hatched plus solid areas in Figure 3.6. Klein [1978] developed correlations for $\bar{\phi}$ as a function of the monthly average daily clearness

index, $\overline{K}_t$, a geometric factor, and a monthly average critical radiation ratio, $\overline{X}_C$. Knowing $\bar{\phi}$, the monthly average daily useful energy gain across the collector is easily calculated as:

$$\overline{Q}_u = A\, F_r(\overline{\tau\alpha})\, \overline{H}_T\, \bar{\phi} \qquad (3.6)$$

Hence, monthly average daily utilizability further reduces the number of calculations required to determine system performance to one calculation per month.

As with monthly average hourly utilizability, monthly average daily utilizability is limited in application. The critical radiation ratio, $\overline{X}_C$, is based upon a constant critical radiation level, and therefore constant collector inlet temperature, throughout the entire month.

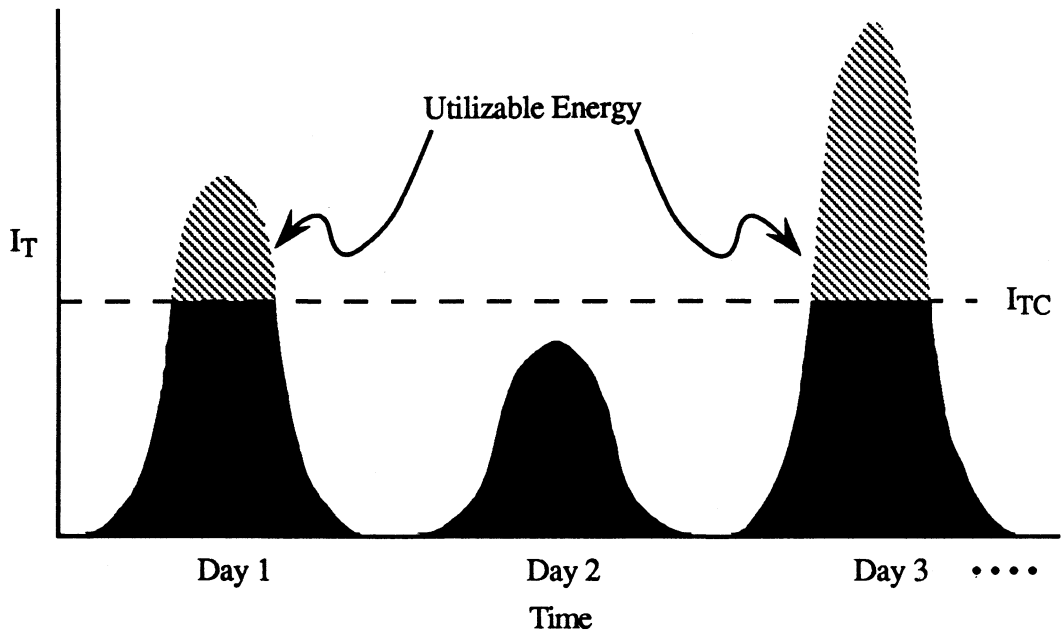


Figure 3.6 Visual Representation of Daily Utilizability

The idea of representing an entire month with the monthly average day is the other short-cut simulation method which should be viewed in the context of utilizability. Simulations using a monthly average day are usually repeated with the final conditions at the end of one simulation used as the initial conditions for the next simulation until a 24 hour periodic steady-state condition has been achieved in order to eliminate the effects of initial conditions on simulation results. Figure 3.7 is a hypothetical hourly radiation cumulative frequency distribution curve for the case when each day is equal to he monthly average day. A horizontal line at an $I_T/\bar{I}_T$ ratio of 1.0 is simply the cumulative frequency curve for a series of monthly average days. Integrating Equation (3.3) using various lower limits and the cumulative frequency distribution of Figure 3.7 results in the straight line in Figure 3.8. The realistic utilizability curve of Figure 3.3 representative of actual radiation data is also shown in Figure 3.8 for comparison purposes. The approximate and correct curves approach agreement only at low critical radiation ratios.

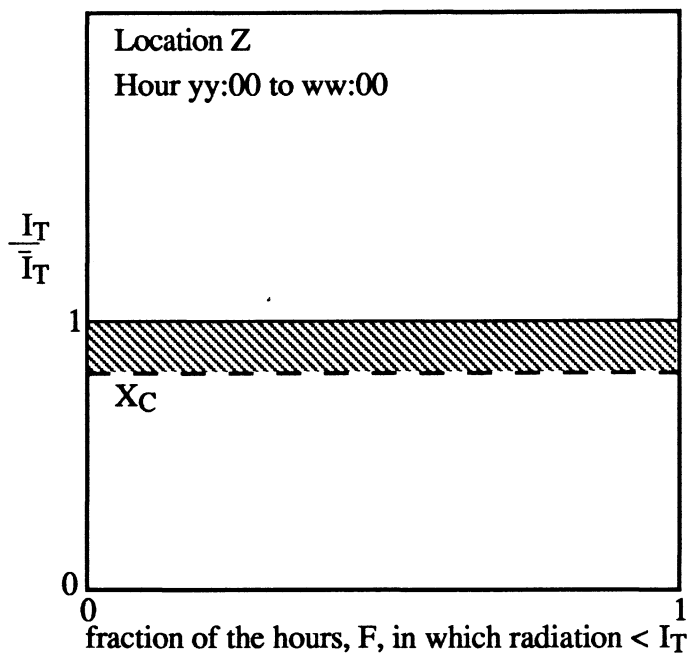


Figure 3.7 Hypothetical Hourly-Radiation Cumulative Frequency Distribution Assuming Monthly Average Days

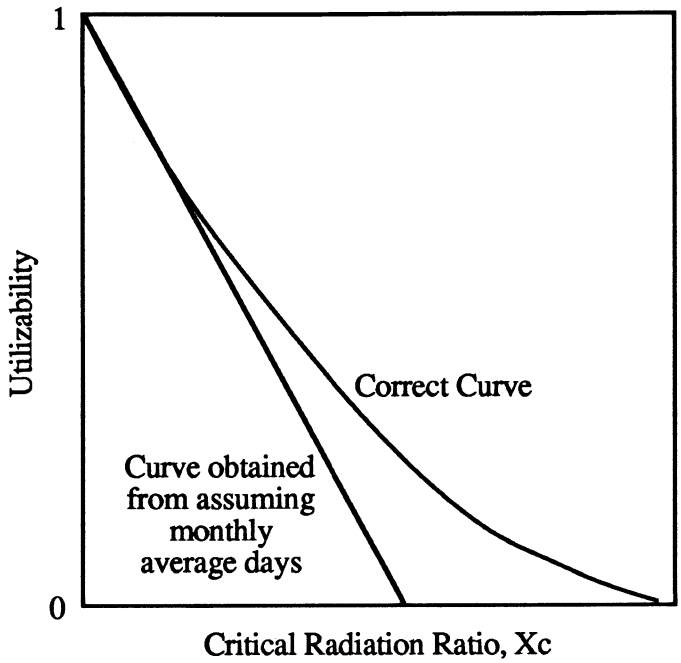


Figure 3.8 Hypothetical ϕ Curves

Similarly, Figure 3.9 is a hypothetical cumulative distribution for a month having a high ($\approx$ 0.7 or greater) $\overline{K}_t$. A cumulative frequency distribution curve approaching a horizontal line at an $I_T/\bar{I}_T$ ratio of 1.0 is characteristic of a month having a high $\overline{K}_t$. Integration of Equation (3.3) using various lower limits and the frequency distribution of Figure 3.9 results in the utilizability curve of Figure 3.10. The utilizability curve from Figure 3.8 representative of monthly average days is also shown in Figure 3.10 for comparison purposes. The utilizability curves show reasonable agreement.

In conclusion, representing an entire month by the monthly average day is only applicable when the critical radiation ratio is low, or when the monthly average daily clearness index is high. Representing a month with the monthly average day under any other circumstances is not advisable.

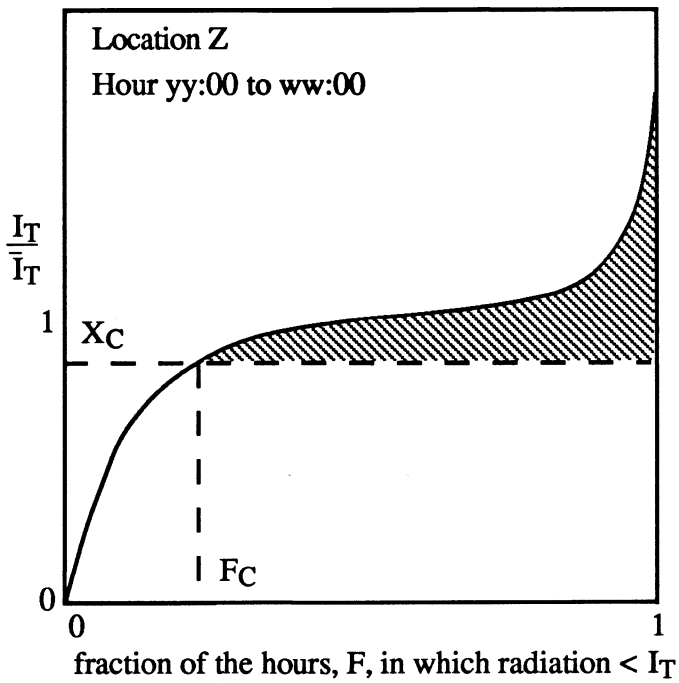


Figure 3.9 Hypothetical Hourly-Radiation Cumulative Frequency Distribution for a High $\overline{K_t}$

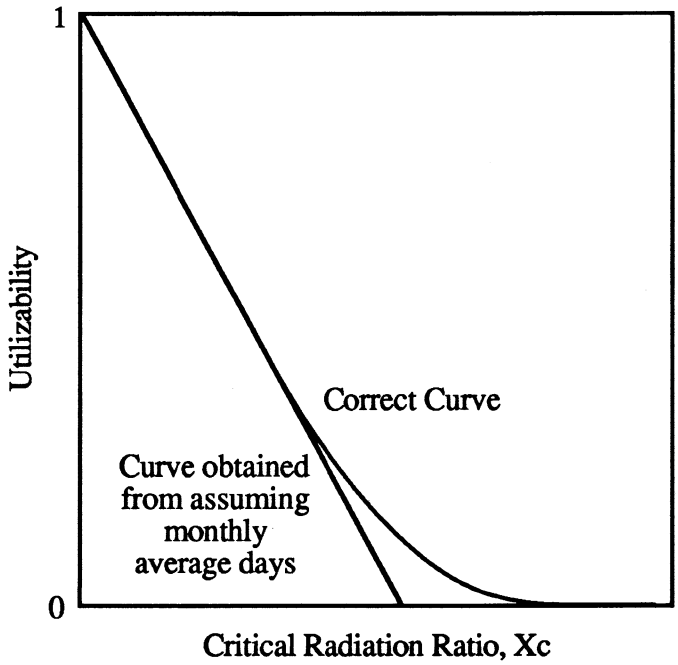


Figure 3.10 Hypothetical ϕ Curves

3.3.3 $\bar{\phi}$, *f-Chart*

The $\bar{\phi}$, *f*-Chart method [Beckman et al., 1976] is the final short cut simulation method to be discussed. The $\bar{\phi}$, *f*-Chart method, as the name implies, is a combination of monthly average daily utilizability and empirical correlations. $\bar{\phi}$, *f*-Chart assumes energy from the storage tank is always delivered at a temperature, T, where T is greater or equal to some minimum useful temperature, T_{min}. The method calculates the maximum possible monthly average daily utilizability, which is the system utilizability assuming a constant collector inlet temperature of T_{min}. Empirical correlations correct for the fact that the collector inlet temperature may be greater than T_{min} during the month. Substitution of the maximum possible monthly average daily utilizability into the empirical correlations results in an implicit equation for the monthly solar fraction. Further corrections are applied to account for tank heat losses and finite load heat exchanger size.

The original $\bar{\phi}$, *f*-Chart method of Klein and Beckman is restricted to closed-loop solar systems like the one pictured in Figure 3.11. Braun et al. [1983] later extended $\bar{\phi}$, *f*-Chart for applicability to open-loop, as well as closed-loop, SDHW systems. The $\bar{\phi}$, *f*-Chart method has a wider range of applicability than does the *f*-Chart method. However, as with *f*-Chart, the $\bar{\phi}$, *f*-Chart algorithms are based upon the assumption of a fully mixed solar storage tank and therefore tend to underestimate the performance of highly stratified systems.

Two different studies have lead to proposals for modifying the $\bar{\phi}$, *f*-Chart method in order to take into account tank stratification. The first of these methods was developed by Copsey and is applicable to the *f*-Chart as well as $\bar{\phi}$, *f*-Chart method. Copsey [1984] developed relationships based upon the collector and load flow rates and yearly solar fraction as predicted by $\bar{\phi}$, *f*-Chart or *f*-Chart which indirectly modify the collector heat removal factor, F_r, and loss coefficient, U_L. Re-evaluating the yearly solar fraction via

$\bar{\phi}$, *f*-Chart or *f*-Chart using the modified parameters results in a solar fraction in which the proper amount of tank stratification has been considered. Copsey reports an rms error between the modified methods and TRNSYS simulations of 2.07% for the $\bar{\phi}$, *f*-Chart method and 3.15% for the *f*-Chart method for collector flow rates between 10-60 kg/hr-m^2. Larger discrepancies are common for both the modified $\bar{\phi}$, *f*-Chart and *f*-Chart methods at flow rates less than 10 kg/hr-m^2. The correction is not applicable to systems using high flow rates which maintain a high degree of tank stratification due to the presence of stratification-enhancing devices, since the amount of stratification is correlated to the collector and load flow rates.

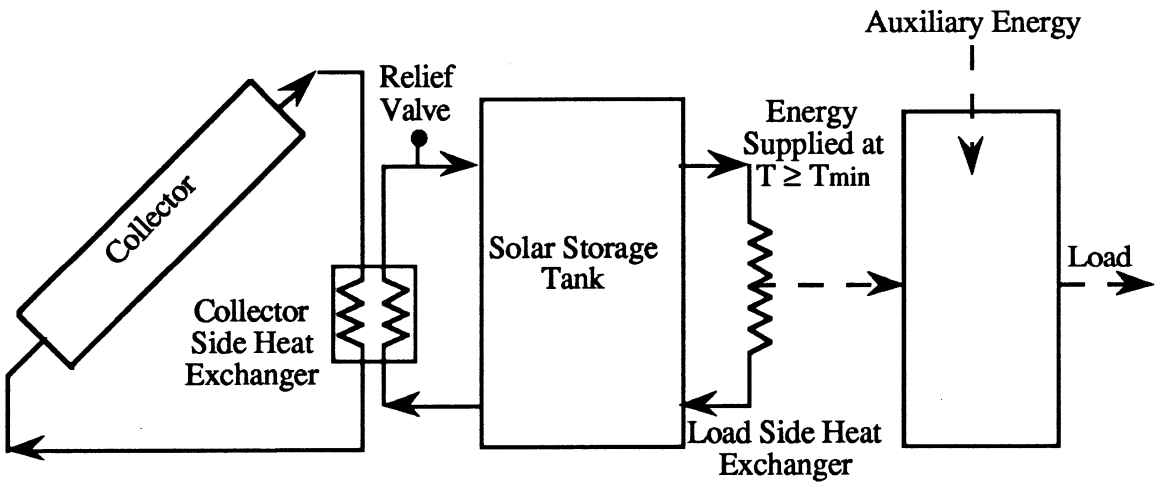


Figure 3.11 Closed-Loop SDHW System

Pagnier [1986] changed the $\bar{\phi}$, *f*-Chart method by developing modified correlations for the monthly average tank temperature and the monthly average useful energy gain applicable to fully stratified SDHW systems. Pagnier reports an rms error between the new $\bar{\phi}$, *f*-Chart algorithm and TRNSYS simulations using a completely stratified solar storage tank of 1.5% for yearly solar fractions and 3.4% for monthly solar fractions for Madison, WI. However, the Pagnier correlations are restricted to fully stratified open-loop systems.

In addition, accuracy of the Pagnier method when the daily integrated collector flow is greater than the daily integrated load flow is questionable.

3.4 OBTAINING SYSTEM PARAMETERS

Obtaining the system parameters to be used as input to the simulation model is the first step along the simulation path. Four possible ways to determine the system parameters are presented in this section.

3.4.1 Individual Parameter Knowledge

Gathering individual parameter knowledge involves testing each of the system components to determine the values of the physical parameters necessary for modeling. Manufacturer rated values should not be used for simulation input. Even commonly manufacturer specified measurements, such as tank volumes and tank and pipe insulation heat loss resistances, tend to be optimistic. Hence, all dimensions, heat loss coefficients, flow rates, temperature dead bands, collector properties, heat exchanger effectivenesses, etc necessary to construct a system model need to be determined via component tests. The ASHRAE 93-77 collector test guidelines [ASHRAE, 1977] is an example of a commonly used component test procedure.

3.4.2 Minnerly

The Minnerly method [1989] uses results obtained from a single ASHRAE-95/SRCC short-term test to determine the set of parameters representative of the system. Minnerly began with a mass and energy balance on the solar storage tank, as illustrated in Figure 3.12.

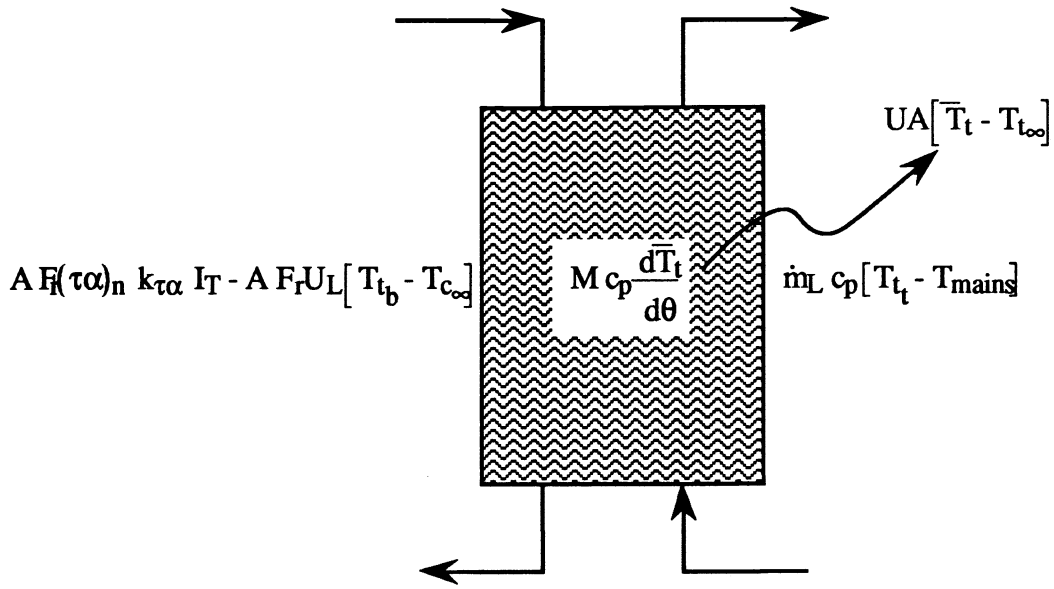


Figure 3.12 Solar Storage Tank Energy and Mass Balance

In equation form:

$$A F_r(\tau\alpha)_n k_{\tau\alpha} I_T - A F_r U_L [T_{t_b} - T_{c_\infty}] = \tag{3.7}$$

$$UA[\bar{T}_t - T_{t_\infty}] + \dot{m}_L c_p [T_{t_t} - T_{mains}] + M c_p \frac{d\bar{T}_t}{d\theta} M c_p$$

where

A	=	Collector aperture area
c_p	=	Fluid specific heat
$F_r(\tau\alpha)_n$	=	Collector gain coefficient at normal incidence
$F_r U_L$	=	Collector loss coefficient
I_T	=	Radiation on the collector surface
$k_{\tau\alpha}$	=	Incidence angle modifier
M	=	Mass of fluid in the tank
$\dot{m}_L$	=	Load mass flow rate
T_{c_∞}	=	Collector ambient temperature
T_{mains}	=	Mains water temperature
$\bar{T}_t$	=	Tank bulk average temperature

T_{t_b} = Temperature at the bottom of the tank

T_{t_t} = Temperature at the top of the tank

T_{t_∞} = Tank ambient temperature

UA = Tank heat loss - surface area product

The collector parameters, $F_r(\tau\alpha)_n$ and F_rU_L, have been modified to take into account the presence of heat exchangers and pipes as outlined in Duffie and Beckman [1980]. Integrating over a day results in:

$$\Delta U = A\left[F_r(\tau\alpha)\, H_{T,on} - F_rU_L\left(\overline{T}_{t_b} - T_{c_\infty}\right)\Delta\theta_{on}\right] - \tag{3.8}$$

$$UA\left[\widehat{T}_t - T_{t_\infty}\right]\Delta\theta_{tot} - M_d\, c_p\left[\overline{T}_{t_t} - T_{mains}\right]$$

where

ΔU = Energy storage within the tank integrated over the entire day

$H_{T,on}$ = Total incident radiation during pump operation

M_d = Total load draw

$\Delta\theta_{on}$ = Total time of pump operation during the day

$\Delta\theta_{tol}$ = Total integration period (1 day)

$\widehat{T}_t$ = Bulk average tank temperature averaged over the entire day

$\overline{T}_{t_b}$ = Bottom tank temperature averaged over the period of pump operation

$\overline{T}_{t_t}$ = Top tank temperature averaged over the periods of load draws

Minnerly presents correlations for calculating the averaged temperatures as a function of the mains temperature, set temperature, test day solar fraction, collector area, solar tank volume and daily integrated tilted surface radiation.

The ASHRAE-95/SRCC test is periodic steady state, requiring ΔU in Equation (3.8) to be approximately zero. The $H_{T,on}$ and $\Delta\theta_{on}$ terms were eliminated by introducing utilizability into the daily integrated energy balance as follows:

$$0 = A\,F_r(\tau\alpha)\,\phi\,H_T - UA\left[\hat{T}_t - T_{t_\infty}\right]\Delta\theta_{tot} - M_d\,c_p\left[\overline{T}_{t_t} - T_{mains}\right] \qquad (3.9)$$

where

ϕ = Test day utilizability

Minnerly approximated the ASHRAE-95/SRCC stepped radiation profile by a triangular profile having the same area as shown in Figure 3.13. The test day utilizability for the triangular radiation profile if the critical radiation level, I_{TC}, is constant throughout the test day is:

$$\phi = 1 - 2\frac{I_{TC}}{I_{max}} + \left[\frac{I_{TC}}{I_{max}}\right]^2 \qquad (3.10)$$

Substitution of Equation (3.10) into Equation (3.9) and rearranging results in the following expression:

$$I_{TC} = I_{max}\left[1 - \sqrt{\frac{UA\left(\hat{T}_t - T_{t_\infty}\right)\Delta\theta_{tot} + M_d\,c_p\left(\overline{T}_{t_t} - T_{mains}\right)}{A\,F_r(\tau\alpha)\,H_T}}\right]$$

or

$$F_rU_L = \frac{F_r(\tau\alpha)\,I_{max}}{\left[\overline{T}_{t_b} - T_{t_\infty}\right]}\left[1 - \sqrt{\frac{UA\left(\hat{T}_t - T_{t_\infty}\right)\Delta\theta_{tot} + M_d\,c_p\left(\overline{T}_{t_t} - T_{mains}\right)}{A\,F_r(\tau\alpha)\,H_T}}\right] \qquad (3.11)$$

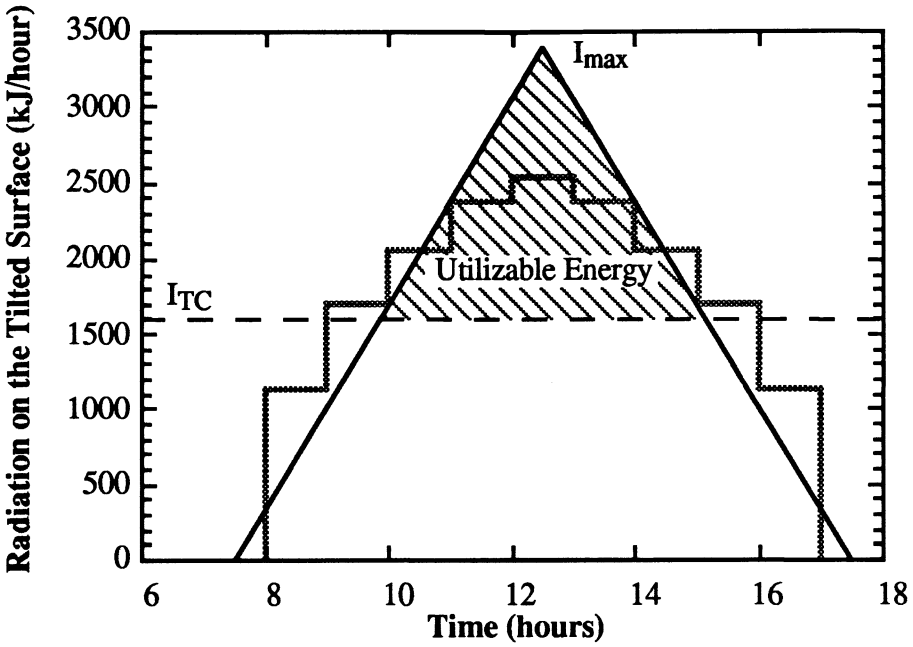


Figure 3.13 Actual and Approximate ASHRAE-95/SRCC Radiation Profile

Any pair of collector parameters, $F_r(\tau\alpha)$ and F_rU_L which satisfy Equation (3.11), given the tank loss coefficient, UA, represent the SDHW system of interest. Minnerly compared *f*-Chart results calculated using various $F_r(\tau\alpha)$ and F_rU_L pairs to the actual long-term performance of an active SDHW system. The actual system performance was monitored and evaluated by the National Bureau of Standards. A maximum absolute error of 3% (6.5% relative error) was observed by Minnerly between the predicted and actual yearly solar fraction.

3.4.3 Buckles

The Buckles rating method [1983], like the Minnerly method, is intended to find the collector parameters representative of the SDHW system from short-term test data. The Buckles and Minnerly methods do significantly differ, however, in that the Buckles method may be used in conjunction with in situ performance data. The Minnerly method, on the

other hand, is specifically designed to coincide with the ASHRAE-95/SRCC short-term test.

Buckles simplified the solar storage tank energy balance, Equation (3.7), by eliminating the load terms via requiring a zero load flow throughout the testing period. The load flow was eliminated in order to make the monitoring and installation of a flow meter in the delivery piping unnecessary. The simplified equation is:

$$A\,F_r(\tau\alpha)_n\,k_{\tau\alpha}\,I_T - A\,F_rU_L\left[T_{t_b} - T_{c_\infty}\right] = UA\left[\overline{T}_t - T_{t_\infty}\right] + \frac{d\overline{T}_t}{d\theta}\,M\,c_p \qquad (3.12)$$

Equation (3.12) may be rearranged and integrated over the time period $\Delta\theta$ to yield:

$$F_r(\tau\alpha)_n - \frac{F_rU_L\left[T_{t_b} - T_{c_\infty}\right]}{I_T\,k_{\tau\alpha}} \approx \frac{M\,c_p\left[\overline{T}_t(\theta+\Delta\theta) - \overline{T}_t(\theta)\right] + UA\left[\widehat{T}_t - T_{t_\infty}\right]\Delta\theta}{A\,I_T\,k_{\tau\alpha}\,\Delta\theta} \qquad (3.13)$$

where

$\widehat{T}_t$ = Bulk average tank temperature averaged over the time period $\Delta\theta$.

The right hand side of Equation (3.13) may be calculated at various times if the following information is known:

1. Tank capacitance, $M\,c_p$
2. Bulk average tank temperatures, $\overline{T}_t$'s, (the time-averaged bulk-averaged tank temperature, $\widehat{T}_t$, may be calculated from the $\overline{T}_t$'s)
3. Tank heat loss coefficient - surface area product, UA
4. Tank ambient temperature, T_{t_∞}
5. Collector area, A

6. Incident radiation, I_T
7. Collector incident angle modifier, $k_{\tau\alpha}$
8. Time step, $\Delta\theta$ ($\Delta\theta$ does not have to be constant throughout the testing period)

Buckles noticed Equation (3.13) is of the form of a straight line when the abscissa is taken as the collector temperature difference to radiation ratio. In other words, Equation (3.13) is equivalent to the following expression for a straight line:

$$b + m\,x = y \tag{3.14}$$

where

b = "y' intercept, $F_r(\tau\alpha)_n$
m = slope, F_rU_L
x = abscissa, $[T_{t_b} - T_{c_\infty}]/[I_T\,k_{\tau\alpha}]$
y = ordinate, right hand side of Equation (3.13).

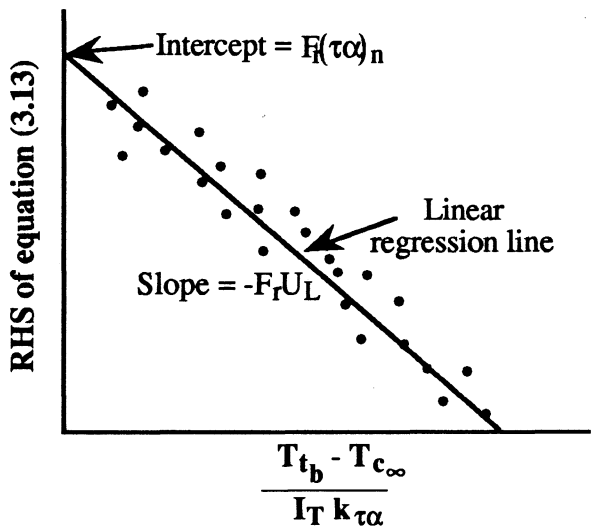


Figure 3.14 Hypothetical Linear Regression Fit to Equation (3.13)

The modified collector parameters, $F_r(\tau\alpha)_n$ and F_rU_L, may be obtained from the linear regression line to the right hand side (i.e., "y") data points as shown in Figure 3.14.

Barker [1990] investigated the Buckle approach and concluded long-term performance results obtained from simulations using the Buckle modified collector parameters are repeatable to within a standard deviation of approximately 5%.

3.4.4 Spirkl

The Spirkl rating method [1990] is analogous to the Buckles method. Both the Spirkl and Buckles methods are intended to find a set of parameters which characterize the SDHW system of interest based upon in situ performance results and a regression analysis. Unlike the buckles method, however, the Spirkl method is applicable for non-zero load flow conditions.

Sprikl, as with Minnerly and Buckles, began with a solar tank mass and energy balance, Equation (3.7). Spirkl assumed only the top tank temperature, T_{t_t}, is known. The following equations were introduced in order to account for tank stratification:

$$M c_p \frac{dT_o}{d\theta} = - UA\left[T_o - T_{t_\infty}\right] - \dot{m}_L c_p \left[T_o - T_{mains}\right] \tag{3.15}$$

$$T_{t_b} = T_o + k_s \left[\overline{T}_t - T_o\right] \tag{3.16}$$

$$T_{t_t} = T_o + \frac{\left[\overline{T}_t - T_o\right]}{k_s} \tag{3.17}$$

where

T_o = Reference temperature

K_s = Stratification constant. For a fully mixed tank, K_s=1 and T_o=$\overline{T}_t$=T_{t_b}=T_{t_t}. For a stratified tank, 0<K_s<1 and T_o<$\overline{T}_t$<T_{t_b}<T_{t_t}.

Equations (3.7) and (3.15) through (3.17) may be combined and solved for the top tank temperature, T_{t_t}, at time $\theta_o+\Delta\theta$ in terms of the collector parameters, $A^*F_r(\tau\alpha)_n$ and $A^*F_rU_L$, tank heat loss coefficient, UA, tank capacitance, M^*c_p, and stratification constant, K_s, if the following information is known at time θ_o:

1. Incident radiation, I_T
3. Collector ambient temperature, T_{c_∞}
4. Tank ambient temperature, T_{t_∞}
5. Mains water temperature, T_{mains}
6. Load flow rate, $\dot{m}_L$
7. Incident angle modifier, $k_{\tau\alpha}$

Spirkl reasoned it should be possible to determine the set of unknown system parameters via a multi-variable non-linear regression algorithm such that the following objective function in minimized:

$$\chi^2(\bar{\xi}) = \sum_{u=1}^{u=n} \left[\frac{T_u - T(\theta_u,\bar{\xi})}{\sigma_u} \right]^2 \qquad (3.18)$$

where

χ^2 = The objective function. χ^2 is a function of the vector $\bar{\xi}$ where $\bar{\xi}$ contains the five unknown parameters

u = Number of measured data points

T_u = Measured top tank temperature at time θ_u

$T(\theta_u,\bar{\xi})$ = The predicted top tank temperature at time θ_u

σ_u = Standard deviation of T_u due to measurement error

Spirkl recommends filtering the measured tank temperatures with a 24 hour time constant in order to eliminate the effects of short-term dynamics on the regression solution. Short-term dynamics may be caused by such phenomena as collector and pipe capacitance, and controller hysterias.

The method was tried for eight different simulated SDHW systems. The simulations are idealistic in that measurement and random errors are not present. The systems differ in collector parameters, $F_r(\tau\alpha)_n$ and F_rU_L, and tank heat loss coefficient, UA. The systems contain an external heat exchanger, $\varepsilon = 0.44$. The tank volume is 0.25 m^3, and the incidence angle modifier constant, b_o, is 0.1. The collector has an area of 5.556 m^2 and is tilted at an angle of 45° due south. The collector and tank loop flow rates are 0.010 and 0.017 kg/sec-m^2. Water is used in both the collector and tank loops. A gain controller is present with temperature sensors located at the collector and solar tank exits. The upper controller dead band is 11.11 °C with the lower dead band temperature being 2.78 °C. A schematic is shown in Figure 3.15.

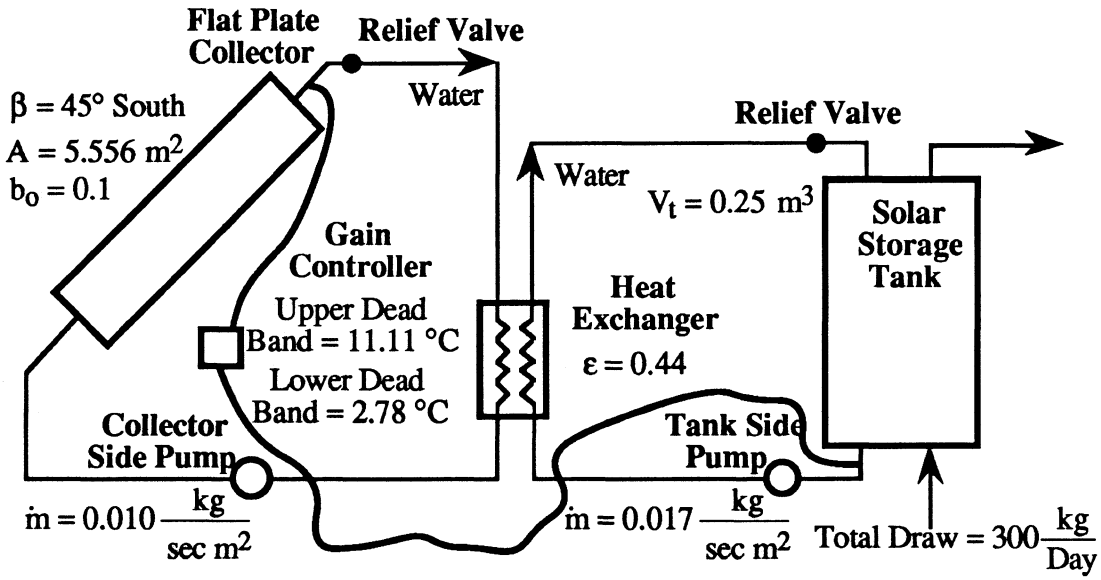


Figure 3.15 Actual SDHW System

Yearly TRNSYS simulations were performed for each system using Madison, WI TMY weather data as input. The mains and tank ambient temperatures were constant throughout the yearly simulations at 12 °C and 22 °C, respectively. The ground reflectance was also constant at 0.2. A 300 kg/day RAND load flow as pictured in Figure 3.16 was specified. *The solar tank was modeled as being fully mixed at all times.* Instantaneous performance data were recorded during July 3 in addition to the yearly performance results. The July 3 data acquisition began at 9:00 A.M. and stopped at 5:00 P.M., with data being recorded every 3 minuets. Figures 3.17 and 3.18 illustrate the July 3 solar radiation and ambient temperature profiles, respectively.

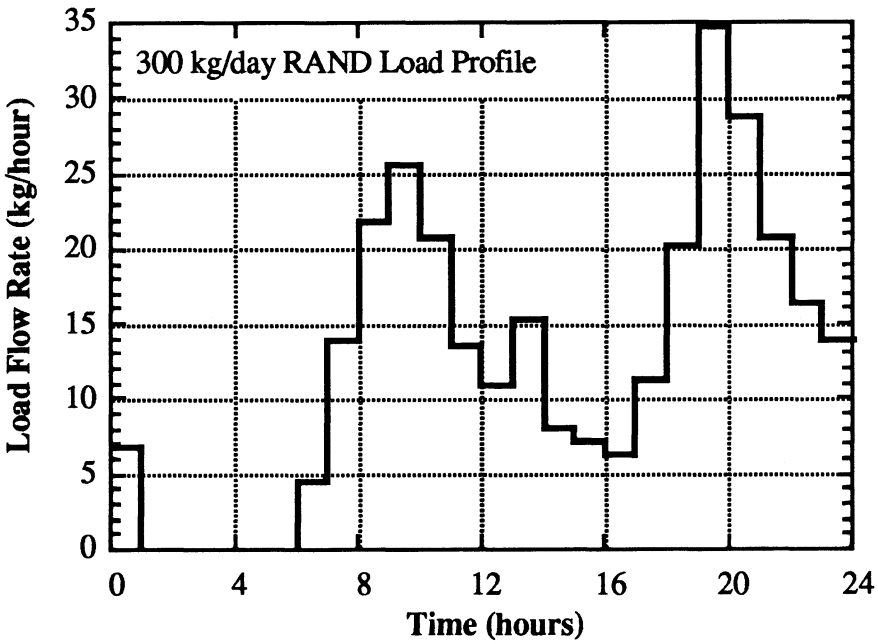


Figure 3.16 RAND Hot Water Draw Profile

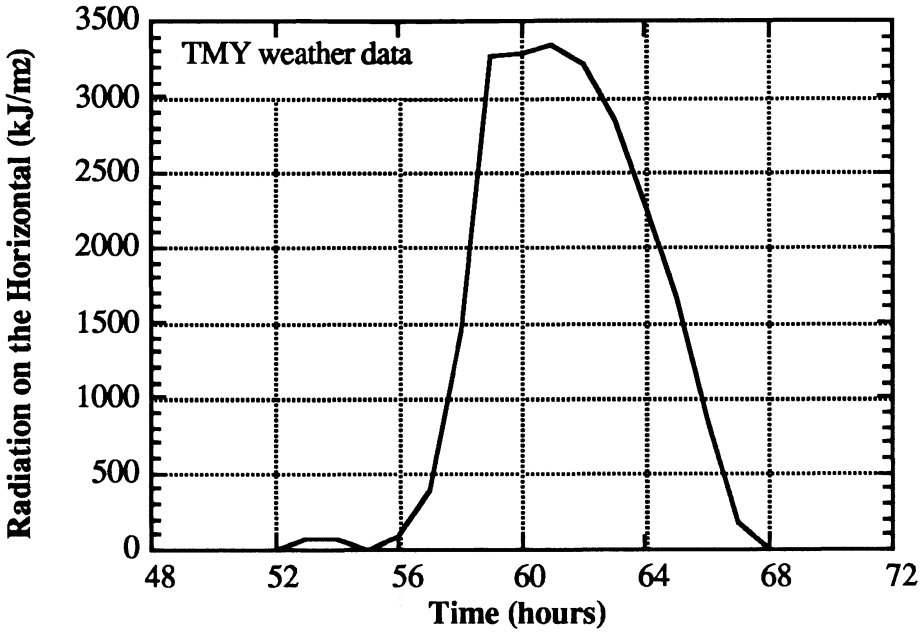


Figure 3.17 Madison, WI July 3 TMY Radiation Profile

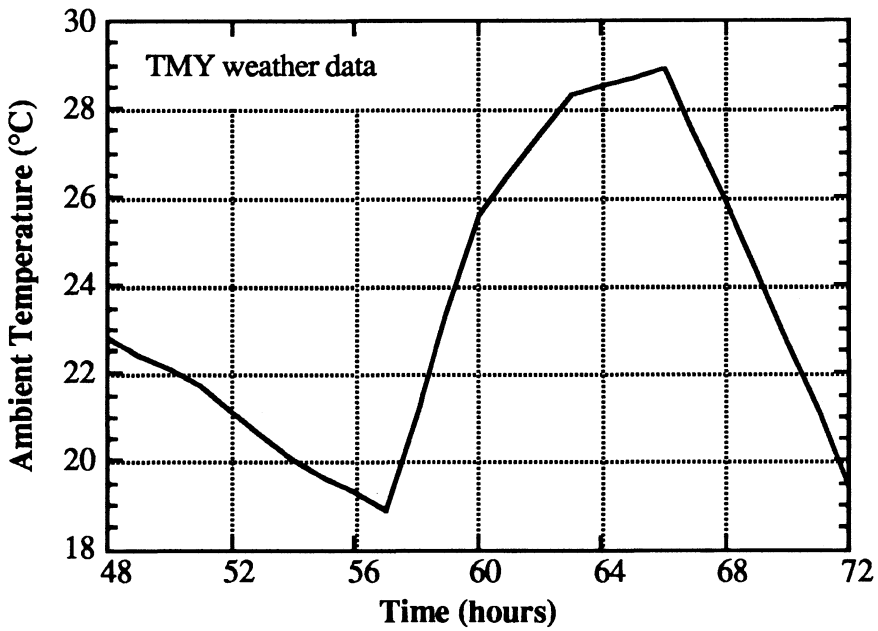


Figure 3.18 Madison, WI July 3 TMY Temperature Profile

The weather and tank temperatures for July 3 were used as input to a Levenberg-Marquardt regression algorithm [Press et al., 1986]. The tank capacitance was treated as known rather than as a variable. The tank temperatures were not filtered since the TRNSYS collector and pipe models used did not consider thermal capacitance. Two regression analyses were performed for each system. The first regression analysis was designed to find the solar tank heat loss coefficient, UA, as well as the modified collector parameters. The second set of regressions were used to find the modified collector parameters, with the solar tank heat loss coefficient being fixed at the correct value.

Yearly TRNSYS simulations were subsequently performed with a simplified system (i.e., no pipes or heat exchangers present) using the determined parameters as input. A schematic of the simplified system is shown in Figure 3.19. The collector parameters were altered as outlined in Duffie and Beckman [1980] to account for the difference in flow rates between "test" and "actual" conditions. The July 3 data (i.e., the data used in the regression analyses) were gathered from systems having a collector loop flow rate of 0.010 kg/sec-m^2 and a tank loop flow of 0.017 kg/sec-m^2. The simplified systems, on the other hand, only have one loop, and hence only one flow rate. A 0.017 kg/sec-m^2 flow rate was assigned to the simplified systems, with modification being applied to the collector regression results for a difference between the "test" and "actual" flow rates (the "test" flow rate being 0.010 kg/sec-m^2). Figures 3.20 and 3.21 compare the yearly delivered solar energies obtained from the actual and simplified system simulations. The relative error in delivered solar energy averaged amongst the eight trials in which the tank UA was determined by the regression analysis is 2.6%. In contrast, the average relative error amongst the eight trials in which the tank UA was fixed at the appropriate value during the regression analysis is slightly greater at 4.9%.

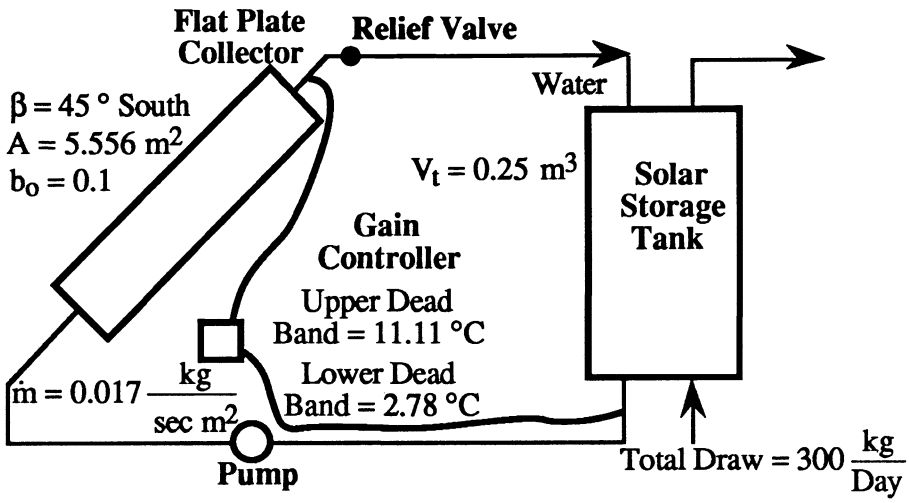


Figure 3.19 Equivalent Simplified SDHW System

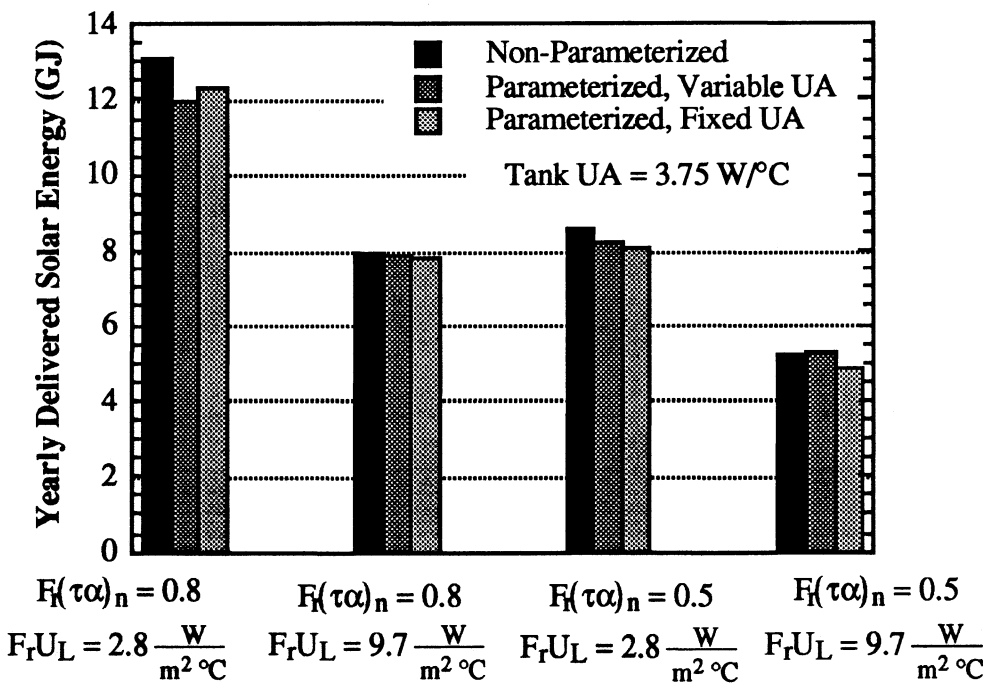


Figure 3.20 Yearly Delivered Solar Energies Obtained from Simulations of Complete and Simplified SDHW Systems

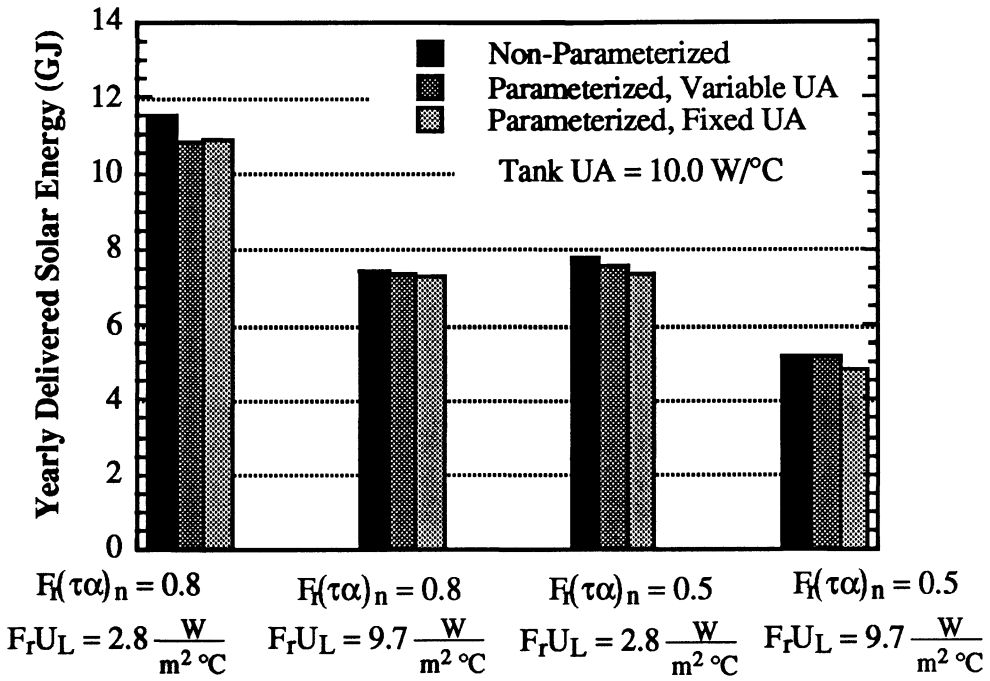


Figure 3.21 Yearly Delivered Solar Energies Obtained from Simulations of Complete and Simplified SDHW Systems

3.5 SUMMARY

SDHW system simulations involve two distinct steps. The first step is to gather the necessary information about the system from which a model may be constructed. Gathering system information requires the performance of physical tests upon the system. The Minnerly, Buckles, and Spirkl methods utilize test data from a complete, operable SDHW system to determine the representative parameters. Obtaining the necessary system parameters by individually testing each of the system components is an option not requiring, nor should be utilized with, an assembled system. All four methods appear to have similar accuracy.

Performing the system simulation using the determined parameters as input is the second simulation step. Detailed simulations can yield accurate results, but require appreciable time and computer facilities. "Short-cut" simulation methods require much less

time as compared to detailed simulations. Also, several short-cut simulation methods, such as *f*-Chart or $\bar{\phi}$, *f*-Chart, do not require computer facilities at all, but may be performed "by hand". Use of the short-cut simulation methods discussed in this chapter over the range of typical system designs yields results comparable to those obtained from detailed simulations. The *f*-Chart method has also been shown to compare favorably with actual long-term system performance data [Duffie and Mitchell, 1983].

Chapter 4

Comparison Between Experimental and Simulated Short-Term Test Results

4.1 INTRODUCTION

Short-term experimental testing of solar domestic hot water (SDHW) systems is a critical part of the Solar Rating and Certification Corporation's (SRCC) current method for system certification. Even short-term testing, however, is time consuming and expensive. Replacing the experimental tests with computer simulations is one way to avoid the disadvantages associated with physical testing. Furthermore, short-term (such as daily) comparisons between experimental tests and simulations are indicative of long-term (such as yearly) simulation accuracy which may be of more interest. Comparisons between short-term experimental and simulated test results for one well-instrumented SDHW system are presented.

