SIMULATION OF GROUND COUPLED VERTICAL U-TUBE HEAT EXCHANGERS

by

Steven P. Rottmayer

A thesis submitted in partial fulfillment of

the requirements for the degree of

MASTER OF SCIENCE
(MECHANICAL ENGINEERING)

at the

UNIVERSITY OF WISCONSIN-MADISON
1997

ABSTRACT{ TC "ABSTRACT" \l 1 }

Ground coupled heat pumps are an efficient alternative to conventional methods of conditioning homes, because instead of using the ambient air they utilize the ground as an energy source or sink. However, ground coupled heat pumps have high installation costs that makes it critical to design the system to maximize performance. Vertical u-tube heat exchangers are commonly used as the ground coupled heat exchanger, but estimating their performance is difficult because of the unique heat transfer conditions of this configuration.

This thesis focuses on modeling the vertical u-tube heat exchanger. Several initial attempts to model the heat exchanger were made, and finally an explicit euler finite difference numerical technique was employed. The ground storage volume is divided axially into sections, and each section is a two dimensional cylindrical mesh representing the fluid, tubes, grout, and soil at a specific depth. The tubes are approximated by non circular sections of the mesh, and is accurate to within 8%. A local coupling factor can increase this accuracy to 3% for most systems, and comparisons with an existing model showed good agreement. The finite difference model has provided an approach that is fundamental and readily extended to more realistic conditions. It is accurate and fast enough to be useful as both a comparison to existing models and as a design tool.

ACKNOWLEDGMENTS{ TC "ACKNOWLEDGMENTS" \l 1 }

4

TABLE OF CONTENTS{ TC "TABLE OF CONTENTS" \l 1 }

List of Figures{ TC "List of Figures" \l 1 }

List of Tables{ TC "List of Tables" \l 1 }

Nomenclature

Variables:

α	Thermal diffusivity
c	Specific Heat
δ	Thickness
R	Radial distance between nodes
t	Time step
t_1	Time step for the fluid nodes
t_2	Time step for the soil nodes
t_{TRNSYS}	TRNSYS time step for the soil nodes
$\Delta\theta$	Angular distance between nodes
$\Delta\theta_{mid}$	Angular distance between nodes
EWT	Exiting water temperature
F	Fall
h	Fluid to tube heat transfer coefficient
int_1	Number of fluid temperature updates between ground temperature updates
int_2	Fraction of ground temperature updates between TRNSYS calls
int_3	Fraction of ground temperature updates between TRNSYS calls
k	Thermal conductivity of soil
L	Length of bore hole
m	Radial position of the tubes and edge of grout
$\dot{m}$	Mass flow rate

n	Number of multiples of $\Delta\theta$ contained in 90
N_i	Nondimensional conductance between hot and cold legs of u-tubes
N_o	Nondimensional conductance between hot and cold legs of u-tubes
ρ	Density
π	Mathematical constant pi
$\dot{Q}$	Heat transfer
Q_{out}	Heat transfer from tubes
$Q_{coupling}$	Heat transfer from tubes
r	Radius of node
r_m	Radius half-way between nodes
R	Resistance
R_{fs}	Resistance from fluid to ground
r_{half}	Radius between the center and r(2)
s	Center to center spacing between u-tubes
Sp	Spring
Su	Summer
T	Temperature
$(UA)_i$	Thermal conductance between hot and cold legs of u-tubes
$(UA)_o$	Thermal conductance between hot/cold legs and the ground
V	Volume
x	current axial position in u-tube
W	Winter
$\dot{W}$	Compressor work
Z	Axial distance between nodes

Array Variables:

i	Radial position of the nodes
j	Circumferential position of the nodes
k	Time step
z	Axial position of the nodes
z'	Coupling node

Subscripts:

c	Cold leg of u-tube
f	Fluid
g	Grout
h	Hot leg of u-tube
s	Soil
t	Tube

Chapter 1

Introduction{ TC "Introduction" \l 1 }

Residential heating and cooling account for more than 25% of the nations electrical energy consumption. Since the consumer demand for electricity is continuously rising, improvements in technology must be made. Air Source heat pumps (ASHP) have proven to be an economical means of heating and air conditioning homes in southerly climates. However, these heat pumps employ air to air heat exchangers and their performance is highly dependent on the environment in which they operate. In regions with low winter temperatures the efficiency of an ASHP decreases significantly. This had limited their market to regions with primarily mild winters.

Ground source or geothermal heat pumps (GSHPs) exchange heat with the ground, and maintain a high level of performance even in colder climates. This results in more efficient use of energy. For this reason many public utilities endorse the use of geothermal heat pumps and are active in an effort to persuade the HVAC industry to increase the number installed.

The use of GSHPs, unfortunately, remains low. One major reason for this is the uncertainty in the design of the ground coupled heat exchanger. There is a high installation cost associated with the heat exchanger so it is probably the most important component.

The lack of accurate models is especially apparent for vertical u-tube heat exchangers. These are expensive, but commonly used because they are easier to install in the available space. The u-tube heat exchanger usually consists of a plastic tube 1 inch in diameter with a bend in the middle that

reverses the direction of fluid flow. Many models have been created to simulate their operation, but it is difficult to accurately represent the unique conditions of this heat transfer problem. Most of today's design tools still rely mainly on a primitive model referred to as the "line source theory" (Ingersoll 1954).

The potential energy and economical savings from the use of GSHPs is substantial, but they are also very expensive to install. Poor design can negate any savings and therefore the lack of adequate modeling undermines the confidence of industry in the design tools and GSHPs in general.