4.2 EXPERIMENTAL SET UP

The experimental tests were conducted in accordance to the ASHRAE-95/SRCC guidelines by Colorado State University (CSU), Fort Collins, Colorado [Carlson, 1991]. A schematic of the SDHW system under investigation is shown in Figure 4.1. Water is the working fluid throughout the system. An electric boiler, rather than a solar collector, is used to transfer energy into the system. The CSU tests differed from the ASHRAE-95/SRCC specifications in that the load draws continued until 16603 kJ, rather than 14400

kJ, of energy were extracted. An agreement in auxiliary energy input to within 3% between two successive days was the criteria for periodic steady-state conditions. The CSU tests use a constant set temperature of 54 °C.

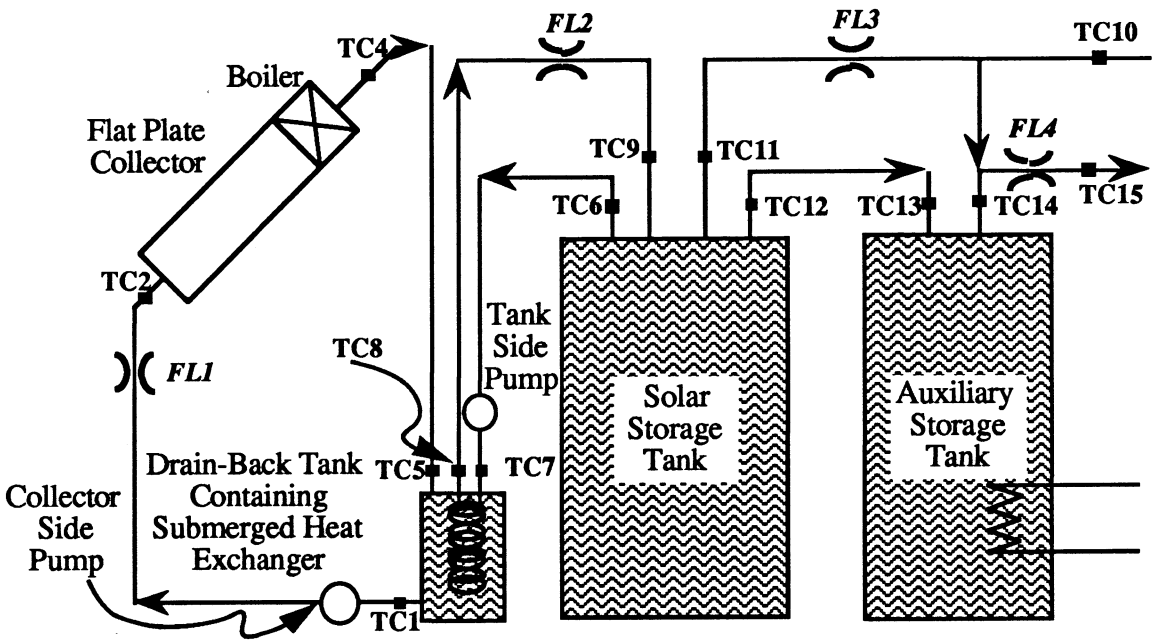


Figure 4.1 SDHW System

A total of 16 experimental tests were performed. The tests differ in collector area, fluid flow rates, and solar tank volume. In addition, for eight of the tests the solar tank is fitted with a manifold to help promote thermal stratification within the tank. The manifold is not present for the other eight tests. Table 4.1 summarizes the 16 test runs.

Test #	Coll Flow (kg/hr)	Tank Flow (kg/hr)	Area (m^2)	Tank Vol (m^3)	Tank Design
1	205.2	169.2	2.78	0.223	Basic
2	410.4	169.2	2.78	0.223	Manifold
3	205.2	342.0	2.78	0.223	Manifold
4	410.4	342.0	2.78	0.223	Basic
5	205.2	169.2	5.56	0.223	Manifold
6	410.4	169.2	5.56	0.223	Basic
7	205.2	342.0	5.56	0.223	Basic
8	410.4	342.0	5.56	0.223	Manifold
9	205.2	169.2	2.78	0.272	Manifold
10	410.4	169.2	2.78	0.272	Basic
11	205.2	342.0	2.78	0.272	Basic
12	410.4	342.0	2.78	0.272	Manifold
13	205.2	169.2	5.56	0.272	Basic
14	410.4	169.2	5.56	0.272	Manifold
15	205.2	342.0	5.56	0.272	Manifold
16	410.4	342.0	5.56	0.272	Basic

Table 4.1 Experimental Test Summary

4.3 SIMULATION MODLES AND PARAMETERS

The computer program TRNSYS (version 13.1) [Klein et al., 1990] was used to perform the computer simulations. Standard models found in the TRNSYS library were used to replicate the SDHW system, with two exceptions. The first exception concerns the pumps. The TRNSYS pump model, Type 3, does not take into consideration any effect the pump work may have on the temperature of the circulating fluid. Type 3 was modified such that a specified percentage of the pump work, η, acts to raise the fluid temperature according to the following equation:

$$T_{out} = T_{in} + \frac{w * \eta}{\dot{m} * c_p} \tag{4.1}$$

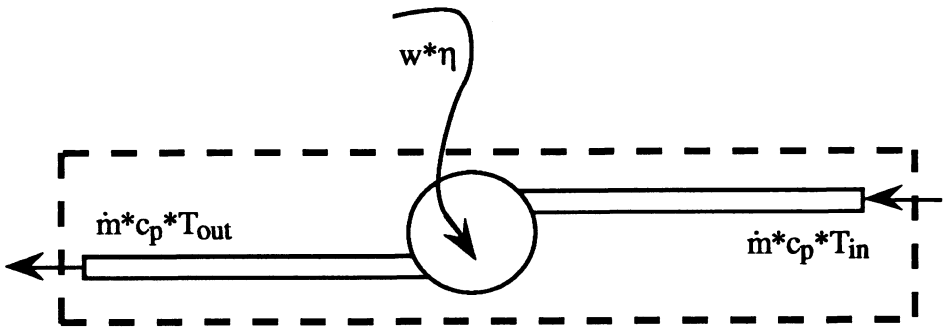


Figure 4.2 Pump Energy Balance

The pump work, w, is the same pump work as calculated by the standard Type 3 model, and is the pump power multiplied by the time of operation. The CSU experimental pump powers for both the collector side and tank side loops are nearly constant, regardless of flow rate.

The second non-standard TRNSYS model concerns the load-flow on-off controller. A specific TRNSYS-compatible FORTRAN subroutine was written to model the load-flow controller. The subroutine turns "on" the flow at the time step closest to the specified specified time (8:00, 12:00, or 17:00) and turns "off" the flow at the time step in which the energy draw is closest to 16603 kJ.

All but a few of the system parameters required for simulation purposes were obtained from CSU. A complete listing of the system parameters, and, where necessary, an indication of how CSU obtained the parameters, follows.

4.3.1 Fluid Properties

The experimental tests take into account the specific heat and density dependence of water upon temperature by the following two polynomial relationships:

$$c_p\left(\frac{J}{kg\text{-}^\circ C}\right) = 4226. - 3.9*T + 0.1125*T^2 - 1.6094x10^{-3}*T^3 + 1.25x10^{-5}*T^4 - 3.9062x10^{-8}*T^5 \quad (4.2)$$

$$\rho\left(\frac{kg}{L}\right) = 0.993 + 1.0087x10^{-4}*T + 8.9392x10^{-6}*T^2 + 5.7928x10^{-8}*T^3 - 1.9531x10^{-10}*T^4 \quad (4.3)$$

Figure 4.3 compares specific heat and density values for various temperatures. Simulations suggest the temperature dependence will affect daily integrated energy flows by at most 0.1%. The simulations thus use constant values of 1.0 kg/L and 4190 J/kg-°C for the fluid density and specific heat, respectively.

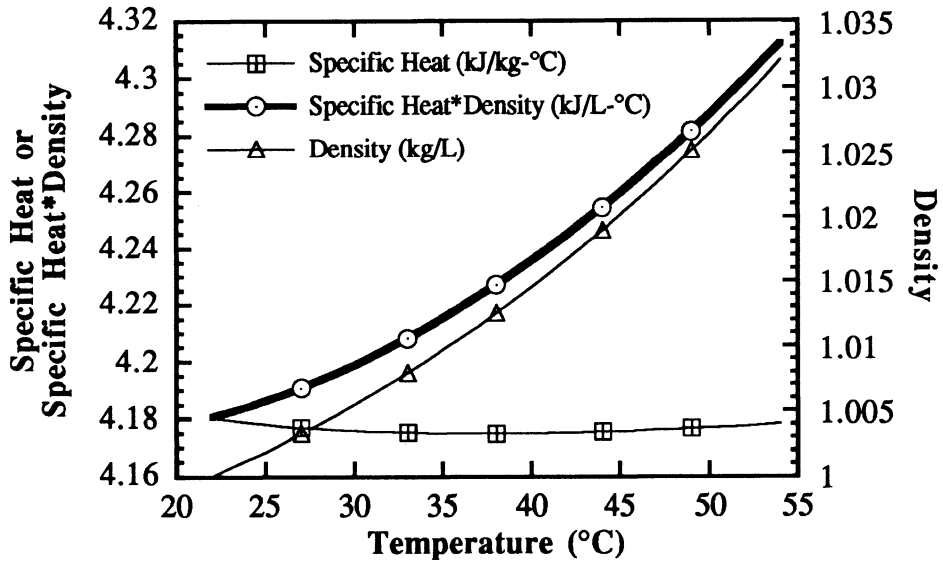


Figure 4.3 Property Dependency Upon Temperature

4.3.2 Flat Plate Collector

The CSU tests use an electric heater, instead of a flat plate collector, to transfer energy into the system. The Hottel-Whillier equation is the governing equation controlling

the amount of energy delivered into the system for both the experimental and simulation cases. The synthesized collector properties are shown in Figures 4.4 and 4.5.

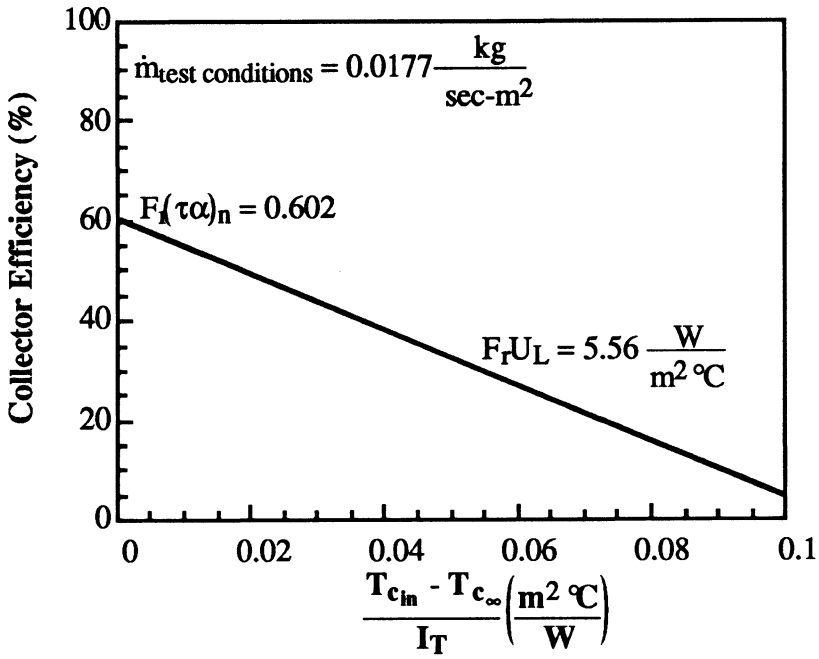


Figure 4.4 Collector Efficiency Performance

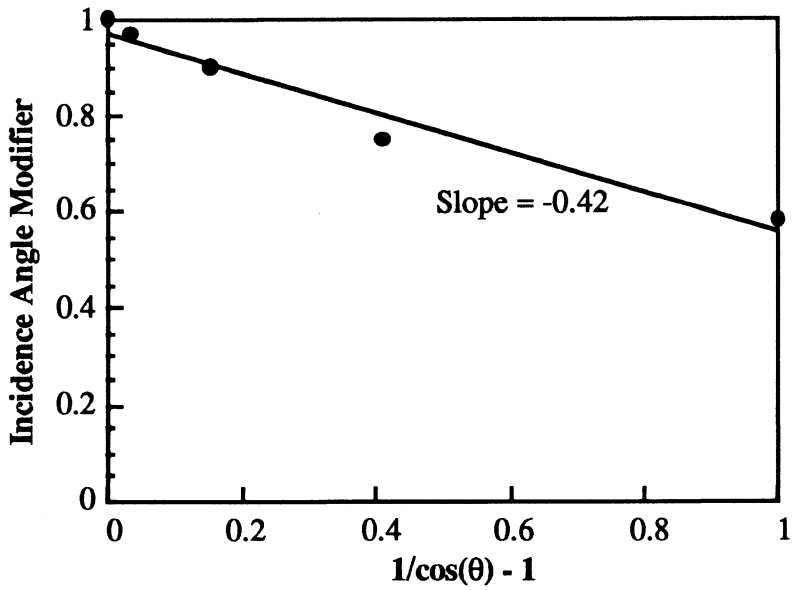


Figure 4.5 Collector Incidence Angle Modifier Data

The collector area is 1.852 m^2/panel. The number of panels varies between 1.5 and 3.0 depending upon the test, with all panels being connected in parallel. The collector is at a slope of 45° and the ground reflectance is 0.0.

4.3.3 Solar Tanks

The experimental tests use either a manufacturer rated "80 gallon" or "65 gallon" solar storage tank, depending upon the test reference number. CSU measured the tank volumes to actually be 71.9 gallons (272 liters) and 58.9 gallons (223 liters). Both tanks are identical except in diameter. Figure 4.6 is a tank vertical cross section. CSU also measured the tank loss coefficient terms by cool down tests. The $M*c_p$ terms are calculated from the measured volumes, a density of 0.998 kg/L, and a specific heat of 4180 J/kg-°C. These capacitance terms are subsequently increased by 1% to take into account the mass of the tank, itself. The heat loss coefficients are thus found to be 3.74 W/°C (1.46 W/m^2-°C) and 3.41 W/°C (1.49 W/m^2-°C) for the large and small tanks, respectively.

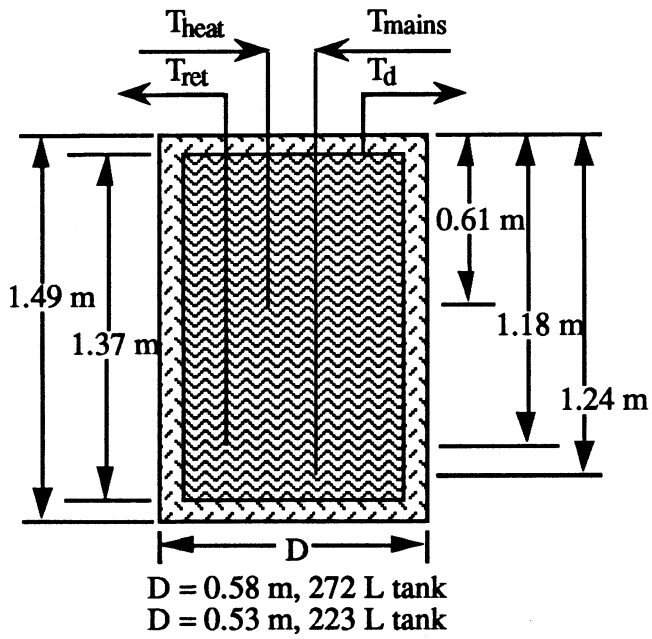


Figure 4.6 Solar Tank Vertical Cross Section

A major simulation question concerns what type of tank model to use. Kleinbach [1990] investigated instantaneous test data for the eight non-diffused tests. The instantaneous test data consists of temperatures, volume flow rates, energy usage, and energy delivery monitored throughout the system. The data were recorded every 15 seconds during a load draw, 15 minuets during simulated daylight hours, and every 30 minutes during overnight periods. Kleinbach simulated various tank models, forcing the input flow rates and temperatures (T_{heat}, T_{mains}, $T_{t\infty}$, $\dot{m}_{heat}$, $\dot{m}_L$) at their measured values. Figure 4.7 illustrates the flow rates and temperatures pertaining the the solar tank. A "P" statistic was defined as follows to evaluate the validity of each tank model:

$$P = \frac{\sqrt{\sum_{\theta=\theta_{on}}^{\theta_{off}} \dot{m}_{heat}^2 \, c_p^2 \, (T_{ret,exp} - T_{ret,sim})^2 \, \Delta\theta^2}}{\int_{day} \dot{m}_{heat} \, c_p \, (T_{heat} - T_{ret,exp}) \, d\theta} \qquad (4.4)$$

where

θ_{on} = time when the flow from the heat source begins

θ_{off} = time when the flow from the heat source ends.

Kleinbach concluded multi-node fixed-inlet tank models having two to five nodes result in lower P values in comparison to the other tank models investigated. The simulations use a three node

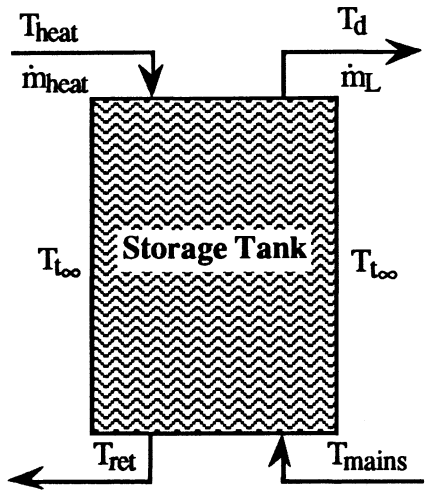


Figure 4.7 Solar Tank Flow Rates and Temperatures

tank. Sufficient data to conduct similar simulations on the eight diffused tanks are not available.

Figure 4.8 shows daily integrated energy flows for test #3 using 5, 10 and 15 node solar tank models. The correspondence between the energy flows shown in Figure 4.8 to the actual system is illustrated in Figure 4.9. Q_u is the energy gain across the collector (boiler), Q_{par} is the sum of the two pump works, Q_s is the energy delivered from the solar tank, Q_{aux} is the auxiliary energy supplied to the auxiliary tank, and Q_{del} is the total energy, solar plus auxiliary, delivered from the system. The solar fraction is defined as:

$$sf = \frac{Q_s - Q_{par}}{Q_{del}} \tag{4.5}$$

Figure 4.8 indicates increasing the number of nodes beyond ten results in small simulation differences. A multinode, fixed-inlet tank model having ten nodes was used in the diffused tank simulated short-term tests.

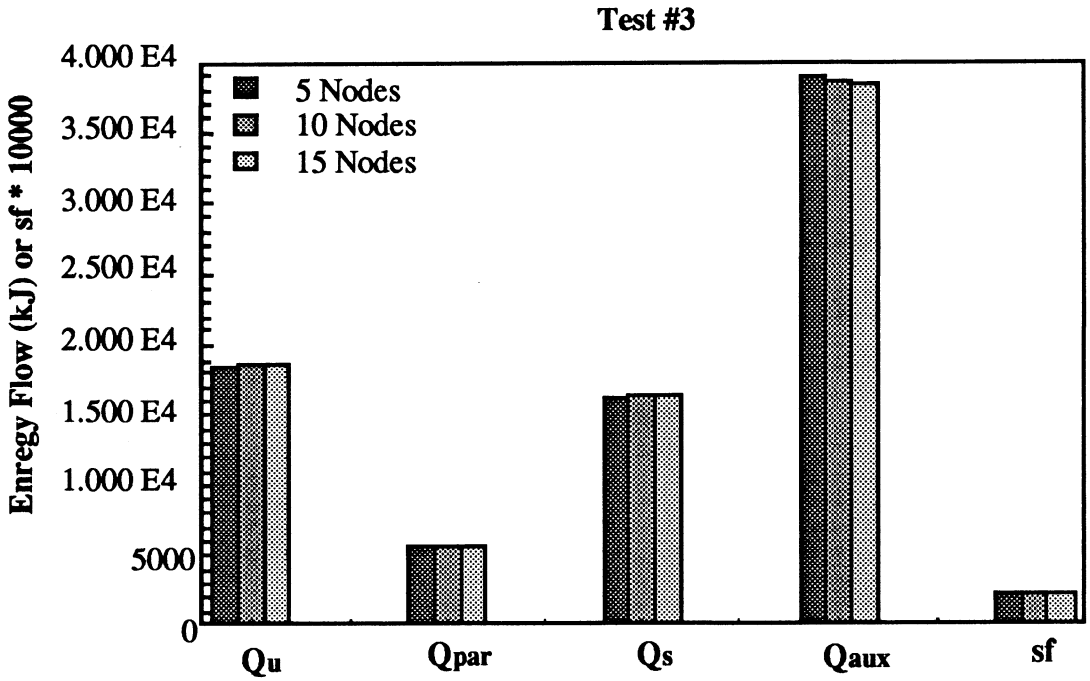


Figure 4.8 Simulation Results as a Function of the Number of Solar-Tank Nodes

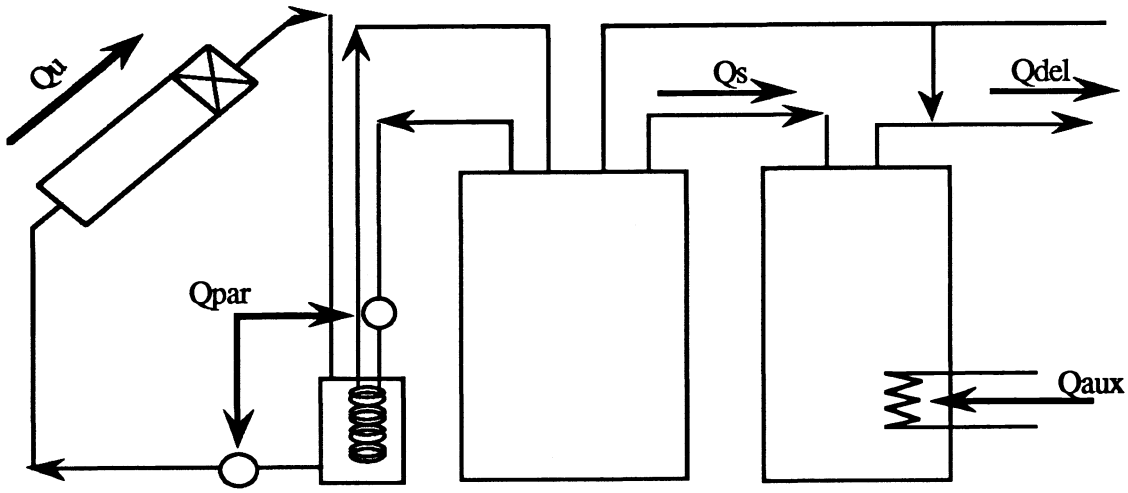


Figure 4.9 SDHW System Energy Flows

4.3.4 Auxiliary Tank

All of the experimental tests use a manufacturer rated "44 gallon" auxiliary storage tank. The actual volume of the auxiliary tank was not measured. It is assumed the actual tank volume is 10% less than the rated volume, resulting in a volume of 143 L. The outside tank height is 1.26 m. The difference between the outer and inner tank dimensions is assumed to be 0.12 m (recall the solar tanks' inner and outer dimension difference is 0.12 m). CSU calculated the tank's heat loss coefficient by the following equation:

$$UA = \frac{Q}{\Delta T} \tag{4.6}$$

where

Q = Energy input required to maintain the tank at a constant temperature

ΔT = Average tank temperature minus average ambient temperature

The calculated heat loss coefficient is thus 1.9 W/°C (1.13 W/m^2-°C). The tank is modeled as being fully mixed.

4.3.5 Drain-Back Tank

The experimental tests use a manufacturer rated "8 gallon" drain-back tank. The tank's outer height is measured to be 0.6 m. No other information about the tank is available. It is again assumed the tank's actual volume is 10% less than the rated volume, and the inner height is 2 inches less than the outer height due to the tank's wall and insulation thickness. The tank volume and height used for simulation purposes are thus 27 L and 0.55 m, respectively. A guess of 0.67 W/°C (1.27 W/°C-m^2) was made as to the tank's loss coefficient.

4.3.6 Pipes

The TRNSYS simulations only considered the pipe losses between the collector and drain-back tank. Piping in the system other than that of the collector loop is relatively small. The pipe's inner diameter is 0.785" whereas the simulations mistakenly used a 0.75" (0.01905 m) diameter. The pipe lengths leading to and from the collector are 11.77 and 13.41 m, respectively. The pipe's insulation's rated conduction heat-loss resistance is 4.7 hr-ft-°F/Btu (1.21 W/°C-m^2). The simulations used a greater heat loss coefficient of 5.78 W/°C-m^2. The pipe heat loss effect on simulation results is discussed in Section 4.7.

4.3.7 Heat Exchanger

A submerged heat exchanger coil is located within the CSU drain-back tank. The submerged-exchanger is modeled as a constant effectiveness exchanger located outside of the tank. Figure 4.10 illustrates the heat exchanger/drain-back tank modification made for simulation purposes. The heat exchanger effectiveness for the various flow rate combinations was calculated from the recorded instantaneous test data shown in Figure 4.11.

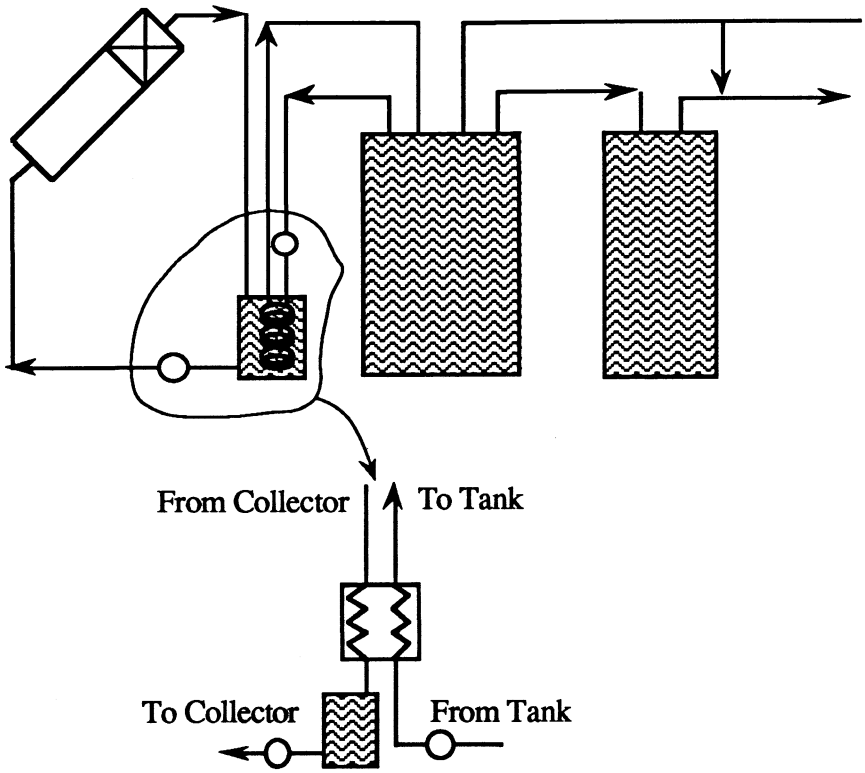


Figure 4.10 Experimental vs. Simulated Heat Exchanger/Drain-Back Tank Arrangement

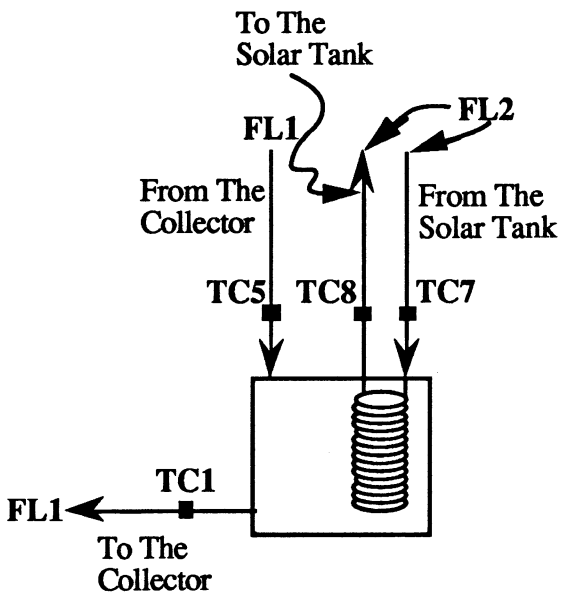


Figure 4.11 CSU Drain-Back Tank and Heat Exchanger

By definition,

$$\varepsilon = \frac{q}{q_{\text{max possible}}}$$

$$= \frac{FL1*\rho_{5-1}*c_{p_{5-1}}*(TC5 - TC1)}{\max[FL1*\rho_{5-1}*c_{p_{5-1}}, FL2*\rho_{7-8}*c_{p_{7-8}}]*(TC5 - TC7)} \quad (4.7)$$

$$= \frac{FL2*\rho_{7-8}*c_{p_{7-8}}*(TC8 - TC7)}{\max[FL1*\rho_{5-1}*c_{p_{5-1}}, FL2*\rho_{7-8}*c_{p_{7-8}}]*(TC5 - TC7)} \quad (4.8)$$

The flow rates FL1 and FL2, and the temperature differences TC5-TC1 and TC8-TC7 are all recorded during the experimental tests. The TC5-TC7 quantity may be obtained by subtraction of recorded thermocouple measurements (recorded temperature differences such as TC1-TC5 and TC8-TC7 are thermopile measurements). The final test day data were used in the effectiveness calculations.

Test Ref #	Collector Side ε	Tank Side ε
7	0.44	0.44
11	0.38	0.38
1	0.41	0.39
13	0.43	0.41
4	0.40	0.41
16	0.39	0.38
6	0.60	0.56
10	0.53	0.53

Table 4.2 Heat Exchanger Effectivenesses

Figures 4.12 and 4.13 show the calculated heat exchanger effectiveness values for tests #7 and #13. The collector side flow rate values (i.e., the dots) were calculated using Equation (4.7). The tank side values (i.e., the squares) were calculated using Equation (4.8). Generally the effectiveness values based upon the collector side data are greater than those based upon the tank side data. Some of the difference between the effectiveness values is due to heat losses form the tank. Tank heat losses decrease TC1 and

hence increase the TC5-TC1 temperature difference in the numerator of Equation (4.7). The increased numerator results in a larger than actual effectiveness. In contrast, tank heat losses act to reduce TC8, and hence reduce the numerator of Equation (4.8). The reduced numerator results in a smaller than actual effectiveness. A summary of the time averaged collector-side and tank-side effectiveness values is presented in Table 4.2. An average of the two calculated effectiveness values is more indicative of the true effectiveness than either of the individually calculated values. Figures 4.14 and 4.15 show the calculated effectivenesses for various flow rate combinations.

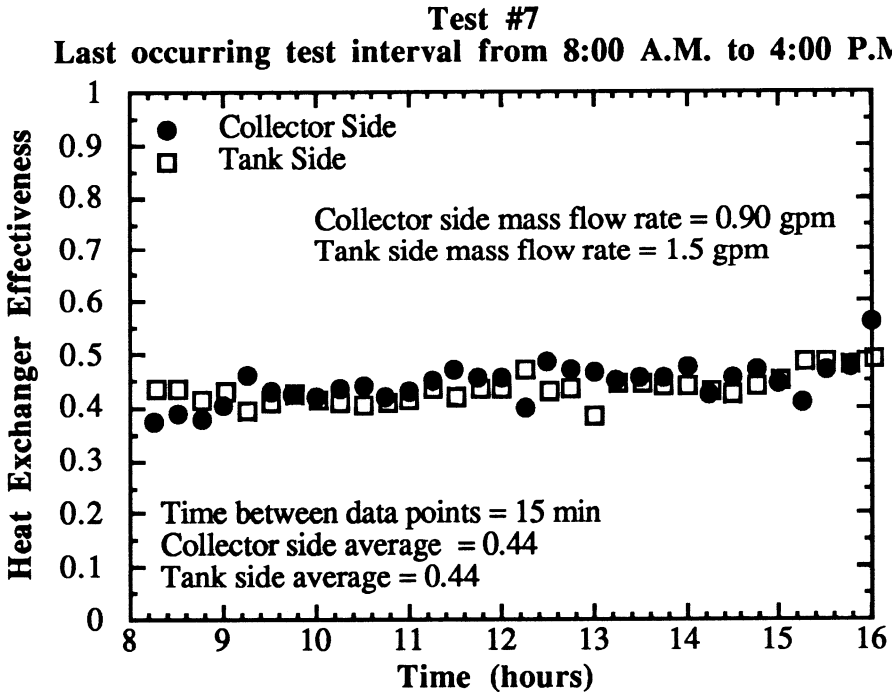


Figure 4.12 Calculated Heat Exchanger Effectiveness for Test #7

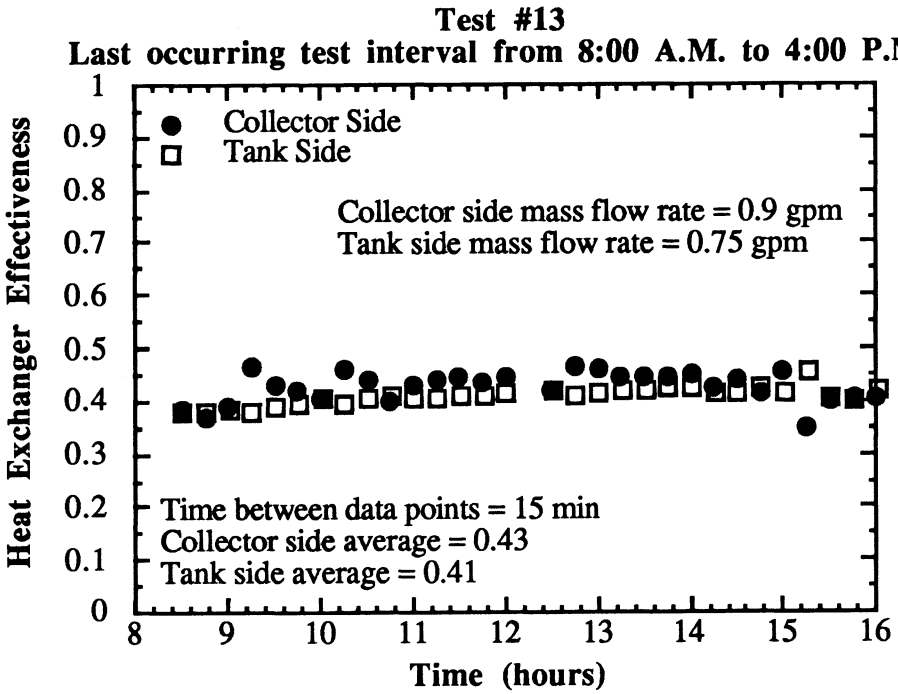
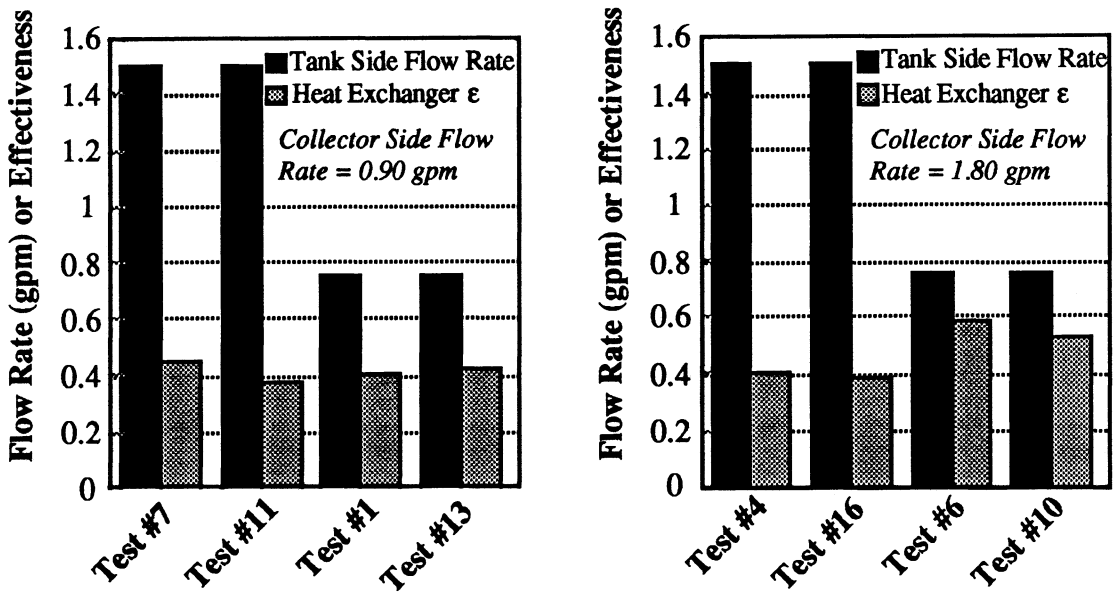


Figure 4.13 Calculated Heat Exchanger Effectiveness for Test #13



Figures 4.14 and 4.15 Summary of Calculated Heat Exchanger Effectivenesses

4.3.8 Pumps

The total energy consumption for the collector and tank side pumps is 433.8 kJ/hr and 241.2 kJ/hr, respectively, independent of mass flow rate. The standard TRNSYS pump model, Type 3, was modified to account for a fluid temperature increase due to the transferal of pump work to the fluid. It is assumed 85% of the pump work acts to raise the fluid temperature (i.e., $\eta = 0.85$ in Equation 4.1). The maximum fluid temperature increase due to pump work across the collector side pump is 0.4 °C. Similarly, the maximum temperature increase across the tank side pump is 0.3 °C.

4.3.9 Gain Controller

The CSU upper controller temperature input is calculated as 75% of the boiler exit temperature and 25% of the simulated collector's theoretical static temperature. A second collector array having no flow through it is added to the TRNSYS deck to obtain the collector static temperature. The lower controller temperature is the solar tank exit temperature on the collector loop side. The turn on and turn off dead band temperatures are 11.11 °C and 2.78 °C respectively.

4.3.10 Load Controller

The TRNSYS-compatible FORTRAN subroutine listed in Appendix A was written to "turn on" and "turn off" the load draw. A load draw occurs at 8:00, 12:00, and 17:00 each day. A draw continues until the following criteria is met:

$$\int_{time} \dot{m}_L * c_p \, (T_{set} - T_{mains}) \, d\theta = 16603 \text{ kJ} \qquad (4.9)$$

where

$T_{mains} = 22\,°C$

$$T_{set} \geq 48.9\ °C$$

A constant set temperature of 54 °C is used for the CSU tests. The SRCC specifications call for a flow rate, $\dot{m}_L$, of 0.20 kg/sec. An error in delivered energy on the order of 1% was observed using a simulation time step, $\Delta\theta$, of 0.005 hours. The error is due to the theoretical time at which the load flow is to turn off not coinciding with a simulation time, as shown in Figure 4.16.

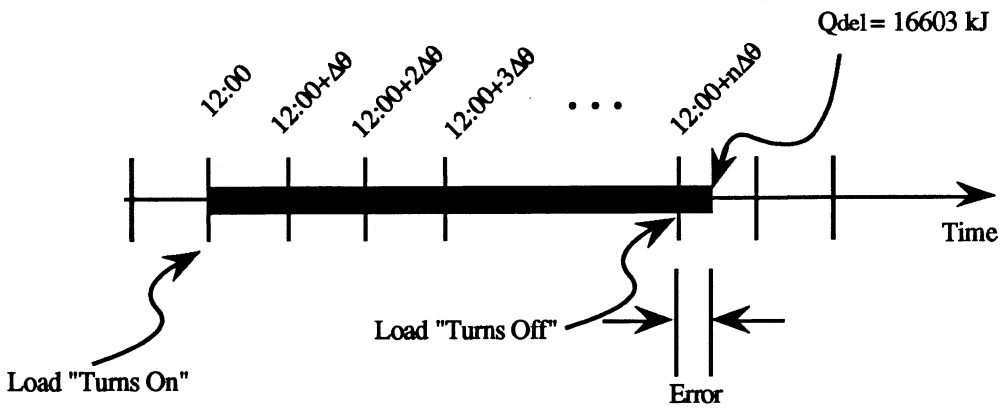


Figure 4.16 Load Draw Error Due to Time Step Size

Load draw energy error may be avoided by slightly modifying the load flow rate. Integrating Equation (4.9) and substituting some of the known values results in the following relationship:

$$\dot{m}_L \left(\frac{kg}{hr}\right) * 4.19 \left(\frac{kJ}{kg\text{-}°C}\right) * \{54 - 22\}\,(°C) * 0.005\,(hr) * n = 16603\,(kJ)$$

or

$$\dot{m}_L * n = 24765.81 \left(\frac{kg}{hr}\right) \qquad (4.10)$$

where

n = number of time steps during the draw period.

Adjusting the mass flow rate to 728.4 kg/sec results in an even number of time steps during each draw (i.e., n = 34) and hence an energy draw error of zero. The modified flow rate of 728.4 kg/sec was used in the TRNSYS simulations. A schematic of the TRNSYS SDHW system is shown in Figure 4.17.

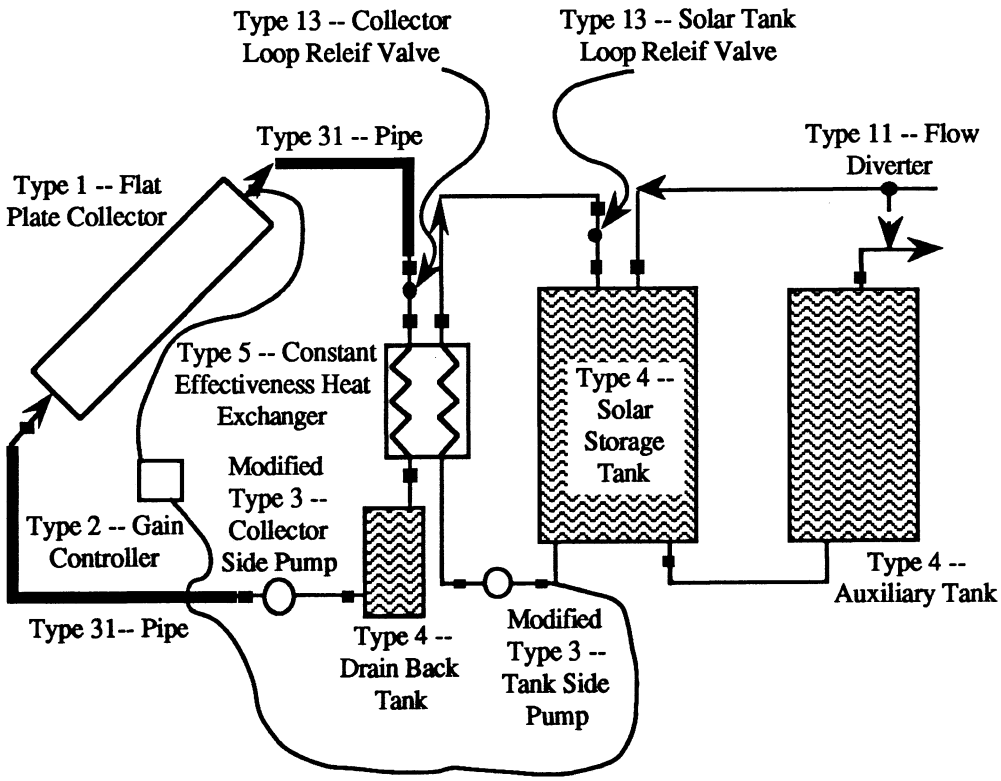


Figure 4.17 TRNSYS SDHW System

4.4 COMPARISON OF RESULTS

Even well-controlled experimental tests will not achieve periodic state conditions. It is possible, on the other hand, for simulations to obtain periodic steady-state conditions. Hence, an initial bias exists when results from a non-periodic steady-state experiment are compared to results form a periodic steady-state simulation. It is assumed a majority of the experimental energy storage occurs in the solar storage tank. The energy stored over the

final test day was calculated by subtracting the energy present in the tank at the end of the last, and second to last test days. The tank energy was taken to equal the capacitance-bulk average temperature product. The results obtained are shown in Figure 4.18.

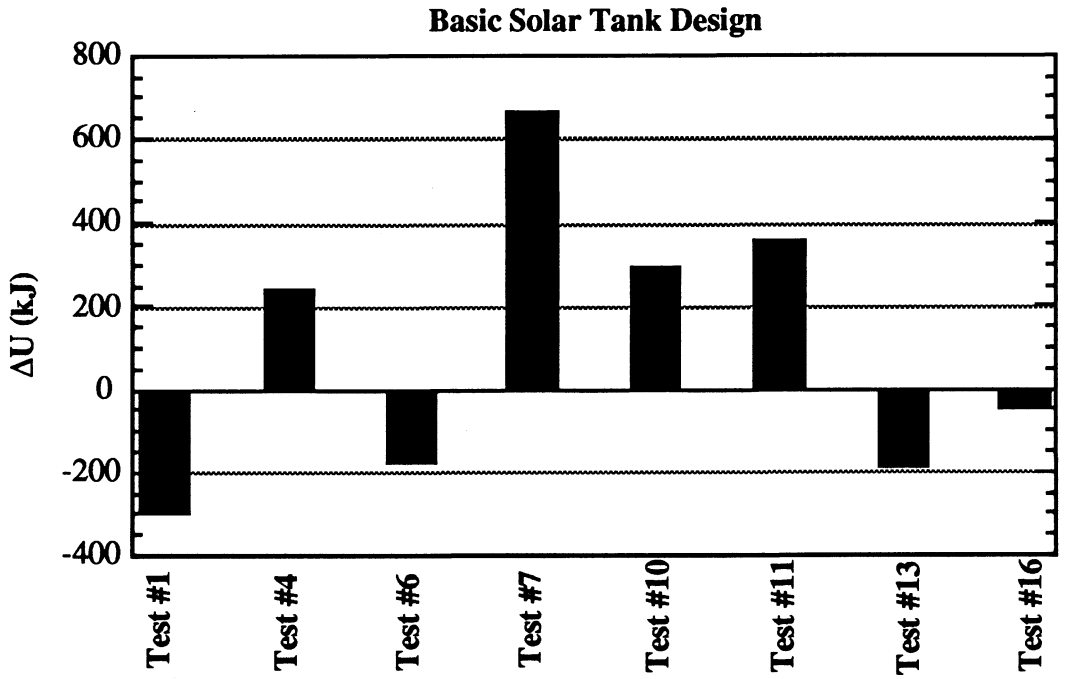


Figure 4.18 Energy Storage Within the Solar Tank

Addition of the storage terms, ΔU, to the measured delivered solar energy, Q_S, results in a better indication of the true collected solar energy. If, for example, the energy delivery form the solar tank is measured to be 16000 kJ, and the solar tank storage is calculated as 600 kJ, the total solar energy collected over the day is 16000+600 kJ or 16600 kJ. It is the total collected solar energy, 16600 kJ in this example, which may potentially be delivered under periodic steady-state conditions. The following basic solar tank test comparisons take into account the energy storage by addition of the storage to the delivered energy. The percentage of modification for the eight tests is listed in Table 4.3.

The range of modification varies between an upper value of +2.7% to a lower value of -1.8%. Sufficient data to determine the eight experimental diffused solar tank test integrated energy storage terms are not available. Hence, the diffused solar tank test comparisons include non-corrected experimental test results.

The first set of comparisons, Figures 4.19 through 4.23 and Table 4.4, concerns the tests involving the basic solar tank. Figure 4.19 compares the important energy flows and solar fraction for the arbitrarily selected test #11. The energy flows are defined as shown in Figure 4.9, and the solar fractions are calculated using Equation (4.5). Figures 4.20 through 4.23 compare various experimental and simulated results over the entire field of eight tests. The experimental solar heat losses of Figure 4.20 were calculated by Kleinbach [1990] via the following equation:

Test Ref #	Modification to Q_s (%)
1	-1.8
4	1.4
6	-0.7
7	2.7
10	1.7
11	2.1
13	-0.8
16	-0.2

Table 4.3 Energy Storage Modification to Q_s

$$Q_{loss} = Q_{in} - Q_s - \Delta U \tag{4.11}$$

The energy delivered from the tank, Q_s, is one of the reported experimental results. The tank energy input, Q_{in} was calculated by Kleinbach by numerically integrating recorded instantaneous test measurements via the trapezoid rule. Kleinbach concluded the calculated losses, Q_{loss}, for tests #1 and #6 should be about 2500 kJ and 4000 kJ, respectively, to agree with the losses observed in the other tests. The calculated losses of approximately 800 and 2000 kJ for tests #1 and #6 are indicative of errors in either Q_{in}, Q_{del}, or ΔU

outside of the range of specified measurement errors. A tabular listing of the test results is found in Table 4.4.

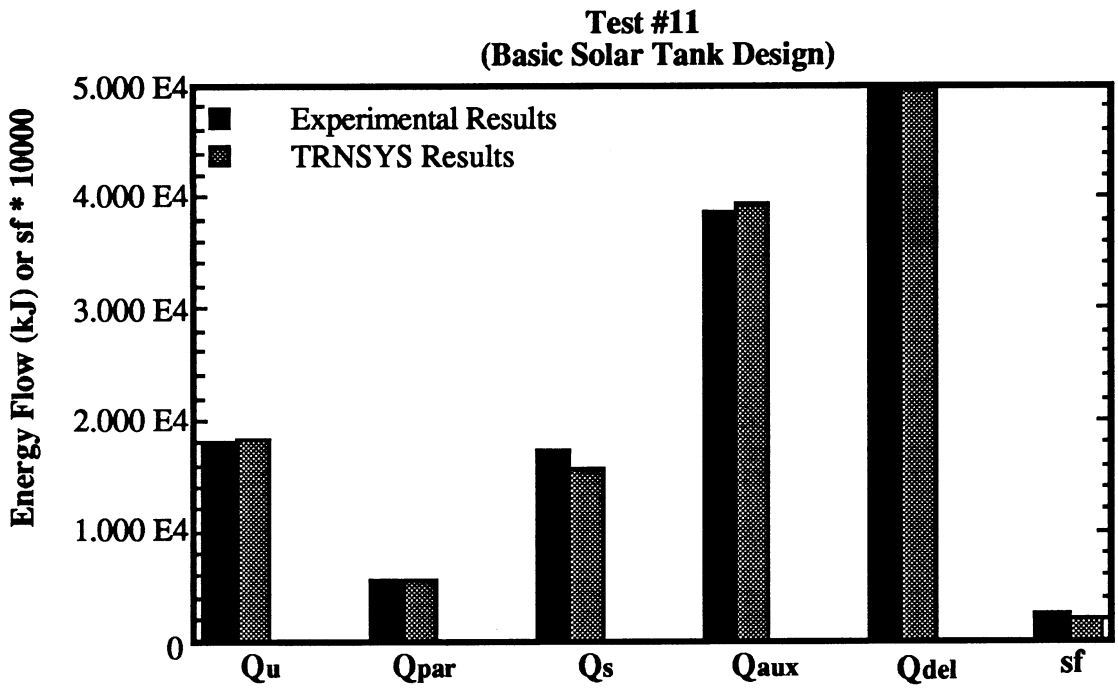


Figure 4.19 Experimental vs. Simulated Test Results

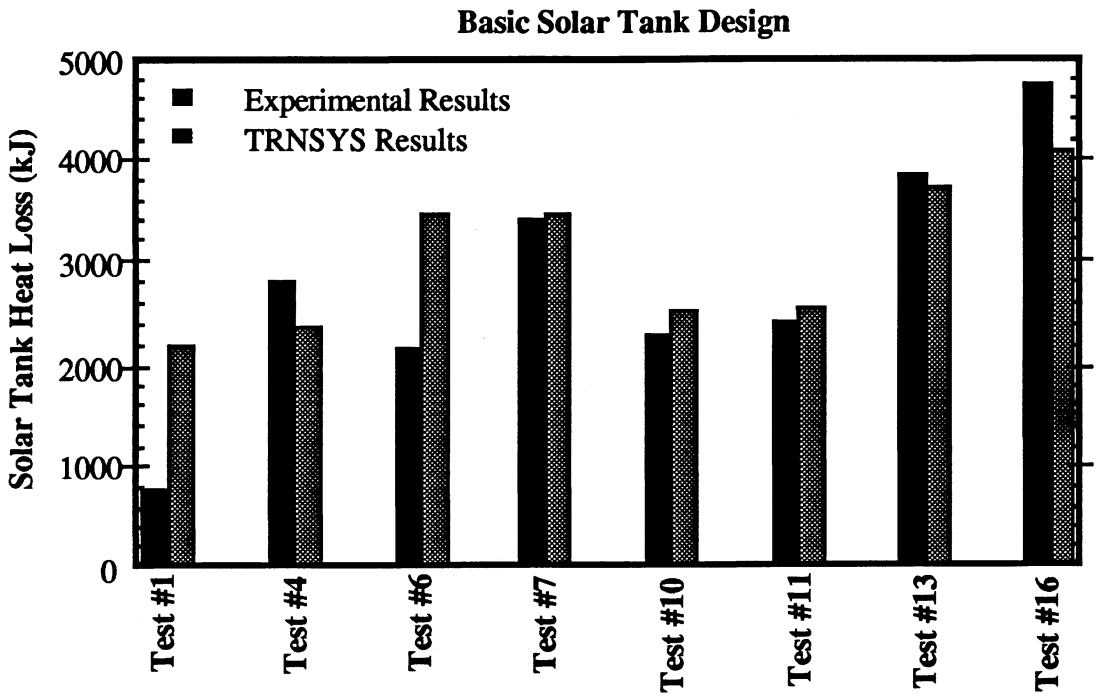


Figure 4.20 Experimental vs. Simulated Tank Heat Losses

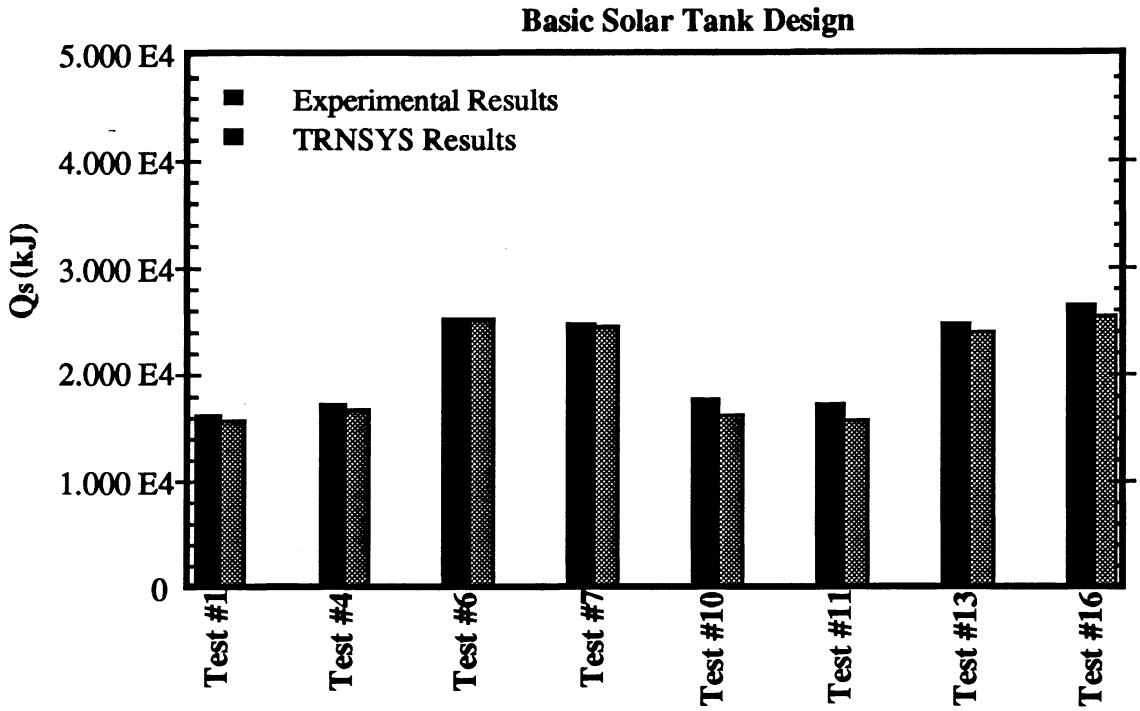


Figure 4.21 Experimental vs. Simulated Delivered Solar Energy

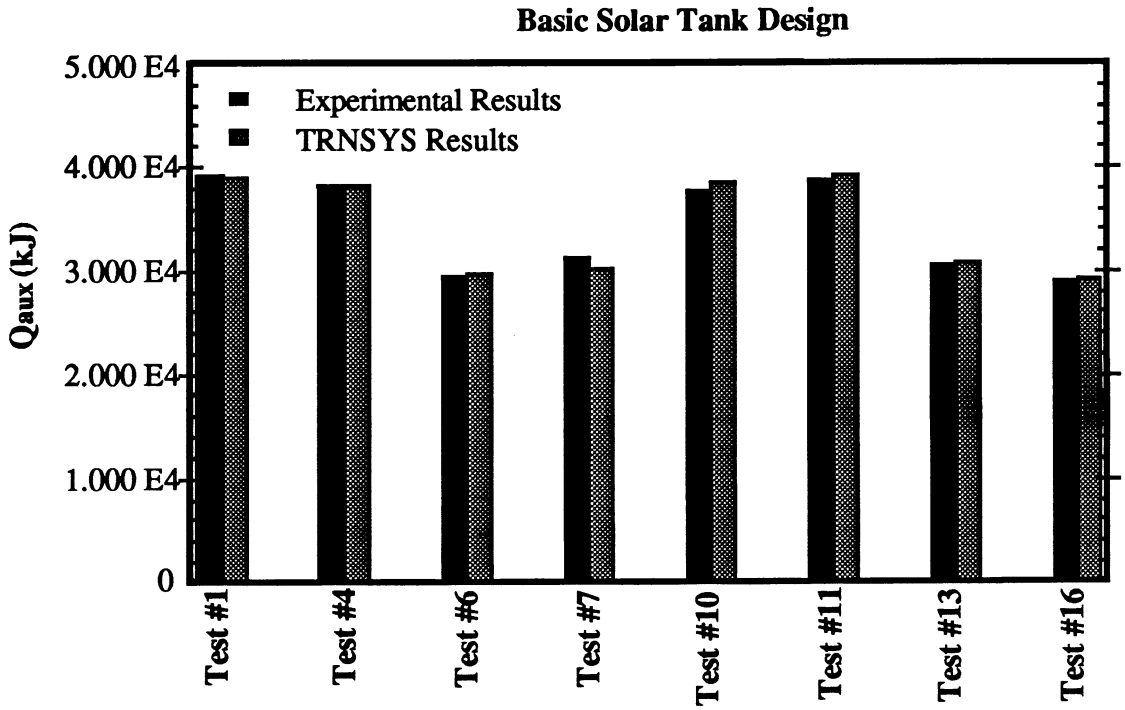


Figure 4.22 Experimental vs. Simulated Auxiliary Energy Input

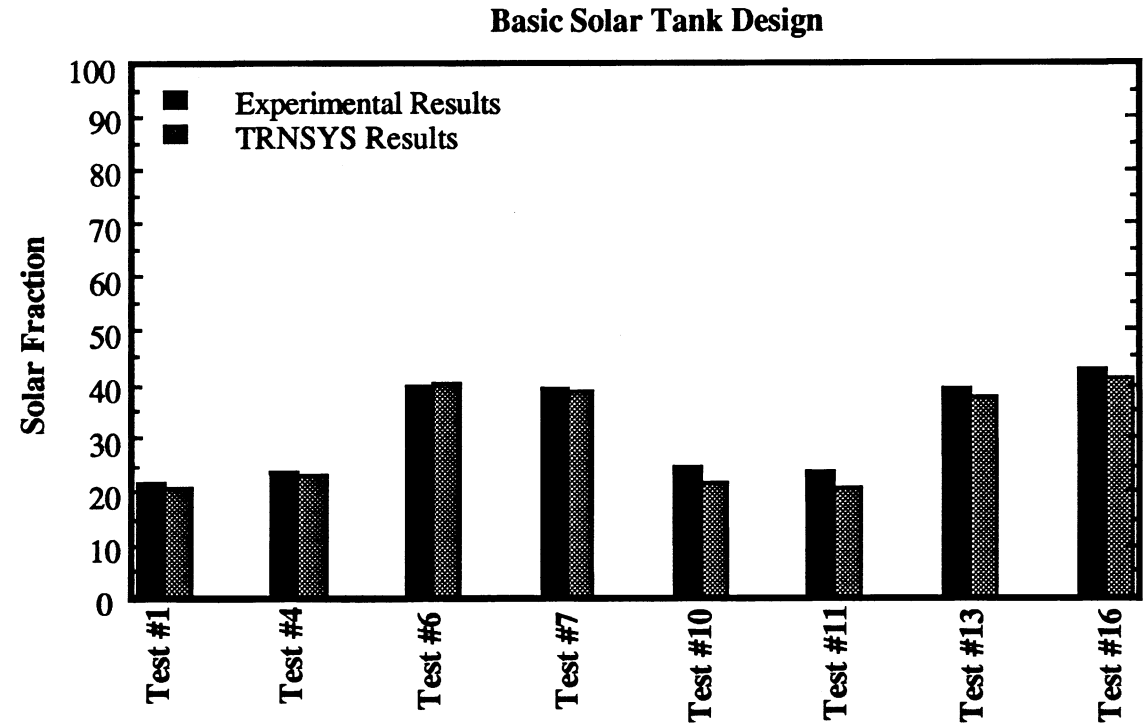


Figure 4.23 Experimental vs. Simulated Solar Fractions

Experimental vs. Simulated SDHW Performance (Basic Solar Tank Design)

Test #1	Q_u (kJ)	Q_s (kJ)	ΔU (kJ)	Q_{par} (kJ)	Q_{net} (kJ)	Q_{aux} (kJ)	Q_{del} (kJ)	sf (%)
CSU	17554±1103	16647±173	-297	5435±66	11212±185	39329±216	49820±393	22.5
Cr CSU		16350			10915			21.9
TRNSYS	18130	15900	0	5535	10365	39140	49780	20.8
Cr CSU-TRN	-576	450	-297	-100	550	189	40	1.1
Test #4								
CSU	18401±1031	16996±174	241	5532±64	11464±186	38182±207	49821±393	23.0
Cr CSU		17237			11705			23.5
TRNSYS	18870	16850	0	5400	11450	38180	49780	23.0
Cr CSU-TRN	-469	387	241	132	255	2	41	0.5
Test #6								
CSU	30681±1657	25471±231	-175	5463±64	20008±240	29529±162	49815±393	40.2
Cr CSU		25296			19833			39.8
TRNSYS	30970	25380	0	5400	19980	29660	49780	40.1
Cr CSU-TRN	-289	-84	-175	63	-147	-131	35	-0.3
Test #7								
CSU	29794±1647	24260±223	664	5369±64	18890±232	31449±171	49829±393	37.0
Cr CSU		24924			19554			39.2
TRNSYS	30370	24600	0	5400	19200	30440	49780	38.6
Cr CSU-TRN	-576	324	664	-31	354	1009	49	0.7
Test #10								
CSU	18083±1051	17717±179	295	5724±67	11993±191	37753±204	49822±392	24.1
Cr CSU		18012			12288			24.7
TRNSYS	18650	16410	-1	5538	10872	38630	49780	21.9
Cr CSU-TRN	-567	1602	296	186	1416	-877	42	2.8
Test #11								
CSU	18088±1069	16968±173	360	5656±67	11312±186	38675±207	49816±392	20.7
Cr CSU		17328			11672			23.4
TRNSYS	18290	15840	-1	5532	10308	39200	49780	20.7
Cr CSU-TRN	-202	1488	361	124	1364	-525	36	2.7
Test #13								
CSU	29796±1658	25113±230	-188	5408±65	19706±239	30635±165	49811±393	39.6
Cr CSU		24925			19518			39.2
TRNSYS	30140	24150	-1	5400	18750	30890	49780	37.7
Cr CSU-TRN	-344	775	-187	8	768	-255	31	1.5
Test #16								
CSU	31746±1630	26712±241	-47	6484±64	21228±249	29013±158	49822±393	42.6
Cr CSU		26665			21181			42.5
TRNSYS	31650	25690	0	5400	20290	29350	49780	40.8
Cr CSU-TRN	96	975	-47	1084	891	-337	42	1.8

Table 4.4 Basic Solar Tank Test Results

The following set of comparisons, Figures 4.24 through 4.27 and Table 4.5, concern the tests involving the manifold solar tank. Figure 4.24 compares the energy

flows and solar fraction for the arbitrarily selected test #9. The energy flows are defined as shown in Figure 4.10 and the solar fractions are calculated using Equation (4.5). Figures 4.25 through 4.27 compare various experimental and simulated results over the entire field of eight tests. The experimental delivered solar energy values, Q_S, of Figures 4.24 and 4.25 have not been modified to take into account energy storage within the solar tank. A tabular listing of the test results is found in Table 4.5.

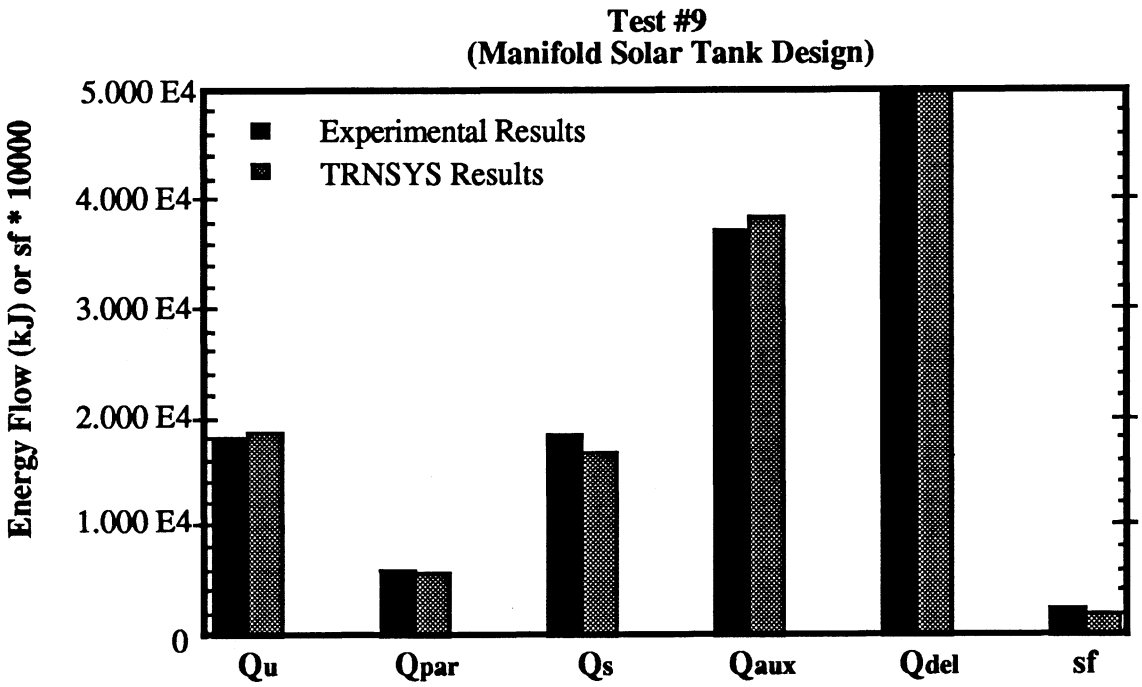


Figure 4.24 Experimental vs. Simulated Test Results

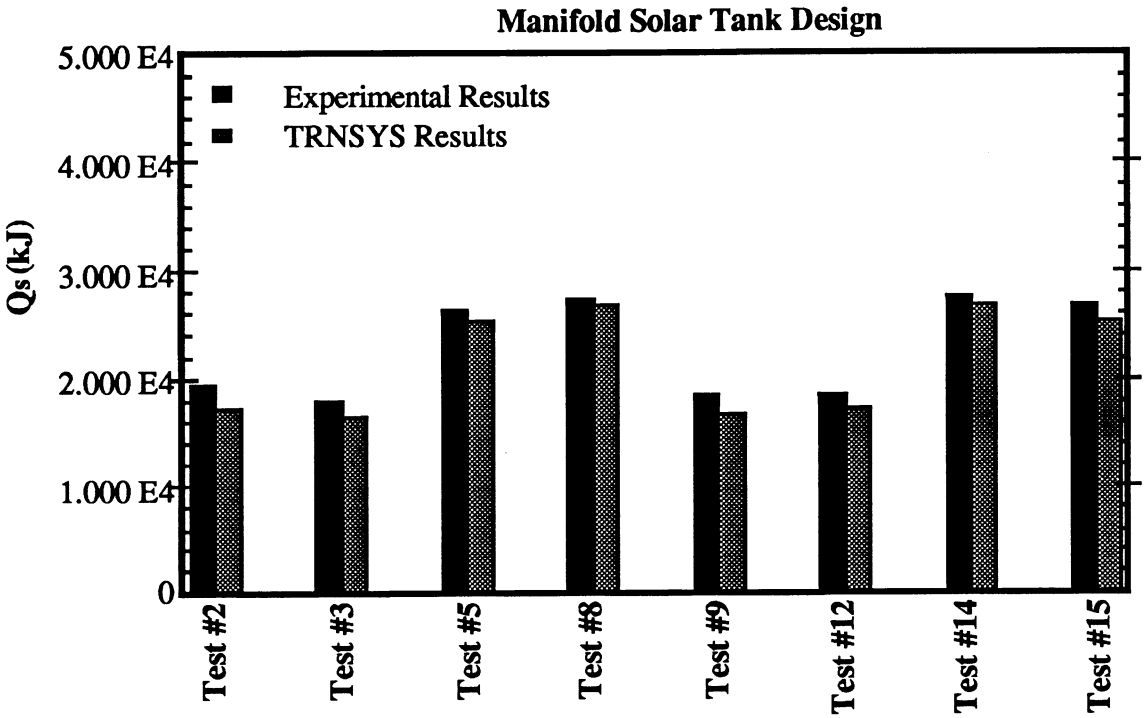


Figure 4.25 Experimental vs. Simulated Delivered Solar Energy

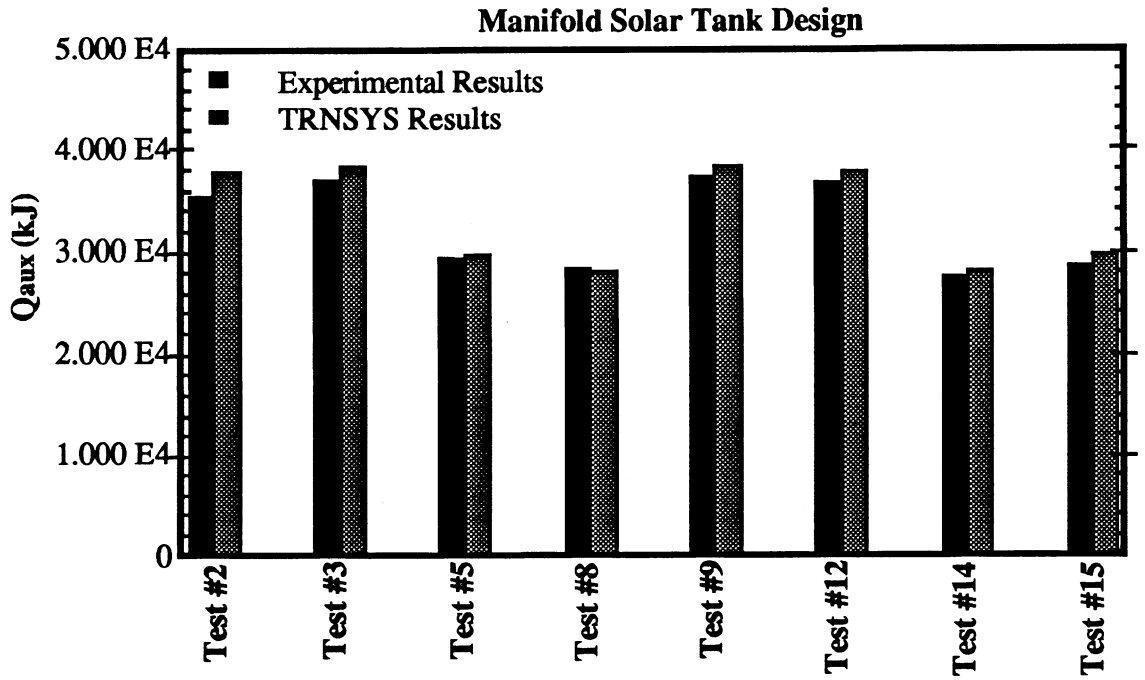


Figure 4.26 Experimental vs. Simulated Auxiliary Energy Input

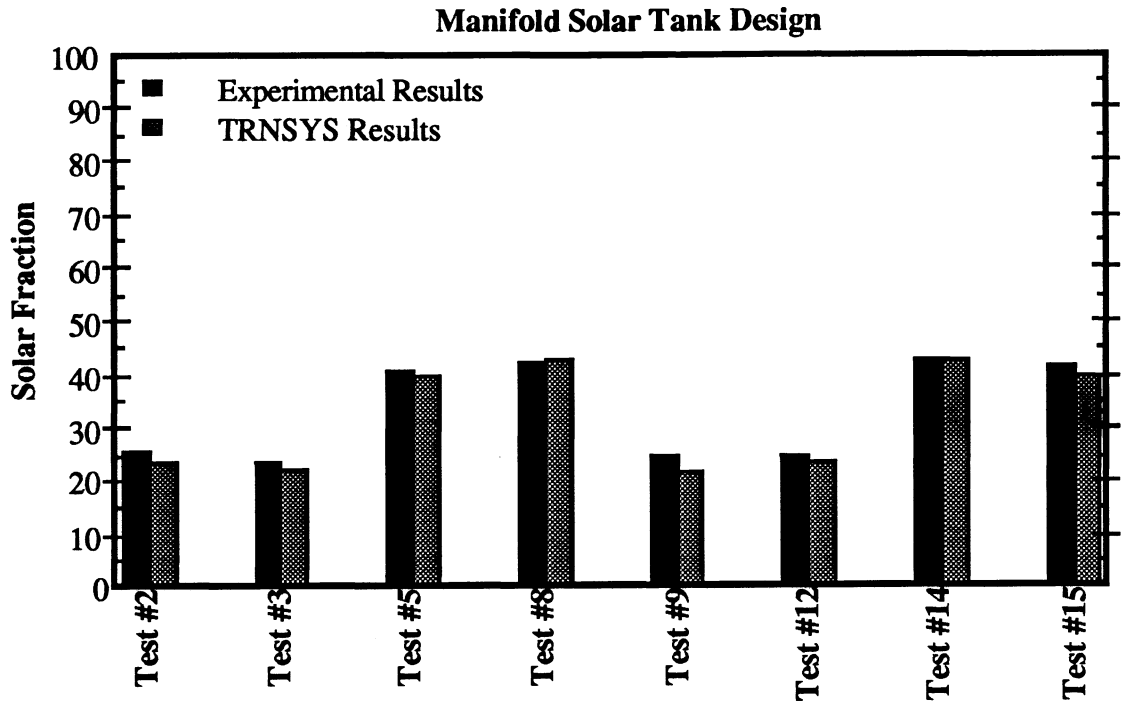


Figure 4.27 Experimental vs. Simulated Solar Fractions

Experimental vs. Simulated SDHW Performance
(Manifold Solar Tank Design)

Test #2	Q_u (kJ)	Q_s (kJ)	ΔU (kJ)	Q_{par} (kJ)	Q_{net} (kJ)	Q_{aux} (kJ)	Q_{del} (kJ)	sf (%)
CSU	18196±1103	19545±192	?	6564±77	12981±207	35480±190	49815±393	26.0
Cr CSU		19545			12981			26.1
TRNSYS	18980	17350	0	5440	11910	37690	49780	23.9
Cr CSU-TRN	-784	2195	0	1124	1071	-2210	35	2.1
Test #3								
CSU	18254±1113	17975±181	?	6028±72	11946±195	37107±198	49811±393	24.0
Cr CSU		17975			11946			24.0
TRNSYS	18490	16470	0	5400	11070	38570	49780	22.2
Cr CSU-TRN	-236	1505	0	628	876	-1463	31	1.8
Test #5								
CSU	29369±1756	26382±239	?	6051±72	20331±249	29312±157	49811±393	40.8
Cr CSU		26382			20331			40.8
TRNSYS	30620	25370	0	5400	19970	29670	49780	40.1
Cr CSU-TRN	-1251	1012	0	651	361	-358	31	0.7
Test #8								
CSU	30047±1736	27273±245	?	6202±72	21072±255	28370±153	49811±393	42.3
Cr CSU		27273			21072			42.3
TRNSYS	32110	26830	0	5400	21430	28210	49780	43.1
Cr CSU-TRN	-2063	443	0	802	-358	160	31	-0.8
Test #9								
CSU	18223±1094	18459±184	?	6066±72	12394±198	37265±199	49811±391	24.9
Cr CSU		18459			12394			24.9
TRNSYS	18680	16690	0	5771	10919	38350	49780	21.9
Cr CSU-TRN	-457	1769	0	295	1475	-1085	31	2.9
Test #12								
CSU	19084±1118	18482±184	?	6227±72	12255±198	36684±193	49822±392	24.6
Cr CSU		18482			12255			24.6
TRNSYS	19200	17250	-1	5400	11850	37790	49780	23.8
Cr CSU-TRN	-116	1232	1	827	405	-1106	42	0.8
Test #14								
CSU	31304±1744	27686±247	?	6213±72	21474±257	27642±148	49815±393	43.1
Cr CSU		27686			21474			43.1
TRNSYS	32280	26890	0	5400	21490	28150	49780	43.2
Cr CSU-TRN	-976	796	0	813	-16	-508	35	-0.1
Test #15								
CSU	30125±1754	26956±242	?	6079±72	20877±253	28681±153	49815±392	41.9
Cr CSU		26956			20877			41.9
TRNSYS	31160	25360	0	5400	19960	29680	49780	40.1
Cr CSU-TRN	-1035	1596	0	679	917	-999	35	1.8

Table 4.5 Manifold Solar Tank Test Results

4.5 SIMULATION SENSITIVITY

Simulations were performed over a range of values for five of the parameters. The parameters selected for parametric study are:

1. Heat exchanger effectiveness
2. Collector loop flow rate
3. Tank loop flow rate
4. Solar tank heat loss coefficient
5. Collector loss coefficient

The five parameters were individually varied at -40%, -10%, +10%, and +40% of their nominal values. A change in performance in simulation results was defined at follows:

$$\% \text{ Change} = \frac{\text{Changed Result - Base Result}}{\text{Base Result}} \tag{4.12}$$

The simulation results are illustrated in Figures 4.28 through 4.32.

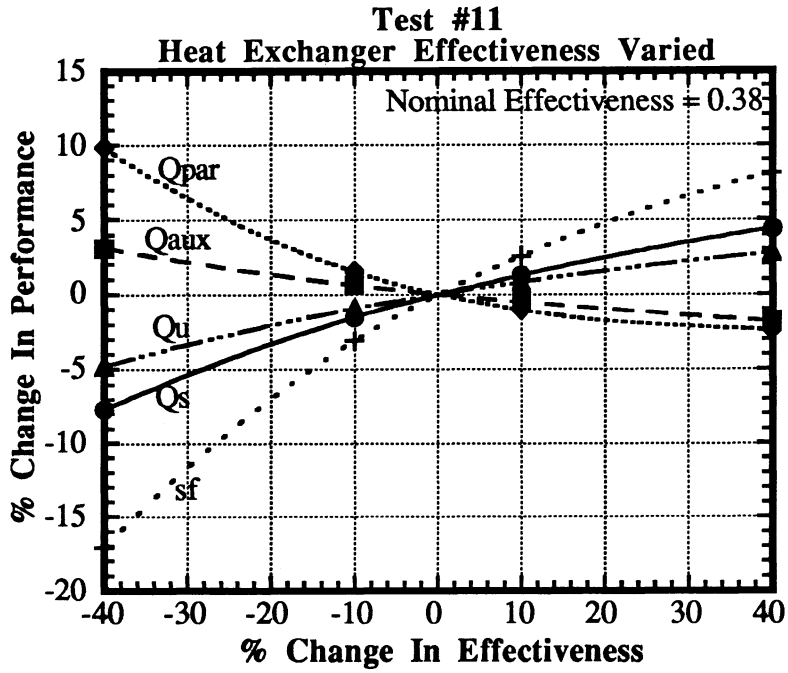


Figure 4.28 Variation in Performance with Heat Exchanger Effectiveness

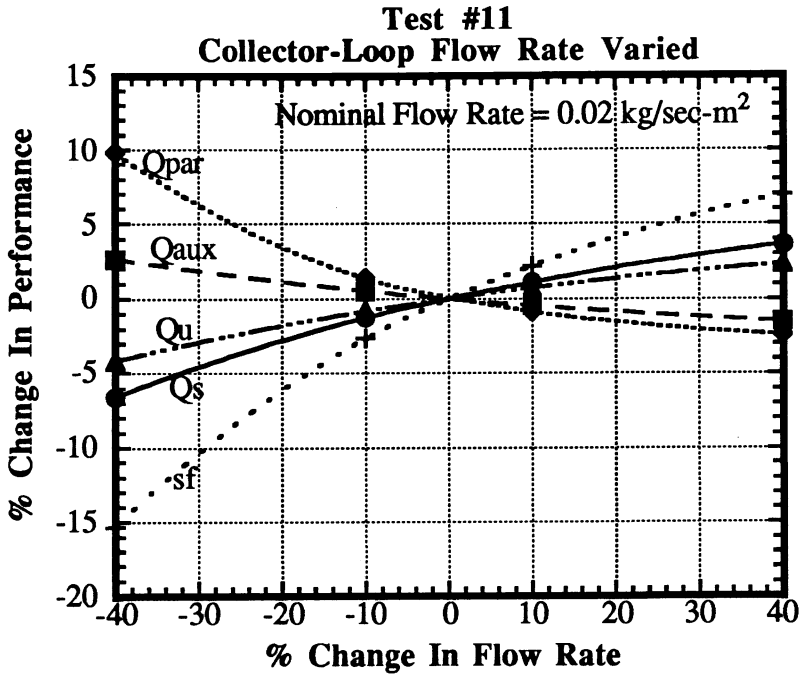


Figure 4.29 Variation in Performance with Collector-Loop Flow Rate

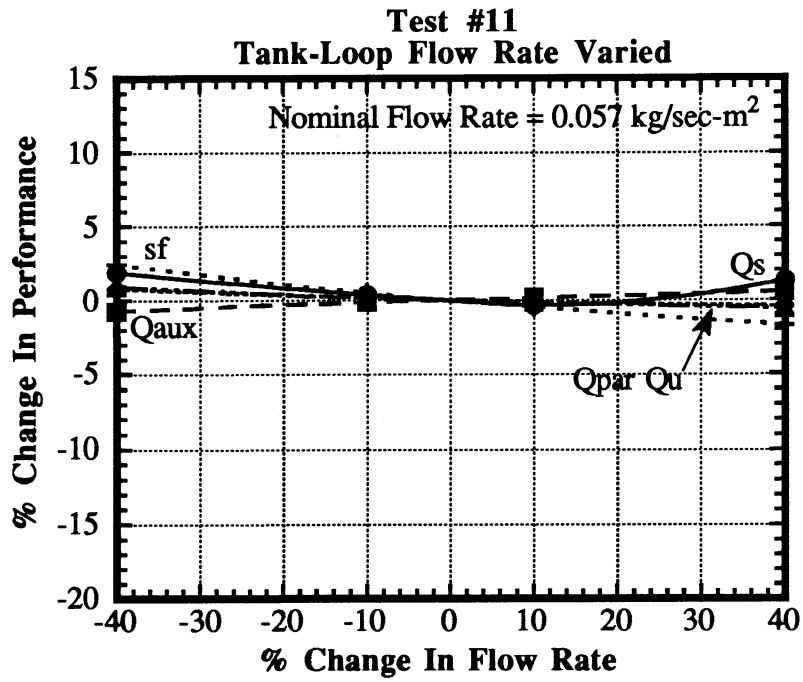


Figure 4.30 Variation in Performance with Tank-Loop Flow Rate

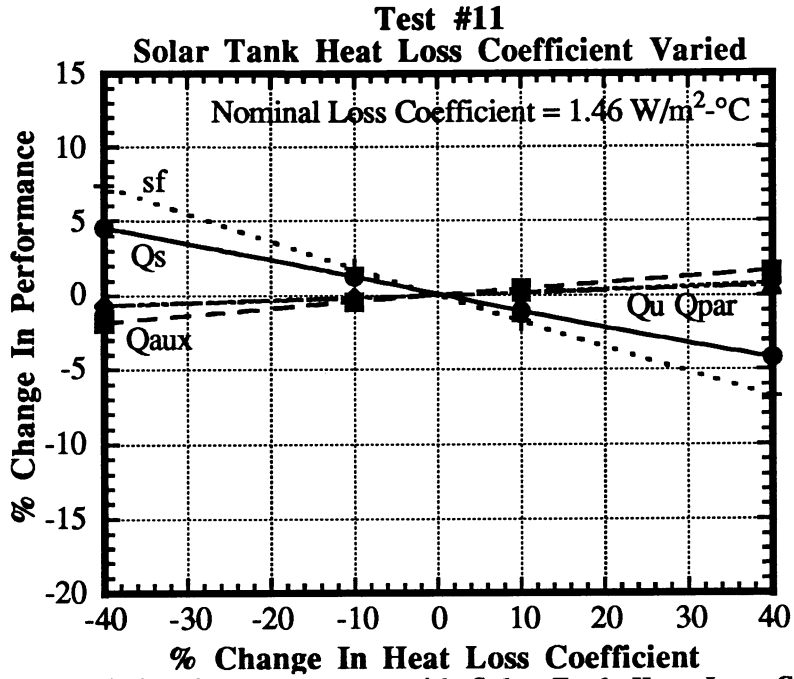


Figure 4.31 Variation in Performance with Solar Tank Heat Loss Coefficient

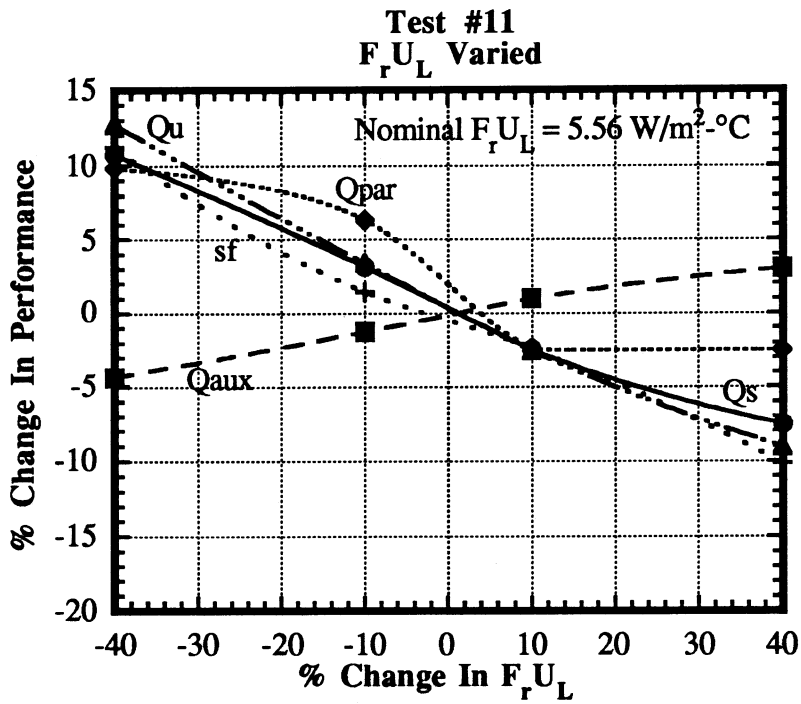


Figure 4.32 Variation in Performance with Collector Heat Loss Coefficient

In addition, simulations were performed with 12 of the system parameters being reduced one at a time by 10% in order the investigate the performance sensitivity to these parameters. The parameters investigated are:

Solar tank's heat loss coefficient per unit area (U_{solar})
Solar tank's volume (V_{solar})
Heat exchanger effectiveness (Hx eff)
Collector gain coefficient at normal irradiance ($F_r(\tau\alpha)_n$)
Collector loss coefficient per unit area (F_rU_L)
Collector area per panel (Area/Panel)
Collector loop flow rate (Coll Flow)
Tank loop flow rate (Tank Flow)
Percentage of pump work which acts to raise the fluid's temperature (Pump %)
Drain-back tank's heat loss coefficient per unit area ($U_{d\text{-}b}$)
Auxiliary tank's volume (V_{aux})

Tables 4.6 and 4.7 show the sensitivity results obtained for tests #11 and #6, respectively. The top set of numbers in each table pertain to the parameter values used. The parameter decreased by 10% is indicated by **bold** face. For example, the first simulation used a solar heat loss coefficient, U_{solar}, of 5.37 kJ/hr-m^2-°C, a solar tank volume of 0.223 m^3, etc... continuing across the first row through an auxiliary volume of 0.143 m^3. None of the first row values are bold faced, indicating these are the base case (i.e., assumed to be the correct) values. The results obtained from the first row simulation are listed in the first row of the second set of numbers. The second row of the first set of numbers indicates the solar tank heat loss coefficient, U_{solar}, was reduced from the base case value of 5.37 to 4.833 kJ/hr-m^2-°C, and all other parameters remained unchanged. The results for this second row simulation are listed in the second row of the second set of numbers, etc.... Tank heat loss coefficients per unit area are changed along with tank

volumes in rows 3 and 13 in order to maintain the same overall heat-loss - surface-area product as in the base case.

The second set of numbers indicate the percentage change in performance. The percentage change is calculated as follows:

$$\% \text{ Change} = \frac{\left\{\dfrac{\text{Changed Result - Base Result}}{\text{Base Result}}\right\}}{\left\{\dfrac{\text{Base Parameter - Changed Parameter}}{\text{Base Parameter}}\right\}} \tag{4.13}$$

Physically, Equation (4.13) is the slope about zero for plots similar the those of Figures 4.28 through 4.32. The base case values are listed in the first row of the second set of numbers. Figures 4.33 through 4.35 visually illustrate the information presented in tables 4.8 and 4.9.

Test #11

Heat loss coefficients (U and F_rU_L) are in units of kJ/hr-m^2-°C

U solar	V solar (m^3)	Hx eff	$F_r(\tau\alpha)_n$	F_rU_L	Area/Panel (m^2)	Coll Flow (kg/hr)	Tank Flow (kg/hr)	Pump %	U_{d-b}	U aux	V aux (m^3)
5.26	0.272	0.38	0.602	20.016	1.852	204.39	340.65	0.85	4.555	4.06	0.143
4.73	0.272	0.38	0.602	20.016	1.852	204.39	340.65	0.85	4.555	4.06	0.143
5.59	**0.245**	0.38	0.602	20.016	1.852	204.39	340.65	0.85	4.555	4.06	0.143
5.26	0.272	**0.34**	0.602	20.016	1.852	204.39	340.65	0.85	4.555	4.06	0.143
5.26	0.272	0.38	**0.542**	20.016	1.852	204.39	340.65	0.85	4.555	4.06	0.143
5.26	0.272	0.38	0.602	**18.014**	1.852	204.39	340.65	0.85	4.555	4.06	0.143
5.26	0.272	0.38	0.602	20.016	**1.667**	204.39	340.65	0.85	4.555	4.06	0.143
5.26	0.272	0.38	0.602	20.016	1.852	**183.95**	340.65	0.85	4.555	4.06	0.143
5.26	0.272	0.38	0.602	20.016	1.852	204.39	**306.59**	0.85	4.555	4.06	0.143
5.26	0.272	0.38	0.602	20.016	1.852	204.39	340.65	**0.77**	4.555	4.06	0.143
5.26	0.272	0.38	0.602	20.016	1.852	204.39	340.65	0.85	**4.1**	4.06	0.143
5.26	0.272	0.38	0.602	20.016	1.852	204.39	340.65	0.85	4.555	**3.65**	0.143
5.26	0.272	0.38	0.602	20.016	1.852	204.39	340.65	0.85	4.555	**4.3**	**0.129**

Changed Variable	Q_s	Q_{aux}	Q_{par}	Q_u	sf
Base Case	15840 (kJ)	39200 (kJ)	5532 (kJ)	18290 (kJ)	20.7 (%)
U_{solar}	9.58	-3.87	-1.77	-1.46	15.98
V_{solar}	-4.58	1.85	-9.51	-2.98	-1.63
Hx eff	-13.64	5.51	13.67	-8.37	-27.99
$F_r(\tau\alpha)_n$	-78.13	31.57	-12.74	-96.31	-112.95
F_rU_L	28.40	-11.48	57.09	30.50	13.32
Area/Panel	-54.04	21.84	15.31	-69.96	-90.98
Coll Flow	-11.36	4.59	12.04	-7.38	-23.62
Tank Flow	3.41	-1.38	1.46	1.48	4.76
Pump %	-14.77	5.97	0.98	4.92	-22.92
U_{d-b}	1.71	-0.69	0.00	-0.99	2.93
U_{aux}	0.00	-12.04	0.00	0.00	0.31
V_{aux}	0.00	-0.24	0.00	0.00	0.32

Table 4.6 Test #11 Sensitivity Results (*100)

Test #6

Heat loss coefficients (U and F_rU_L) are in units of kJ/hr-m^2-°C

U solar	V solar (m^3)	Hx eff	$F_r(\tau\alpha)_n$	F_rU_L	Area/Panel (m^2)	Coll Flow (kg/hr)	Tank Flow (kg/hr)	Pump %	$U_{d\text{-}b}$	U aux	V aux (m^3)
5.37	0.223	0.58	0.602	20.016	1.852	408.78	170.325	0.85	4.555	4.06	0.143
4.833	0.223	0.58	0.602	20.016	1.852	408.78	170.325	0.85	4.555	4.06	0.143
5.7	**0.201**	0.58	0.602	20.016	1.852	408.78	170.325	0.85	4.555	4.06	0.143
5.37	0.223	**0.52**	0.602	20.016	1.852	408.78	170.325	0.85	4.555	4.06	0.143
5.37	0.223	0.58	**0.542**	20.016	1.852	408.78	170.325	0.85	4.555	4.06	0.143
5.37	0.223	0.58	0.602	**18.014**	1.852	408.78	170.325	0.85	4.555	4.06	0.143
5.37	0.223	0.58	0.602	20.016	**1.667**	408.78	170.325	0.85	4.555	4.06	0.143
5.37	0.223	0.58	0.602	20.016	1.852	**367.9**	170.325	0.85	4.555	4.06	0.143
5.37	0.223	0.58	0.602	20.016	1.852	408.78	**153.29**	0.85	4.555	4.06	0.143
5.37	0.223	0.58	0.602	20.016	1.852	408.78	170.325	**0.77**	4.555	4.06	0.143
5.37	0.223	0.58	0.602	20.016	1.852	408.78	170.325	0.85	**4.1**	4.06	0.143
5.37	0.223	0.58	0.602	20.016	1.852	408.78	170.325	0.85	4.555	**3.65**	0.143
5.37	0.223	0.58	0.602	20.016	1.852	408.78	170.325	0.85	4.555	**4.3**	**0.129**

Changed Variable	Q_s	Q_{aux}	Q_{par}	Q_u	sf
Base Case	25380 (kJ)	29660 (kJ)	5400 (kJ)	30970 (kJ)	40.1 (%)
U_{solar}	8.51	-7.28	0.00	-1.45	11.64
V_{solar}	-7.56	6.47	0.00	-5.90	-8.78
Hx eff	-18.09	15.48	0.00	-12.79	-22.17
$F_r(\tau\alpha)_n$	-82.22	70.35	0.00	-94.50	-103.71
F_rU_L	29.43	-25.18	0.00	33.70	38.23
Area/Panel	-53.61	45.87	0.00	-61.39	-67.34
Coll Flow	1.77	-1.52	0.00	1.45	3.08
Tank Flow	-13.12	11.23	0.00	-9.88	-15.86
Pump %	-7.80	6.68	0.00	4.65	-9.10
$U_{d\text{-}b}$	1.78	-1.52	0.00	-1.16	3.08
U_{aux}	0.00	-15.91	0.00	0.00	0.81
V_{aux}	0.00	-0.62	0.00	0.00	0.84

Table 4.7 Test #6 Sensitivity Results (*100)

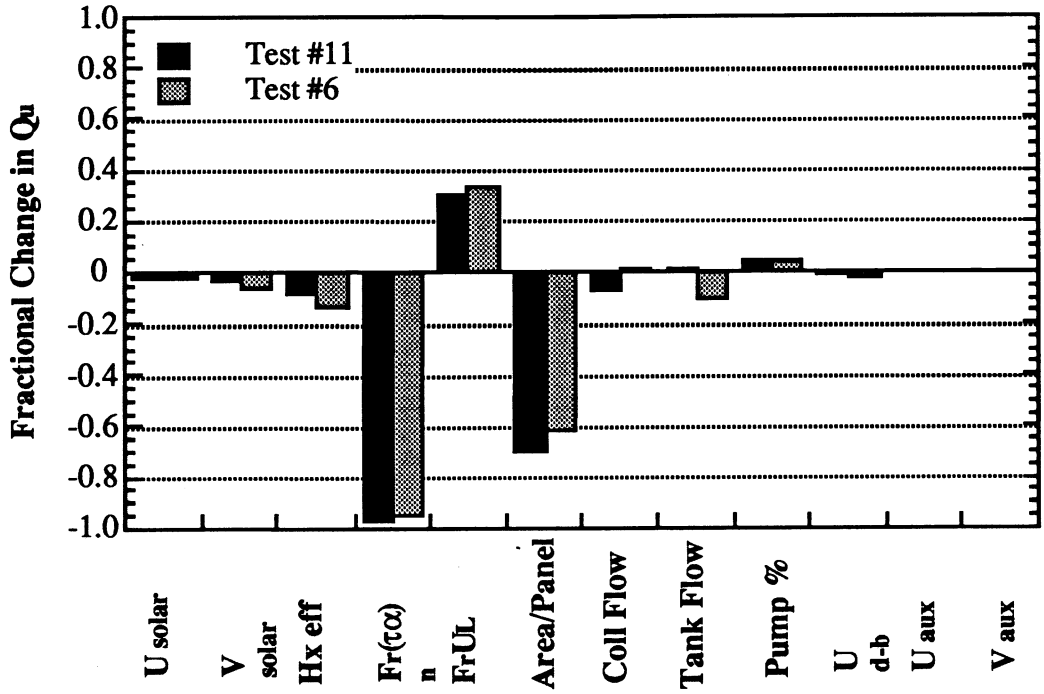


Figure 4.33 Sensitivity of Q_u to Various System Parameters

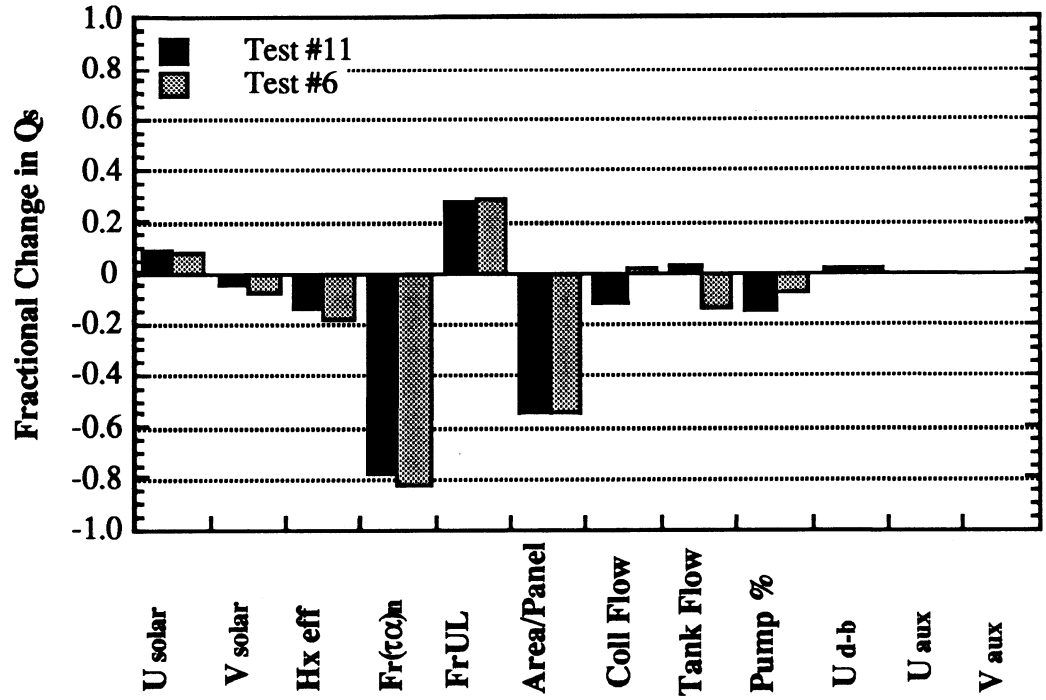


Figure 4.34 Sensitivity of Q_s to Various System Parameters

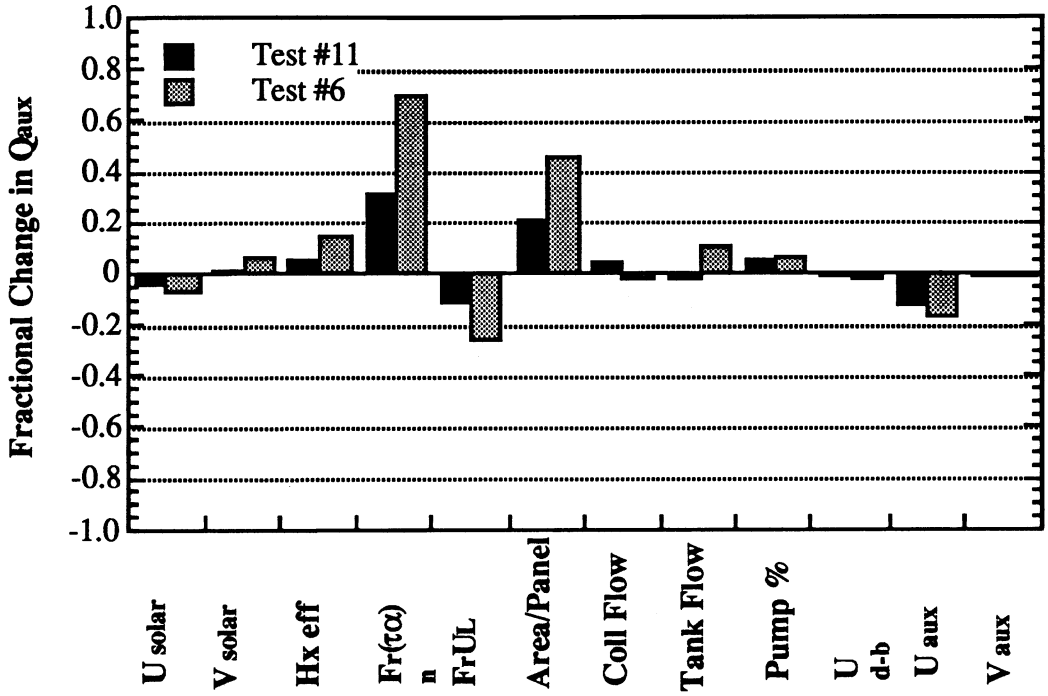


Figure 4.35 Sensitivity of Q_{aux} to Various System Parameters

4.6 DISCUSSION

Five daily integrated energy flows are if interest. The important energy flows are:

Q_u	=	Energy gain across the collector (boiler)
Q_{par}	=	Sum of the two pump works
Q_s	=	Energy delivered from the solar tank
Q_{aux}	=	Auxiliary energy supplied to the auxiliary tank
Q_{del}	=	Total energy, solar plus auxiliary, delivered from the system.

The total delivered energy, Q_{del}, is specified by the testing standards to be 49809 kJ/day. Both the experimental and simulated tests deliver the correct total energy within an acceptable error tolerance. The pump work, Q_{par}, is typically small for a SDHW system. The CSU system under investigation is atypical in that the pump work is substantial, being approximately 10% of the total delivered energy. However, Q_{par} for the system is merely

an indication of pump-on time since the power required by both pumps during operation is constant.

The other three energy quantities (Q_u, Q_s, and Q_{aux}) may by substantial for a typical SDHW system. The Q_u experimental error is approximately ±6% of the measured value. The Q_u simulation results, except for test #8, all fall within the respective experimental error tolerances. The Q_s and Q_{aux} experimental error tolerances are much smaller than those of Q_u. The experimental errors on Q_s and Q_{aux} are typically ±1% and ±0.5% of the measured values, respectively. The Q_s and Q_{aux} simulation results generally do not fall within the experimental error bounds. Figure 4.36 compares the experimental and simulated Q_s and Q_{aux} results for the eight basic tank tests. The values shown are the upper or lower CSU result, minus the TRNSYS result, divided by the TRNSYS result. The CSU Q_s values have been modified to take into account the energy storage within the solar tank by addition of the numerically integrated storage terms, ΔU, to the experimentally reported Q_s values. For example, values shown in Figure 4.36 for test #7 were calculated as follows:

Q_s experimentally reported	=	24260 ± 223 kJ
ΔU calculated from instantaneous test measurements	=	664 kJ
Q_s TRNSYS	=	24600 kJ
Q_{aux} experimentally reported	=	31449 ± 171 kJ
Q_{aux} TRNSYS	=	30440 kJ

$$Q_s \text{ value} = \frac{\{(24260 - 223) + 664\} - 24600}{24600} * 100\% = 0.41\%$$

$$Q_{aux} \text{ value} = \frac{(31449 - 171) - 30440}{30440} * 100\% = 2.75\%$$

A value of zero indicates the simulated result falls within the experimental error tolerance. Figure 4.36 indicates that, except for tests #10 and #11, the simulated Q_s results lie within 3% of the experimental values. Figure 4.36 also shows a 3% agreement between the experimental and simulated Q_{aux} results.

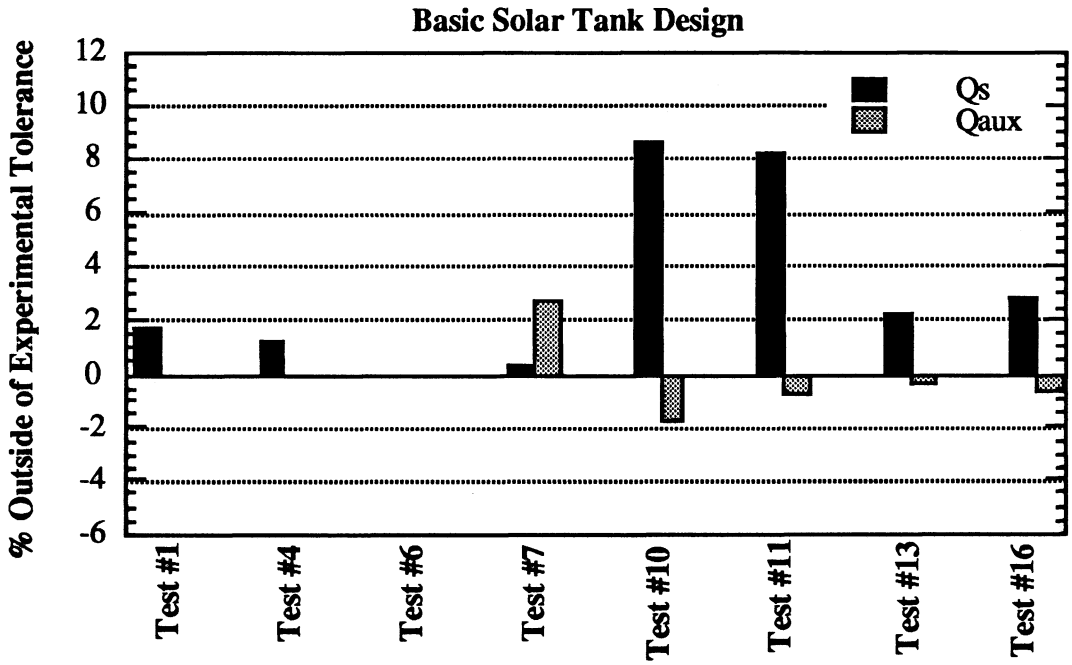


Figure 4.36 Comparison Between Simulated and Experimental Energy Flows

Figure 4.37 is the analogous plot to Figure 4.36 for the manifold tank test results, with one major exception. Sufficient data to calculate the experimental daily integrated solar tank energy storage terms for the eight manifold tank tests are not available. Hence, the experimental results used in calculating the values of Figure 4.37 are not corrected for energy storage. The Q_s comparisons of Figure 4.37 generally lie outside of the 3% mark observed for the basic tank tests of Figure 4.36. The larger discrepancy seen in the manifold test comparisons may be partially accounted for if energy storage was taken into account, as in the basic tank test comparisons. The manifold tank test Q_{aux} comparisons of

Figure 4.37, as with the basic tank tests of Figure 4.36, generally lie within 3% of the experimental results.

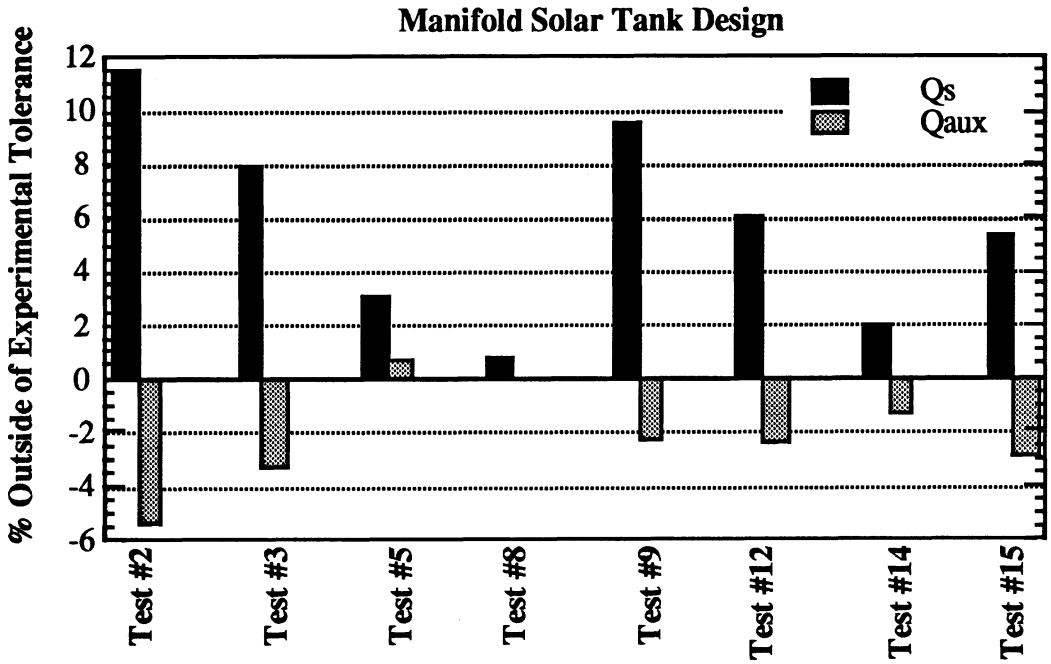


Figure 4.37 Comparison Between Simulated and Experimental Energy Flows

The sensitivity simulations indicate the collector parameters ($F_r(\tau\alpha)_n$, F_rU_L, and area/panel) are the most critical of the parameters investigated on system performance. A small error in the collector parameters may result in large simulation errors. Parameters having a secondary, yet still important, effect on system performance include the heat exchanger effectiveness, fluid low rates, and tank heat loss coefficients. A change in tank volumes only slightly affect simulation results.

4.7 CONCLUSIONS

The comparisons presented in this chapter should not be interpreted as an indication as to the accuracy in which the computer program TRNSYS can simulate a SDHW system. Simulating SDHW systems, as illustrated in Figure 1.5, involves two distinct steps. The first step includes the determination of those system parameters required for modeling. The system parameters are then used as input to the second, or actual simulating, step. Obviously, numerical errors (such as convergence errors, round off errors, etc...) are inherently present in any numerical simulation routine. Furthermore, all of the physical phenomena observed in real life cannot be numerically modeled. Precise methods for modeling some phenomena are, as to this date, undetermined, requiring the attention of present and future researchers and modelers. In other cases, the modeling of some phenomena, even when possible, must be neglected in order to have a model which will execute within a reasonable amount of time. The TRNSYS program and component subroutines used to model the CSU drain-back SDHW system have numerical and modeling limitations. However, as demonstrated in this chapter, the results calculated by TRNSYS are strongly dependent upon the system parameters. Hence, the TRNSYS results are inherently no more accurate than the system parameters used as input to the program.

The system data required to generate those results presented in this chapter were obtained by gathering information about each individual system component (the first option presented in Figure 1.5). The knowledge of 37 system parameters were required. In some cases, such as with the heat exchanger effectiveness, solar tank volumes, and solar and auxiliary tank heat loss coefficients, the students at CSU measured via single experiments. However, although the best indication possible, an experimentally determined value is not equivalent to an actual, and sometimes dynamic, value.

In other cases, the determination of the parameters required a simple phone call to CSU. For example, CSU "dialed" into the actual gain controller upper and lower dead band temperatures of 11.11 and 2.78 °C. Hence, the same values were "dialed" into the TRNSYS input list. Still in other cases, such as with the drain-back tank volume, drain-back tank heat loss coefficient, auxiliary tank volume, pipe heat loss coefficient, and fraction of pump energy which is transfered into the fluid, the value of the parameters are unknown. Intelligent estimates for unknown parameters were made based upon the manufactures rated values and/or experimentally determined information obtained for similar components.

The use of assumed values inherently adds uncertainty to the simulation results. For example, the solid circles in Figures 4.38 and 4.39 show the CSU experimental Q_s and Q_{aux} results for the entire set of 16 tests. The *D*'s along the abscissa indicate the diffuser tests. The non-diffuser Q_s terms were modified to account for energy storage within the solar tank over the last test day. An experimental error in Q_u input will propagate through the system, affecting energy flows "downstream" of the collector-boiler arrangement. The "CSU Max" error in Figures 4.38 and 4.39 is the Q_s or Q_{aux} measured value with the Q_s or Q_{aux} measurement error plus Q_u error added on. Likewise, the "CSU min" values are the experimental values minus the respective plus Q_u error sum. However, as stated earlier, the Q_s errors are approximately ±1% and the Q_{aux} errors ±0.5% of the measured values. Hence, any noticeable error tolerance in Figures 4.38 and 4.39 is due to the Q_u error addition.

The pipe heat loss coefficient is one of the parameters which is unknown. The squares in Figures 4.38 and 4.39 represent the simulated results using an assumed pipe heat loss coefficient. The figures indicate the simulations generally fall within the maximum-minimum error band, but tend to *underestimate* the solar energy contribution.

The hollow circles indicate simulations performed using the rated pipe heat loss coefficient. Again, the simulation results tend to fall within the error bands. However, the rated pipe heat loss simulations tend to *overestimate* the solar energy contribution. In either case, the simulations follow the same trends as experiments.

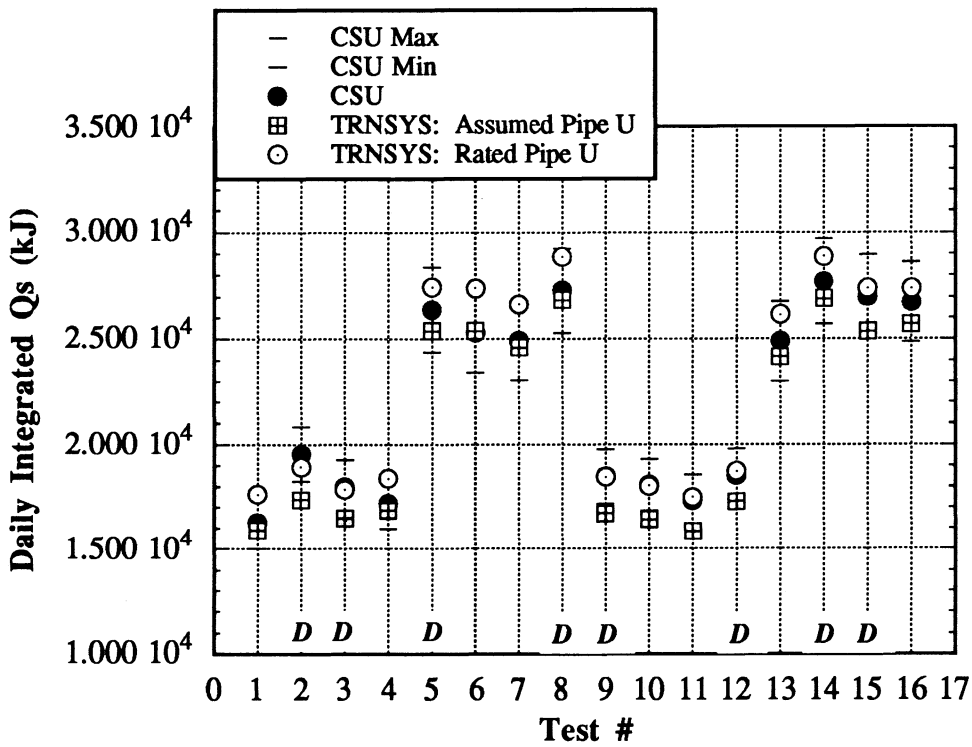


Figure 4.38 Change in Simulated Q_s with Pipe Heat Loss Coefficient

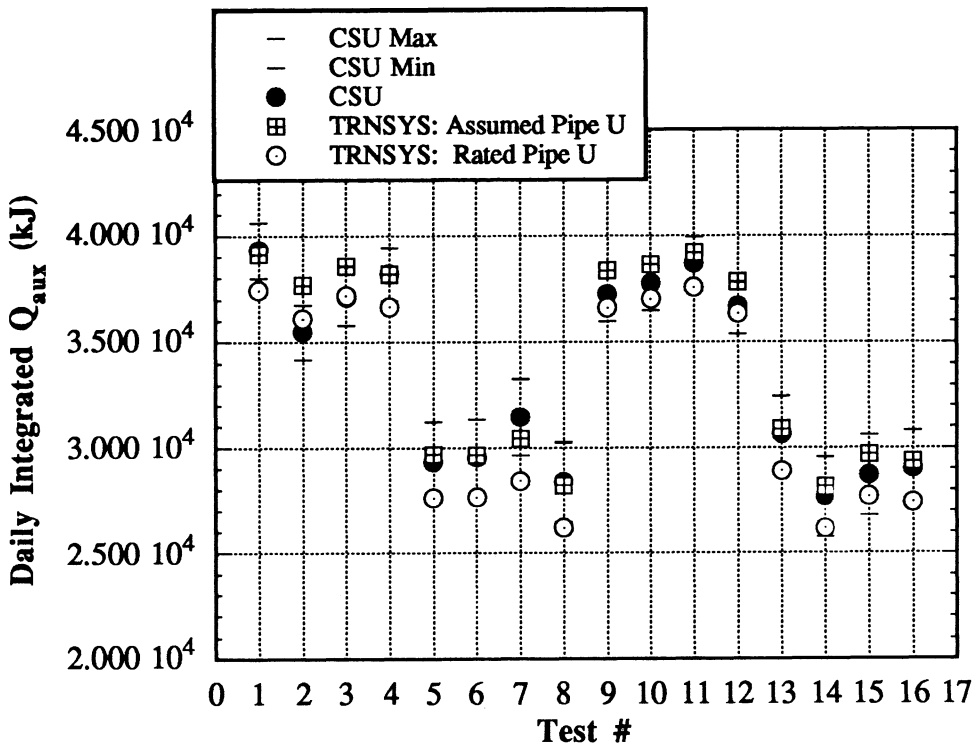


Figure 4.39 Change in Simulated Q_{aux} with Pipe Heat Loss Coefficient

Figures 4.36 and 4.37 demonstrate the simulated and experimental results generally agree to within a 3% relative error. Some of the difference is due to modeling limitations. However, some of the error is also caused by erroneous values being used as input to the simulation routine. The sensitivity results presented in this chapter indicate small variations to most, if not all, of the parameters will, either single handedly, or in combination with other parameters, result in at least a 3% change in system performance, leading one to believe most of the error is due to uncertainty in system parameters, and not caused by limitations in the chosen modeling routine.

Chapter 5

The Use of "Compressed" Weather in Detailed SDHW Simulations

5.1 INTRODUCTION

As stated previously, the recommendation of a long-term SDHW performance prediction method is the goal of this theses study. Figure 1.5 outlines two possible "paths" which may be followed to arrive at the long-term system performance. One path involves basing long-term performance upon short-term test results. The time expenses encountered along the "test and extrapolate" path have eliminated the testing methods from consideration. The other path involves using computer simulations to predict the long-term performance. Chapter 3 lists four common simulation methods as alternatives to the testing methods. The Solar Rating and Certificate Corporation (SRCC) has expressed a favoritism toward using detailed simulations (i.e., TRNSYS [Klein et al., 1990]) if a simulation method is indeed to be recommended. SRCC feels the flexibility allowed by detailed simulations makes this simulation technique best suited for the purposes they have in mind. The time expense associated with performing multiple TRNSYS simulations of a particular SDHW system to show the variation of performance with location is a major drawback to using detailed simulations. Hence, a way to "speed up" the simulations is needed.

SDHW detailed simulations basically consists of two parts. One part involves the actual system model. Using a "sped up" SDHW combined model rather than individual

component models will sped up simulations by a factor of 3.7. The 3.7 increase in speed is definitely welcomed, but is not by itself an answer in that the simulations are still too slow. The ambient temperature and radiation profiles are the second part the SDHW system simulations. Typical Metrological Year (TMY) data is is a common data base used in solar simulations. TMY data consists of typical months of actual hourly-spaced weather data for 26 U.S. locations. The variation of system performance with location is easily determined by repetitively simulating the system using a different TMY data set for each simulation. However, it is unlikely performance results for only 26 locations are sufficient to accurately show how the system performance will vary across the entire U.S.. The other possibility is to use hourly data generated from longer term, such as monthly average daily, weather data. Monthly average weather data for numerous U.S. locations is easily accessible, and requires comparatively little computer memory to store.

This chapter is divided into two parts. The first part explains one method for generating temperature and solar radiation profiles from the respective monthly average daily values. The second part of the chapter explains the use of "compressed" weather as a means of speeding up detailed SDHW simulations.

5.2 RADIATION AND TEMPERATURE GENERATION

Solar radiation and ambient temperature profiles may be viewed as consisting of the sum of a deterministic and a random component [Knight, 1988]. The deterministic component is defined as the average weather statistic for the time period in question. The random component is defined as the difference between the actual and average weather values. The method for creating short-term (i.e., hourly) weather from long-term (i.e., monthly average daily) weather used by the TRNSYS Type 54 weather generator [Knight, 1988] is outlined below. The weather generating method consists of representing the

weather statistic by an autoregressive model involving a normally distributed variable, χ. The cumulative frequency distribution of the weather statistic is equated to that of a normal distribution, resulting in a relationship between the weather statistic and the variable, χ.

5.2.1 Radiation (k_t) Generation

Radiation generation is fundamentally an exercise in hourly clearness index (k_t) generation. The hourly radiation on a horizontal surface, I, may be obtained from the k_t values and the extraterrestrial radiation, I_o (methods are outlined in Duffie and Beckman [1980] for calculating I_o based upon the time of day). The fraction of duffuse radiation, I_d, may be calculated from the k_t and I values by the Erbs [1980], or other such correlations. Generation of k_t's consists of two distinct steps. The first step involves determining the daily clearness indexes (K_t's). The second step involves reducing the daily K_t values into hourly k_t values.

The daily clearness indexes may be determined from the monthly average daily clearness index via the K_t cumulative frequency distribution curve. Figure 5.1 depicts the Bendt et al. [1981] K_t distribution. The daily clearness indexes are determined by calculating the fraction of days within the month (F) for which the K_t values for these days are less than or equal to a specified K_t value. For example, specifying a K_t value of 1.0 results is an F value of 1.0 since all days within a month have, by definition, K_t values less than or equal to unity. The process is reversed in the case of weather generation and the F values are specified instead of the K_t's since the K_t's are unknown. For example, suppose a month consists of three days, and has a monthly average daily clearness index $\overline{K_t}$ equal to 0.4. Logical values of F for the three day month are the values which divide the F axis in Figure 5.1 into three equal parts. Hence, the F values are calculated as follows:

$$F_1 = \frac{0+1}{2*3} = \frac{1}{6} \qquad (5.1)$$

$$F_2 = \frac{1+2}{2*3} = \frac{3}{6} \text{ , and} \tag{5.2}$$

$$F_3 = \frac{2+3}{2*3} = \frac{5}{6.} \tag{5.3}$$

Equation (5.4) is a general expression for the cumulative distribution frequency statistic where i is the i^{th} day of the month, and N is the number of days within the month:

$$F_i = \frac{(i-1)+i}{2*N} = \frac{2*i-1}{2*N} \tag{5.4}$$

The corresponding K_t values for this three day month are determined from Figure 5.1 to be 0.27, 0.40, and 0.61. The days are subsequently ordered in the sequence which most closely results in the 0.25 to 0.30 lag one autocorrelation between K_t values observed in actual weather data.

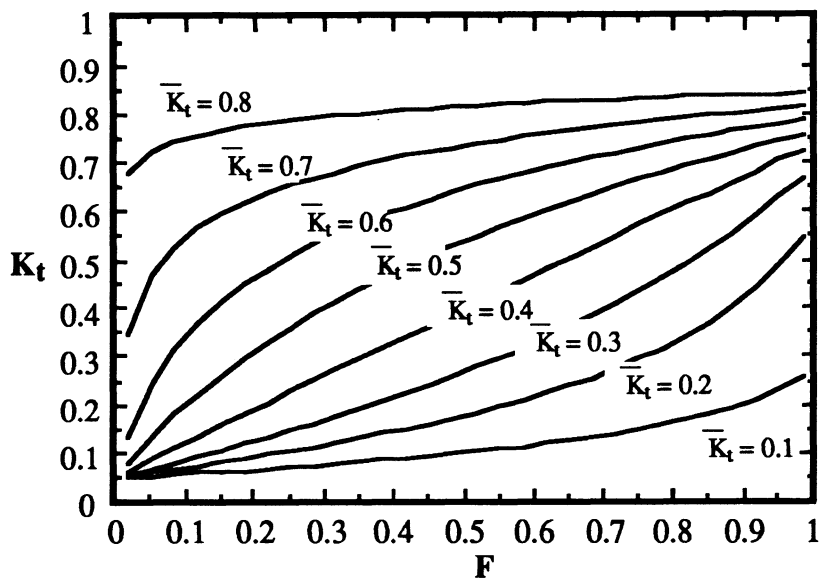


Figure 5.1 Bendt K_t Distribution

The second step in radiation generation involves reducing the daily K_t values into hourly k_t values. The k_t values consist of the superposition of a monthly average hourly value, k_{tm}, and a random value. The k_{tm} values may be calculated via:

$$k_{tm} = \frac{I}{I_o} = \frac{H}{H_o} * \frac{I}{H} * \frac{H_o}{I_o} = K_t * \frac{r_t}{r_d} \tag{5.5}$$

The variables r_t and r_d are functions of the time of day and time of sunset. Knight determined a k_t cumulative frequency distribution, Figure 5.2. The variable σ_a in Figure 5.2 is the standard deviation of the random k_t component. Graham [1985] defined σ_a as:

$$\sigma_a = 0.1557 \sin\left(\frac{\pi K_t}{0.9330}\right) \tag{5.6}$$

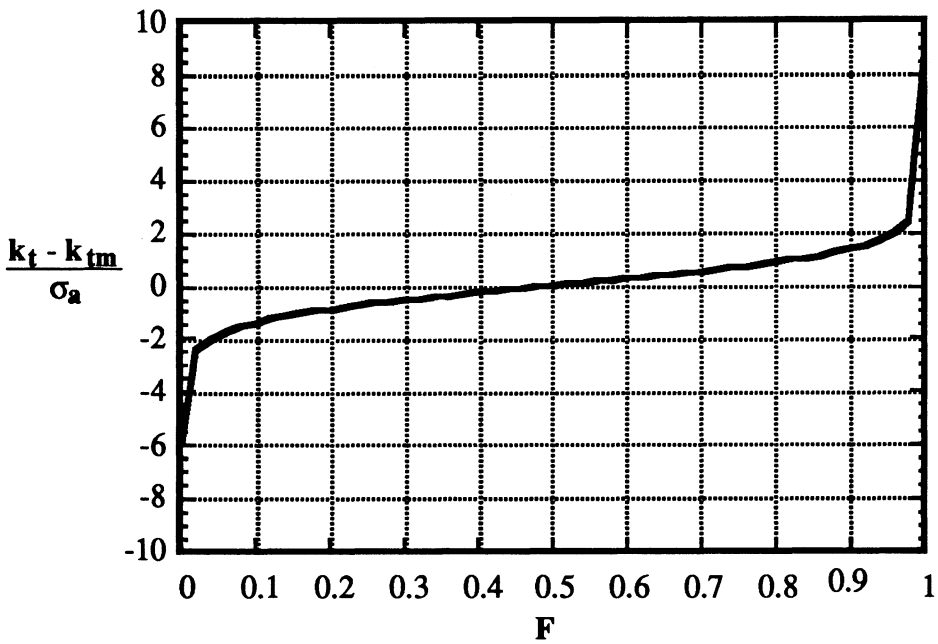


Figure 5.2 Knight k_t Distribution

Graham transformed k_t values for various times and locations into the normally distributed variable, χ, having a mean 0 and variance 1. The k_t values were transformed in order to fit an autoregressive model to the data. Graham found the following model best represents the transformed data:

$$\chi_i = \phi^*\chi_{i-1} + \varepsilon_i \qquad (5.7)$$

where

$$\phi = 0.3455 + 1.0745*K_t - 1.1327*K_t^2$$

ε_i is a randomly selected value from a normal distribution with mean 0 and a variance equal to $1 - \phi^2$.

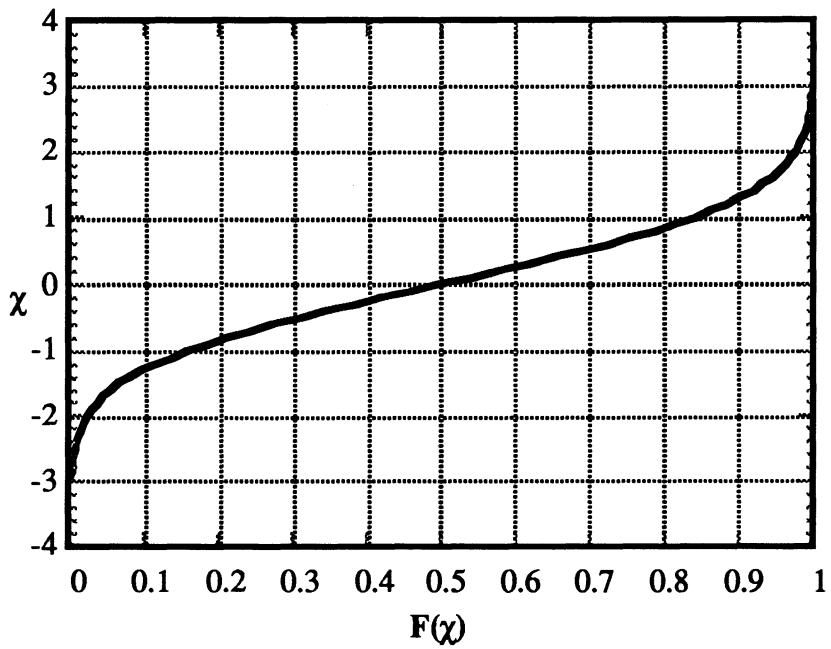


Figure 5.3 Normal Distribution

Equating the cumulative distribution for χ, Figure 5.3, to that of k_t, Figure 5.2, results in the following expression for k_t.

$$k_t = k_{tm} - \frac{\sigma_a}{1.585} * \ln\left[\frac{1}{0.5\left(1 + \text{erf}\left(\frac{\chi}{\sqrt{2}}\right)\right)} - 1\right] \tag{5.8}$$

5.2.2 Temperature Generation

Ambient temperature profiles, as with solar radiation profiles, may be thought of as being made up of a random component superimposed upon a deterministic component. One method of generating hourly temperature data, analogous to radiation generation, consists of fitting an autoregressive model to normally transformed data. The untransformed temperature cumulative frequency distribution is equated to that of a normal distribution, resulting in an expression for the hourly temperature values.

Erbs [1984] developed an expression for the deterministic temperature profile (i.e., the monthly average hourly temperature, $\overline{T}_h$) as a function of the monthly average daily temperature, $\overline{T}_m$, and an amplitude, A. The Erbs monthly average hourly temperature profile is shown in Figure 5.4. The amplitude, A, is a function of $\overline{K}_t$, and defined as follows:

$$A = 25.8\,\overline{K}_t - 5.21 \tag{5.9}$$

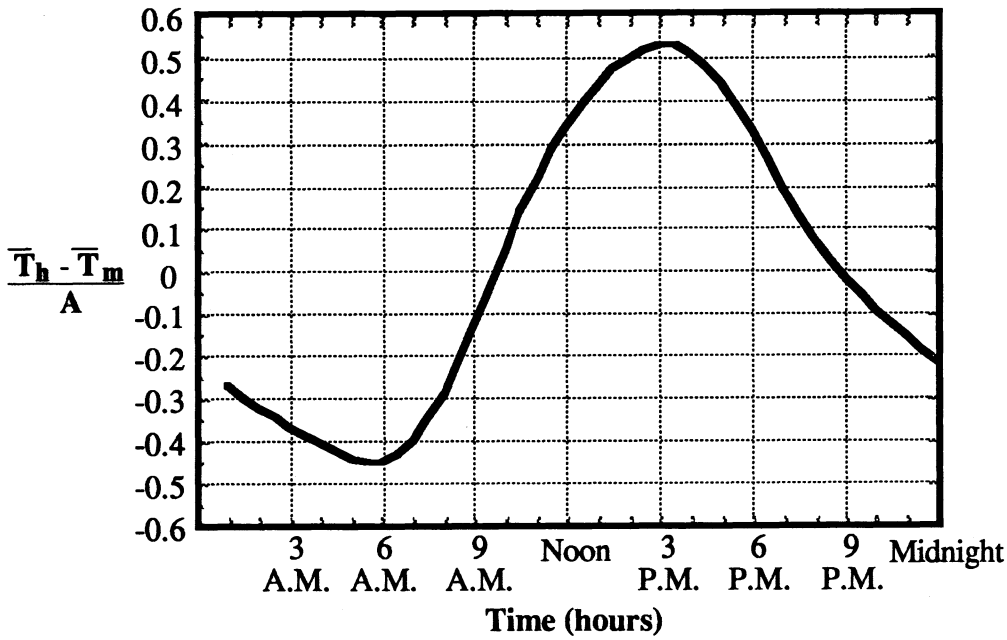


Figure 5.4 Variation of Monthly Average Hourly Ambient Temperature with Time of Day

Erbs also developed an expression for the ambient temperature cumulative frequency distribution, shown in Figure 5.5. The variable σ_m in Figure 5.5 is the standard deviation of $\overline{T}_m$ about the long-term monthly average daily temperature. Erbs estimated σ_m as follows:

$$\sigma_m = 1.45 - 0.0290\,\overline{T}_m + 0.0664\,\sigma_{yr} \qquad (5.10)$$

where

σ_{yr} = Standard deviation of the 12 $\overline{T}_m$'s about the yearly average daily temperature.

The variable N in Figure 5.5 is equal to the number of hours in the month.

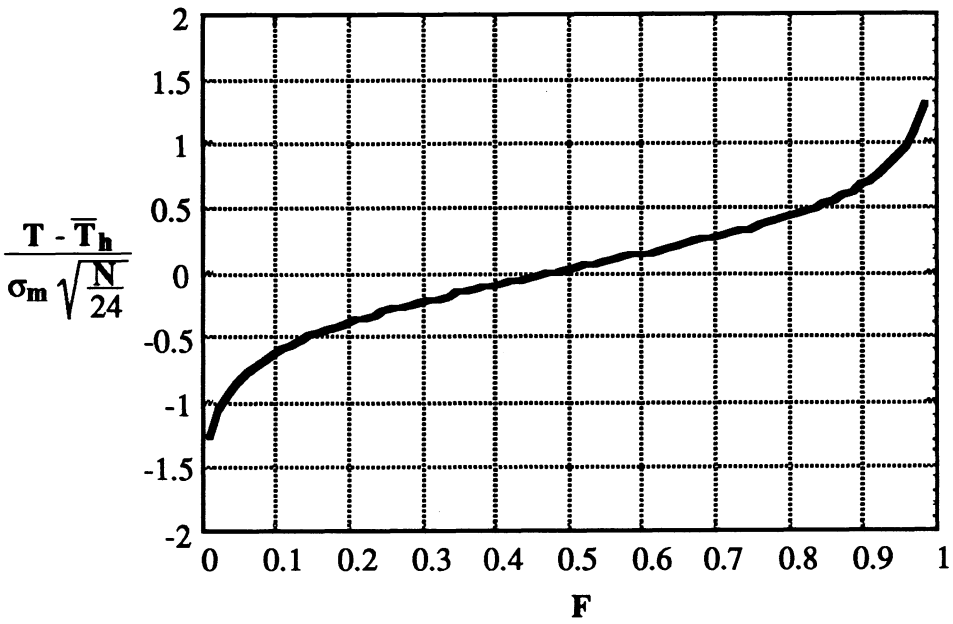


Figure 5.5 Erbs Ambient Temperature Distribution

Knight transformed temperature data for various times and locations into the normally distributed variable χ having a mean 0 and variance 1. Knight found the following autoregressive model best fit the transformed temperature data:

$$\chi_i = \phi_1 \chi_{i-1} + \phi_2 \chi_i + \varepsilon_i \tag{5.11}$$

where

$\phi_1 = 1.178$

$\phi_2 = -0.202$

ε_i is a randomly selected variable from a normal distribution with mean zero and a variance, σ^2, equal to:

$$\sigma^2 = 1 - \phi_1 \left[\frac{\phi_1}{1 - \phi_2} \right] - \phi_2 \left[\phi_2 + \frac{\phi_1^2}{1 - \phi_2} \right] \tag{5.12}$$

Equating the normal cumulative distribution frequency curve, Figure 5.3, to that of ambient temperature, Figure 5.5, results in the following expression for the ambient temperature, T:

$$T = \overline{T}_h - \frac{\sigma_m}{1.698} \ln\left[\frac{1}{0.5\left[1 + \text{erf}\left(\frac{\chi}{\sqrt{2}}\right)\right]} - 1\right] \qquad (5.13)$$

Figures 5.6 through 5.9 compare Nashville, TN, TMY to TRNSYS Type 54 generated weather. The input to Type 54 was monthly average daily TMY radiation and temperature values. Figures 5.6 and 5.7 show the daily horizontal surface radiation for the months of January and July, respectively, plotted in ascending order. Figures 5.8 and 5.9 are the analogous plots to Figures 5.6 - 5.7 for the ambient temperature. The figures show comparable distributions between the TMY and generated weather data. Monthly solar fractions obtained from simulations of a typical SDHW system using the generated and TMY data are shown in Figures 5.10 and 5.11. The two different pairs of curves on each plot are for two different collector areas. The generated and TMY weather simulations show good agreement, with the average relative error in yearly solar fraction for the four pairs of simulations being 3.1%.

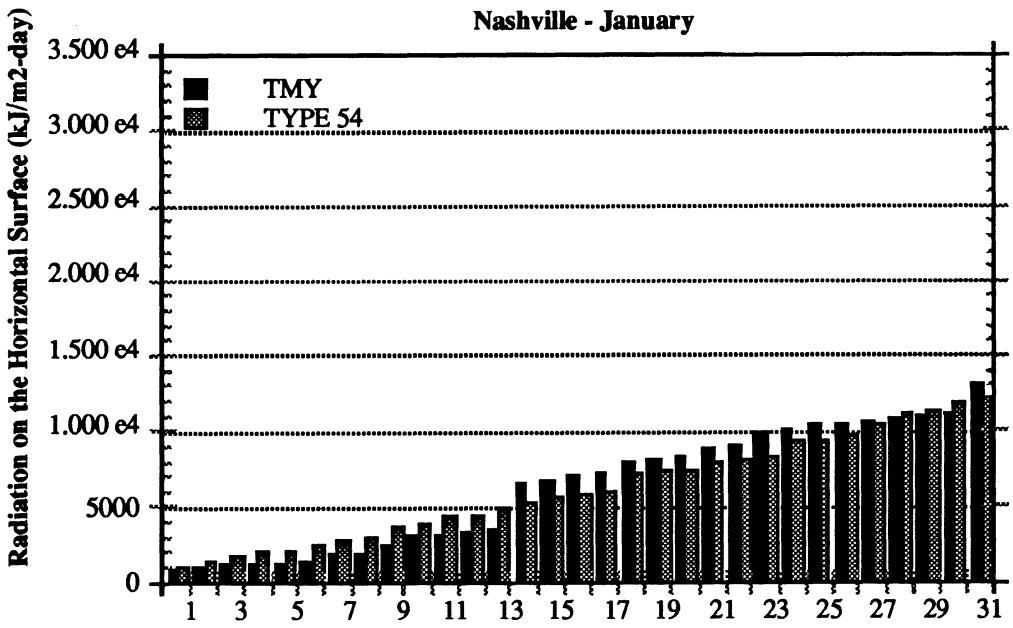


Figure 5.6 Comparison Between TMY and TRNSYS Generated Horizontal Surface Radiation

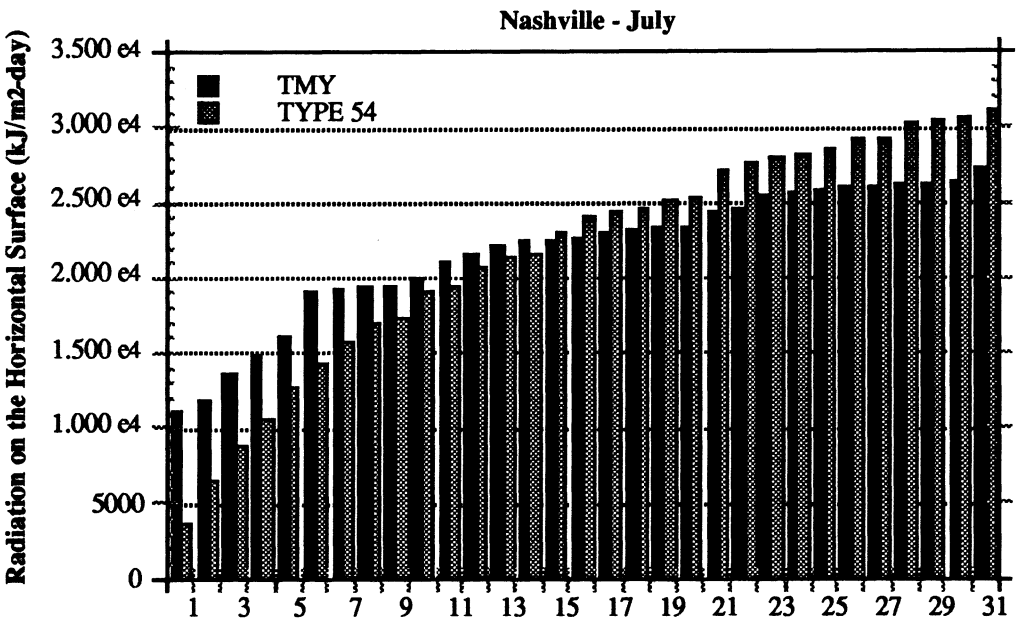


Figure 5.7 Comparison Between TMY and TRNSYS Generated Horizontal Surface Radiation

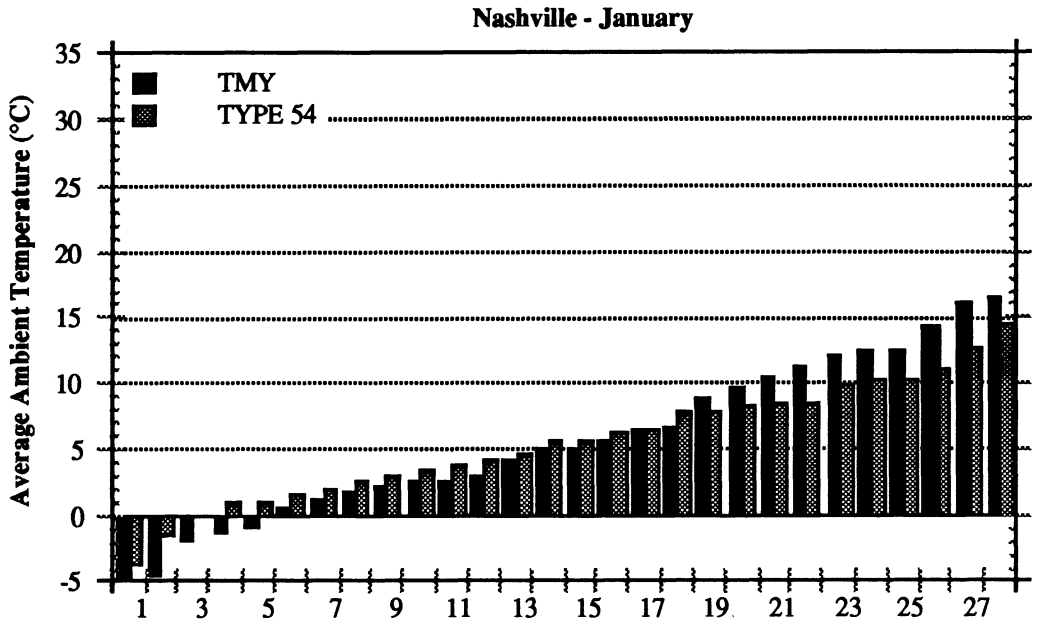


Figure 5.8 Comparison Between TMY and TRNSYS Generated Ambient Temperature

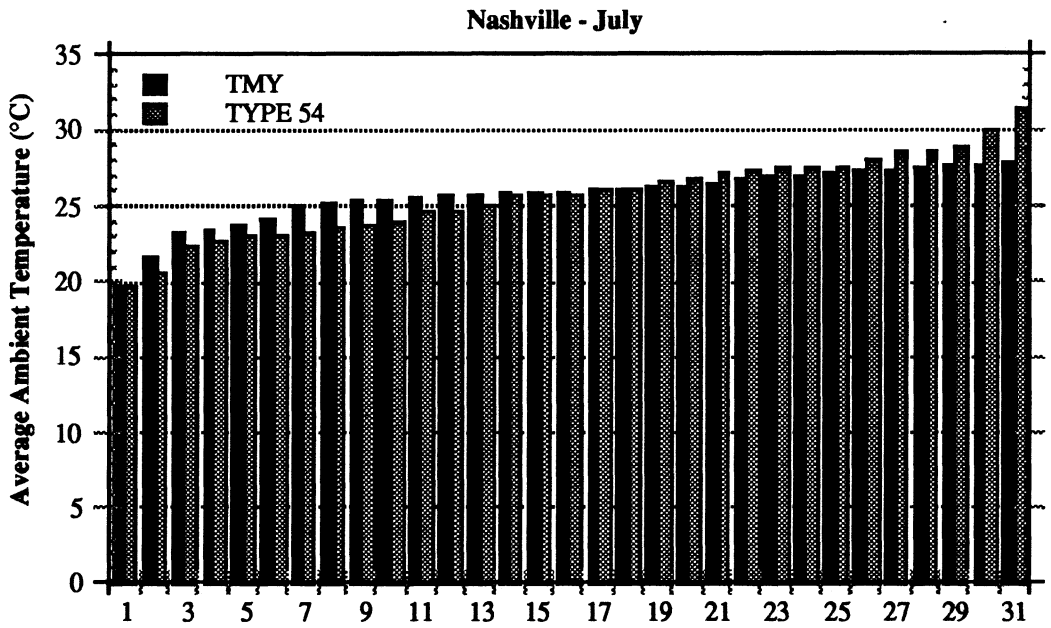


Figure 5.9 Comparison Between TMY and TRNSYS Generated Ambient Temperature

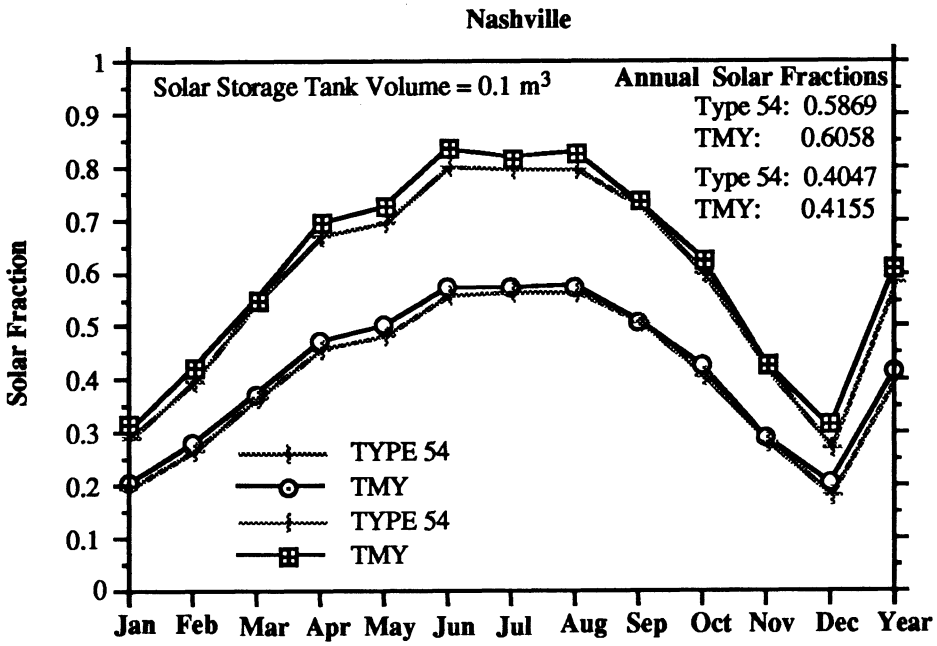


Figure 5.10 Comparison Between SDHW System Simulations Using TMY and TRNSYS Generated Weather Data

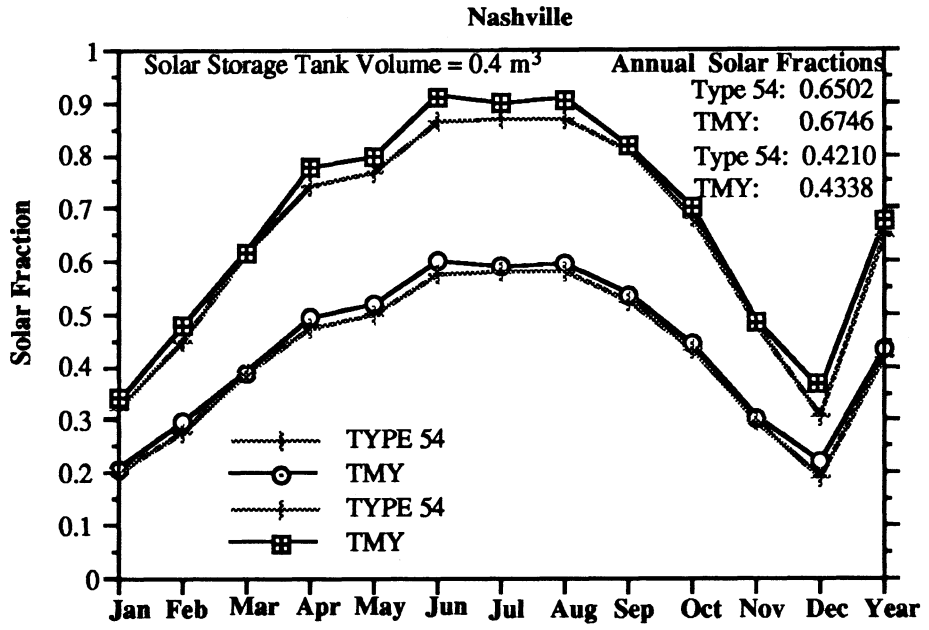


Figure 5.11 Comparison Between SDHW System Simulations Using TMY and TRNSYS Generated Weather Data

5.3 "COMPRESSED" WEATHER

"Compressed" weather means representing an entire month of days with N statistically chosen days where N is less than the actual number of days in the month. The reason compressed weather is investigated in this thesis is to "speed up" (i.e., reduce the computational effort) required to perform detailed simulations. Compressed weather is directly related to weather generation in that the N chosen days are usually constructed, and are not actual, weather profiles. Several aspects concerning compressed weather are unclear, however. The most predominant questions are:

1. Is the random component in radiation and ambient temperature profiles important in the simulation of SDHW systems?
2. How many days are required to accurately represent a month?
3. In which order should the days be arranged?

These three questions are addressed below.

5.3.1 The Importance of the Random Component

Neglecting the random component observed in radiation and temperature profiles is desirable in the calculation of compressed weather for a couple of reasons. First of all, neglecting the random component reduces the number of calculations required since only the deterministic component needs to be calculated. The net effect of reduced computations is reduced simulation times which, if utilized in the generation of a "compressed" month, further enhances the purpose behind using compressed weather. Secondly, it is unknown if weather generation methods which retain the random component will still be statistically meaningful if the data base (i.e., number of days in the month) is reduced.

5.3.1.1 Temperature

Hollands et al. [1989] investigated the effect of the random temperature component on the performance of active solar hot water systems. Hollands performed detailed simulations using the computer program WATSUN [Chandrashekar et al., 1990] over a range of system parameters, locations, and hot water load profiles. One set of simulations used TMY temperature data, whereas the other set used the deterministic profile pictured in Figure 5.4. Systems were found to perform better when subjected to the deterministic temperature profile rather than the TMY profile in all cases. However, the difference was slight, with maximum relative errors in yearly solar fractions between the two sets of simulation being on the order of +1%.

Hollands also performed a third set of simulations using a constant ambient temperature equal to the monthly average daily temperature. The maximum relative error in solar fraction between simulations using the constant verses TMY ambient temperature was +5%. Hollands concluded neglecting the random temperature temperature component is acceptable for standard solar hot water system designs. However, neglecting both the random and deterministic components is not advisable.

5.3.1.2 Radiation

A process similar to Hollands' was used to investigate the importance of the random radiation component on SDHW system performance. Simulations were performed using actual and "smoothed" TMY radiation data. Smoothed TMY radiation data is TMY data in which the random component has been removed. In other words, the smoothed radiation data used in the study is deterministic radiation profiles generated from TMY based K_t values. One important difference between the use of actual and symmetrical (or smoothed) horizontal radiation data should be pointed out. The fraction of hourly diffuse

radiation (i.e., I_d/I) may be calculated directly if the hourly clearness index, k_t, is known by such relationships as the Erbs hourly correlation (solid line in Figure 5.12) [Erbs, 1980]. Knowledge of the diffuse radiation allows for the calculation of tilted surface radiation via one of the methods presented in Duffie and Beckman [1980].

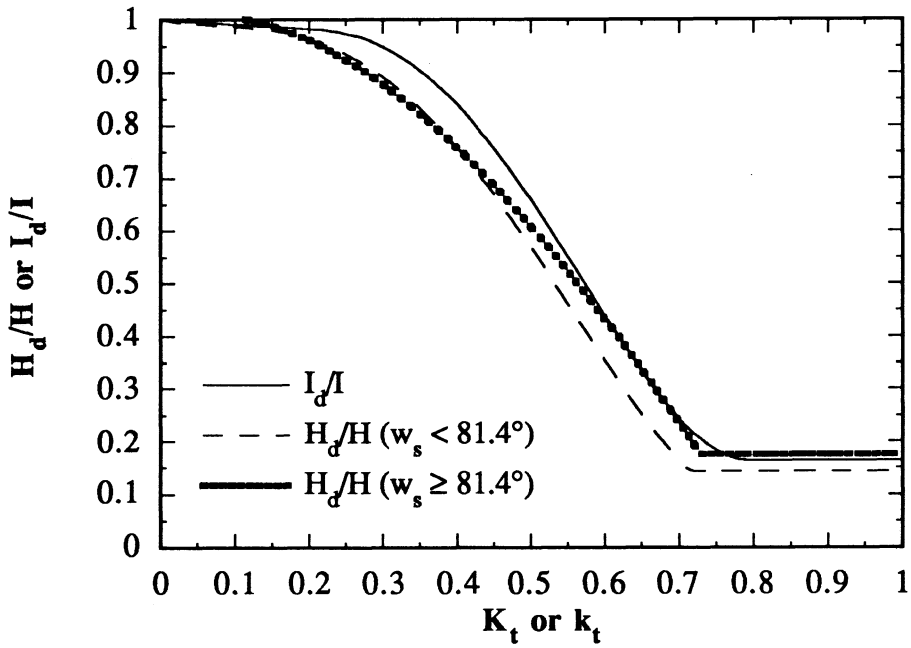


Figure 5.12 Erbs Diffuse Correlations

Erbs concluded substitution of k_{tm} for k_t into an hourly diffuse correlation will yield erroneous results due to the difference between the assumed (i.e., monthly average hourly) and actual k_t distributions. A better estimate for the diffuse radiation may be obtained by calculating the fraction of daily diffuse radiation (i.e., H_d/H) based upon the daily clearness index, K_t, and such relationships as the Erbs daily correlations (dashed lines in Figure 5.12). The hourly radiation values, I and I_d, may then be determined from

the daily integrated values, H and H_d, and the statistics r_t and r_d. r_t is the ratio of total hourly to total daily radiation (i.e., I/H), and r_d is equal to the ratio of hourly diffuse to daily diffuse radiation (i.e., I_d/H_d). Both r_t and r_d are functions of the time of day and time of sunset, and are calculated as explained in Duffie and Beckman [1980].

The daily radiation on the horizontal surface, H, will be the same for the actual and smoothed radiation profiles due to the days having the same K_t value. The radiation on the tilted surface will not be the same, however, because of differences in the fraction of diffuse radiation. Two distinct cases exist for comparison purposes. The first case involves adjusting all the K_t values by a multiplication factor such that the total monthly radiation on a tilted surface for the actual and smoothed months are equivalent. The K_t adjustment will destroy the agreement between total horizontal radiation values. Furthermore, adjusting the K_t's by a fixed constant such that the targeted monthly integrated tilted surface radiation is obtained will not necessarily result in the same daily integrated tilted surface radiation between the symmetrical actual days. The second possibility is to leave the K_t values unmodified with the realization that differences exist between the TMY and smoothed tilted surface radiation on a daily, as well as monthly, basis. Figures 5.13 through 5.16 illustrate the TMY, symmetrical with a modified K_t, and symmetrical with an unmodified K_t value total horizontal, diffuse horizontal, beam horizontal, and tilted surface radiation profiles for July 3 in Madison, WI. The figures demonstrate that the actual and symmetrical profiles may significantly differ.

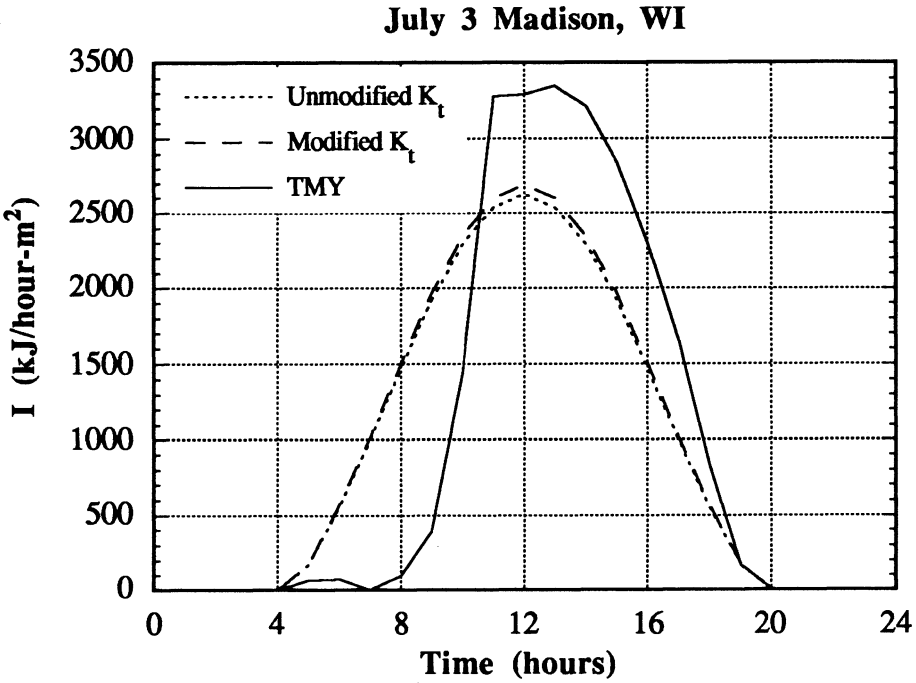


Figure 5.13 Total Horizontal Surface Radiation

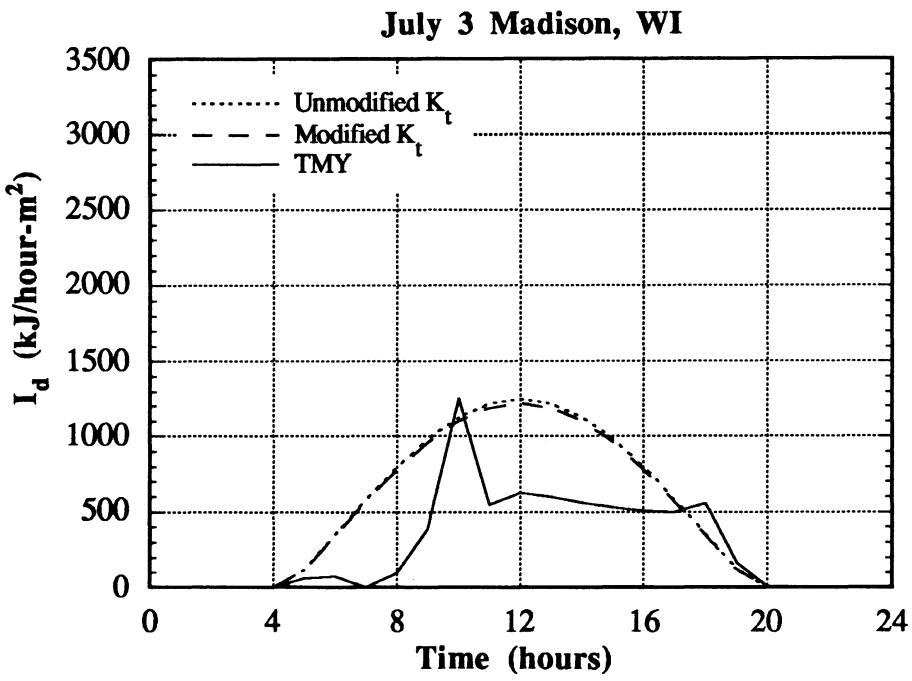


Figure 5.14 Diffuse Horizontal Surface Radiation

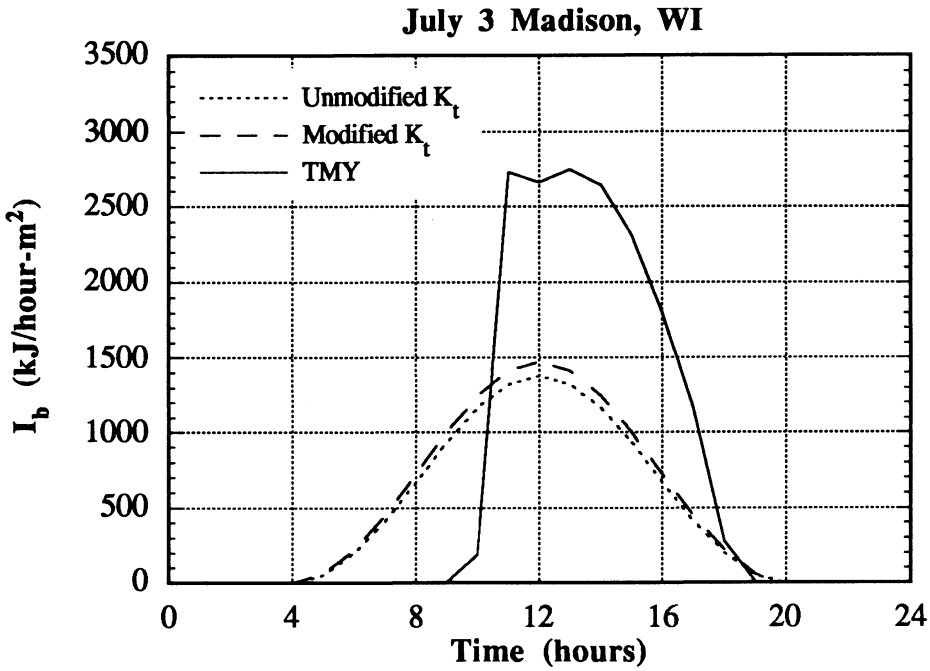


Figure 5.15 Beam Horizontal Surface Radiation

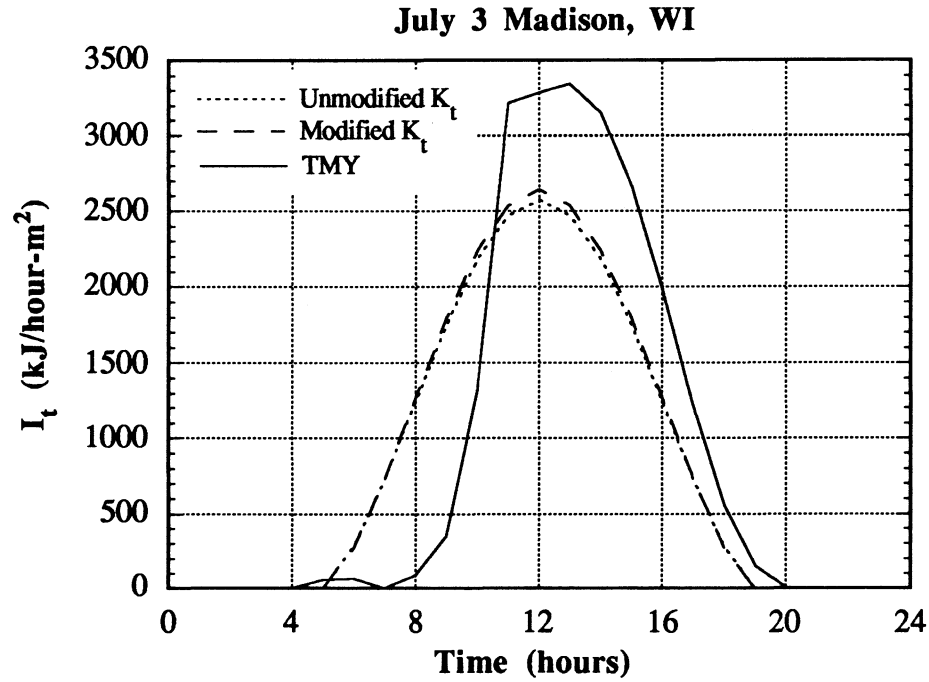


Figure 5.16 Tilted Surface Radiation

Whillier [1953] concluded that, in general, a solar system will perform better when subjected to actual, rather than symmetrical, radiation profiles if the K_t's for the actual and symmetrical days are identical. The argument presented by Whillier is illustrated in Figure 5.17. Two fictitious horizontal surface radiation profiles, one symmetrical and the other not, having the same total radiation (i.e., the same K_t) are shown in Figure 5.17A. Figure 5.17B illustrates how the two radiation profiles may look on the tilted surface. The total daily tilted radiation for the symmetrical and non-symmetrical profiles are not necessarily the same. For the critical radiation level, I_{TC}, the utilizable energy (i.e., cross-hatched area) is greater for the non-symmetrical day than for the symmetrical day. The greater utilizable energy directly results in better system performance. Enhanced performance due to random fluctuations in the radiation profile may be generalized to any system where the critical radiation level, I_{TC}, is relatively significant, as is the case in Figure 5.17. The foregoing argument does not apply, however, when the integrated horizontal surface radiation, and hence K_t's, are not equivalent.

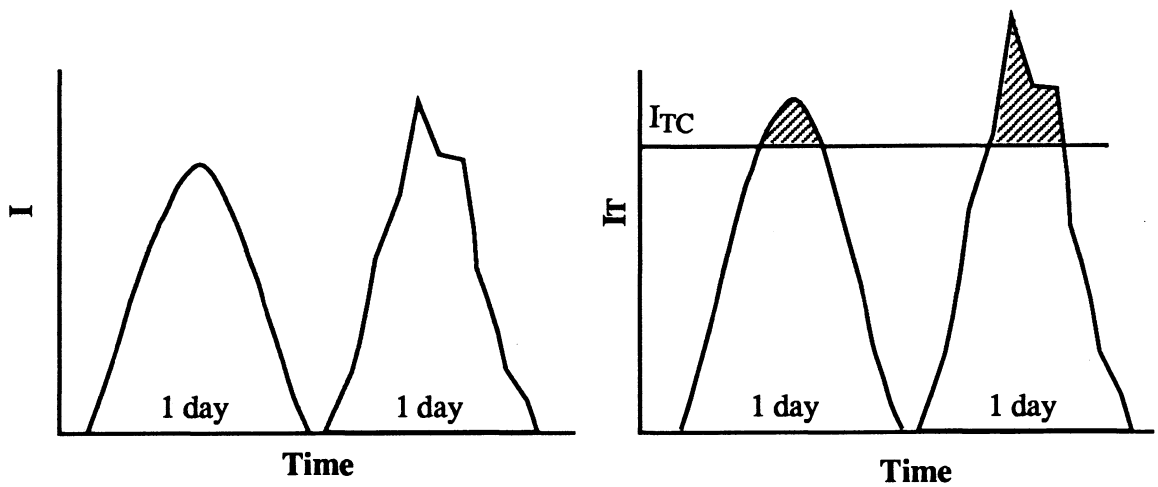


Figure 5.17 A and B Symmetrical vs. Non-Symmetrical Radiation Profiles

Simulations were conducted using actual TMY weather data, symmetrical days with modified K_t's, and symmetrical days with unmodified K_t's. The simulations were performed for a SDHW system similar to the CSU system (test #7 conditions) over a range of parameter variables and locations. The TRNSYS compatible combined SDHW model mentioned in Chapter 3 was used to simulate the system. The simulated system differs from the actual CSU system in that the mains temperature is 10 °C (not 22 °C), the system is subjected to a 200 L/day RAND [Mutch, 1974] load profile (not the ASHRAE-95/SRCC specified load profile), the controller dead bands are slightly different and not a function of the collector stagnation temperature, the incidence angle modifier is 0.0, and the slope is equal to the latitude of the location being simulated. In addition, the combined system model does not allow for the presence of a drain-back tank, or pipe capacitance. Some of the system parameters are listed in Table 5.1

System Parameter	Value
Water load	200 L/day RAND profile
$F_r(\tau\alpha)_n$	0.6
F_rU_L	5.56 W/m^2-°C
Collector loop flow rate	0.0102 kg/sec-m^2
Tank loop flow rate	0.0170 kg/sec-m^2
Solar tank volume/collector area	0.0446 m^3/m^2
Dead band	11.5 °C

Table 5.1 SDHW System Parameters

Simulations were performed for an Albuquerque, NM, Madison, WI, and Seattle, WA, January and July. Albuquerque, Madison, and Seattle were chosen because these locations have comparably differing climatic conditions. Table 5.2 lists the various $\overline{H}$'s and $\overline{K}_t$'s for the months simulated.

Month	TMY $\overline{H}$ (kJ/m^2)	TMY $\overline{K}_t$
Albuquerque, January	11,120	0.61
Albuquerque, July	28,220	0.69
Madison, January	5,913	0.44
Madison, July	21,748	0.54
Seattle, January	3,163	0.30
Seattle, July	22,678	0.56

Table 5.2 Monthly Average Daily Weather Data

The cloudiest of the six months is Seattle in January having a TMY based $\overline{H}$ and $\overline{K}_t$ equal to 3,163 kJ/m^2 and 0.30, respectively. The sunniest month is Albuquerque in July with an $\overline{H}$ and $\overline{K}_t$ of 28,220 kJ/m^2 and 0.69, respectively. Seventeen simulations were performed for each location and month. The first simulation consisted of the base case discussed above. Simulations two through seventeen differ from the base case as listed in Table 5.3. The ambient temperature is constant at 20 °C and the initial system temperature is 0 °C for all simulations.

Figures 5.18 and 5.19 show the relative error in delivered solar energy, Q_s, between symmetrical and TMY days for the case where the K_t's are *not* adjusted. The delivered solar energy is the energy gain across the solar tank and calculated via:

$$Q_s = \int^{month} \dot{m}_L \, c_p \, (T_d - T_{mains}) \, d\theta \tag{5.14}$$

where

$\dot{m}_L$ and T_d are the solar tank to auxiliary tank flow rate and temperature

The relative error is defined as:

$$\text{Relative Error} = \frac{Q_{s,symmetrical} - Q_{s,TMY}}{Q_{s,TMY}} * 100\% \tag{5.15}$$

Test #	Changed Parameter
1	None
2	400 L/day RAND load
3	100 L/day RAND load
4	$F_r(\tau\alpha)_n = 0.8$
5	Constant load throughout day
6	Constant load between 9 A.M. and 3 P.M.
7	Constant load between 3 P.M. and 9 P.M.
8	Tank flow = 0.034 kg/sec-m^2 (i.e., doubled tank flow)
9	Tank flow = 0.0085 kg/sec-m^2 (i.e., halved tank flow)
10	Collector flow = 0.0204 kg/sec-m^2 (i.e., doubled collector flow)
11	Collector flow = 0.0051 kg/sec-m^2 (i.e., halved collector flow)
12	Volume/area = 0.0892 m^3/m^2 (i.e., doubled volume and load)
13	Volume/area = 0.0223 m^3/m^2 (i.e., halved volume and load)
14	Dead band = 23 °C
15	Dead band = 0 °C
16	$F_r(\tau\alpha)_n$ and F_rU_L = 0.8 and 3.056 W/m^2-°C, respectively
17	$F_r(\tau\alpha)_n$ and F_rU_L = 0.5 and 8.056 W/m^2-°C, respectively

Table 5.3 Test Cases Investigated

Figures 5.18 and 5.19 illustrate that, with the exception of Seattle in July, the solar system performs better when subjected to the non-symmetrical as compared to the symmetrical days, as expected. The difference between symmetrical and TMY tilted surface radiation for the months other than Seattle in July are less than ±2.75%. However, the smoothed Seattle July tilted surface radiation is 4.2% greater than the corresponding TMY value. A large difference in diffuse radiation between the symmetrical and TMY months is also observed for Seattle in July. The TMY month, using Erbs hourly diffuse correlations, has a monthly average diffuse component of 8216 kJ/m^2, whereas the

symmetrical month has, using Erbs daily correlations, an average diffuse radiation value of 8867 kJ/m^2. The difference in diffuse radiation is one reason for the discrepancy observed in tilted surface radiation. Also, there is a difference of 0.88% between the smoothed and TMY total horizontal surface radiation values for this month. The slightly greater symmetrical month horizontal surface radiation is another contributor to the tilted surface radiation discrepancy. The tilted surface radiation discrepancy is, in turn, contributing to the approximately +2% enhanced performance observed for the symmetrical month.

The Figures also indicate that, with a few exceptions, the relative errors for any given month over the 17 trials fall within a band of ±1%. Hence, the relative error caused by using symmetrical days in SDHW system simulations appears to be fairly independent of the 16 system parameters varied. The largest deviation from this independence appears to be the test #16 simulations. The test #16 simulations are all characterized by relatively small (less than 1%) errors. The oddity of test #16 is especially noticeable for the July Albuquerque simulation where the errors for the other tests corresponding to this month are on the order of -1.5%. Test #16 is characterized by a high $F_r(\tau\alpha)_n$ and low F_rU_L which, when combined with high radiation levels such as present in an Albuquerque summer, results in low critical radiation ratios. The effect of the random radiation component will be less of an influence in the case of low critical radiation ratios.

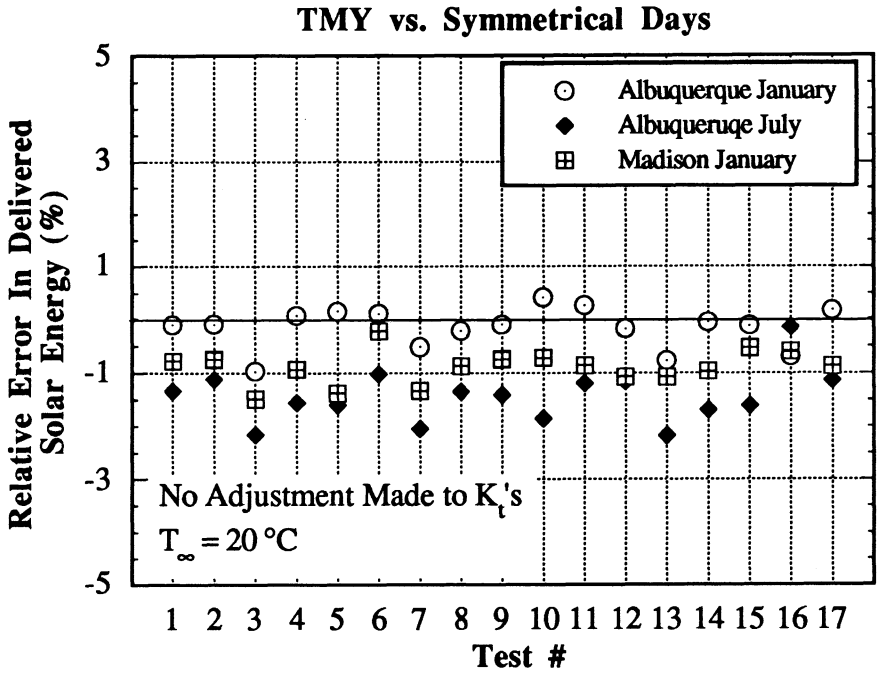


Figure 5.18 Performance Difference: TMY and Non-Adjusted K_t Symmetrical Days

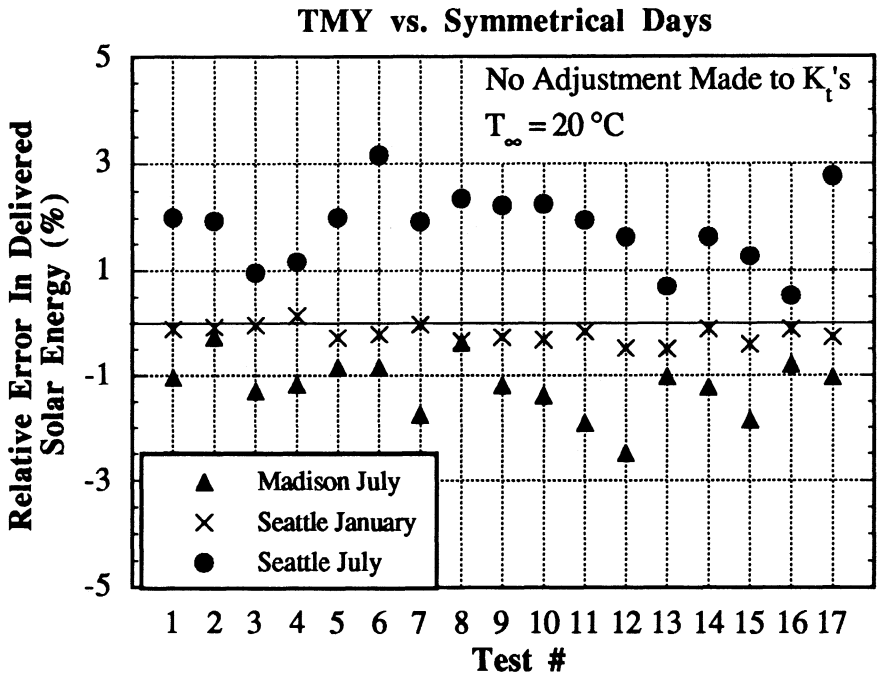


Figure 5.19 Performance Difference: TMY and Non-Adjusted K_t Symmetrical Days

Figures 5.20 and 5.21 are the analogous plots to Figures 5.18 and 5.19 for the case where the K_t's are adjusted. As in Figures 5.18 and 5.19, the 17 test cases for each month tend to fall around a band of ±1%. The agreement between test points indicates the error caused by using symmetrical days in SDHW system simulations is fairly independent of the 16 parameters investigated. Again, the largest deviation from this conformity is the test #16 Albuquerque July. Unlike the previous simulations, the Figure 5.20 and 5.21 errors tend to be either positive or negative, rather than strictly negative. A correlation between the change in K_t values and the error is present. Months in which the K_t's were increased have errors greater than zero (i.e., are delivering more energy than TMY months). In contrast, months in which the K_t's were decreased have errors less than zero (i.e., deliver less energy).

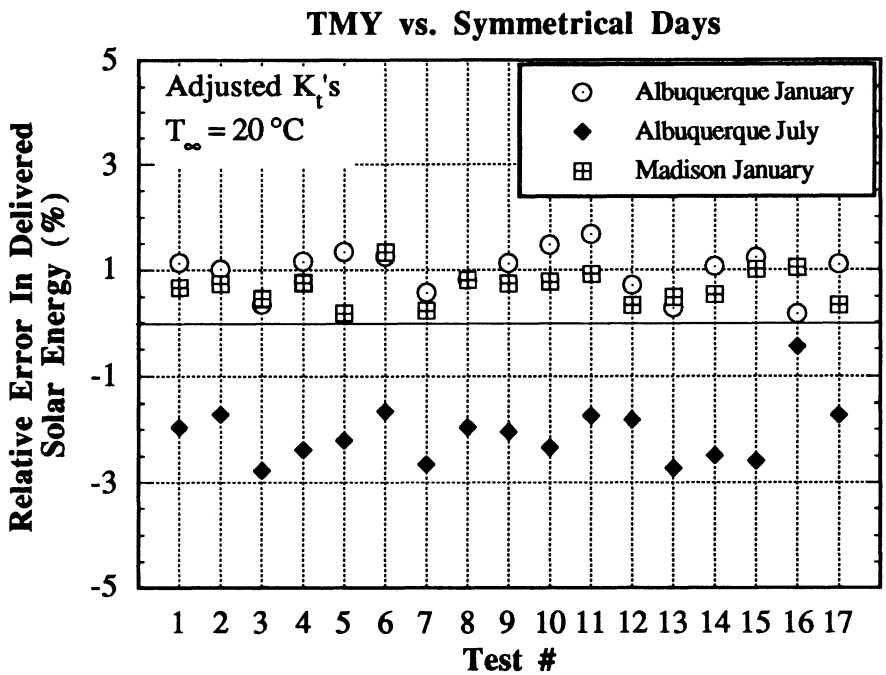


Figure 5.20 Performance Difference: TMY and Adjusted K_t Symmetrical Days

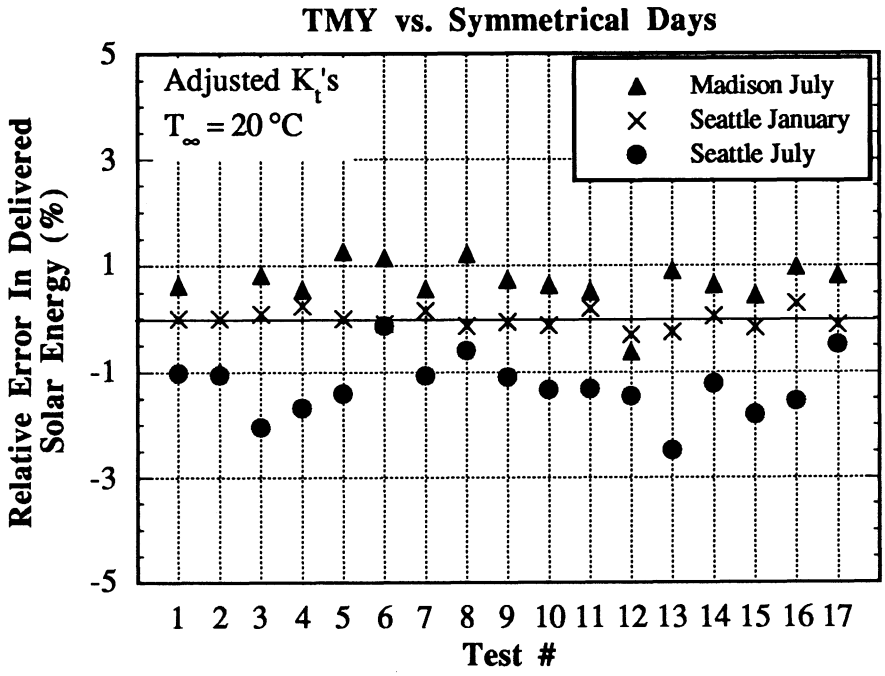


Figure 5.21 Performance Difference: TMY and Adjusted K_t Symmetrical Days

Figures 5.22 and 5.23 show the TMY solar fractions for the test cases investigated. The solar fractions range from a low of 20% (Seattle, January) to a high of 98% (Albuquerque, July). Figures 5.18 through 5.21 demonstrate the maximum relative error in delivered solar energy encountered by neglecting the random radiation component is on the order of ±3%.

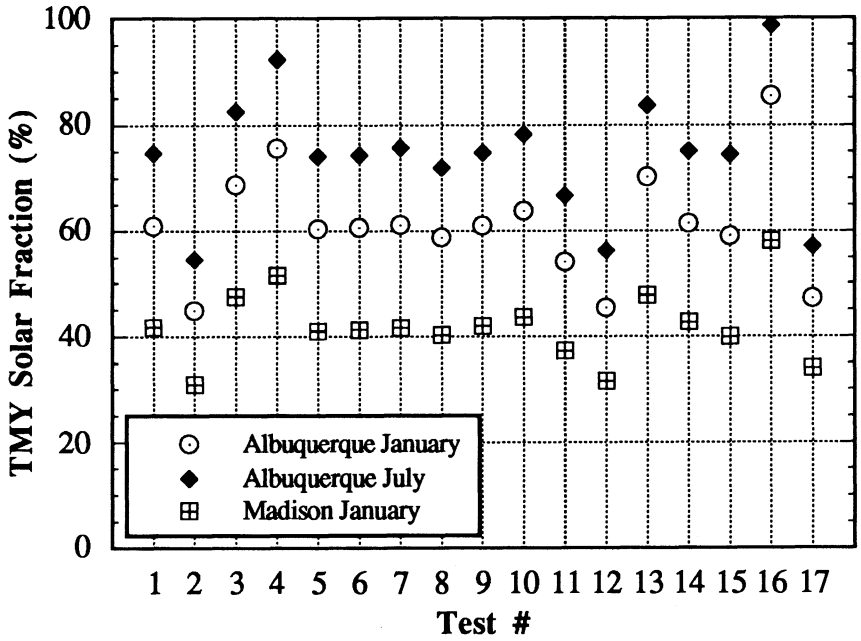


Figure 5.22 TMY Monthly Solar Fractions

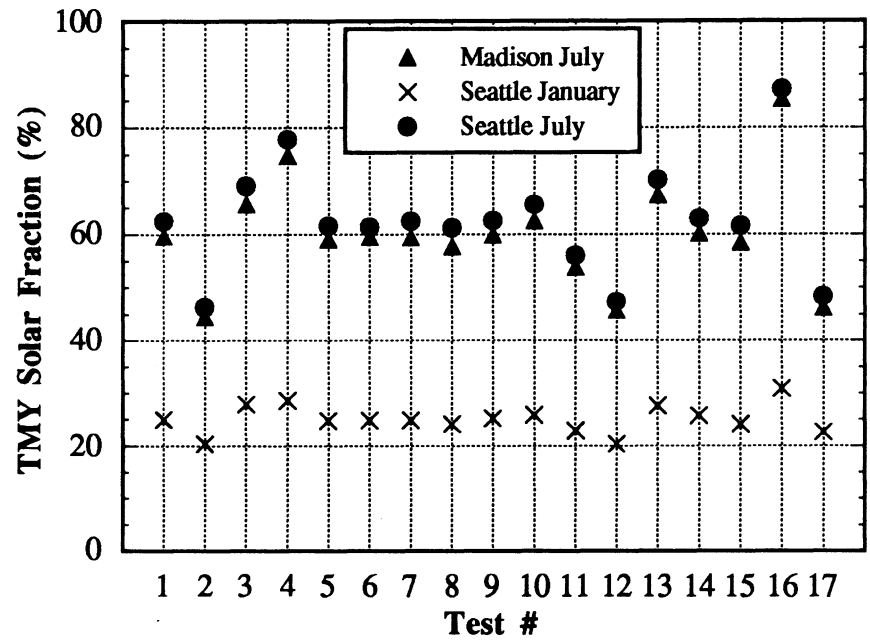


Figure 5.23 TMY Monthly Solar Fractions

5.3.2 Number and Order of Days Required to Represent a Month

Sets of simulations were presented in the previous section in order to investigate the effect of the random radiation component on SDHW system simulations. Analogously, simulation results are presented in this section to investigate the order and number of symmetrical days required to represent a month.

"Compressed" weather means representing a month of days with N statistically chosen days where N is less than the actual number of days within the month. The purpose of compressed weather is to "speed up" system simulations. An approximation of the monthly average system performance may be obtained by simulating the system over the N chosen days, and averaging the N daily results. One drawback with shrinking a month is that, due to the limited number of days, the initial conditions may have a big influence upon the final results. Simulations using compressed weather in this section are repeated in order to eliminate errors caused by the initial conditions. In other words, the simulation of the "month" is repeated using the conditions present at the end of one simulation as the initial conditions for the next simulation. The repetition is continued until periodic-steady state conditions are achieved.

Repeating the series of days also acts to reduce the number of different daily combinations possible For example, suppose a month is to be represented by a series of three days consisting of a poor, moderate, and good day. A statistics book would say there are 3! or six possible combinations for the days. Figure 5.24 graphically shows the six 3-day combinations. Figure 5.25 shows how the "1-2-3" series (i.e., upper left series in Figure 5.24) "looks" when repeated a number of times. What becomes obvious in Figure 5.25 is that the repeated "1-2-3" series also contains the "2-3-1" and "3-1-2" series. Similarly, a repeated "1-3-2" series contains the "2-1-3" and "3-2-1" series. Repetition of the series has reduced the number of possible combinations from six to two.

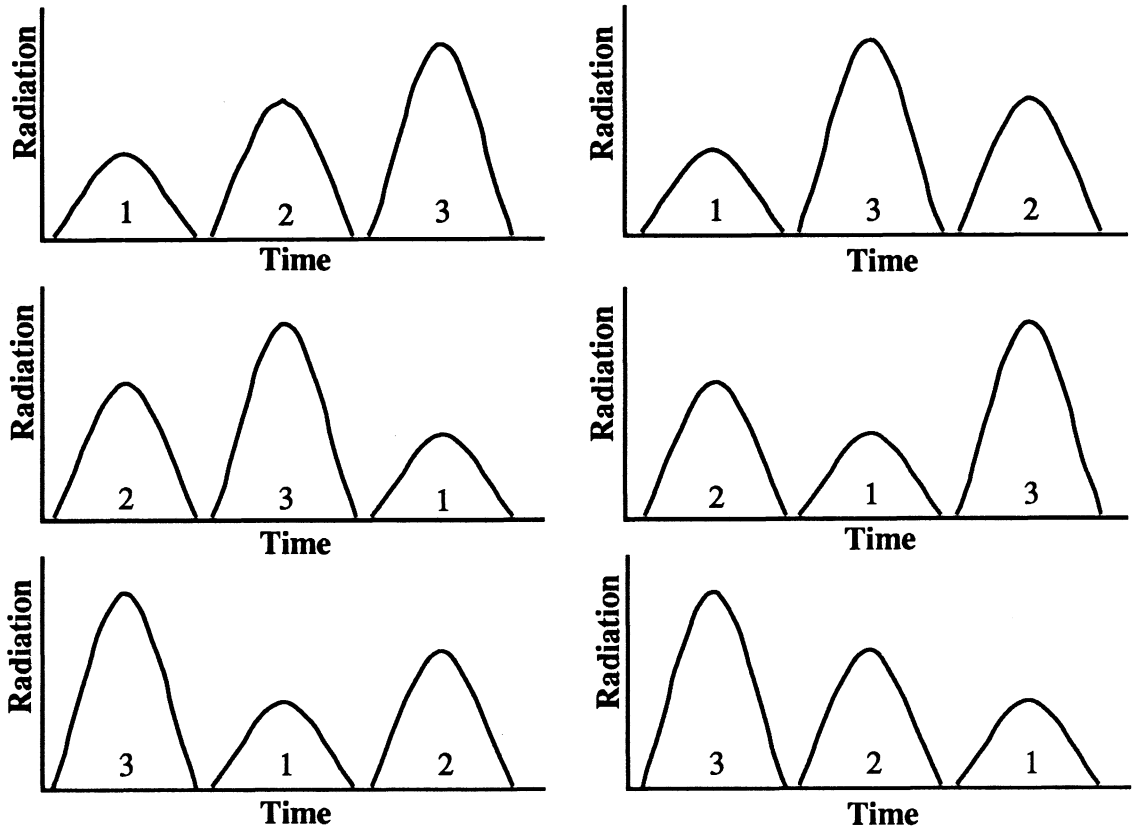


Figure 5.24 Six 3-Day Combinations

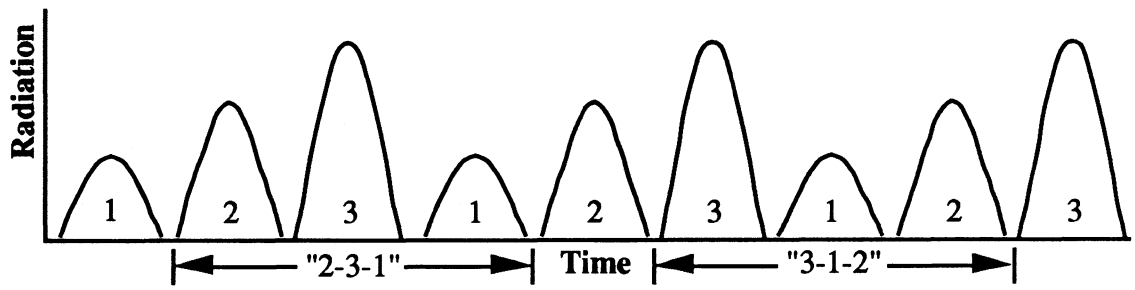


Figure 5.25 Repeated "1-2-3" Three-Day Series

Section 5.3.1.2 shows comparisons between full monthly simulations involving TMY and smooth TMY radiation profiles for two distinct cases. One case is characterized by the TMY and symmetrical days having the same K_t's. The other case is characterized by the TMY and symmetrical months having the same monthly tilted surface radiation.

Comparisons are presented in this section between SDHW system simulations using deterministic-compressed monthly and full monthly TRNSYS Type 54 weather profiles. The TRNSYS Type 54 weather generator creates temperature and radiation profiles having both a deterministic and random component, and orders the days via the Knight sequence. The Knight sequence orders the days such that an autocorrelation of 0.25 to 0.30 is achieved between the K_t values. Neither of the two conditions from the previous section are necessarily met in the comparisons presented in this section.

Table 5.4 outlines the steps followed by the TRNSYS Type 54 weather generator and a TRNSYS compatible compressed month deterministic weather generator. Both the Type 54 and compressed weather generators begin by selecting appropriate K_t's from Figure 5.1. Type 54 further reduces the K_t's into k_{tm} values via Figure 5.2. The k_t values are calculated using Equation (5.8) and, with knowledge of the hourly clearness indexes, the hourly horizontal radiation, I, is determined. An optional feature is to correct the K_t's such that the original daily K_t values are retained. Correction of the K_t's requires recalculation of the horizontal radiation values. The last step is to compute the diffuse radiation from the Erbs hourly correlations. The proposed compressed weather generator starts off by correcting the selected K_t's such that the monthly average value is retained. The total daily and diffuse daily radiation values are calculated from the K_t's. The daily radiation values are further broken down into hourly values. The last step is to correct the hourly values such that the long-term average radiation values are obtained.

Full Month Deterministic + Random Component	Compressed Month Deterministic Component Only
$K_t = \text{funct}(\overline{K_t})$	$K_t = \text{funct}(\overline{K_t})$
$k_{tm} = K_t \frac{r_t}{r_d}$	Correct K_t's such that $\sum^{\text{"month"}} \frac{K_t}{N} = \overline{K_t}$
$k_t = \text{funct}(k_{tm})$	$H = \frac{H}{H_o} H_o = K_t H_o$
$I = \frac{I}{I_o} I_o = k_t I_o$	$H_d = \text{funct}(H,K_t)$ *{Erbs daily correlations}*
Correct k_t's such that $\frac{\sum^{day} I}{H_o} = K_t$ *{Optional}*	$I = \frac{I}{H} H = r_t H$
Recalculate I's	$I_d = \frac{I_d}{H_d} H_d = r_d H_d$
$I_d = \text{funct}(I,k_t)$ *{Erbs hourly correlations}*	Correct I's and I_d's such that $\sum^{days} \sum^{hours} I = \overline{H} N$

Table 5.4 TRNSYS Type 54 and Compressed Weather Generator Algorithms

Obviously, the daily K_t values calculated by the Type 54 weather generator are not going to coincide with the N values chosen by the compressed weather generator when N is less than the actual number of days in the month. However, the K_t values will not coincide even if N is equal to the actual number of days in the month due to the correction of the K_t's used in the compressed weather subroutine. (It should be pointed out that the monthly average daily total horizontal surface radiation values, $\overline{H}$, will coincide even though the K_t's are not equivalent because of the corrections used in both routines to force this condition.) Neither will the monthly average daily tilted surface radiation values, $\overline{H}_t$,

be equivalent. The subroutines use two different procedures to calculate the total hourly and daily diffuse radiation values. As a consequence, even though $\overline{H}$ is the same in each case, the Type 54 and N symmetrical hourly radiation values, I and I_d, will not be equivalent at any particular time. The difference in horizontal surface radiation will result in different monthly tilted surface radiation values which will yield different values when summed and averaged over the month.

Yearly simulations were conducted for each of the possible one to four day weather profiles over the 17 tests outlined in Table 5.3. The ten series investigated are listed in Table 5.5. The tests presented in this section are slightly different than those in the previous in that the incidence angle modifier constant, b_0, has been changed from 0.0 to 0.1. The simulations were performed for Albuquerque, NM, Madison, WI, and Seattle, WA. Albuquerque, Madison, and Seattle were chosen because of their differing climatic conditions. The maximum series length was limited to four days for several reasons. First of all, including five day sequences increases the number of series from 10 to 34. Secondly, a five day "month" repeated twice will only decrease computational efforts by approximately 10/30. Finally, as will be shown later, a four day "month" can represent an actual month with little, if any, additional benefit found in increasing N beyond four.

Figures 5.26 through 5.28 show the Madison compressed-month simulation results. The full months in Figure 5.28 (i.e., the circles) are complete (i.e., 28, 29, 30 or 31 day) deterministic months ordered according the the Knight sequence. The difference observed between "months" having the same number of days for the same test number is due to the finite solar tank storage capacity. The purpose of the storage tank is to dampen out phase shifts between the charging (i.e., radiation) and discharging (i.e., load) profiles. An ideal storage tank will dampen out all charging-discharging phase shifts, and hence all forcing functions (i.e., radiation and load profiles) will "look" the same to the system. An

actual storage tank, however, will not be able to dampen out all charging-discharging shifts. In other words, a "1-2-3" charging profile will "look" different to the system than a "1-3-2" profile. The difference in charging profiles is the cause for the differences in system response if all else is the same.

Figures 5.26 through 5.28 indicate the simulation errors for a particular weather profile over the 17 test runs are within about ±1% of each other, with four exceptions. The agreement between test runs means there is little correlation between the system parameters investigated (i.e., those parameters listed in Table 5.3) and the weather profile, with the four exceptions. Test #16 runs all fall around an error value of zero, regardless of trends shown amongst the corresponding data points. For example, the three day "1-2-3" test #16 point falls at +0.2% where as most of the other three day "1-2-3" points lie around -1.5% error. Test #16 is characterized by a high $F_r(\tau\alpha)_n$ and low F_rU_L, and therefore has a low critical radiation level. It was pointed out in the utilizability discussion of Chapter 3 that a month may be represented by a single mean day if the critical radiation ratio is low enough. The test #16 critical radiation ratio is not low enough to use just one day. However, the ratio is low enough that the month may be excellently approximated by just two (or more) days.

# Of Days In "Month"	Possible Combinations
1	"1"
2	"1-2"
3	"1-2-3"
	"1-3-2"
4	"1-2-3-4"
	"1-3-4-2"
	"1-3-2-4"
	"1-4-3-2"
	"1-2-4-3"
	"1-4-2-3"

Table 5.5 Possible Combinations for One to Four-Day Months

Test #17 is another test which deviates from the others, with the abbreviated month simulation results showing large errors. The two to four day "months" all lie around -3%, where as the full month point is at +0.1% error. Test #17 is characterized by a low $F_r(\tau\alpha)_n$ and high F_rU_L which results in a high critical radiation level. Again, utilizability shows

months having a high critical radiation ratio and a low to moderate average clearness index cannot be as well represented by an abbreviated series of days.

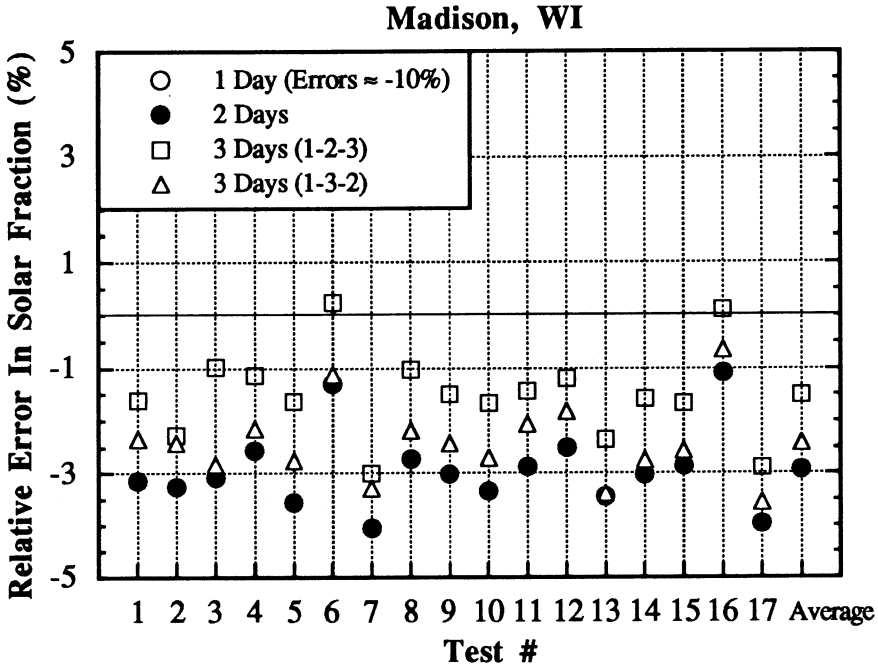


Figure 5.26 System Performance for Compressed vs. TRNSYS Type 54 Weather

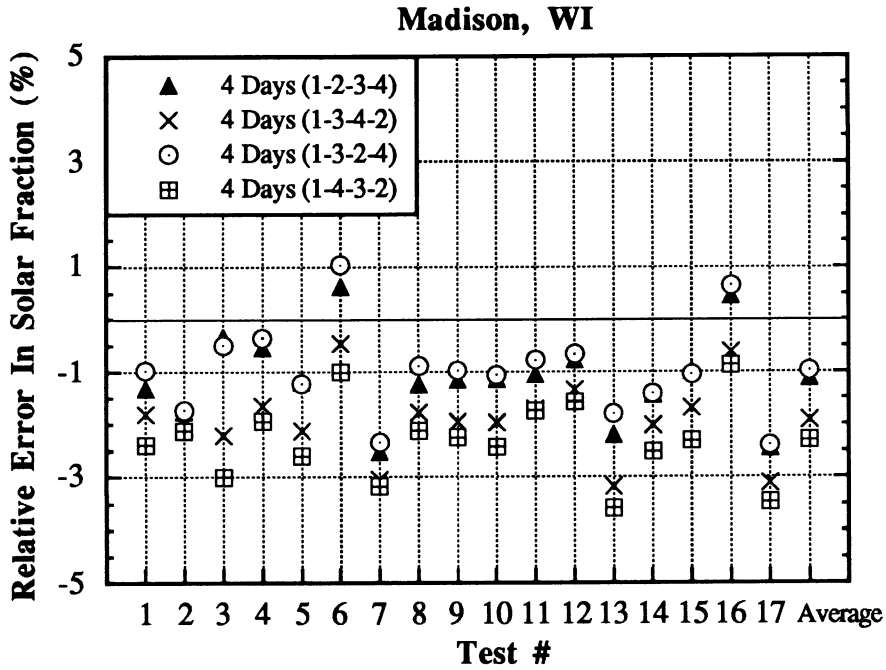


Figure 5.27 System Performance for Compressed vs. TRNSYS Type 54 Weather

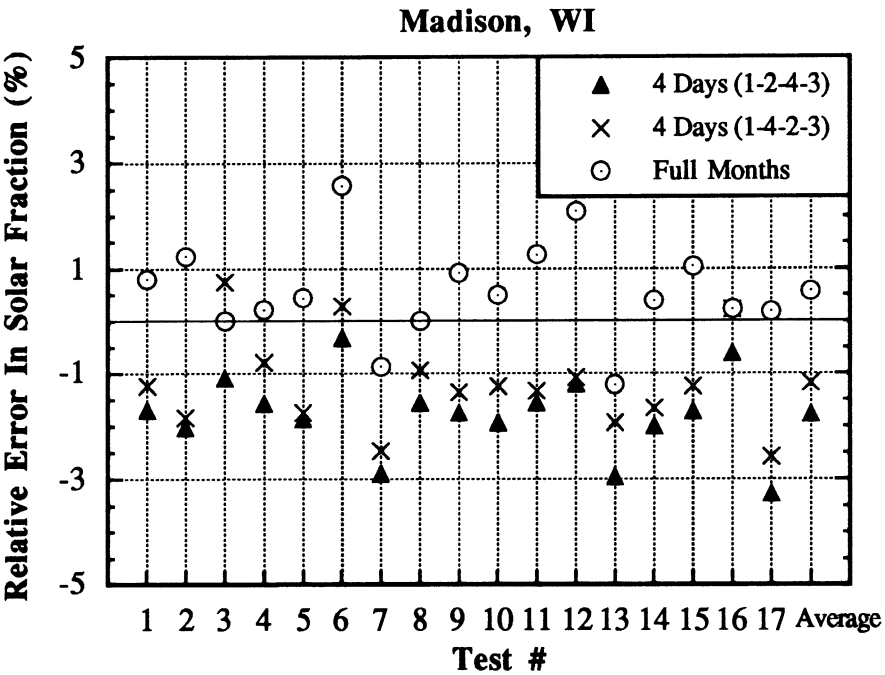


Figure 5.28 System Performance for Compressed vs. TRNSYS Type 54 Weather

Tests #6 and #7 are the other two tests which show deviations from the other corresponding test points. Test #6 shows a "jump" and #7 a "dip" if the points for a particular weather profile were connected. Tests #5, #6, and #7 are characterized by unique load profiles. As stated previously, a change in the discharging (i.e., load) profile may have an impact upon the system performance for the same charging (i.e., radiation) profile. The test #5 constant load profile, test #6 day profile, and test #7 evening load profile are shown in Figure 5.29. The RAND load profile used for tests #1 through #4 and and #8 through #17 is also shown in Figure 5.29.

The Type 54 July 3 Madison, WI bottom node tank temperature profiles for tests #1, #5, #6, and #7 are shown in Figure 5.30. The Type 54 July 3 is a typical day, having an H of 27,870 kJ/m^2 ($\overline{H}$ is equal to 21,953 kJ/m^2) and an average temperature of 22.7 °C (the monthly average daily temperature is 21.0 °C). The temperature profiles corresponding to a constant and RAND load are similar. The similar temperature profiles means the change from a RAND to constant load had little effect upon the system response. The fact the RAND and constant load profiles basically "look" the same to the system is further born out in that the relative errors between tests #1 and #5 are approximately equal for any given weather sequence. The day and evening loads, however, do show dissimilar tank temperature profiles as compared to the RAND load. The differing temperature profiles means switching between a RAND, day, and evening load profile significantly affects the charging-discharging relationship. The altered relationship will, in turn, affect the system response.

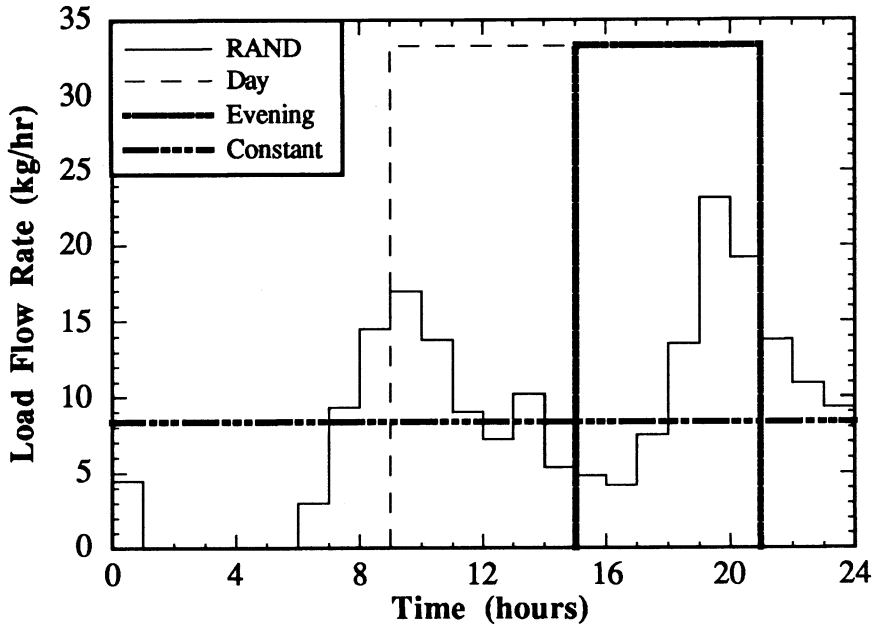


Figure 5.29 Load Profiles Investigated

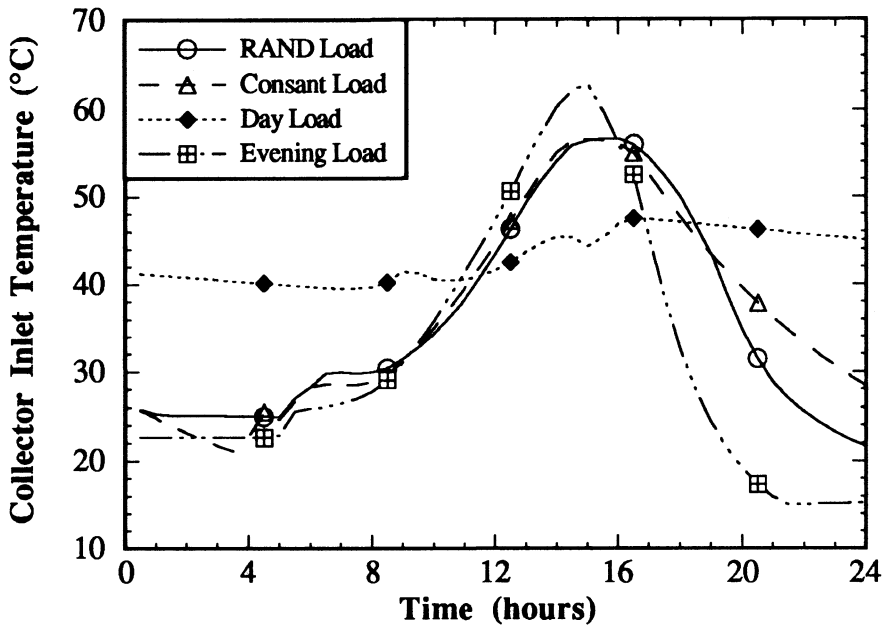


Figure 5.30 Madison July 3 Bottom Node Tank Temperatures

Some observations regarding the accuracy of compressed months may also be drawn. All of the deterministic months tend to under predict system performance, except for the full month case. The full month simulations tend to over predict the Type 54 results because of the modification made to he daily K_t values (i.e., step two in Table 5.4). Using one mean day to represent months with low to moderate $\overline{K}_t$'s and moderate to high I_{TC}'s is not appropriate, with errors being on the order of -10%. Increasing the number of days per month from one to two reduces errors which were on the order of -10% to -3%. Increasing the number of days beyond two becomes more complex because the sequence in which the days are taken becomes a factor. As stated previously, the daily sequence appears to be fairly independent of the system parameters, but is dependent upon the load (i.e., discharging) profile. No one sequence will out perform the others for every test case. A shift in trends is especially noticeable for tests #6 and #7. Hence, only generalizations may be made.

A four day series is not always better than a three day one. For example, a three day "1-2-3" series appears to give better results than a four day "1-4-3-2" series. Over all, two series appear to perform better than the others. The two preferred weather profiles are the four day "1-2-3-4" and "1-3-2-4" sequences, with the "1-3-2-4" sequence being slightly better than the other. The "1-3-2-4" error over the RAND load tests is around -1%. The test #6 error is +1.0% and that for #7 is -2.3%. The worst case is test #17 with an error on the order of -2.5%.

Figures 5.31 through 5.33 present the Seattle, WA, results. Similar conclusions as presented in the Madison case may be drawn. The Albuquerque results, being, with a few exceptions, less than or equal to ±1% are not shown. The excellent agreement between compressed and full monthly Albuquerque simulations is due to the high monthly $\overline{K}_t$'s. The Type 54 solar fractions for all three locations are shown in Figure 5.34.

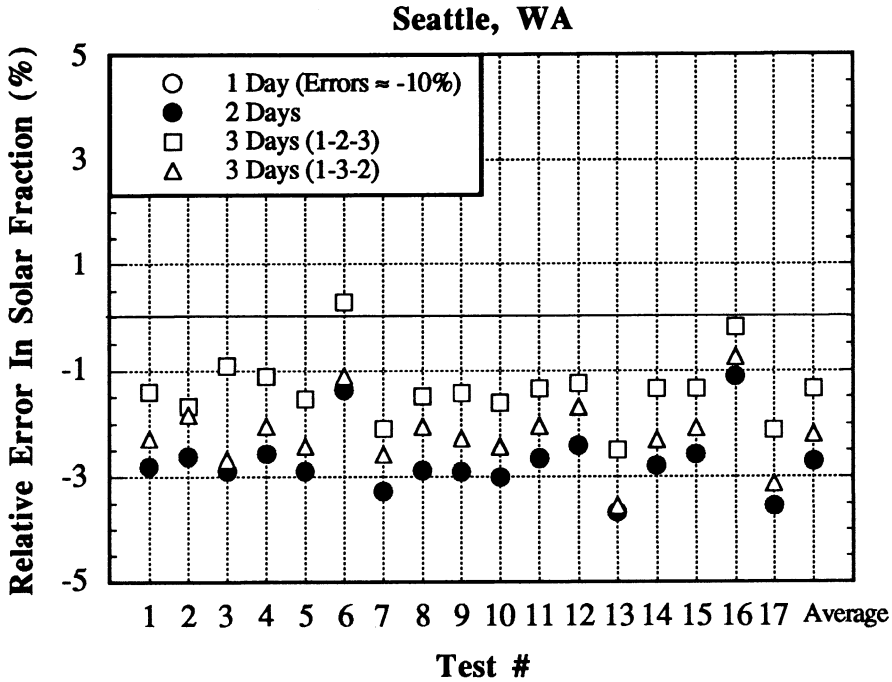


Figure 5.31 System Performance for Compressed vs. TRNSYS Type 54 Weather

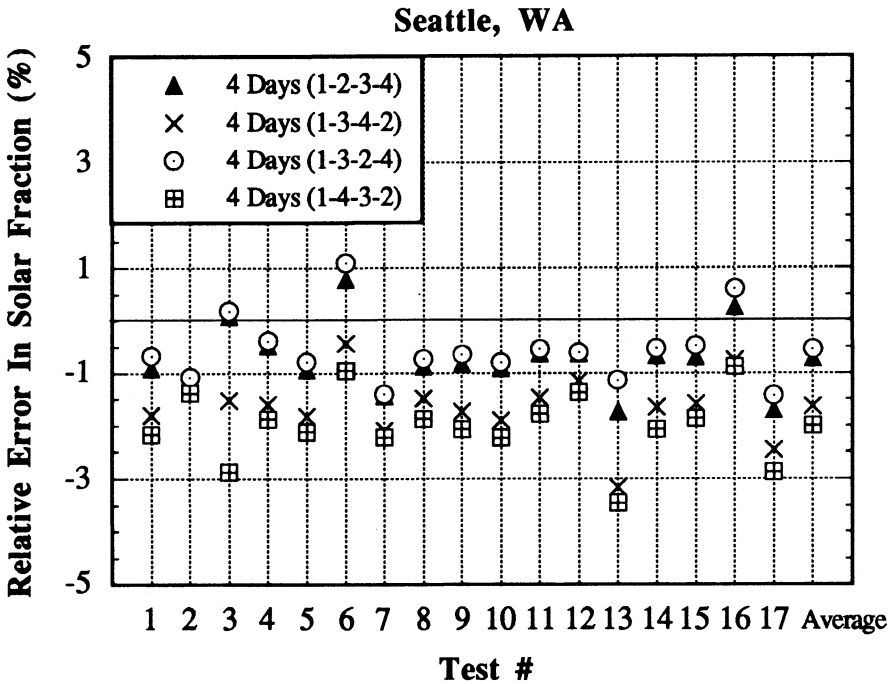


Figure 5.32 System Performance for Compressed vs. TRNSYS Type 54 Weather

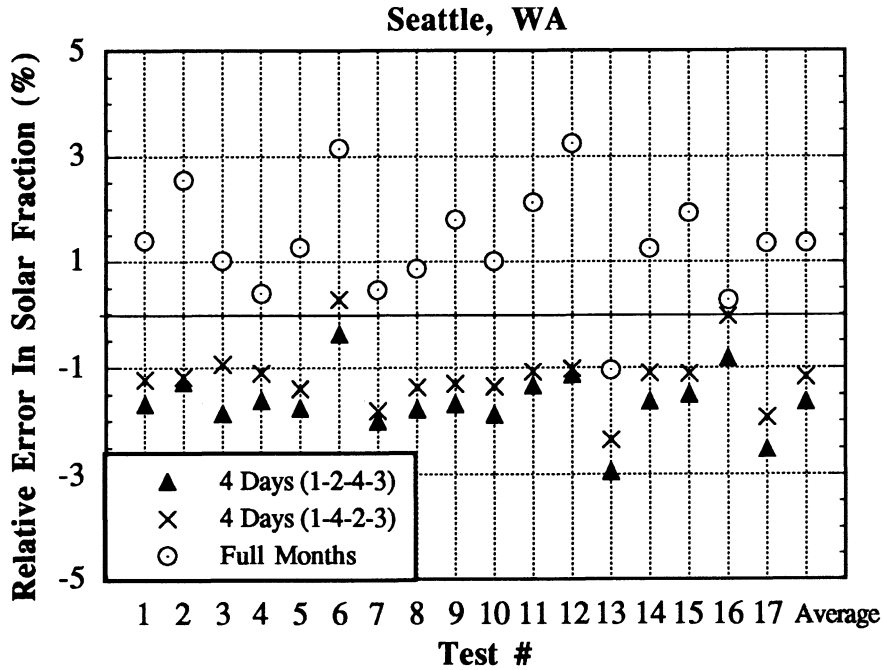


Figure 5.33 System Performance for Compressed vs. TRNSYS Type 54 Weather

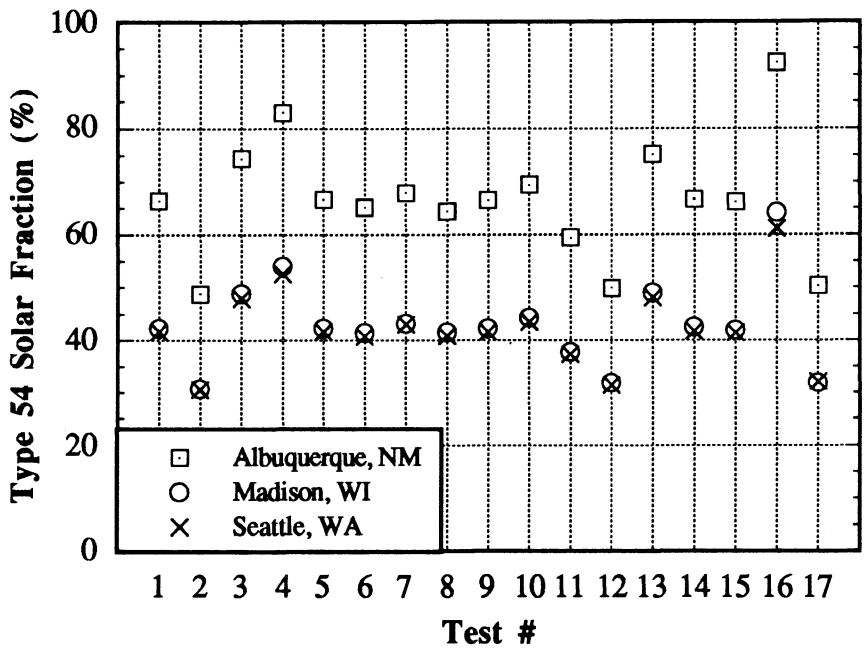


Figure 5.34 TRNSYS Type 54 Solar Fractions

5.3.3 Initial Condition Effects

The previous compressed month simulations were all repeated in order to eliminate the effects of initial conditions on simulation results. A month of N days was resimulated using the conditions at the end of the last day as the initial conditions for a repeat simulation(s) before moving on to the next "month". Figure 5.35 illustrates the simulation process used for the case of a two day "month". The "month" begins with an assumed initial tank temperature, T_i. The tank temperature at the end of the two days has changed from T_i to T_1*. The "month" is resimulated, but using T_1 rather than T_i as the initial condition, resulting in a final temperature of T_2. A final simulation with T_2 as the initial system temperature recovers T_2 as the final temperature. The recovering of T_2 means periodic steady-state conditions have been achieved, and the simulation can progress to the next month. "Months" with three or fewer days were generally found to reach the periodic steady-state solution on the third repeat simulation, requiring at least three simulations per month. Four day "months" generally reach the periodic steady-state solution on the second repeat simulation, requiring two simulations per month.

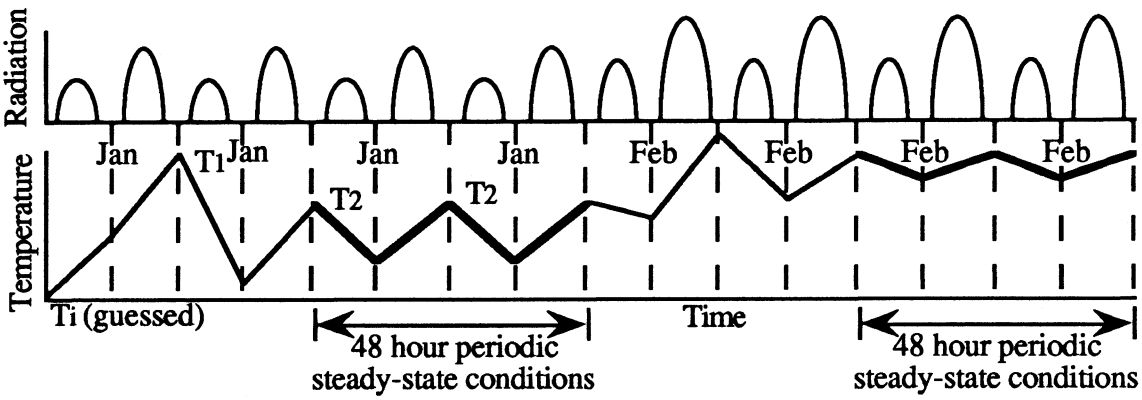


Figure 5.35 Repeated Two-Day "Month"

* Figure 5.35 is intended for illustrative purposes only. The linear temperature changes depicted should not be mistaken for actual temperature variations in terms of magnitude or shape.

A further reduction in computational effort can be achieved by not repeating simulations in order to damped out the initial conditions effects. Ability to neglect repetitious simulations is based upon the accuracy in which the initial "monthly" temperature can be predicted. For example, referring to Figure 5.35, the two day January simulation would not have had to been repeated if the simulation began with an initial temperature of T_2 instead of T_i. The ability to neglect repetitious simulations is also related to the tank volume. Figure 5.36 shows periodic steady-state bulk average tank temperatures for a large, moderate, and small tank volume using a four day "1-3-2-4" Madison, WI July month. The large volume, (i.e., solid line) is characterized by a fairly flat profile, implying a large time constant. The solid line in Figure 5.37 is the large tank volume January periodic steady-state simulation solution. The dashed line shown in Figure 5.37 is the first iteration tank profile. The assumed initial tank temperature is 0 °C, whereas the correct temperature is approximately 22 °C. Figure 5.37 illustrates that the entire four days are required for the simulation to reach a periodic steady-state temperature because of the large time constant.

A small volume (i.e., heavy dashed line in Figure 5.36) is characterized by much variation in temperature, implying a short time constant. The solid line in Figure 5.38 is the four day Madison, January periodic steady-state bulk average tank temperature for the small volume simulation. The dashed line in Figure 5.38 is the initial iteration temperature profile. The initial guessed temperature is 0 °C. Figure 5.38 illustrates that, due to the short time constant, the simulation reaches a periodic steady-state temperature by the beginning of the second day even though there is a 17 °C temperature difference between the assumed and correct initial temperatures.

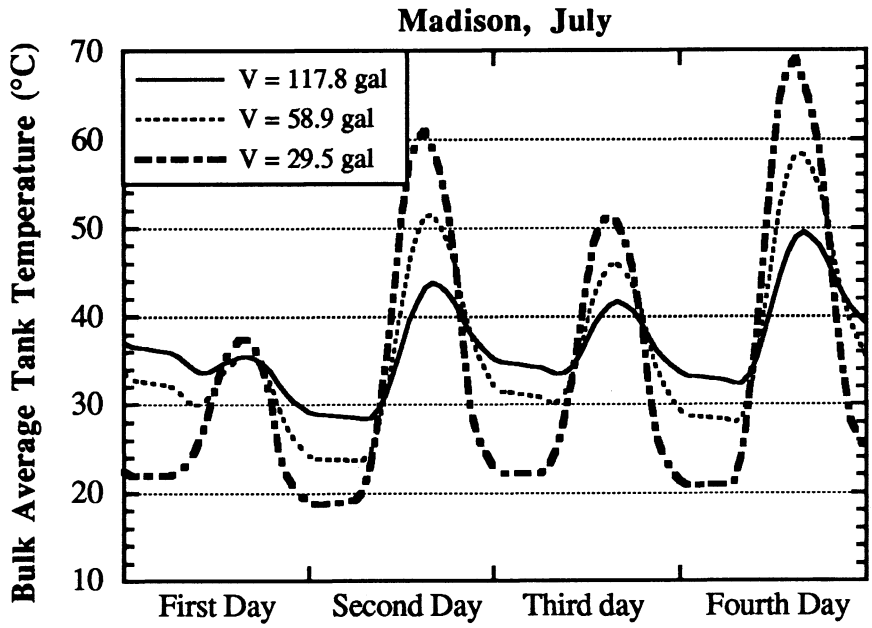


Figure 5.36 Temperature Fluctuations as a Function of Tank Volume

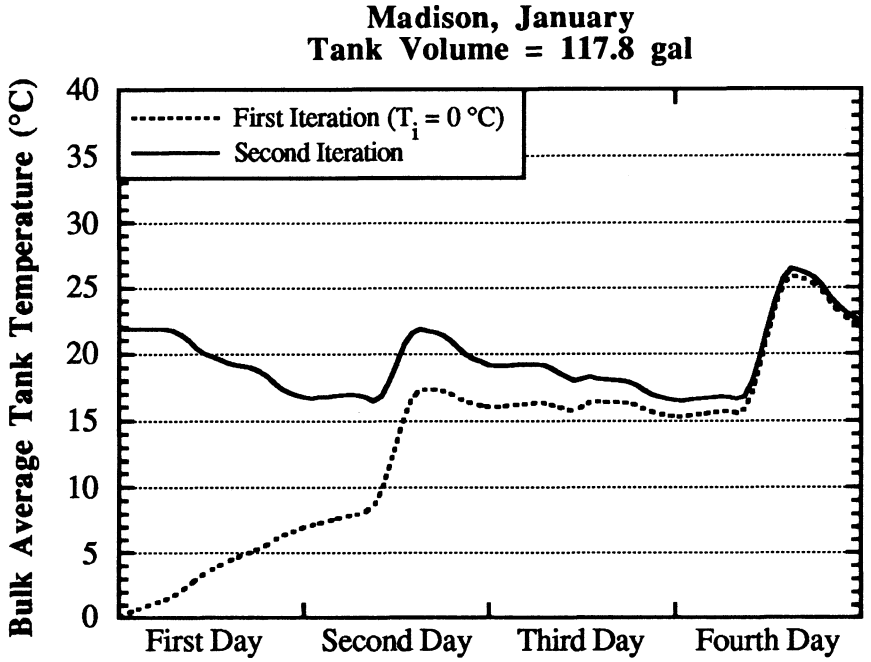


Figure 5.37 Transient and Steady-State Tank Temperature Profiles

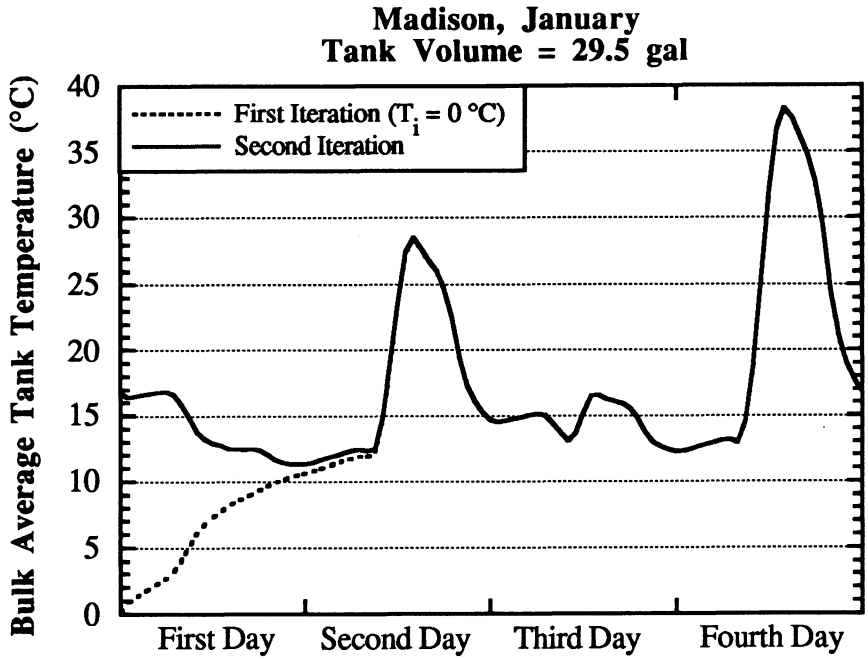


Figure 5.38 Transient and Steady-State Tank Temperature Profiles

Errors due to neglecting initial condition effects is thus related to the tank volume and accuracy in which the initial monthly temperatures can be predicted. Basing the "monthly" initial temperatures upon previous "monthly" performance is one idea for obtaining initial conditions. Variation in system performance between two consecutive months is small due to similar weather conditions. The tank temperature profile at the end of the last day composing the month was chosen as being a representative monthly tank temperature. The final temperature will equal the initial "monthly" temperature under periodic steady-state conditions. Hence, the final temperature should be both the beginning and end temperature, and therefore is a fairly representative monthly temperature.

Temperature	Value (°C)
T_{t_∞}	22
T_{mains}	10
$(T_{mains}+T_{set})/2$	32
T_{set}	54

Table 5.6 Various Representative System Temperatures

Selecting an initial January temperature is one problem with basing the initial monthly temperature upon the previous monthly performance since January precedes all other months. The base case and the arbitrary selected tests #3, #6, and #16 (tank volume is equal to 58.9 gal) were re-simulated to investigate the effect of initial January temperature on simulation results. The simulations used a "1-3-2-4" weather profile along with Albuquerque, Madison, and Seattle monthly average daily weather data. Four simulations differing in initial January tank temperature were performed for each location and test-case combination. One simulation began with a uniform initial solar tank temperature equal to the tank ambient temperature, T_{t_∞}. The other three simulations have a uniform initial temperature equal to the mains water temperature, T_{mains}, average of the mains and set water temperatures, and set water temperature, T_{set}, respectively. Table 5.6 lists the different initial temperatures and their values. The initial tank temperature profile for months other than January was equal to the profile at the end of the previous month. None of these monthly simulations were repeated.

The system being simulated has a three node solar and single node auxiliary tank. The auxiliary tank, being fully mixed in this case, has a temperature equal to the set temperature at all times. Figures 5.39 through 5.41 show the relative error in yearly solar fraction between the non-repetitive four-day month and TRNSYS Type 54 generated months. The error between periodic steady-state four-day months and TRNSYS Type 54 months is also shown for completeness. The error associated with using four-day repetitious months for the selected test cases is, as also shown in Figures 5.27 and 5.32, within ±1%, except for the Albuquerque test #16. The Albuquerque test #16 shows an

error of approximately +1.7%, as well as a yearly solar fraction of 92.5% (Figure 5.34). The high solar fraction indicates energy dumping is probably occurring throughout the summer months, if not most of the year. The higher Albuquerque test #16 error is probably traceable to a lack of agreement between discarded energy.

The figures also illustrate substantial errors in Madison and Seattle when the initial January tank temperature is equal to the set temperature. The set temperature is an indication of the upper system temperature, but has no effect upon the solar tank in this case. The system being simulated has a separate auxiliary tank "down stream" from the solar tank into which the auxiliary energy is being added. Therefore, the solar tank does not reach temperatures comparable to the set temperature except when the delivered solar energy is high, as in the Albuquerque simulations. Similarly, the Seattle tests #3, #6, #16 and Madison tests #6 and #16 indicate an average between the set and mains temperature tends to over predict the actual tank temperature. The mains temperature, unlike the set temperature in this system, does have an influence upon the solar tank temperature. The mains temperature, however is a representative lower system temperature, but not necessarily an indication of the January tank temperature. Simulations indicate the mains temperature tends to under predict the actual tank temperature.

Using the tank ambient temperature as the initial January tank temperature is the final option investigated. The January tank temperature is likely to be cool, yet warmer than the mains temperature, for most U.S. locations. The simulations demonstrate approximating the initial January temperature by the room temperature results in errors less than ±1% for all but two cases. The Seattle test #6 error of +1.2% suggests the tank ambient temperature is slightly optimistic for this simulations. A slightly greater error is also observed in the Albuquerque test #16 where, as stated previously, energy dumping may be having an effect on simulation results.

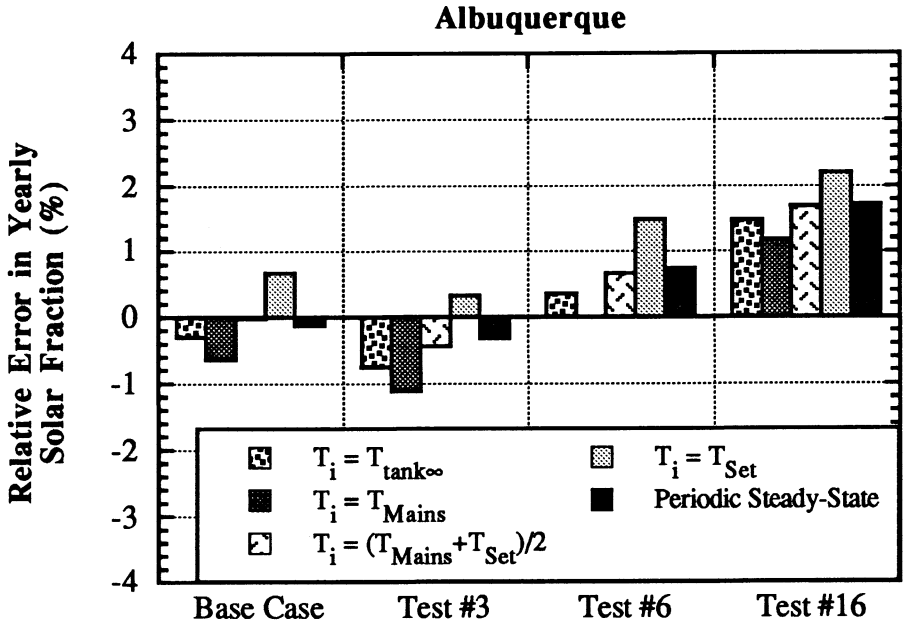


Figure 5.39 Effect of Initial Tank Temperature on Compressed Weather Simulations

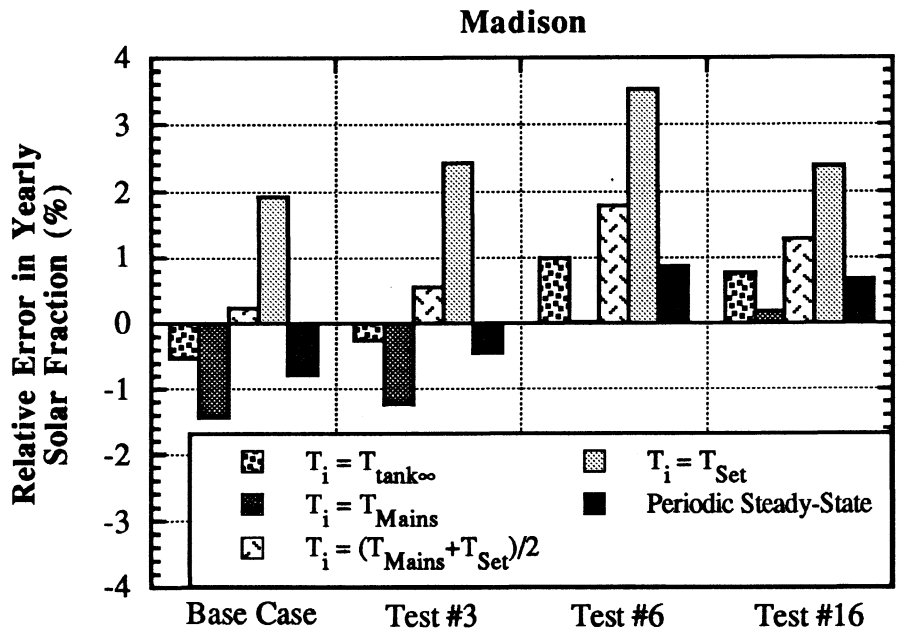


Figure 5.40 Effect of Initial Tank Temperature on Compressed Weather Simulations

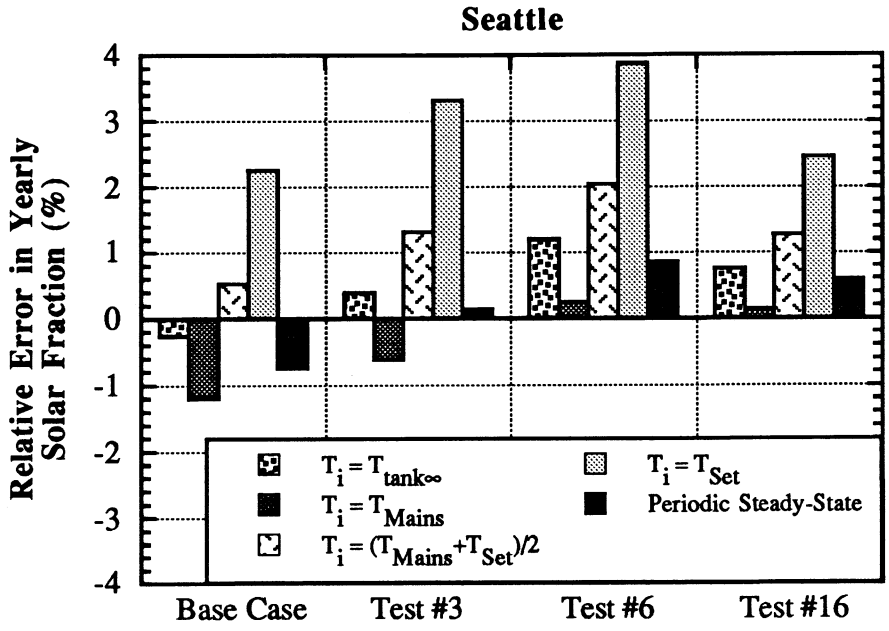


Figure 5.41 Effect of Initial Tank Temperature on Compressed Weather Simulations

The volume effect was investigated by performing base case simulations over a solar tank volume range of 18.6 gal to 115.0 gal. Figure 5.42 shows the relative difference in yearly solar fraction between repeated and non-repeated four day monthly simulations. The initial January tank temperature is set equal to the room temperature of 22 °C. The initial tank temperature profile for months other than January is equal to the temperature profile at the end of the previous "month". As expected, the results of Figure 5.42 move away from zero with increasing tank capacitance. The differences at large tank sizes is due to the longer time required by the system to reach a periodic steady-state temperature. The difference, however, is slight, with the values of Figure 5.42 all lying within ±0.8%.

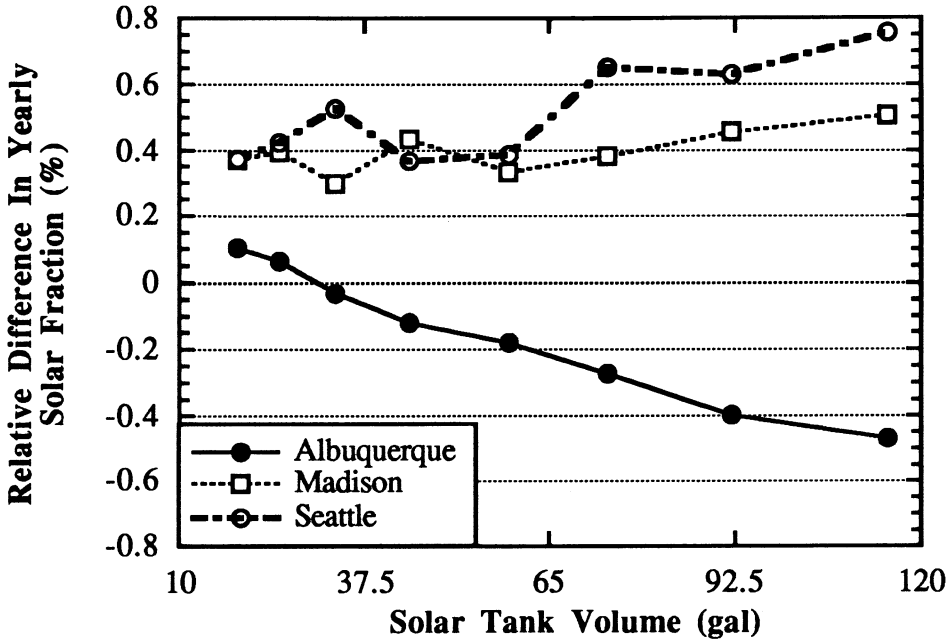


Figure 5.42 Effect of Storage Capacitance on Non-Repetitive Compressed Weather Simulations

Figure 5.43 shows the relative error in yearly solar fraction between non-repetitive four-day and TRNSYS Type 54 generated full months over the range of tank volumes. The moderate to low Madison and Seattle $\overline{K}_t$'s combined with small tank volumes enhance the importance of the random weather component on system performance. As expected, the four-day Madison and Seattle deterministic "months" under predict performance for systems having small tank volumes. The Madison and Seattle error dampens out to within ±0.5% with larger volumes. Albuquerque, having relatively high $\overline{K}_t$'s, is approximated to within ±1.0% throughout the range of tank values by a non-repetitive "1-3-2-4" month.

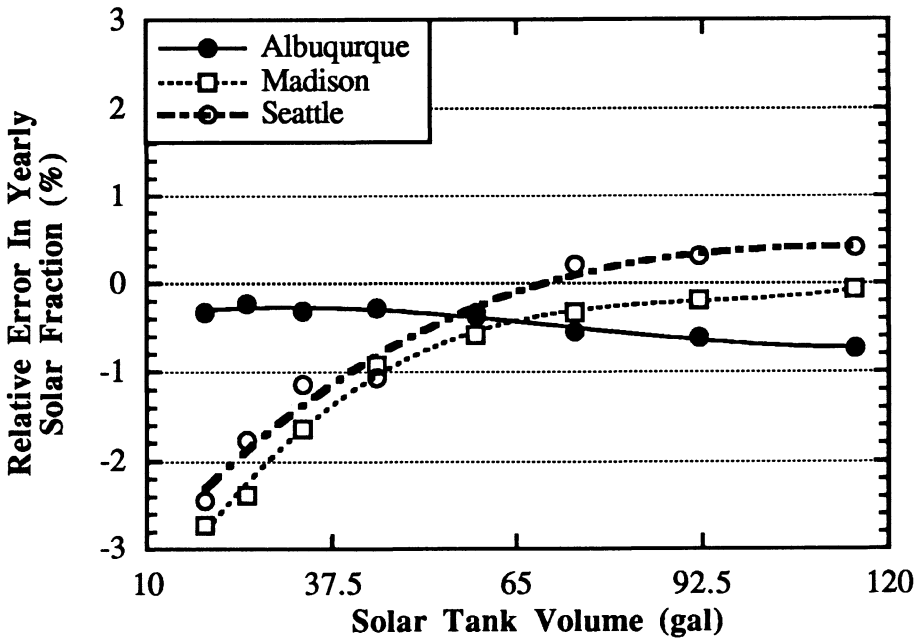


Figure 5.43 Comparison Between Non-Repetitive Deterministic Four-Day and TRNSYS Type 54 Simulation Results

Setting the initial January solar tank temperature to the room temperature yields accurate results for a two tank system. The upper tank node would be equal to the set temperature in single tank systems. The greater upper tank temperature will increase lower nodal temperatures. An average between the mains and set temperature may be more appropriate for the initial January lower tank nodal temperatures in such cases. However, any reasonable assumed temperature distribution along with a reasonable solar tank volume will yield results within engineering accuracy.

5.4 CONCLUSIONS

The purpose of this chapter is to illustrate that SDHW system simulations performed using a "1-3-2-4" non-repetitive weather profile will yield results comparable to simulations performed with full monthly hour-by-hour weather data. The maximum error in using abbreviated months is on the order of ±3%. Reducing simulation times is the purpose of using compressed weather rather than full monthly weather data. Figure 5.44 illustrates the CPU time required for four different simulations. The ordinate in Figure 5.44 is the required CPU time by the method the perform a yearly simulation divided by the CPU time required to perform a yearly simulation using non-repetitive four-day "months". Thus, the fourth entry has an ordinate value of 1.0 by definition.

Using individual TRNSYS Types and full (i.e., 28, 29, 30, or 31 day) months is the first entry. The second entry is for a combined system simulation using full months. A reduction in CPU time by a factor of 3.7 is obtained by using a combined system rather than individual TRNSYS Types. The third entry is for a combined system simulation using four-day "months" and two iterations per month. A reduction in CPU times by a factor of 5.7 is observed between the non-compressed and compressed-repetitive simulations. The final entry is for a combined system using four-day non-repetitive "months". An additional decrease in simulation times by a factor of 1.8 is associated by ignoring the monthly repetitions.

The main comparison to be made is between combined system non-compressed months (i.e., the second entry) and combined system compressed non-repetitive months (i.e., the fourth entry) where simulation times have decreased by a factor of 10. Hence, the use of four-day deterministic "months" decreases CPU times by a factor of 10, while yielding results with a relative accuracy of ±3%.

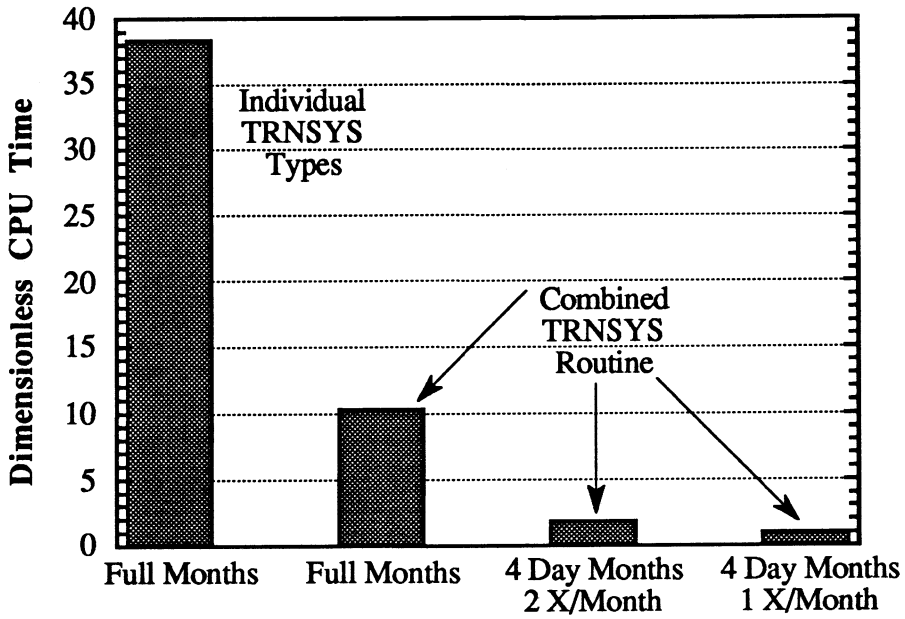


Figure 5.44 Simulation CPU Times

Chapter 6

Conclusions

6.1 RECOMMENDATIONS

As stated in Chapter 1, the purpose of this study is to recommend a long-term SDHW system performance method. Computer simulation methods are accurate, inexpensive, and require less time when compared to short-term test and extrapolate methods. The Solar Rating & Certification Corporation has expressed a favoritism towards the detailed simulation method if a simulation method is to be recommended. Detailed simulations are accurate, but require a sufficient amount of time to show the variation in system performance with location. A significant allotment of time is also required if the performance sensitivity to a parameter value or values is to be investigated by performing multiple detailed simulations. It was demonstrated in Chapter 5 that four properly chosen deterministic days accurately represents an entire month of weather for SDHW simulation purposes. Use of an abbreviated "month" reduces computational times by a factor of 10. The maximum relative difference in performance between SDHW systems subjected to non-repetitive deterministic-compressed and TRNSYS Type 54 months is 3%. This study recommends the use of a combined SDHW system model subjected to four day non-repetitive months to predict system long-term performance for a general rating purpose.

6.2 FUTURE WORK

It was shown in Chapter 5 that four deterministic days can represent a month for purposes of simulating an active SDHW system. Investigating the accuracy of compressed weather as input to non-active, such as thermosyphon, systems is one possible area for future work. Also, there are instances where more than four days may be desired to represent the month. For example, one may want extend the weather period such that it coincides with a given load period. Hence, another area of future work involves investigating the use of compressed weather in SDHW system simulations when N is greater than four.

Finally, incorporating the recommended performance prediction method with commercial software packages to produce a SDHW "rating tag" is a final recommendation. A SDHW rating tag would be analogous to the energy guides attached to many new home appliances. The promotion of SDHW sales by clearly showing yearly system cost is the purpose behind the rating tag. An attached rating tag will allow a perspective buyer to consciously compare the operational casts between solar aided and non-solar hot water systems. Figure 6.1 is an example of how a rating tag may appear.

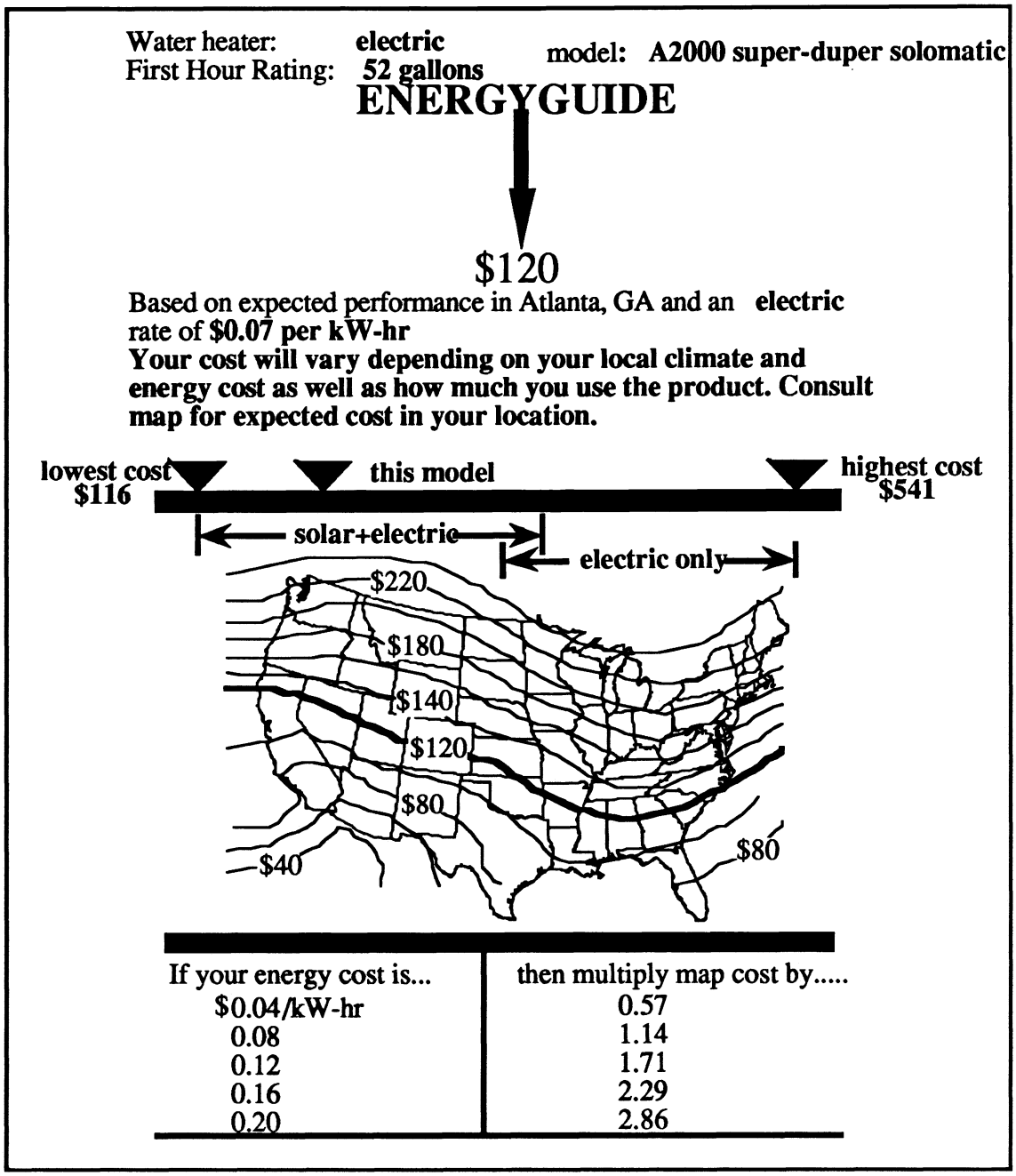


Figure 6.1 Example SDHW Rating Tag

Appendix A

FORTRAN Listings

A.1 INTRODUCTION

Appendix A contains one FORTRAN program and four TRNSYS [Klein et al., 1990] compatible subroutines. The FORTRAN program "spirkl" finds the lumped collector parameters, $F_r(\tau\alpha)_n$ and F_rU_L, and solar tank heat loss coefficient, UA, via the Spirkl method for SDHW systems having a fully mixed storage tank. The subroutine Type 75 generates a year of deterministic-compressed weather months. Type 74 is a combined active SDHW system model for either indirect or direct, one or two tank systems. Type 65 calculates the monthly average hourly energy gain across a solar collector using the Clark correlations for monthly average hourly utilizability. Finally, Type 71 models the ASHRAE-95/SRCC load flow profile.

A.2 SPIRKL ALGORITHM FOR FULLY MIXED SOLAR STORAGE TANKS

The program "spirkl" finds the lumped collector parameters, $F_r(\tau\alpha)_n$ and F_rU_L, and solar tank heat loss coefficient, UA, via the Spirkl method [1990]. "Spirkl" is only applicable for SDHW systems having a fully mixed storage tank. The method solves for the unknown system parameters via the Levenberg-Marquardt method such that the following objective function is minimized:

$$\chi^2(\bar{\xi}) = \sum_{u=1}^{u=n} \left[\frac{T_u - T(\theta_u, \bar{\xi})}{\sigma_u} \right]^2 \qquad \text{(A.1)}$$

where

χ^2 = The objective function. χ^2 is a function of the vector $\bar{\xi}$ where $\bar{\xi}$ contains the unknown parameters

u = Number of measured data points

T_u = Measured tank temperature at time θ_u

$T(\theta_u, \bar{\xi})$ = The predicted tank temperature at time θ_u

σ_u = Standard deviation of T_u due to measurement error

The program requires the following information from the file 'input.dat':

1. Filter frequency, f. All frequencies having a relative percent of the absolute maximum frequency less than f are set to zero.
2. Initial guess for tank UA
3. Initial guess for $A*F_r(\tau\alpha)_n$
4. Initial guess for $A*F_rU_L$
5. Number of parameters to be found (i.e., 1 to 3)
6. Agreement required between two successive χ^2 values before the parameter search is ended
7. Numbers corresponding to the parameters to be found. For example, an input of "1, 3" wound be specified if the first (i.e., UA) and third (i.e., $A*F_rU_L$) parameters are to be found.
8. Time step between data points, $\Delta\theta$
9. Collector slope, β
10. Ground reflectance, ρ

11. Incidence angle modifier constant, b_o
12. Solar storage tank capacitance, $M*c_p$

In addition, the following set of data is required for each time step:

1. Time
2. Tank temperature
3. Load mass flow rate, $\dot{m}_L$
4. Radiation on the tilted surface, I_T
5. Tank ambient temperature, T_{t_∞}
6. Collector ambient temperature, T_{c_∞}
7. Mains water temperature, T_{mains}
8. Incident angle, θ
9. Total radiation on the horizontal surface, I
10. Diffuse radiation on the horizontal surface, I_d
11. Standard deviation of the tank temperatures due to measurement errors

The measured tank temperatures are filtered to eliminate the effects of short-term dynamics on the regression analysis. The program utilizes the IMSL math routines *df2trf* and *df2trb* to filter the temperatures [IMSL, 1987]. The subroutines *Mrqmin* and *Mrqcof* from Press et al. [1986] are used to implement the optimization method. The predicted tank temperatures are found from a mass and energy balance upon the tank. The resulting differential equation is solved via the fourth order Runge Kutta method [Cheney and Kincaid, 1985]. The tank temperature derivatives required by the Levenberg-Marquardt

algorithm are calculated by the following second order error approximation along with Richardson extrapolation [Cheney and Kincaid, 1985]:

$$\frac{dT}{d\xi_k} \approx \frac{T(\xi_k + \Delta\xi_k) - T(\xi_k - \Delta\xi_k)}{2\,\Delta\xi_k} \qquad \text{(A.2)}$$

The value of $\Delta\xi_k$ is adjusted by changing the value of "h" in the subroutine *funcs*. The subroutine *Deriv* reduces the second order error by computing the Richardson two-dimensional triangular array. The number of array columns computed is adjusted by changing the value of "n" within the subroutine.

```
      program sprikl
c     dynamic parameter evaluation of a solar domestic hot water system
      implicit none

      integer n, unknow, dum
      integer i, j, flag, mfit

      parameter  (n = 500, unknow = 3, dum = n*2+15)

c     inputs
      real*8 Itotal(n), Id(n), temp(n), rho, bo, slope, mcp
      real*8 mfL(n), It(n), Tamb(n), Tca(n), Tmains(n), theta(n)
      real*8 sig(n), time(n)
      real*8 wfftr(dum), tempnew(n), tempfreq(n), step, CpL, tol
      real*8 a(unknow), beta(unknow), da(unknow), ss(unknow)
      real*8 Qu(n), tguess(n), dTda(n,unknow), beta2(unknow)
      real*8 covar(unknow,unknow), alpha(unknow,unknow)
      real*8 chisq, alamda, chiold, limit

      integer lista(unknow), ll(unknow)

      common /misc/step, wfftr, tol
      common /misc2/CpL, mcp
      common mfL, It, Tamb, Tca, Tmains
```

```
      data flag/0/, CpL/4.19/
      data alamda/-1./, chiold/1.e6/

c     ***read in data***

      open(20, file = 'input.dat', status = 'old')
      open(21, file = 'output4.dat',status = 'unknown')
      read(20,*)tol, a(1), a(2), a(3), mfit, limit
      read(20,*)(lista(j), j = 1, mfit)
      read(20,*)step, slope, rho, bo, Mcp
      do 50 j = 1, n
            read (20,*, end = 60)Time(j), Temp(j), mfl(j), It(j), Tamb(j), Tca(j),
     +            Tmains(j), theta(j), Itotal(j), Id(j), sig(j)
50    continue
60    continue
      flag = j-1

c     ****modify for incidence angle****

      call angle(flag, slope, Itotal, Id, It, bo, theta, rho)

      call transform(flag, wfftr, temp, tempnew, tempfreq, tol)

80    continue
      call mrqmin(tempnew, sig, flag, a, unknow, lista, mfit, ll, ss, covar, alpha, beta,
     +      beta2, da, chisq, alamda, tguess, dTda)

      write(6,1030)a(1), a(2), a(3), chisq

      if (dabs(chiold - chisq) .ge. limit) then
            chiold = chisq
            goto 80
      else
            alamda = 0.
      call mrqmin(tempnew, sig, flag, a, unknow, lista, mfit, ll, ss, covar, alpha, beta,
     +      beta2, da, chisq, alamda, tguess, dTda)

      write(21,1030)a(1), a(2), a(3), chisq

      do 110 j = 1, flag
            if (it(j) .gt. 0) then
                  Qu(j) = a(2)*It(j) - a(3)*(tguess(j) - tca(j))
            else
                  Qu(j) = 0
            end if
            write(21,1010)j,temp(j),tempnew(j),tguess(j),Qu(j)
110         continue
      end if

      close(20)
      close(21)
```

```
1010  format (i5,2x,f10.5,2x,f10.5,2x,f10.5,2x,f10.3,2x,f10.3)
1030  format (f10.5, 2x, f10.5, 2x, f10.5, 2x, f10.5)
      stop
      end
c+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
      subroutine angle(ndata, beta, Itotal, Id, It, bo, theta, rho)

c     modify for incidence angle
c     flat plates given bo only
c     routine returns It where It = It*kta

      implicit none

      integer ndata, j

      real*8 theta(ndata), Itotal(ndata), Id(ndata), It(ndata)
      real*8 effsky, effgnd, cosslp, fsky, fgnd, Idsky, Idgnd
      real*8 katb, katds, katdg, taualf
      real*8 rads, bo, beta, kta, rho

      data rads/.0174533/

      taualf(theta)=1.-bo*(1./dmax1(dble(0.5),dcosd(theta))-1.) - (1.-
     +      bo)*(dmax1(dble(60.),theta)-60.)/30.

c     use relations of Brandemuehl for effective incedence angles for diffuse
      do 65 j = 1, ndata
      effsky = 59.68-0.1388*beta+0.001497*beta*beta
      effgnd = 90.-0.5788*beta+0.002693*beta*beta
      cosslp = cos(beta*rads)
      fsky = (1.+ cosslp)/2.
      fgnd = (1.- cosslp)/2.
      Idsky = fsky*Id(j)
      Idgnd = rho*fgnd*Itotal(j)

c     use constant form ASHRAE tests

      katb = taualf(theta(j))
      katds = taualf(effsky)
      katdg = taualf(effgnd)
      if (It(j) .gt. 0) then
            Kta = (katb*(It(j) - Idsky - Idgnd) + katds*Idsky + katdg*Idgnd) / It(j)
            It(j) = It(j)*kta
      end if
65    continue

      return
      end
c+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
      subroutine transform(ndata, wfftr, ytime, ytrans, yfreq, tol)
```

```
      integer ndata, dum, n, j

      parameter (n = 500, dum = n*2+15)

      real*8 wfftr(dum), ytime(ndata), yfreq(ndata), ytrans(ndata)
      real*8 tol, max

c     ***transform and filter actual data***

      call dfftri (ndata, wfftr)
      call df2trf (ndata, ytime, yfreq, wfftr)
      max = 0
      do 70 j = 1, ndata
            if (dabs(yfreq(j)) .gt. max) max = dabs(yfreq(j))
70    continue
      do 80 j = 1, ndata
            if (dabs(yfreq(j)) .lt. max - max*tol) yfreq(j) = 0.
80    continue

c     ***re-transform (so that can see what is going on)***

      call df2trb(ndata, yfreq, ytrans, wfftr)
      do 90 j = 1, ndata
            ytrans(j) = ytrans(j)/float(ndata)
90    continue

      return
      end
c+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
      subroutine funcs (Temp, flag, x, tguess, dtdx, ma)

c     funcs evaluates the predicted temperatures and their derivatives;
c     used in the optimization routine

      implicit none

      integer ma, flag, n
      integer i, j

      real*8 x(ma), Temp(flag), tguess(flag), dtdx(flag,ma), h

      data h/.5/

      call funcs2(Temp, flag, x, tguess, ma)
      call deriv(Temp, flag, x, h, dtdx, ma)

      return
      end
c+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
      subroutine funcs2 (Temp, flag, x, tgnew, mx)
```

```
c       finds unknown temperatures

        implicit none

        integer mx, flag, k, dum, n

        parameter (n = 500, dum = n*2+15)

        real*8 x(mx), step, T, tol
        real*8 Temp(n), Tguess(n), Tgnew(flag), tgfreq(n)
        real*8 wfftr(dum)

        common /misc/step, wfftr, tol

c       find temperature values for all time steps (NOTE: using origional
c       temperature data in finding the 'predicted' temperatures)

        tgnew(1) = Temp(1)
        do 100 k = 2, flag
              call rk4(k, Temp(k-1), T, mx, x, step)
              tgnew(k) = T
 100    continue

        return
        end
c+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
        subroutine rk4(k, Temp, T, mx, x, step)

        implicit none

        integer mx, k, n

        parameter (n = 500)

        real*8 x(mx), Temp, T
        real*8 step, Cpl, Mcp
        real*8 mfL(n), It(n), Tamb(n), Tca(n), Tmains(n)
        real*8 Tcabar, Tambbar, Tmbar, mfLbar, Itbar
        real*8 F1, F2, F3, F4, dTdt

        common /misc2/ CpL, mcp
        common mfL, It, Tamb, Tca, Tmains

        Tcabar = ( Tca(k-1) + Tca(k) )/2.
        Tambbar = ( Tamb(k-1) + Tamb(k) )/2.
        Tmbar = ( Tmains(k-1) + Tmains(k) )/2.
        mfLbar = ( mfL(k-1) + mfL(k) )/2.
        Itbar = ( It(k-1) + It(k) )/2.

        F1 = step*dTdt(x, mx, Temp, Tca(k-1), Tamb(k-1), Tmains(k-1), cpL,
     +          mfL(k-1), mcp,It(k-1))
```

```
      F2 = step*dTdt(x, mx, Temp+F1/2., Tcabar, Tambbar, Tmbar, cpL, mfLbar, mcp,
     +     Itbar)

      F3 = step*dTdt(x, mx, Temp+F2/2., Tcabar, Tambbar, Tmbar, cpL, mfLbar, mcp,
     +     Itbar)

      F4 = step*dTdt(x, mx, Temp+F3, Tca(k), Tamb(k), Tmains(k), cpL, mfL(k),
     +     mcp, It(k))

      T = Temp + .1666667*(F1 + 2.*F2 + 2.*F3 + F4)

      return
      end
c+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
      real*8 function dTdt(x,mx,T,Tca,Tamb,Tmains,cpL,mfL,mcp,It)

      implicit none

      integer mx

      real*8 x(mx), T, Tca, Tamb, Tmains, cpL, mfL, mcp, It

      dTdt = (x(2)*It - x(3)*(t - Tca) - x(1)*(T - Tamb) - mfL*cpL* (T - Tmains)) / mcp

      return
      end
c+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
      subroutine deriv(Temp, ndata, a, h, dyda, ma)

      implicit none

      integer points, ndata, x, n, ma, k, l, j, i, w

      parameter (points = 500, x = 3, n = 4)

      real*8 ahold(x), hold, h, aminus, aplus
      real*8 d(points,x,n,n), a(ma), y2(points)
      real*8 y1(points), dyda(ndata, ma), q, Temp(ndata)

      hold = h
      do 20 k = 1, ma
            h = hold
            do 2 w = 1, ma
                  ahold(w) = a(w)
2           continue

            do 15 i = 1, n
                  aplus = a(k) + h
                  ahold(k) = aplus
                  call funcs2(Temp, ndata, ahold, y2, ma)
```

```
                aminus = a(k) - h
                ahold(k) = aminus
                call funcs2(Temp, ndata, ahold, y1, ma)
                do 10 l = 1, ndata
                    d(l,k,i,1) = (y2(l) - y1(l)) / (2.*h)
                    q = 4.
                    do 5 j = 1, i-1
                        d(l,k,i,j+1) = d(l,k,i,j) + (d(l,k,i,j) - d(l,k,i-1,j)) /
                            (q-1.)
                        q = 4.*q
5                   continue
10              continue
                h = h/2.
15          continue
20      continue
        do 40 l = 1, ndata
            do 30 k = 1, ma
                dyda(l,k) = d(l,k,n,n)
30          continue
40      continue

        return
        end
c++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
        subroutine mrqmin(y, sig, ndata, a, ma, lista, mfit, ll, ss, covar, alpha, beta, beta2,
    +       da, chisq, alamda, tguess, dyda)

        implicit none

        integer mmax, mfit, ma, ndata, ihit, lista(ma)
        integer k, kk, j, ll(mfit)

        parameter (mmax = 3)

        real*8 y(ndata), sig(ndata), a(ma), tguess(ndata)
        real*8 covar(mfit,mfit), alpha(mfit,mfit), atry(mmax)
        real*8 da(mfit), beta(mfit), ss(mfit), dyda(ndata,ma)
        real*8 alamda, chisq, ochisq, beta2(mfit)

        if (alamda .lt. 0) then
            kk = mfit+1
            do 12 j = 1, ma
                ihit = 0
                do 11 k = 1, mfit
                    if (lista(k) .eq. j) ihit = ihit+1
11              continue
                if (ihit .eq. 0) then
                    lista(kk) = j
                    kk = kk + 1
                else if (ihit .gt. 1) then
                    pause 'Improper permutation in LISTA'
```

```
                end if
12          continue
            if (kk .ne. (ma+1)) pause 'Impmroper permutation in LISTA'
            alamda = 0.001
            call mrqcof(y, sig, ndata, a, ma, lista, mfit, alpha, beta, chisq, tguess,
     +          dyda)
            ochisq = chisq
            do 13 j = 1, ma
                atry(j) = a(j)
13          continue
        end if
        do 15 j = 1, mfit
            do 14 k = 1, mfit
                covar(j,k) = alpha(j,k)
14          continue
            covar(j,j) = alpha(j,j)*(1. + alamda)
            beta2(j) = beta(j)
15      continue
        call gauss(covar, mfit, beta2, da, ll, ss)
        if (alamda .eq. 0.) then
            return
        end if
        do 16 j = 1, mfit
            atry(lista(j)) = a(lista(j))+da(j)
16      continue
        call mrqcof(y, sig, ndata, atry, ma, lista, mfit, covar, da, chisq, tguess, dyda)
        if (chisq .lt. ochisq) then
            alamda = .1*alamda
            ochisq = chisq
            do 18 j = 1, mfit
                do 17 k = 1, mfit
                    alpha(j,k) = covar(j,k)
17              continue
                beta(j) = da(j)
                a(lista(j)) = atry(lista(j))
18          continue
        else
            alamda = 10.*alamda
            chisq = ochisq
        end if

        return
        end
c++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
        subroutine mrqcof(y, sig, ndata, a, ma, lista, mfit, alpha, beta, chisq, ymod, dyda)

        implicit none

        integer mpoints, mfit, ma, lista(mfit), ndata
        integer j, k, i
```

```
      parameter (mpoints = 500)

      real*8 y(ndata), sig(ndata), alpha(mfit,mfit)
      real*8 beta(mfit), dyda(ndata,ma), a(ma), ymod(mpoints)
      real*8 chisq, sig2i, dy, wt

      do 12 j = 1, mfit
            do 11 k = 1, j
                  alpha(j,j) = 0.
11          continue
            beta(j) = 0.
12    continue
      chisq = 0.

      call funcs(y, ndata, a, ymod, dyda, ma)

      do 15 i = 1, ndata
            sig2i = 1./(sig(i)*sig(i))
            dy = y(i) - ymod(i)
            do 14 j = 1, mfit
                  wt = dyda(i,lista(j)) * sig2i
                  do 13 k = 1, j
                        alpha(j,k) = alpha(j,k)+wt*dyda(i,lista(k))
13                continue
                  beta(j) = beta(j) + dy*wt
14          continue
            chisq = chisq+dy*dy*sig2i
15    continue
      do 17 j = 2, mfit
            do 16 k = 1, j-1
                  alpha(k,j) = alpha(j,k)
16          continue
17    continue

      return
      end
c+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
      subroutine gauss(a, n, b, x, L, S)

c     solves a*x = b via gauss elimination with scaled partial piviting

      implicit none

      integer n, i, j, k, lk, L(n)
      real*8 a(n,n), b(n), x(n), S(n)
      real*8 smax, rmax, r, xmult, sum

c     set up "S"

      do 3 i = 1, n
            L(i) = i
```

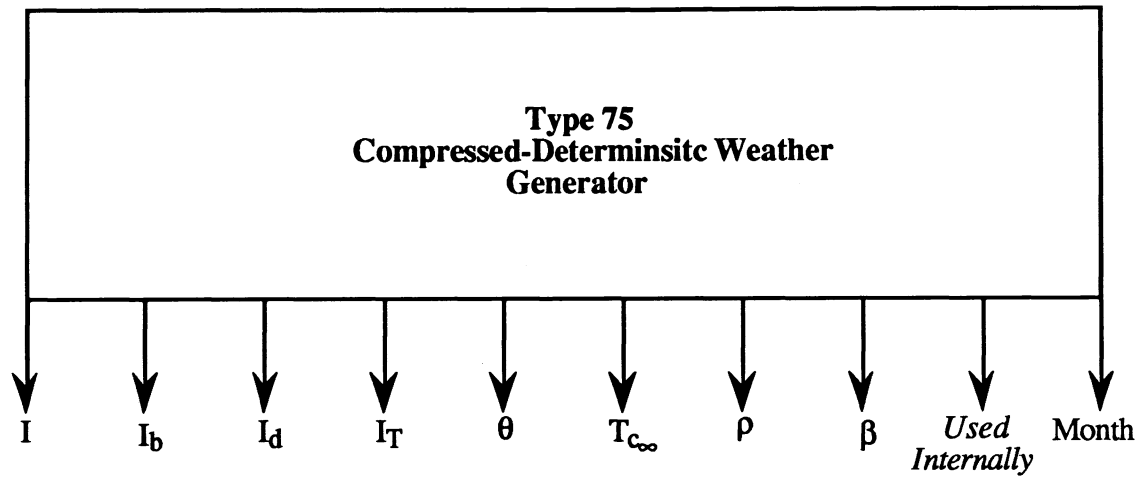
```
            smax = 0.
            do 2 j = 1, n
                  smax = dmax1(smax, dabs(a(i,j)))
2           continue
            s(i) = smax
3     continue

c     set up "L" and fix "a"

      do 7 k = 1, n-1
            rmax = 0.
            do 4 i = k, n
                  r = dabs(a(l(i),k))/s(l(i))
                  if (r .le. rmax) goto 4
                  j = i
                  rmax = r
4           continue
            lk = l(j)
            l(j) = l(k)
            l(k) = lk
            do 6 i = k+1, n
                  xmult = a(l(i),k)/a(lk,k)
                  do 5 j = k+1, n
                        a(l(i),j) = a(l(i),j) - xmult*a(lk,j)
5                 continue
                  a(l(i),k) = xmult
6           continue
7     continue

c     fix "b"

      do 15 k = 1, n-1
            do 10 i = k+1, n
                  b(l(i)) = b(l(i)) - a(l(i),k)*b(l(k))
10          continue
15    continue

c     back substitution to find x

      x(n) = b(l(n))/a(l(n),n)
      do 25 i = n-1, 1, -1
            sum = b(l(i))
            do 20 j = i+1, n
                  sum = sum - a(l(i),j)*x(j)
20          continue
            x(i) = sum/a(l(i),i)
25    continue

      return
      end
```

A.3 TYPE 75: COMPRESSED-DETERMINISTIC WEATHER GENERATOR

The TRNSYS compatible Type 75 generates a "year" of deterministic-compressed weather months. The subroutine does not have any inputs, but does read in the following information from the file 'weather.dat':

1. Latitude
2. Longitude
3. Collector Slope, β
4. Ordering of the days (only if the number of days/month < 28)
5. $\overline{H}$'s
6. $\overline{K}_t$'s
7. $\overline{T}_m$'s
8. ρ's

The program automatically generates full (i.e., 28, 29, 30, or 31 day) months if the number of days (i.e., parameter number one) is greater than 28. Days composing full months are ordered according to the Knight sequence [Knight, 1988]. The days are chosen from the Bendt et al. [1981] cumulative radiation frequency curves. The diffuse radiation is calculated according the the Erbs horizontal surface radiation correlations [1980]. The tilted surface radiation is calculated using the Reindl correlations [1988]. The temperature profile consists of monthly average hourly values calculated according to the Erbs relationship [1984]. Figure A.1 illustrates the Type 75 information flow.



Parameters

1. #Days/Month
2. # Iterations/Month
3. $\Delta\theta$

Figure A.1 Type 75 Information Flow Diagram

```
subroutine type75(time, xin, out, t, dtdt, par, info)

implicit none

dimension out(20), par(3), info(10), s(5000)

dimension order(31), day(31), f(31), dkt(31)
dimension ws(31), wrise(31), delta(31)
dimension Hbar(12), Ktbar(12), Tbar(12), rho(12)
dimension It(31,24), Th(31,24)
dimension I(31,24), Io(31,24), Ib(31,24), Id(31,24)
dimension cdelta(31), sdelta(31), cw(31,24)
dimension mprof(3,31)
dimension mday(12), length(12), begin(12)

real time, xin, out, t, dtdt, par, s

real cf, delt, hour
real Gsc, rdconv
real beta, phi, rho, long
real delta, ws, tset, wrise, trise, daylen
real w, theta, w1, w2
real Hbar, I, Id, Ib, Io, It, Itilt, Ho, H, Hd
real a, b, rt, rd, Rb, Rtilda
```

```
      real f, Ktbar, dKt, hkt
      real sumkt, sumI, sumIt
      real bendt, interp
      real Ai, ff
      real Tbar, Tstar, th, amplit

      real halfbeta, s3hlfb, cbeta, sbeta, cphi, sphi
      real cpmb, spmb, cdelta, sdelta, cws, sws
      real cw, dKt2, dKt3, dKt4
      real mday, length, begin, rep, iter, clock

      integer iday, ndays, day, order, start, stop
      integer k, j, flag, month, lower, upper, xday
      integer mprof, isq, info, nstore, iav

      parameter (Gsc=1.367, rdconv=.017453)

      data (mday(j), j=1,12)/17.,47.,75.,105.,135.,162.,198.,228.,
     +                       258.,288.,318.,344./
      data (length(j), j=1,12)/31.,28.,31.,30.,31.,30.,31.,31.,30.,
     +                       31.,30.,31./
      data (begin(j), j=1,12)/1.,32.,60.,91.,121.,152.,182.,213.,
     +                       244.,274.,305.,335./

      data mprof/24,28,11,19,18,3,2,4,9,20,14,23,8,16,21,26,
     1           15,10,22,17,5,1,6,29,12,7,31,30,27,13,25,
     2  24,27,11,19,18,3,2,4,9,20,14,23,8,16,21,7,
     3           22,10,28,6,5,1,26,29,12,17,31,30,15,13,25,
     4  24,27,11,4,18,3,2,19,9,25,14,23,8,16,21,26,
     5           22,10,15,17,5,1,6,29,12,7,31,20,28,13,30/

      common/store/nstore, iav, s

      if (info(7) .eq. -1) then
            info(6) = 20
            info(10) = 4512
            call typeck(1, info, 0, 3, 0)
      end if

      ndays = int(par(1)+.1)
      rep = par(2)
      delt = par(3)

      if (info(8) .eq. 1) then
            iter = float(ndays)*rep*24.
            month = 0
            clock = 5000.

            open(3, file='weather.dat', status='old')
            read(3,*)phi, long, beta
            if (ndays .lt. 28) read(3,*) (order(j), j = 1, ndays)
```

```
            beta = beta*rdconv
            halfbeta = beta/2.
            s3hlfb = sin(halfbeta)*sin(halfbeta)*sin(halfbeta)
            cbeta = cos(beta)
            sbeta = sin(beta)

            phi = phi * rdconv
            cphi = cos(phi)
            sphi = sin(phi)
            cpmb = cos(phi - beta)
            spmb = sin(phi - beta)

            flag = 0

            read(3,*) (Hbar(j), j = 1, 12)
                  do 2 j = 1, 12
                        s(info(10)+flag) = hbar(j)
                        flag = flag + 1
2           continue

            read(3,*) (Ktbar(j), j = 1, 12)
                  do 3 j = 1, 12
                        s(info(10)+flag) = ktbar(j)
                        flag = flag + 1
3           continue

            read(3,*) (Tbar(j), j = 1, 12)
            do 4 j = 1, 12
                  s(info(10)+flag) = Tbar(j)
                  flag = flag + 1
4           continue

            read(3,*) (rho(j), j = 1, 12)
            do 5 j = 1, 12
                  s(info(10)+flag) = rho(j)
                  flag = flag + 1
5           continue

            close(3)
      else
            beta = out(8)
            iter = out(9)
            month = out(10)
            clock = out(11)
            phi = out(12)
            long = out(13)
            s3hlfb = out(14)
            cphi = out(15)
            sphi = out(16)
            cbeta = out(17)
            sbeta = out(18)
```

```
            cpmb = out(19)
            spmb = out(20)
        end if

        if (info(7) .eq. 0) clock = clock + delt

        if (clock .gt. iter) then
            if (info(8) .eq. 1) then
                clock = -delt
            else
                clock = delt
            end if
            month = month + 1

            if (month .ge. 13) goto 130

c           ****select days out of the month****
            if (ndays .lt. 28) then
                if (ndays .eq. 1) then
                    day(1) = mday(month)
                else
                    day(1) = begin(month) + nint(length(month) / (2.*float(ndays)))
                    do 10 k = 2, ndays
                        day(k) = day(k-1) + nint(length(month)/float(ndays))
10                  continue
                end if
            else
                ndays = int(length(month)+.1)
                iter = float(ndays)*rep*24.
                do 15 k = 0, ndays-1, 1
                    day(k) = begin(month) + k
15              continue
            end if

c           ****calculate Ho (assuming Ho constant throughout month)****

            delta(1) = 23.45 * sin(360.*(284.+mday(month))/365.*rdconv)
            delta(1) = delta(1) * rdconv
            ws(1) = acos(-tan(phi)*tan(delta(1)))
            Ho = 37595.2 * (1.+.033*cos(360.*mday(month)/365.*rdconv))
            Ho = Ho * (cphi*cos(delta(1))*sin(ws(1)) + ws(1)*sphi*
     +          sin(delta(1)))

c           ****main loop of the program****

            sumkt = 0.
            sumI = 0.
            do 20 k = 1, ndays
                f(k) = (2.*float(k) - 1.)/(2.*float(ndays))
                dKt(k) = bendt(f(k), s(info(10)+11+month))
                delta(k) = 23.45 * sin(360.*(284.+float(day(k)))/365.*rdconv)
```

```
                delta(k) = delta(k) * rdconv
                ws(k) = acos(-tan(phi)*tan(delta(k)))
                tset = ws(k)/.2618 + 12.
                daylen = .1333 * ws(k)/rdconv
                trise = tset - daylen
                wrise(k) = (trise - 12.) * .2618
                sumkt = sumkt + dkt(k)

20          continue

c           ****employ correction factor to daily Kt values****

            do 30 k = 1, ndays
                dkt(k) = dkt(k)*s(info(10)+11+month)*float(ndays)/sumkt
30          continue

c           ****generate a daily Kt series****
c           ****use Knight ordering if substantial # of days present****

            if (ndays .lt. 28) then
39              continue
            else
                isq = 2
                if (s(info(10)+11+month) .le. .45) isq = 1
                if (s(info(10)+11+month) .ge. .55) isq = 3
                flag = 0
                do 40 k = 1, 31
                    if (mprof(isq,k) .le. ndays) then
                        flag = flag + 1
                        order(flag) = mprof(isq,K)
                    end if
40              continue
            end if

c           ****calculate the hourly-horizontal-surface radiation profile****

            do 60 iday = 1, ndays
                k = order(iday)
                cdelta(k) = cos(delta(k))
                sdelta(k) = sin(delta(k))
                cws = cos(ws(k))
                sws = sin(ws(k))
                a = .409 + .5016*sin(ws(k)-1.0472)
                b = .6609 - .4767*sin(ws(k)-1.0472)
                H = dKt(k)*Ho

c               ****use Erbs correlations****

                dkt2 = dkt(k)*dkt(k)
                dkt3 = dkt2*dkt(k)
                dkt4 = dkt3*dkt(k)
```

```
                if (ws(k) .lt. 1.4207) then
                    if (dkt(k) .lt. .715) then
                        Hd = 1. - .2727*dkt(k) + 2.4495*dkt2
     +                          - 11.9514*dkt3 + 9.3879*dkt4
                        Hd = Hd * H
                    else
                        Hd = H*(.143)
                    end if
                else
                    if (dkt(k) .lt. .722) then
                        Hd = 1. + .2832*dkt(k) - 2.5557*dkt2
     +                          + .8448*dkt3
                        Hd = Hd * H
                    else
                        Hd = H*(.175)
                    end if
                end if

                start = int(wrise(k)*3.82 + 12.)
                stop = int(ws(k)*3.82 + 12.) + 1

                do 50 j = start, stop, 1
                    w = (j - 12.) * .2618
                    cw(k,j) = cos(w)
                    rd = .1309*(cw(k,j) - cws) / (sws - ws(k)*cws)
                    rt = rd*(a + b*cw(k,j))
                    w2 = ((j + .5) - 12.) * .2618
                    w1 = ((j - .5) - 12.) * .2618
                    Io(k,j) = 18797.6*(1. + .033*cos(360.
     +                      *float(day(k))/365.*rdconv))
                    Io(k,j) = Io(k,j) * (cphi*cdelta(k)
     +              *(sin(w2) - sin(w1)) + (w2-w1)*sphi*sdelta(k))
                    I(k,j) = rt*H
                    if ( I(k,j) .lt. 0. ) I(k,j) =  0.
                    sumI = sumI + I(k,j)
                    if ((I(k,j) .gt. 0.) .and. (Io(k,j) .gt. 0.)) then
                        Id(k,j) = rd*Hd
                        if (Id(k,j) .lt. 0.) Id(k,j) = 0.
                        Ib(k,j) = I(k,j) - Id(k,j)
                        if (Ib(k,j) .lt. 0.) Ib(k,j) = 0.
                    end if

50              continue
60          continue

c           **** employ correction factor****

            cf = s(info(10)+month-1)*float(ndays)/sumI
            flag = 0
            do 80 iday = 1, ndays
                do 70 j = 1, 24
```

```
                    k = order(iday)
                    I(k,j) = I(k,j)*cf
                    s(info(10)+48+flag) = I(k,j)
                    Ib(k,j) = Ib(k,j)*cf
                    s(info(10)+792+flag) = Ib(k,j)
                    Id(k,j) = Id(k,j)*cf
                    s(info(10)+1536+flag) = Id(k,j)
                    flag = flag + 1
70              continue
80          continue

c           ****calculate the tilted surface radiation****
c           ****gamma = 0 and HDR model****

            flag = 0
            do 100 iday = 1, ndays
                k = order(iday)
                do 90 j = 1, 24
                    if ((I(k,j) .gt. 0.) .and. (Io(k,j) .gt. 0.)) then
                        Rb = cpmb*cdelta(k)*cw(k,j) + spmb*sdelta(k)
                        Rb = Rb / (cphi*cdelta(k)*cw(k,j) + sphi*sdelta(k))
                        Ai = Ib(k,j) / Io(k,j)
                        ff = sqrt(Ib(k,j) / I(k,j))

                        It(k,j) = (Ib(k,j) + Id(k,j)*Ai)*Rb + Id(k,j)*(1 - Ai)*
     +                          (1.+cbeta)/2.*(1. + ff*s3hlfb)
                        It(k,j) = It(k,j)+I(k,j)*s(info(10)+35+month)*(1.-cbeta)/2.
                        if (It(k,j) .lt. 0.) It(k,j) = 0.
                        s(info(10)+2280+flag) = It(k,j)

c                       ****find theta****

                        theta = sdelta(k)*sphi*cbeta - sdelta(k)*cphi*sbeta +
     +                      cdelta(k)*cphi*cbeta*cw(k,j) + cdelta(k)*sphi*sbeta*cw(k,j)
                        theta = acos(theta)/rdconv
                        s(info(10)+3024+flag) = theta

                    else
                        It(k,j) = 0.
                        s(2280+flag) = It(k,j)
                        theta = 90.
                        s(info(10)+3024+flag) = theta
                    end if
                    flag = flag + 1
90              continue
100         continue

c           ****calculate the temperature profile****

            flag = 0
            do 120 iday = 1, ndays
```

```
                k = order(iday)
                do 110 j = 1, 24
                    amplit = 25.8*s(info(10)+11+month) - 5.21
                    Tstar = .2618*(j - 1.)
                    Th(k,j) = .4632*cos(Tstar - 3.805) + .0984*cos
     +                  (2.*Tstar - .36) + .0168*cos(3.*Tstar - .822)
                    Th(k,j) = amplit* (Th(k,j) + .0138*cos(4.*Tstar - 3.513))
                    Th(k,j) = Th(k,j) + s(info(10)+23+month)
                    s(info(10)+3768+flag) = Th(k,j)
                    flag = flag + 1
110             continue
120         continue

        end if

        if (ndays .eq. 1) then
            xday = 1
        else
            xday = 1 + nint(mod(float(int(clock)/24), float(ndays)))
        end if
        hour = mod(clock,24.)
        lower = max(1, int(float((xday-1))*24. + hour))
        upper = lower + 1
        out(1)=interp(hour,s(info(10)+47+lower),s(info(10)+47+upper))
        out(2)=interp(hour,s(info(10)+791+lower),s(info(10)+791+upper))
        out(3)=interp(hour,s(info(10)+1535+lower),s(info(10)+1535+upper))
        out(4)=interp(hour,s(info(10)+2279+lower),s(info(10)+2279+upper))
        out(5)=interp(hour,s(info(10)+3023+lower),s(info(10)+3023+upper))
        out(6)=interp(hour,s(info(10)+3767+lower),s(info(10)+3767+upper))
        out(7)=s(info(10)+35+month)
        out(9) = iter
        out(10) = month
        out(11) = clock

        if (info(8) .eq. 1) then
            out(8) = beta/rdconv
            out(12) = phi
            out(13) = long
            out(14) = s3hlfb
            out(15) = cphi
            out(16) = sphi
            out(17) = cbeta
            out(18) = sbeta
            out(19) = cpmb
            out(20) = spmb
        end if

130     continue
        return
        end
c=======================================================================
```

```
      REAL FUNCTION BENDT(F,KTBAR)
C ***COMPUTES THE VALUE OF KT CORRESPONDING TO F USING BENDT***
C ***EXPRESSION FOR THE LIU AND JORDAN KT DISTRIBUTION **********
      REAL F,KTBAR,KT,KMIN,KMAX,CGAM,GAM

      KMIN = 0.05
      KMAX = .6313 + .267*KTBAR - 11.9*(KTBAR-.75)**8
      CGAM = (KMAX-KMIN) / (KMAX-KTBAR)
      GAM = -1.498 + (1.184*CGAM - 27.182*EXP(-1.5*CGAM))/(KMAX-KMIN)
      BENDT = LOG( (1.-F) * EXP(GAM*KMIN) + F * EXP(GAM*KMAX) ) / GAM

      RETURN
      end
c=============================================================
      real function interp(a, b, c)

c     linear interpolation given two values

      real a, b, c, na

      na = a - int(a)

      interp = na*(c - b) + b

      return
      end
```

A.4 TYPE 74: COMBINED SDHW SYSTEM MODEL

The TRNSYS Type 74 models a liquid collector-storage SDHW system. The model is compatible with either compressed or non-compressed weather data. The model simulates an indirect system when the value of the collector-tank heat exchanger (parameter number 22) is less than 1. A direct system results by setting the effectiveness to a value of unity. A one or two tank system may be specified. The auxiliary is added at the top of the solar tank in the case of a single tank system, or exclusively in the second tank when a two tank system is indicated. A one to ten node solar tank may be specified. The auxiliary tank, if present, is modeled as being fully mixed. A daily periodic RAND load profile [Mutch, 1974] is the default load condition. However, any load profile may be specified via the last input.

No instantaneous output is generated when compressed weather is used as input. Rather, the following monthly average daily values are written to a file:

1. Bulk average tank temperature, $\hat{T}_t$
2. Radiation on the horizontal surface, $\overline{H}$
3. Radiation on the tilted surface, $\overline{H}_T$
4. Collector ambient temperature, $\overline{T}_{c_\infty}$
5. Energy gain across the collector
6. Load side energy gain across the solar tank
7. Heat loss from solar tank
8. Auxiliary energy input
9. Auxiliary tank heat loss
10. Total load draw

Figure A.2 illustrates the Type 74 non-compressed and compressed information flow. A list of parameters is given in Table A.1

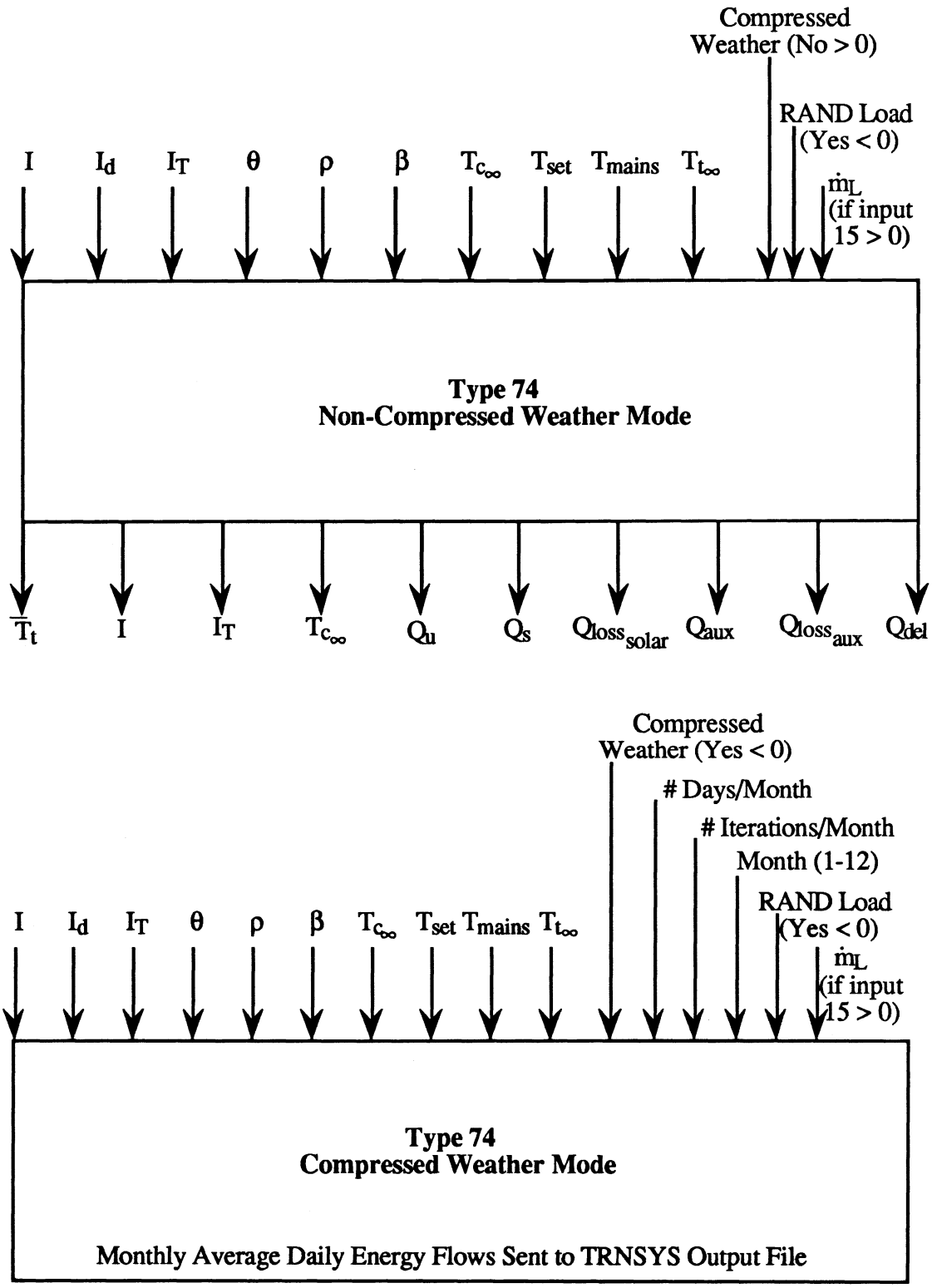


Figure A.2 Type 74 Information Flow Diagram

Type 74 Parameters

1. $\Delta\theta$	11. $\dot{m}_{collector,\ test\ conditions}$	21. U_{aux}
2. $\dot{m}_{collector}$	12. A	22. ε
3. $c_{p_{collector}}$	13. # Panels in Series	23. Total Daily Load Draw
4. $\dot{m}_{tank}$	14. # Solar Tank Nodes	24. $\Delta T_{upper\ dead\ band}$
5. $c_{p_{tank}}$	15. $V_{t_{solar}}$	25. $\Delta T_{lower\ dead\ band}$
6. ρ_{tank}	16. ht_{solar}	26. Tempering (Yes < 0)
7. $T_{boil,\ tank}$	17. U_{solar}	27. $L_{pipe,\ in}$
8. $F_R(\tau\alpha)_n$	18. # of Tanks	28. $L_{pipe,\ out}$
9. F_rU_L	19. $V_{t_{aux}}$	29. $Diam_{pipe}$
10. b_o	20. ht_{aux}	30. U_{pipe}
		31. T_i

Table A.1 Type 74 Parameters

```
      subroutine type74(time, xin, out, t, dtdt, par, info)
c     THIS SUBROUTINE MODELS A LIQUID COLLECTOR-STORAGE SYSTEM
      implicit none
      dimension randwl(24), days(12)
      dimension xin(16), out(20), par(31), info(10), s(5000)
      dimension Tsolavg(10), Tisol(10), Tfsol(10)
      dimension aa(10), bb(10), cc(10), dd(10), ee(10)
      real time, xin, out, t, dtdt, par, s
      real Frtan, FrUl, b0, Area, rho, delt
      real I, Id, It
      real beta, beta2, cbeta, theta
      real gtest, gcap
      real r7, r6, r5, r4, r3, r2, r1, xk, FpUl, nser
      real effsky, effgnd, fsky, fgnd, Idsky, Idgnd
```

```
      real Ktab, ktads, ktadg, kta, taualf

      real randwl, daily, days, mfload, capload, capl, capnet
      real mfcoll, cpcoll, capcoll, mftnk, cptnk, captnk, rhot
      real col2, col3, capmin
      real volsol, htsol, Usol, UAtop, volaux, htaux, Uaux
      real diamsol, masssol, capsol, UAsol
      real diamaux, massaux, capaux, UAaux, hxeff
      real udeadb, ldeadb, lin, lout, dpipe, Upipe, UApout, UApin
      real Tboil, TIsol, TFsol, Tsolavg, TIaux, TFaux, Tauxavg
      real tmains, tset, Tbar, TC, TCout, tnnodes, tenv, tamb, tbulk
      real D1, D2, D3, D4, D5
      real aa, bb, cc, dd, ee, U, V, QB
      real rdconv, pi, eps, iter, begin, clock
      real pid2, pidiam, FpUla

      integer info, start, stop, step, j, nx, k, flag
      integer nnodes, ntanks, nstore, iav, month
      integer gamma, stick, mstick, niterate, miterate

      logical mix, comp, Rload

      parameter (pi=3.1415927, rdconv=.017453)

      common/store/nstore, iav, s

      data mstick/11/, miterate/10/, eps/.001/

      data randwl/0.0225,0.0,0.0,0.0,0.0,0.0,0.015,0.0465,0.0725,
     1  0.0849,0.0690,0.0450,0.0360,0.0510,0.0270,0.0240,0.0210,
     2  0.0375,0.0675,0.1160,0.0960,0.0690,0.0546,0.0465/

      data days/31.,28.,31.,30.,31.,30.,31.,31.,30.,31.,30.,31./

      taualf(theta) = 1.-b0*(1./amax1(0.5,cos(theta*rdconv))-1.)
     +              - (1.-b0)*(amax1(60.,theta)-60.)/30.

      info(6) = 10
      info(9) = 1

c     ****input****

      I = xin(1)
      Id = xin(2)
      It = xin(3)
      theta = xin(4)
      rho = xin(5)
      beta = xin(6)
      Tenv = xin(7)
      Tset = xin(8)
      Tmains = xin(9)
```

```
      Tamb = xin(10)
      D2 = xin(11)
      comp = (D2 .lt. 0.)
      if (.not. comp) then
            D3 = xin(12)
            Rload = (D3 .lt. 0.)
            if (.not. Rload) mfload = xin(13)
      else
            month = int(xin(14)+.1)
            D3 = xin(15)
            Rload = (D3 .lt. 0.)
            if (.not. Rload) mfload = xin(16)
      end if
c     ********read in once********

      if (info(8) .eq. 1) then

            mfcoll = par(2)
            cpcoll = par(3)
            mftnk = par(4)
            cptnk = par(5)
            rhot = par(6)

            Frtan = par(8)
            FrUl = par(9)
            gtest = par(11)
            area = par(12)
            nser = par(13)

            volsol = par(15)
            htsol = par(16)
            Usol = par(17)

            volaux = par(19)
            htaux = par(20)
            Uaux = par(21)

            hxeff = par(22)
            lin = par(27)
            lout = par(28)
            dpipe = par(29)
            Upipe = par(30)
            nx = 12
            f (comp) nx = nx + 3
            if (.not. Rload) nx = nx + 1
            info(10) = 101
            call typeck(1, info, nx, 31, 0)
            if (comp) then
                  s(info(10)) = xin(12)*xin(13)*24.
                  s(info(10)+1) = xin(12)*(xin(13)-1.)*24.
```

```
                clock = -par(1)
            end if
        end if

        delt = par(1)
        Tboil = par(7)
        b0 = par(10)
        nnodes = nint(par(14)+.1)
        ntanks = nint(par(18)+.1)
        daily = par(23)
        udeadb = par(24)
        ldeadb = par(25)
        D1 = par(26)
        mix = (d1 .lt. 0.)

        if (Info(8) .eq. 1) then

            Tfaux = Tset
            do 9 j = 1, nnodes
                out(10+j) = par(31)
9           continue
            if (ntanks .eq. 1) out(11) = Tset

            beta2 = beta*beta
            effsky = 59.98 - 0.1388*beta + 0.001497*beta2
            effgnd = 90. - 0.5788*beta + 0.002693*beta2
            Ktads = taualf(effsky)
            Ktadg = taualf(effgnd)
            beta = beta*rdconv
            cbeta = cos(beta)
            Fsky = (1.+cbeta)/2.
            Fgnd = (1.-cbeta)/2.

            if (hxeff .lt. 0.) then
                mftnk = mfcoll
                cptnk = cpcoll
                hxeff = 1.
            end if

            diamsol = sqrt(4.*volsol/(htsol*pi))
            pidiam = pi*diamsol
            pid2 = pidiam*diamsol
            masssol = pid2*htsol*rhot/(4.*float(nnodes))
            capsol = masssol*cptnk
            if (nnodes .eq. 1) then
                UAsol = pidiam*htsol*Usol + Usol*pid2/2.
            else
                UAsol = pidiam*htsol*Usol/float(nnodes)
                UAtop = UAsol + Usol*pid2/4.
            end if
```

```
      if (ntanks .eq. 2) then
            diamaux = sqrt(4.*volaux/(htaux*pi))
            pidiam = pi*diamaux
            pid2 = pidiam*diamaux
            massaux = pid2*htaux*rhot/(4.)
            capaux = massaux*cptnk
            UAaux = pidiam*htaux*Uaux + Uaux*pid2/2.
      end if

      capcoll = mfcoll*cpcoll
      gcap = gtest*cpcoll
      captnk = mftnk*cptnk
      FpUl = -gcap*alog(1.-FrUl/gcap)

      FpUla = FpUl*area

      R1 = capcoll/FpUla*(1.-exp(-FpUla/capcoll))/
     +     (gcap/FpUl*(1.-exp(-FpUl/gcap)))

      FrUl = FrUl*R1
      Frtan = Frtan*R1

      xk = FrUl*area/capcoll
      R2 = (1.-(1.-xk)**nser)/(nser*xk)
      Frtan = Frtan*R2
      FrUl = FrUl*R2

c     ****to calculate temp at collector exit for control purposes****

      s(info(10)+81) = area*Frtan
      s(info(10)+82) = area*FrUl

      lin = lin*pi*dpipe
      UApin = lin*Upipe
      s(info(10)+100) = UApin
      lout = lout*pi*dpipe
      UApout = lout*Upipe
      s(info(10)+101) = UApout

      R3 = 1./(1. + UApout/capcoll)
      Frtan = Frtan*R3
      R3 = (1. - Upipe*lin/capcoll + Upipe*(lin+lout)/
     +     (area*FrUl)) / (1. + UApout/capcoll)
      FrUl = FrUl*R3

      R4 = 1. / (1. + (FrUl*area/capcoll) * (capcoll/(hxeff*
     +     min(capcoll,captnk)) -1.) )
      FrUl = FrUl*R4
      Frtan = Frtan*R4
      FrUl = FrUl * area
      Frtan = Frtan * area
```

```
            s(info(10)+3) = Frtan
            s(info(10)+4) = FrUl
            s(info(10)+5) = mftnk
            s(info(10)+6) = cptnk
            s(info(10)+7) = capsol
            s(info(10)+8) = UAsol
            s(info(10)+9) = UAtop
            s(info(10)+10) = capaux
            s(info(10)+11) = UAaux
            s(info(10)+12) = Fsky
            s(info(10)+13) = Fgnd
            s(info(10)+14) = ktads
            s(info(10)+15) = ktadg

c           ****to calculate collector exit temp for control purposes****

            col2 = capcoll*capcoll
            col3 = col2*capcoll
            capmin = min(capcoll, captnk)
            s(info(10)+83) = UApin*UApout*hxeff*capcoll*capmin
            R1 = hxeff*col2*capmin
            s(info(10)+84) = 2.*UApin*R1
            s(info(10)+85) = 2.*UApout*R1
            s(info(10)+86) = 4.*hxeff*col3*capmin
            s(info(10)+87) = s(info(10)+86)*(UApin+UApout)
            s(info(10)+88) = UApin*UApout*capcoll*captnk
            R1 = 2.*col2*captnk
            s(info(10)+89) = R1*UApin
            s(info(10)+90) = R1*UApout
            s(info(10)+91) = 4.*col3*captnk
            s(info(10)+92) = s(info(10)+91)*(-UApin - UApout)
            s(info(10)+93) = UApin*UApout*hxeff*capmin*captnk
            s(info(10)+94) = s(info(10)+93)*capcoll
            s(info(10)+95) = 2.*hxeff*capcoll*capmin*captnk
            s(info(10)+96) = s(info(10)+95)*capcoll
            s(info(10)+97) = s(info(10)+96)*UApout
            s(info(10)+98) = 4.*hxeff*col3*capmin*captnk

            k = info(10)
            R1 = -s(k+82)*s(k+83) + s(k+82)*s(k+87)/(2.*capcoll) +
     +      s(k+86)*(-s(k+82)-UApin-UApout) + s(k+82)*s(k+88)
            R2 = s(k+82)*s(k+92)/(2.*capcoll) + s(k+91)*(s(k+82)+
     +      UApin+UApout) - s(k+82)*s(k+93)
            R3 = s(k+95)/2.*(UApin*UApout + 2.*s(k+82)*(UApin+UApout)) +
     +      s(k+96)*(-2.*s(k+82) - UApin - UApout)
            R4 = s(k+98)
            s(k+99) = R1 + R2 + R3 + R4

        else
            iter = s(info(10))
```

```
            begin = s(info(10)+1)
            clock = s(info(10)+2)
            Frtan = s(info(10)+3)
            FrUl = s(info(10)+4)
            mftnk = s(info(10)+5)
            cptnk = s(info(10)+6)
            capsol = s(info(10)+7)
            UAsol = s(info(10)+8)
            UAtop = s(info(10)+9)
            capaux = s(info(10)+10)
            UAaux = s(info(10)+11)
            Fsky = s(info(10)+12)
            Fgnd = s(info(10)+13)
            ktads = s(info(10)+14)
            ktadg = s(info(10)+15)
            gamma = s(info(10)+20)
            TC = s(info(10)+21)
            capl = s(info(10)+22)
            capload = s(info(10)+23)
            tbulk = s(info(10)+24)
            qb = s(info(10)+25)
            captnk = s(info(10)+26)
            do 10 j = 1, nnodes
                  Tfsol(j) = s(info(10)+27+j)
                  Tsolavg(j) = s(info(10)+47+j)
                  if (info(7) .eq. 0) then
                        out(10+j) = Tfsol(j)
                  end if
10          continue
            TFaux = s(info(10)+68)
            Tauxavg = s(info(10)+69)
      end if

c     ****set clock****
      if (comp) then
            if (xin(12) .ge. 28.) then
                  begin = 0.
                  iter = days(month)*24.
            end if
            if ((info(7) .eq. 0) .and. (info(8) .ne. 1)) then
                  clock = clock + delt
                  if (clock .gt. begin) then
                        k = info(10)
                        s(k+70) = s(k+70) + tbulk
                        s(k+71) = s(k+71) + I
                        s(k+72) = s(k+72) + It
                        s(k+16) = s(k+16) + It
                        s(k+73) = s(k+73) + Tenv
                        s(k+74) = s(k+74) + captnk*gamma*(TC - tsolavg(nnodes))
                        s(k+75) = s(k+75) + capload*(tsolavg(1) - Tmains)
                        if (nnodes .eq. 1) then
```

```
                        s(k+76) = s(k+76) + UAsol*(Tsolavg(1) - Tamb)
                    else
                        do 12 j = 1, nnodes
                            if (j .eq. 1) then
                              s(k+76) = s(k+76) + UAtop*(Tsolavg(j) - Tamb)
                            else if (j .eq. nnodes) then
                              s(k+76) = s(k+76) + UAtop*(Tsolavg(j) - Tamb)
                            else
                              s(k+76) = s(k+76) + UAsol*(Tsolavg(j) - Tamb)
                            end if
12                      continue
                    end if
                    s(k+77) = s(k+77) + qb
                    s(k+17) = s(k+17) + qb
                    s(k+78) = s(k+78) + UAaux*(tauxavg - Tamb)
                    s(k+18) = s(k+18) + UAaux*(tauxavg - Tamb)
                    s(k+79) = s(k+79) + capl*(Tset - Tmains)
                    s(k+19) = s(k+19) + capl*(Tset - Tmains)
                    if (clock .ge. iter) then
                        if (xin(12) .lt. 28) then
                            d4 = xin(12)/(delt*days(month))
                        else
                            d4 = 1./delt
                        end if
                        write(6,*)month
                        do 13 j = 70, 73, 1
                            if ((j .eq. 70) .or. (j .eq. 73)) then
                              s(info(10)+j)=s(info(10)+j)/(24.*days(month))
                            else
                              s(info(10)+j) = s(info(10)+j)/days(month)
                            end if
13                      continue
                        do 14 j = 70, 79, 1
                            s(info(10)+j) = s(info(10)+j)/d4
                            write(6,*)s(info(10)+j)
14                          continue
                            write(6,*)'**********************************'
                            clock = 0.
                            do 15 j = 70, 79, 1
                        s(info(10)+j) = 0.
15                      continue
                        if (month .eq. 12) then
                            j = info(10)
                            d4 = 1. - s(j+17)/(s(j+19)+s(j+18))
                            write(6,*)d4
                        end if
                    end if
                end if
            end if
        end if
```

```
c       ****use relations of Brandemuehl for effective diffuse rad incidence angles****

        if (It .gt. 0.) then
              Idsky = Fsky*Id
              Idgnd = rho*Fgnd*I
              Ktab = taualf(theta)
              Kta = (Ktab*(It-Idsky-Idgnd) + ktads*Idsky +
     +        ktadg*Idgnd)/It
        end if

c       ****calculate water load based upon time of day****

        if (ntanks .eq. 1) then
              tbar = Tsolavg(1)
        else
              tbar = Tauxavg
        end if

        if (Rload) mfload = daily*randwl(int(mod(time,24.))+1)
        capl = mfload*cptnk
        If ((mix) .and. (Tbar .gt. Tset)) then
              capload = capl*(Tset-Tmains)/(Tbar-Tmains)
        else
              capload = capl
        end if

        do 20 j = 1, nnodes
              tisol(j) = out(10+j)
 20     continue

        stick = 0
        niterate = 0

 30     continue

        niterate = niterate + 1
        tnnodes = tsolavg(nnodes)

c       ****set collector outlet temperature for pump control****

        if (gamma .eq. 1) then
              TC = (It*Frtan*Kta - FrUl*(Tsolavg(nnodes)
     +        -tenv))/captnk + Tsolavg(nnodes)
              k = info(10)
              R1 = s(k+81)*It*kta
              R2 = s(k+82)*tenv
              R7 = s(k+82)*TC
              R3 = (R1+R2)*(s(k+83)+s(k+86)) + (R1-R2)*(s(k+84)+s(k+85)) +
     +        Tenv*s(k+87) + (-R1-R2)*(s(k+88)+s(k+91))
              R4 = (-R1+R2)*(s(k+89)+s(k+90)) + s(k+92)*tenv +
     +        (-R7 + 2.*R2)*s(k+93) + (TC-2.*Tenv)*s(k+94)
```

```
            R5 = (-2.*R2*s(k+101) + R7*(-s(k+100)+s(k+101)))*s(k+95)
     +      + (2.*R7 + TC*s(k+100))*s(k+96)
            R6 = (-TC + 2.*tenv)*s(k+97) - TC*s(k+98)
            TCout = -(R3+R4+R5+R6)/s(k+99)
         else
            TCout = par(8)*Kta*It/par(9) + tenv
         end if

c        ****set pump control function****

         if (TCout - Tsolavg(nnodes) .ge. udeadb) then
c           ****definatly on****
            if (gamma .eq. 0) then
               stick = stick + 1
               if (stick .lt. mstick) then
                  TC = (It*Frtan*Kta - FrUl*(Tsolavg(nnodes)
     +               -tenv))/captnk + Tsolavg(nnodes)
                  if (TC .gt. Tboil) TC = Tboil
                  gamma = 1
               end if
            end if
         else if (TCout - Tsolavg(nnodes) .gt. ldeadb) then
c           ****was on so stay on****
            if (gamma .eq. 1) then
               if (TC .gt. Tboil) TC = Tboil
            end if
c           ****was off so stay off - requires no additional work****
         else
c           ****definatly off****
            if (gamma .eq. 1) stick = stick + 1
            if (stick .lt. mstick) gamma = 0
         end if

c        ****set up and solve tank equations with the following assumptions****
c           1. always fixed inlet and outlet positions (at the top and bottom)
c           2. auxiliary, if present, in top node
c           3. auxiliary always sufficient to make up difference

            if (nnodes .eq. 1) then
               start = 1
               stop = 1
               step = 1
               aa(1) = (-captnk*gamma - capload - UAsol)/capsol
               bb(1) = (captnk*gamma*TC + capload*Tmains + UAsol*Tamb)/capsol

c              ****solve analytically****

               if (abs(aa(1)) .gt. 0.) then
               cc(1) = bb(1)/aa(1)
               dd(1) = aa(1)*delt
               ee(1) = aa(1)*capsol
```

```
                Tsolavg(1) = (TIsol(1) + cc(1))/dd(1)*(exp(dd(1)) - 1.) - cc(1)
            else
                tsolavg(1) = .5*(bb(1)*delt + 2.*TIsol(1))
            end if

        else

            capnet = abs(captnk*gamma - capload)

            if (mftnk*gamma .gt. mfload) then
                D3 = -1
                D4 = 1.
                D5 = 0.
                start = 1
                stop = nnodes
                step = 1
            else
                D3 = 1
                D4 = 0.
                D5 = 1.
                start = nnodes
                stop = 1
                step = -1
            end if

            do 100 j = start, stop, step

                if (j .eq. 1) then
                    aa(j) = (-capload - UAtop - capnet*D4)/capsol
                    if (ntanks .eq. 2) then
                        bb(j) = (captnk*gamma*TC + UAtop*Tamb +
     +                  capnet*D5*Tsolavg(2))/capsol
                    else
                        bb(j) = (UAtop*Tamb + capnet*D5*Tsolavg(2))/capsol
                    end if
                else if (j .eq. nnodes) then
                    aa(j) = (-captnk*gamma - UAtop - capnet*D5)/capsol
                    bb(j) = (capload*Tmains + UAtop*Tamb + capnet*D4
     +              *tsolavg(nnodes-1))/capsol
                else
                    aa(j) = (-UAsol - capnet)/capsol
                    if ((ntanks .eq. 2) .or. (j .ne. 2)) then
                        bb(j) = (UAsol*Tamb + capnet*Tsolavg(j+D3))/capsol
                    else
                        bb(j) = (captnk*gamma*TC + UAsol*Tamb +
     +                  capnet*Tsolavg(j+D3))/capsol
                    end if
                end if

c               ****solve analytically****
```

```
                if (abs(aa(j)) .gt. 0.) then
                    cc(j) = bb(j)/aa(j)
                    dd(j) = aa(j)*delt
                    ee(j) = aa(j)*capsol
                    Tsolavg(j) = (TIsol(j) + cc(j))/dd(j)*(exp(dd(j)) -
     +              1.) - cc(j)
                else
                    solavg(j) = .5*(bb(j)*delt + 2.*TIsol(j))
                end if
100         continue

        end if

c       ****correct for temperature inversions****

101     continue
        flag = 0
        do 103 k = 1, nnodes-1
            if (Tsolavg(k) .lt. Tsolavg(k+1)) then
                Tsolavg(k) = (Tsolavg(k)+Tsolavg(k+1))/2.
                Tsolavg(k+1) = Tsolavg(k)
                flag = 1
            end if
103     continue
        if (flag .eq. 1) goto 101

        if (((abs(tnnodes - tsolavg(nnodes)) .gt. eps)) .and.
     +  (niterate .lt. miterate)) goto 30

        tbulk = 0.
        do 105 j = 1, nnodes
            tbulk = tbulk + tsolavg(j)
105     continue
        tbulk = tbulk/float(nnodes)

c       ****calculate final temperatures****

        do 110 j = start, stop, step
            if (abs(aa(j)) .gt. 0.) then
                TFsol(j) = (Tisol(j) + cc(j))*exp(dd(j)) - cc(j)
            else
                TFsol(j) = bb(j)*delt + Tisol(j)
            end if
110     continue

c       ****corect for temperature inversions****

120     continue
        flag = 0
        do 123 k = 1, nnodes-1
            if (TFsol(k) .lt. TFsol(k+1)) then
```

```
                TFsol(k) = (TFsol(k)+TFsol(k+1))/2.
                TFsol(k+1) = TFsol(k)
                flag = 1
                end if
123     continue
        if (flag .eq. 1) goto 120

        if (ntanks .eq. 1) then
              j = 1
              U = (exp(dd(j))-1.)/ee(j)
              V = ((exp(dd(j))-1.)/dd(j)-1.)/ee(j)
              qb = amax1(0., (tset - tfsol(j)))/u)
              TFsol(j) = U*qb + TFsol(j)
              Tsolavg(j) = V*qb + Tsolavg(j)
        end if

c       ****auxiliary tank with the following assumptions****
c             1. tank fully mixed
c             2. auxiliary always sufficient to make up difference

        if (ntanks .eq. 2) then
              aa(1) = (-UAaux - capload)/capaux
              bb(1) = (capload*Tsolavg(1) + UAaux*Tamb)/capaux
              if (abs(aa(1)) .gt. 0.) then
                    cc(1) = bb(1)/aa(1)
                    dd(1) = aa(1)*delt
                    ee(1) = aa(1)*capaux
                    TIaux = TFaux
                    TFaux = (TIaux + cc(1))*exp(dd(1)) - cc(1)
                    Tauxavg = (TIaux + cc(1))/dd(1)*(exp(dd(1)) - 1.) - cc(1)
              else
                    TIaux = TFaux
                    TFaux = bb(1)*delt + TIaux
                    Tauxavg = .5*(TFaux + TIaux)
              end if

              if (abs(aa(1)) .gt. 0.) then
                    U = (exp(dd(1))-1.)/ee(1)
                    V = ((exp(dd(1))-1.)/dd(1)-1.)/ee(1)
              else
                    U = delt/capaux
                    V = U/2.
              end if
              QB = AMAX1(0., (TSET - TFAUX)/U)

              TFaux = U*qb + TFaux
              Tauxavg = V*qb + Tauxavg

        end if

        if (.not. comp) then
```

```
            out(1) = tbulk
            out(2) = I
            out(3) = It
            out(4) = Tenv
            out(5) = captnk*gamma*(TC - tsolavg(nnodes))
            out(6) = capload*(tsolavg(1) - Tmains)
            if (nnodes .eq. 1) then
                  out(7) = UAsol*(Tsolavg(1) - Tamb)
            else
                  out(7) = 0.
                  do 400 j = 1, nnodes
                        if (j .eq. 1) then
                              out(7) = out(7) + UAtop*(Tsolavg(j) - Tamb)
                        else if (j .eq. nnodes) then
                              out(7) = out(7) + UAtop*(Tsolavg(j) - Tamb)
                        else
                              out(7) = out(7) + UAsol*(Tsolavg(j) - Tamb)
                        end if
400               continue
            end if
            out(8) = qb
            out(9) = UAaux*(tauxavg - Tamb)
            out(10) = capl*(Tset - Tmains)
      end if
      if (info(8) .eq. 1) then
            s(info(10)+3) = Frtan
            s(info(10)+4) = FrUl
            s(info(10)+5) = mftnk
            s(info(10)+6) = cptnk
            s(info(10)+7) = capsol
            s(info(10)+8) = UAsol
            s(info(10)+9) = UAtop
            s(info(10)+10) = capaux
            s(info(10)+11) = UAaux
            s(info(10)+12) = Fsky
            s(info(10)+13) = Fgnd
            s(info(10)+14) = ktads
            s(info(10)+15) = ktadg
            s(info(10)+26) = captnk
      end if
      s(info(10)+2) = clock
      s(info(10)+20) = gamma
      s(info(10)+21) = TC
      s(info(10)+22) = capl
      s(info(10)+23) = capload
      s(info(10)+24) = tbulk
      s(info(10)+25) = qb
      do 500 j = 1, nnodes
            s(info(10)+27+j) = Tfsol(j)
            s(info(10)+47+j) = Tsolavg(j)
500   continue
```

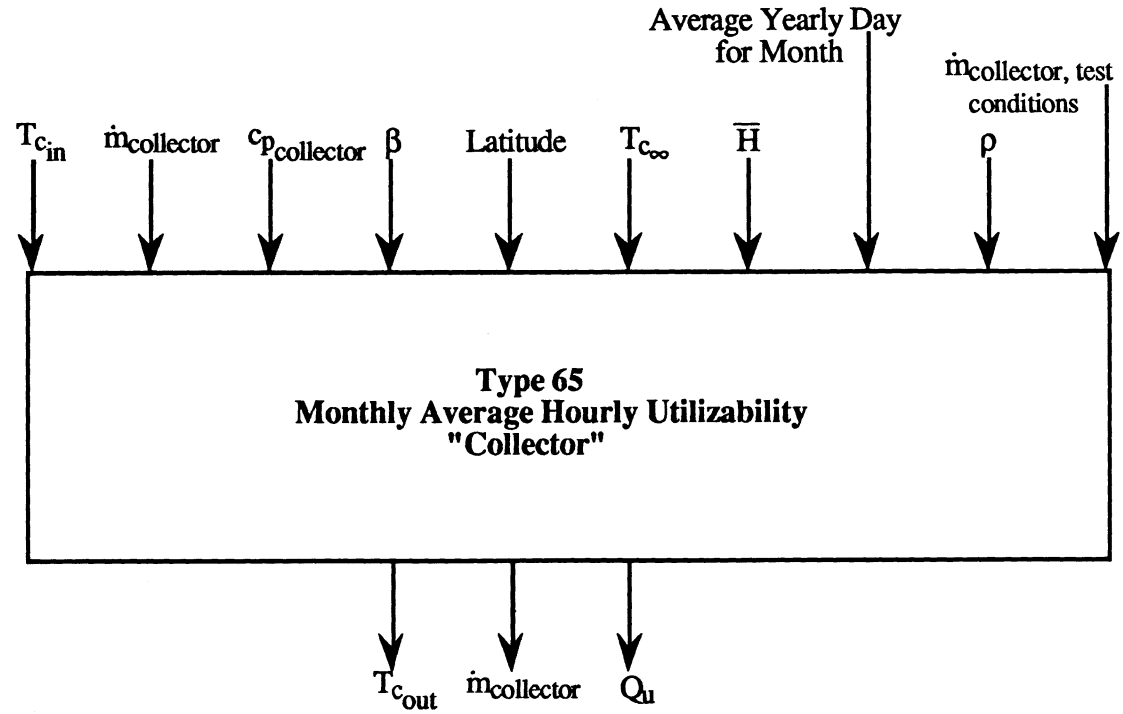
```
s(info(10)+68) = Tfaux
s(info(10)+69) = Tauxavg

return
end
```

A.5 TYPE 65: MONTHLY AVERAGE HOURLY UTILIZABILITY "COLLECTOR"

Type 65 calculates the monthly average hourly useful energy gain across a flat plate collector via monthly average hourly utilizability methods. The performance of a system may be estimated by substituting Type 65 into a TRNSYS solar system model in place of the following components:

1. Data reader (Type 9) or weather generator (Type 54)
2. Radiation processor (Type 16)
3. Flat plate collector (Type 1)

The diffuse radiation component is calculated via the Erbs daily diffuse correlations [1980]. The tilted surface radiation is subsequently calculated assuming isotropic sky conditions (e.g., Lui and Jordan relation) [Duffie and Beckman, 1980]. Finally, the utilizability for an hour is calculated using the Clark et al. correlations [1983]. Figure A.3 illustrates the Type 65 information flow.



Parameters

1. $F_R(\tau\alpha)_n$
2. F_rU_L
3. b_o
4. A
5. $\Delta\theta$

Figure A.3 Type 65 Information Flow Diagram

```
subroutine type65 (time,xin,out,t,dtdt,par,info)
implicit none

dimension xin(10), out(16), par(5), info(10)

real time, xin, out, t, dtdt, par
real Gsc, pi
real tin, mf, cp, beta, phi, ta, rho
real Frtan, FrUl, bo, area, step, rads
real delta, wset, tset, daylen, trise, wrise, w
real Ho, Hbar, Hdbar, ktbar, kt, Io, Ibar, Idbar, Itbar, Ic
real Rb, Rtilda, theta
real ntime, a, b, rt, rd, w2, w1, y2, y1
real xmax, ap, xc
real Frtaualf, utiliz, Qu, tout, gtest, Nser
```

```
      real taualf, effsky, effgnd, cosslp, fsky, fgnd, Idsky, Idgnd
      real katb, katds, katdg, kat
      real FpUl, rtest, r1, r2, r3, ratio, xk

      integer info, iday

      parameter (Gsc=1.367)

      data pi/3.1416/

      taualf(theta)=1.-bo*(1./amax1(0.5,COS(theta*rads))-1.)
     .                  - (1.-bo)*(amax1(60.,theta)-60.)/30.

      Nser = 1.

      tin = xin(1)
      mf = xin(2)
      cp = xin(3)
      beta = xin(4)
      phi = xin(5)
      ta = xin(6)
      hbar = xin(7)
      iday = int(xin(8) + .1)
      rho = xin(9)
      gtest = xin(10)

      Frtan = par(1)
      FrUl = par(2)
      bo = par(3)
      area =  par(4)
      step = par(5)

      rads = pi/180.
      phi = phi * rads
      beta = beta * rads

c     ****calculations done once****

      if (info(8) .eq. 1) then
            info(6) = 16
            info(9) = 1
            call typeck(-1, info, 10, 5, 0)

            delta = 23.45 * sin(360.*(284.+float(iday))/365.*rads)
            delta = delta * rads
            wset = acos(-tan(phi)*tan(delta))
            tset = wset/(15.*rads) + 12.
            daylen = 2./15. * wset/rads
            trise = tset - daylen
            wrise = (trise - 12.) * 15. * rads
            Ho = 24.*3600.*Gsc/pi * (1.+.033*cos(360.*float(iday)/365.*rads))
```

```
            Ho = Ho * (cos(phi)*cos(delta)*sin(wset) + wset*sin(phi)*sin(delta))
            ktbar = hbar/Ho

c           ****use Erbs correlations where applicable****

            if ((ktbar .ge. .3) .and. (ktbar .le. .8)) then
                  if (wset .le. 1.4207) then
                        hdbar = hbar*(1.391 - 3.56*ktbar + 4.189*ktbar
     +                        *ktbar - 2.137*ktbar**3.)
                  else
                        hdbar = hbar*(1.311 - 3.022*ktbar + 3.427*ktbar
     +                  *ktbar - 1.821*ktbar**3.)
                  end if
            else
                  hdbar = hbar*(.775 + .00606*(wset/rads - 90.) - (.505 +
     +                  .00405*(wset/rads - 90.))*cos((115.*ktbar - 103.)*rads))
            end if
      else
            wrise = out(8)
            wset = out(9)
            Hdbar = out(10)
            Hbar = out(11)
      end if

c     ****calculations done each time step****

 10   continue

      ntime = mod(time,24.) + step/2.

      w = (ntime - 12.) * 15. * rads
      if ((w .ge. wrise) .and. (w .le. wset)) then
            a = .409 + .5016*sin(wset-(60*rads))
            b = .6609 - .4767*sin(wset-(60*rads))
            d = pi/24.*(cos(w)-cos(wset))/(sin(wset)-wset*cos(wset))
            rt = rd*(a + b*cos(w))

c           ****find rtilda****

            w2 = ((ntime + .5) - 12.) *15.*rads
            y2 = sin(w2)
            w1 = ((ntime - .5) - 12.) *15.*rads
            y1 = sin(w1)
            Io = 12.*3600.*Gsc/pi * (1. + .033*cos(360.*float(iday)/365.*rads))
            Io = Io*(cos(phi)*cos(delta)*(y2 - y1) + (w2 - w1)*sin(phi)*sin(delta))
            Ibar = rt*Hbar
            kt = Ibar/Io
            Idbar = rd*Hdbar

c           ****isotropic diffuse assumption and gamma = 0****
```

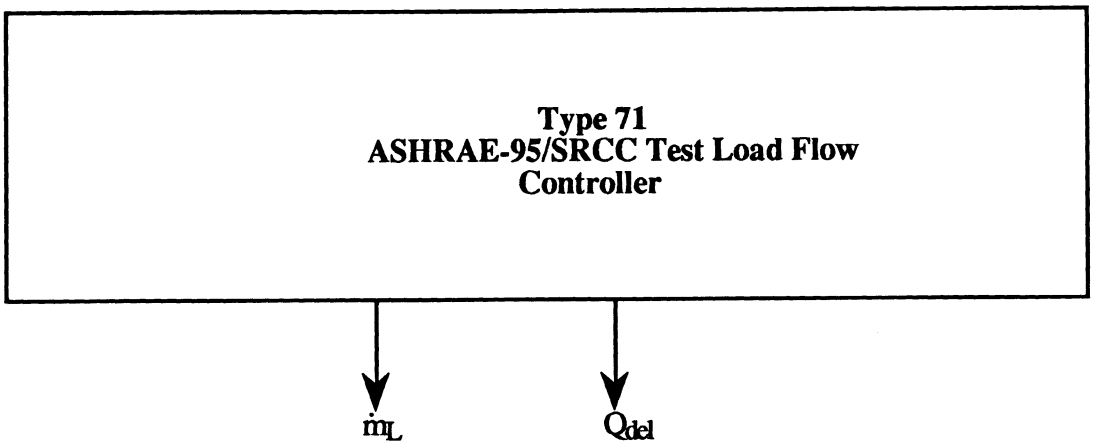
```
      Rb = (cos(phi-beta)*cos(delta)*cos(w) + sin(phi-beta)*sin
     +     (delta)) / (cos(phi)*cos(delta)*cos(w) + sin(phi)*sin(delta))

      Itbar = hbar*( (rt - hdbar/hbar*rd)*Rb + hdbar/hbar*rd*.5*
     +     (1. + cos(beta)) + rho*rt*.5*(1. - cos(beta)) )
      if (Itbar .lt. 0.) Itbar = 0.

      Rtilda = (1. - Idbar/Ibar)*Rb + .5*(Idbar/Ibar)*
     +     (1. + cos(beta)) + .5*rho*(1. - cos(beta))

c     ****Clark's correlations for utilizability****

      xmax = 1.85 + .169*rtilda/(kt*kt) - .0696*cos(beta)/
     +     (kt*kt) - .981*kt/(cos(delta)*cos(delta))
      if (xmax .lt. 0) goto 500

c     ****find theta****

      theta = sin(delta)*sin(phi)*cos(beta) - sin(delta)*cos(phi)
     +     *sin(beta) + cos(delta)*cos(phi)*cos(beta)*cos(w)
      theta = acos(theta + cos(delta)*sin(phi)*sin(beta)*cos(w))

c     ****modify for incidence angle****

c     flat plates given bo only

c     ****use relations of Brandemuehl for effective incidence angles****
c     ****for diffuse****

      if ((info(7) .le. 0) .and. (Itbar .gt. 0)) then
            beta = beta/rads
            effsky = 59.68-0.1388*beta+0.001497*beta*beta
            effgnd = 90.-0.5788*beta+0.002693*beta*beta
            cosslp = cos(beta*rads)
            fsky = (1. + cosslp)/2.
            fgnd = (1. - cosslp)/2.
            IdSKY = fsky*Idbar
            IdGND = rho*fgnd*Ibar

c           ****use constant from ASHRAE test****

            katb = taualf(theta)
            katds = taualf(effsky)
            katdg = taualf(effgnd)
            kat = (katb*(Itbar-IdSKY-IdGND)+katds*IdSKY+katdg
     +           *IdGND)/Itbar
            out(15) = kat
      else
            kat = out(15)
      end if
```

```
            Frtaualf = Frtan * kat

c           ****find critical radiation level****

            if (tin .lt. ta) then
                  Ic = 0
            else
                  Ic = FrUl*(tin - ta)/Frtaualf
            end if

            if (Itbar .le. 0.) goto 500

            xc = Ic/Itbar

            if (xc .ge. xmax) then
                  utiliz = 0
            else if (xmax .eq. 2.) then
                  utiliz = (1. - xc/xmax)*(1. - xc/xmax)
            else
                  ap = (xmax - 1.)/(2. - xmax)
                  utiliz = abs(abs(ap) - sqrt(ap*ap + (1+2.*ap)*
     +            (1.-xc/xmax) * (1. - xc/xmax)))
            end if

      else
500         continue
            Ibar = 0.
            Itbar = 0.
            utiliz = 0.
            theta = 90.
            xc = 0.
      end if

      if (mf .gt. 0) then
            FpUl=-gtest*cp*alog(1.-FrUl/gtest/cp)
            rtest=gtest*cp*(1.-EXP(-FpUl/gtest/cp))
            R1=NSer*mf*cp/area*(1.-exp(-FpUl*area/NSer/mf/cp))/rtest
            xk=R1*area*FrUl/mf/cp/NSer
            R2=(1.-(1. - xk)**NSer)/NSer/xk
            R3=1.
            RATIO=R1*R2*R3
      else
            ratio = 0
      end if

      Qu = ratio*area*Frtaualf*utiliz*Itbar

      if ((Qu .lt. 0) .or. (mf .eq. 0)) Qu = 0
      if ((Qu .gt. 0) .and. (mf .gt. 0)) then
            tout = Qu/(mf*cp) + tin
      else
```

```
        if ((mf .gt. 0) .and. (tin .ne. ta)) then
            if (tin .gt. ta) then
                tout = tin - FrUl*area/(mf*cp)*(tin - ta)
            else
                tout = ta
            end if
        else
            tout = Frtaualf*Itbar/(FrUl) + ta
        end if
    end if
    tout = max(tout,ta)

    out(1) = tout
    out(2) = mf
    out(3) = Qu
    out(4) = ta
    out(5) = Ic
    out(6) = utiliz
    out(7) = Itbar
    out(8) = wrise
    out(9) = wset
    out(10) = Hdbar
    out(11) = Hbar
    out(12) = Ibar
    out(13) = theta
    out(14) = Idbar
    out(16) = xc

    return
    end
```

A.6 TYPE 71: ASHRAE-95/SRCC TEST LOAD FLOW CONTROLLER

Type 71 models the ASHRAE-95/SRCC SDHW short-term test specified load flow profile [Wood, 1989]. Type 71 turns "on" the flow at the time step closest to the specified time (e.g., 8:00 A.M., 12:00 noon, or 5:00 P.M.). Type 71 subsequently turns "off" the flow at the time step in which the delivered energy is closest to the specified value. Figure A.4 illustrates the Type 71 information flow.



Parameters

1. Target Energy Value (e. g. 14100 kJ)
2. Flow During Draw (e. g. 720 kg/hour)
3. $c_p\left(\text{e. g. } 4.19 \frac{kJ}{kg\,°C}\right)$
4. T_{set}
5. T_{mains} (e. g. 22 °C)
6. $\Delta\theta$

Figure A.4 Type 71 Information Flow Diagram

```
      subroutine type71 (time,xin,out,t,dtdt,par,info)
      dimension out(2), par(6), info(10)
      double precision real n, tset, tmains, limit, duration, tm, ton,
     +  step, qdel
      logical on, chkon, chkoff
c
c     set initial conditions
      if (info(7) .eq. -1) then
            call typeck(-1,info,0,6,0)
            info(6)=3
            info(9)=1
            limit = par(1)
            mdot = par(2)
            cp = par(3)
            tset = par(4)
            tmains = par(5)
            step = par(6)
            duration=limit/(mdot*cp*(tset-tmains))
            n=8.
            on=.false.
            chkon = .false.
            chkoff = .false.
      end if
```

```
c
 5      continue
        if (chkon) then
              on = .true.
              ton = n
              if (n .eq. 8.) then
                    n=12.
              else if (n .eq. 12.) then
                    n=17.
              else if (n .eq. 17.) then
                    n=8.
              end if
              chkon = .false.
              goto 10
        end if
        if (chkoff) then
              on = .false.
              chkoff = .false.
              goto 10
        end if
        tm = time-((int(time)/24)*24)
c
        if (.not.(on)) then
              if ((step+tm) .ge. n) then
                    if (abs(tm+step-n) .le. step) then
                          if (abs(tm-n) .lt. abs(tm+step-n)) then
                                on = .true.
                                ton = n
                                if (n .eq. 8.) then
                                      n=12.
                                else if (n .eq. 12.) then
                                      n=17.
                                else if (n .eq. 17.) then
                                      n=8.
                                end if
                          else
                                chkon = .true.
                          end if
                    end if
              end if
        else
              if ((step+tm) .ge. (duration+ton)) then
                    if (abs(tm+step-(duration+ton)) .le. step) then
                          if(abs(tm-(duration+ton)) .lt. abs(tm+step-(duration+ton))) then
                                on = .false.
                          else
                                chkoff = .true.
                          end if
                    end if
              end if
        end if
```

```
10      continue
        if (on) then
              out(1) = mdot
        else
              out(1) = 0
        end if
        qdel = out(1)*cp*(tset-tmains)
        out(2) = qdel
        return
        end
```

Appendix B

CSU Test #7 TRNSYS Deck

```
******************************
*  Drain-Back system         *
*  SRCC performance test     *
******************************
*
EQUATIONS 52
*
*****************************************************
***OPERATING CONDITIONS AND SET POINTS***
*****************************************************
*
***hot water set temperature [°C]***
TSET = 54.
*
***mains water temperature [°C]***
TMAINS = 22.
*
***tank environment temperature [°C]***
TENV = 22.
*
***collector ambient temperature [°C]***
TAMB = 22.
*
***ground reflectance***
RHO = 0.0
*
```

upper controller dead band [°C]
THDELT = 11.111
*
lower controller dead band [°C]
TLDELT = 2.7778
*
heat exchanger effectiveness
HXEFF = 0.44
*
****
COLLECTOR PARAMETERS*******
****
*
number of panels
PANELS = 3
*
number of panels in series
SERIES = 1
*
area per panel [m^2]
AC = 1.852
*
***collector slope [degrees]
SLOPE = 45.
*
collector gain coefficient
FRTAN = 0.602
*
collector loss coefficient [kJ/hr-m^2-°C]
FRUL = 20.016
*
***mass flow rate per unit area during collector test [Kg/hr-m^2]
GTEST = 63.773
*

```
**********************************
***FLUID PARAMETERS**********
**********************************
*
***collector-loop volume flow rate when pump is operating [kg/hr]***
CMDOT = 204.39
*
***tank-side volume flow rate when pump is operating [kg/hr]***
TMDOT = 340.65
*
***collector-loop density [kg/m3]***
CDENS = 1000.
*
***tank-side density [kg/m3]***
TDENS = 1000.
*
***approximate collector-loop specific heat [kJ/kg-°C]***
CPCOLL = 4.19
*
***tank-side specific heat [kJ/kg-°C]***
CPTANK = 4.19
*
***boiling temperature of tank fluid [°C]***
TTBOIL = 100.
*
***boiling temperature of collector-loop fluid [°C]***
TCBOIL = 100.
*
****************************************
***DRAIN-BACK TANK PARAMETERS***
****************************************
*
***tank volume [m3]***
DBVOL = 0.0273
*
```

-tank height [-m]

DBHGHT = -0.549

*

heat loss coefficient [kJ/hr-m^2-°C]

UDRN = 4.555

*

****

STORAGE TANK PARAMETERS

****

*

tank volume [m^3]

VSTANK = .223

*

heat loss coefficient [kJ/hr-m^2-°C]

UTS = 5.37

*

negative of tank height [-m]

HSTANK = -1.37

*

****

AUXILIARY TANK PARAMETERS

****

*

tank volume [m^3]

VATANK = 0.143

*

negative of tank height [-m]

HATANK = -1.15

*

tank loss coefficient [kJ/hr-m^2-°C]

UTA = 4.06

*

maximum rate of auxiliary heater [kJ/hr]

QMAX=10000000.

*NOTE: changed from given value of 12150 kJ/hr

*

```
***tank section containing heater***
NHEAT = 1
*
***tank section containing thermostat***
NTHERM = 1
*
***thermostat dead-band [°C]***
TDB = 0.
*
***UA of gas flue of auxiliary heater [kJ/hr-°C]***
UAF = 0.
*
***average flue temperature when not operating [°C]***
TFAVE = 30.
*
**************************************
***PIPE PARAMETERS**************
**************************************
*
***diameter of pipes [m]***
DIAM = 0.01905
*
***length of pipe leading to collector [m]***
LIN = 11.7665
*
***length of pipe leading from collector [m]***
LOUT = 13.405
*
***U for insulation [kJ/hr-m2-°C]***
UPIPE = 20.82
*
******************************************
***LOAD FUNCTION PARAMETERS******
******************************************
*
```

```
***flow rate during draw [kg/hr]***
DRAW = 728.4
*
***load limit per draw [kJ/draw]***
LIMIT = 16603.
*
*****************************************
***SIMULATION PARAMETERS*********
*****************************************
*
***number of days***
DAYS = 5
*
***time step [hr]***
STEP = 0.005
*
*****************************************************
***FIXED AND CALCULATED PARAMETERS******
*****************************************************
*
LENGTH = 24.*DAYS + 17.5
START = LENGTH - 25.
AREA = AC*PANELS
*****************************************
TEMP=.75*[1,1]+.25*[70,1]
QSDEL=CPTANK*[4,4]*([4,3]-TMAINS)
QDEL=CPTANK*[8,1]*(TSET-TMAINS)
*****************************************
```

SIMULATION 17. LENGTH STEP
LIMITS 50 30
TOLERANCES 0.000001 0.000001

UNIT 15 TYPE 14 IRRADIANCE PROFILE
PARS 44
0,0 8,0 8,1134 9,1134 9,1692 10,1692 10,2052 11,2052 11,2376
12,2376 12,2520 13,2520 13,2376 14,2376 14,2052 15,2052 15,1692
16,1692 16,1134 17,1134 17,0 24,0

UNIT 16 TYPE 14 INCIDENCE ANGLE PROFILE
PARS 44
0,90 8,90 8,60 9,60 9,45 10,45 10,30 11,30 11,15 12,15
12,0 13,0 13,15 14,15 14,30 15,30 15,45 16,45 16,60
17,60 17,90 24,90

UNIT 1 TYPE 1 DRAIN-BACK COLLECTOR
PARAMETERS 13
1 SERIES AREA CPCOLL 1 GTEST FRTAN FRUL -1 CPTANK 2 15 5
INPUTS 10
21,1 21,2 33,2 0,0 15,1 15,1 0,0 0,0 16,1 0,0
TMAINS 0.0 0.0 TAMB 0.0 0.0 0.0 RHO 90.0 SLOPE

UNIT 70 TYPE 1 COLLECTOR WITH NO FLOW THROUGH IT
PARAMETERS 13
1 SERIES AREA CPCOLL 1 GTEST FRTAN FRUL -1 CPTANK 2 15 5
INPUTS 10
21,1 0,0 0,0 0,0 15,1 15,1 0,0 0,0 16,1 0,0
TMAINS 0.0 0.0 TAMB 0,0 0,0 0.0 RHO 90.0 SLOPE

UNIT 21 TYPE 31 PIPE FROM DB TANK TO COLLECTOR
PARAMETERS 6
DIAM LIN UPIPE CDENS CPCOLL 44.
INPUTS 3
3,1 3,2 0,0
0.0 0.0 TENV

UNIT 22 TYPE 31 PIPE FROM COLLECTOR TO DB TANK
PARAMETERS 6
DIAM LOUT UPIPE CDENS CPCOLL 44.
INPUTS 3
1,1 1,2 0,0
0.0 0.0 TENV

UNIT 13 TYPE 13 COLLECTOR LOOP RELIEF VALVE
PARAMETERS 2
TCBOIL CPCOLL
INPUTS 3
22,1 22,2 22,1
TSET 0.0 TSET

UNIT 5 TYPE 5 COLLECTOR/TANK HEAT EXCHANGER
PARAMETERS 4
4 HXEFF CPCOLL CPTANK
INPUTS 4
13,1 13,2 33,1 33,2
TMAINS 0.0 TMAINS 0.0

```
UNIT 40 TYPE 4  DRAIN-BACK TANK
PARAMETERS 6
1 DBVOL CPCOLL CDENS UDRN DBHGHT
INPUTS 5
5,1        5,2      5,1   0,0      0,0
TMAINS 0.0 TMAINS 0.0   TENV
DERIVATIVES 1
44.

UNIT 3 TYPE 3  COLLECTOR LOOP PUMP
PARAMETERS 2
CMDOT 433.8
INPUTS 5
40,1   40,2  2,1      0,0      0,0
TSET    0.0  0.0   CPCOLL .85

UNIT 2 TYPE 2  GAIN CONTROLLER
PARAMETERS 3
3 THDELT TLDELT
INPUTS 3
TEMP          4,1     2,1
TMAINS    TMAINS 0.0

UNIT 12 TYPE 13  TANK RELIEF VALVE
PARAMETERS 2
TTBOIL CPTANK
INPUTS 3
5,3     5,4   5,3
TSET  0.0 TSET

UNIT 4 TYPE 4  SOLAR STORAGE TANK
PARAMETERS 6
1 VSTANK CPTANK TDENS UTS HSTANK
INPUTS 5
12,1    12,2     11,1   11,2   0,0
TSET    0.0  TMAINS 0.0  TENV
DERIVATIVES 3
44.  44.  44.

UNIT 33 TYPE 3  TANK SIDE PUMP
PARAMETERS 2
TMDOT 241.2
INPUTS 5
4,1     4,2 2,1       0,0      0,0
TSET 0.0 0.0   CPTANK .85
```

```
UNIT 44 TYPE 4  AUXILIARY STORAGE TANK
PARAMETERS 13
1 VATANK CPTANK TDENS UTA HATANK
QMAX NHEAT NTHERM TSET TDB UAF TFAVE
INPUTS 5
0,0        0,0    4,3     4,4     0,0
TMAINS  0.0  TMAINS 0.0   TENV
DERIVATIVES 1
TSET

UNIT 11 TYPE 11  TEMP CONTROLLED FLOW DIVERTER
PARAMETERS 2
4 3
INPUTS 4
0,0        8,1     4,3     0,0
TMAINS 0.0 TMAINS  TSET

UNIT 8 TYPE 71 SRCC LOAD PROFILE
PARAMETERS 6
LIMIT DRAW CPTANK TSET TMAINS STEP
INPUTS 0

UNIT 27 TYPE 28  DAILY SUMMARY
PARAMETERS 31
24 0 LENGTH  14  2  1
-11 -4
-12 -4
-13 -4
-14 -4
-15 -4
-16 -3  -17  -3  3  -18  -3  4  -1  -1  1  -19  -3  2  -4
INPUTS 9
4,7  44,5   4,5   1,3   44,8   3,3   33,3    QSDEL   8,2
LABELS 10
DELU ATLOSS STLOSS QU  QAUX COLPMP TNKPMP  QSDEL  QDEL SF
```

UNIT 29 TYPE 28 ENERGY SUMMARY
PARAMETERS 25
24 0 LENGTH 16 2
-11 -4
-12 -4
-13 -4
-14 -4
-15 -4
-16 -4
-17 -4
-18 -4
-19 -4
-20 -4
INPUTS 10
1,3 22,3 5,5 40,5 21,3 4,5 4,9 4,6 44,5 44,6
LABELS 10
QU QLOSS1 QHX QLDR QLOSS2 QLSOL QINSOL QOTSOL QLAUX QOUAUX

END

References

ASHRAE Standard 93-77, "Methods of Testing to Determine the Thermal Performance of Solar Collectors." American Society of Heating, Refrigeration, and Air Conditioning Engineers, New York, 1977.

Barker, G. B., "A Short-Term Testing Method for Active Solar Domestic Hot Water Systems." M.S. Thesis, University of Colorado, Boulder, Colorado, 1990.

Beckman, W. A., S. A. Klein, and J. A. Duffie, "A Design Procedure for Solar Heating Systems." *Solar Energy*, Volume 18, pp. 113-127, 1976

Bendt, P., M. Collares-Pereira, and A., Rabl, "The Frequency Distribution of Daily Insolation Values." *Solar Energy*, Volume 27, pp. 1-5, 1981.

Braun, J. E., S. A. Klein, and K. A. Pearson, "An Improved Design Method for Solar Water Heating Systems." *Solar Energy*, Volume 31, pp. 597-604, 1983.

Buckles, W. E., "Short Term Monitoring and Performance evaluation of Solar Domestic Hot Water Systems." M.S. Thesis, University of Wisconsin, Madison, Wisconsin, 1983.

Bourges, B. , A. Rabl, M. J. Carvalho, and M. Collares-Pereira, *Accuracy of the European Solar Water Heating Test Procedure Part 2: Prediction of Long Term Performance*, European Solar Collector and System Testing Group Contract 33.34-87-12 E D I S P, 1990

Carlson, W. T., "Comparison of Experimental and TRNSYS SRCC Ratings of a Generic Drain Back Solar Water System." M.S. Thesis, Colorado State University, Fort Collins, Colorado, 1991.

Chandrashekar, M., S. D. P. Dixon, and R. S. McKnight, *WATSUN User's Manual and Program Documentation Version 12.0/90-04*, WATSUN Simulation Laboratory, University of Waterloo, Ontario, Canada, 1990.

Cheney, W. and D. Kincaid, *Numerical Mathematics and Computing*, Brooks/Cole Publishing Company, Pacific Grove, California, 1985.

Clark, D. R., S. A. Klein, and W. A. Beckman, "Algorithm for Evaluating the Hourly Radiation Utilizability Function." *Journal of Solar Energy Engineering*, Volume 105, pp. 281-287, 1983.

Copsey, A. B., "A Modification of the f-Chart and $\bar{\phi}$, *f*-Chart Method for Solar Domestic Hot Water Systems With Stratified Storage." M.S. Thesis, University of Wisconsin, Madison, Wisconsin, 1984.

Duffie, J. A., and J. W. Mitchell, "*f*-Chart: Predictions and Measurements." *Journal of Solar Energy Engineering*, Volume 105, pp. 3-9, 1983.

Duffie, J. A., and W. A. Beckman, *Solar Engineering of Thermal Processes*, Wiley Interscience, New York, 1980.

Erbs, D. G., "Methods for Estimating the Diffuse Fraction of Hourly Daily, and Monthly-Average Global Solar Radiation." M.S. Thesis, University of Wisconsin, Madison, Wisconsin, 1980.

Erbs, D. G., "Models and Applications for Weather Statistics Related to Building Heating and Cooling Loads." Ph. D. Thesis, University of Wisconsin, Madison, Wisconsin, 1984.

Graham, V. A., "Stochastic Synthesis of the Solar Atmospheric Transmittance." Ph D. Thesis, University of Waterloo, Ontario, Canada, 1985.

Hollands, K. G. T., L. J. D'Andrea, and I. D. Morrison, "Effect of Random Fluctuations in Ambient Air Temperature on Solar System Performance." *Solar Energy*, Volume 42, pp. 335-338, 1989.

IMSL Math/Library, IMSL, Houston, Volume 2, Chapter 6, 1987.

Klein, S. A., "A Design Procedure for Solar Heating Systems." Ph. D. Thesis, University of Wisconsin, Madison, Wisconsin, 1976.

Klein, S. A., "Calculation of Flat-Plate Collector Utilizability." *Solar Energy*, Volume 21, pp. 393-402, 1978.

Klein, S. A., et al., *TRNSYS 13.1 User's Manual*, Engineering Experiment Station Report 38-13, Solar Energy Laboratory, University of Wisconsin-Madison, 1990.

Klein, S. A.and A. H. Fanney, "A Rating Procedure for Solar Domestic Water Heating Systems", *Journal of Solar Energy Engineering*, Volume 105, pp. 430-439, 1983.

Klein, S. A., and W. A. Beckman, *F-CHART User's Manual*, F-Chart Software, Middleton, WI, 1988.

Kleinbach, E. M., "Performance Study of One-Dimensional Models for Stratified Thermal Storage Tank." M.S. Thesis, University of Wisconsin, Madison, Wisconsin, 1990.

Knight, K. M., "Development and Validation of a Weather Data Generator Model." M.S. Thesis, University of Wisconsin, Madison, Wisconsin, 1988.

Minnerly, B. V., "A Long-Term Performance Prediction Method for Solar Domestic Hot Water Systems". M.S. Thesis, University of Wisconsin, Madison, Wisconsin, 1989.

Mutch, J. J., *Residential Water Heating, Fuel Consumption, Economics and Public Policy*, RAND Report R1498, 1974.

Pagnier, M., "A Modification of the $\bar{\phi}$, f-Chart Method for Solar Domestic Water Heating Systems With No Mixing." M.S. Thesis, University of Wisconsin, Madison, Wisconsin, 1986.

Press, W., B. P. Flannery, S. A. Teukolsky, and W. T. Vetterling, *Numerical Recipes*, Cambridge University Press, 1986.

Reindl, D., "Estimating Diffuse Radiation on horizontal Surfaces and Total Radiation on Tilted Surfaces." M.S. Thesis, University of Wisconsin, Madison, Wisconsin, 1988.

Sprikl, W., "Dynamic Solar Domestic Hot Water Testing." *Journal of Solar Energy Engineering*, Volume 112, pp. 98-101, 1990.

Whillier A., "Solar Energy Collection and It's Utilization for House Heating", Ph. D. Thesis, Massachusetts Institute of Technology, Cambridge, Massachusetts, 1953.

Wood, B., "Operating Guidelines and Minimum Standards for Certifying Solar Water Heating Systems: An Optional SWH System Certification and Rating Program." SRCC Document OG-300-89, Solar Rating & Certificate Corporation, Arlington, Virginia, 1989.