1.1 Literature Review{ TC "1.1 Literature Review" \l 2 }

There have been many models which have been used to simulate the use of a vertical u-tube ground coupled heat exchanger. These models have handled the complex geometry of the system by using a variety of assumptions and approximations. Although many models have been created (twenty are discussed in greater detail in Muraya 1994), the assumptions used by each may be combined into three categories.

The first is a single tube approximation. This can be used with numerical methods or the analytical line source theory. An equivalent single tube diameter has been used with either constant temperature sources or constant heat flux sources. The feasibility of this solution was first investigated by Bose (1984). The model developed by Bose used a constant heat flux source as approximated by the line source theory. Through experimentation an empirical relationship between the single tube and the u-tubes was determined to be $\sqrt{2}$ times the u-tube diameter. This is accurate for some situations, but it is limited to a specific range of conditions. The tubes may be unusually close/far, or the conductivity of the grout material uncommonly high/low. Both of these conditions have been shown to make the relationship inaccurate (Muraya 1994). This model was expanded by Kavanaugh (1992) which employed the cylindrical source solution, adjusted for the u-tube separation, and included the effects of thermal coupling. Kavanaugh empirically determined

another coefficient that was used to calculate an equivalent heat transfer coefficient for the tubes.

A second approach is to use a finite element method of analysis. This is because it is easier to accurately model a two tube system with a finite element grid. The problem with a finite element model is the excessive computer time required to perform three dimensional simulations. For this reason, these models have been limited to two dimensions, and the thermal interference is estimated by using an appropriate guess value for the temperature of the fluid returning to the heat pump the amount of interference can be estimated from the two dimensional model. These results are then extrapolated to three dimensions. The temperatures of the fluid entering and exiting the heat pump are typically used because this results in the largest fluid temperature difference, the most interference, and in effect a worst case scenario. This provides much insight into the effects of interference, but it overpredicts its influence. Often, a two dimensional finite element model is used to provide accurate parameters for other three dimensional models.

The third type of model used to investigate this system uses the line source theory. The heat flux from the tubes to the ground is considered to occur in multiple step pulses. These step pulses are used by the line source theory to approximate the short term transient response of the heat exchanger. The temperature of the ground around the tube is updated and continues to change as time passes. The step pulses decrease in intensity as the temperature of the ground increases and eventually a steady state is reached.

One model using the line source theory was created by the University of Lund in Sweden (Hellstrom 1990). An equivalent fluid to ground thermal resistance is calculated with a sophisticated mathematical procedure called the multipole method (Claesson 1988). Step pulses are applied to this resistance to obtain an average temperature at the bore hole edge. Once this temperature is determined a finite difference method is employed from the bore hole edge to an adiabatic boundary far from the tubes. This model has proven to be quite accurate, but two dimensional testing has shown that the approximation of an average bore hole edge temperature can be inaccurate. The

multipole method also requires the step pulse inputs used by the line source theory. This model is commonly considered to be the best available in the industry.

Another model (Dobson 1991) eliminates the assumption of an average temperature at the bore hole edge, but still uses the cylindrical source theory (Claesson 1988) The heat flux from the two tubes are considered separately and then superimposed. The cylindrical source solution is an evolution of the line source theory so this model still relies on the accuracy of the step pulses. It is also a concern that this model would not accurately account for the thermal interference between the tubes.

Many geothermal heat pump design tools determine the size of the ground coupled heat exchanger with the models developed by the third method. In fact, all the ones researched use the line source or cylindrical source solution to the problem. Some of them use the original idea of line source in which the entire length of the tube is considered to have a constant average heat flux. The others use the idea of a line step in which the tubes are split axially into several sections. Either way an assumed heat flux is required. These models are extremely flexible and thorough in their analysis, but they need to be compared with alternative three dimensional models.

1.2 Project Scope{ TC "1.2 Project Scope" \l 2 }

This project proposes a method of simulation for a vertical u-tube ground coupled heat exchanger. The model will enhance the use of current models by providing a rapid alternative that can be used for design or as a comparison to existing models. The heat exchanger is modeled using an explicit euler finite difference approach. The model is a three dimensional transient heat transfer model that includes the fluid temperature distribution through the tubes. The program allows the user to change the following inputs: bore hole depth, flow rate, properties of the fluid, ground, and grout, and temperatures of the ground and inlet fluid. The model is compatible with TRNSYS, a transient

simulation program, and results obtained from simulations are compared to results from the model created at the University of Lund. The evaluation will provide insight into the accuracy of current design tools.

This project is intended to increase the confidence of industry in the design of ground coupled heat pumps. Taking full advantage of the increase in energy savings provided by ground coupled heat pumps will help ensure that their use will increase and the electrical demands of the nation are met in the future.

Chapter 2

Ground Source Heat Pump Fundamentals{ TC "Ground Source Heat Pump Fundamentals" \l 1 }

This chapter describes the fundamentals of a ground source heat pump. The chapter begins with a description of the basic operation of a GSHP and their advantages over traditional ASHP's. This is followed by a description of the unique heat transfer situation GSHP's present, and concludes with a discussion of the current method for modeling vertical u-tube heat exchangers.

2.1 Basic Heat Pump Operation{ TC "2.1 Basic Heat Pump Operation" \l 2 }

Heat pumps are used to provide heating and cooling for residences using a vapor compression cycle for operation. Figure 2.1 shows an example of an ideal vapor compression cycle. In the ideal cycle refrigerant enters the compressor as saturated vapor (state 1) and is compressed to the condenser pressure. The refrigerant is now a superheated vapor (state 2). It is then cooled in the condenser by an external fluid until it becomes a saturated liquid (state 4). The refrigerant next passes through an expansion valve and is throttled to the evaporator pressure. The temperature of the refrigerant drops below the temperature of a second external fluid (state 5), and is heated by this fluid returning the refrigerant to its original condition (state 1).
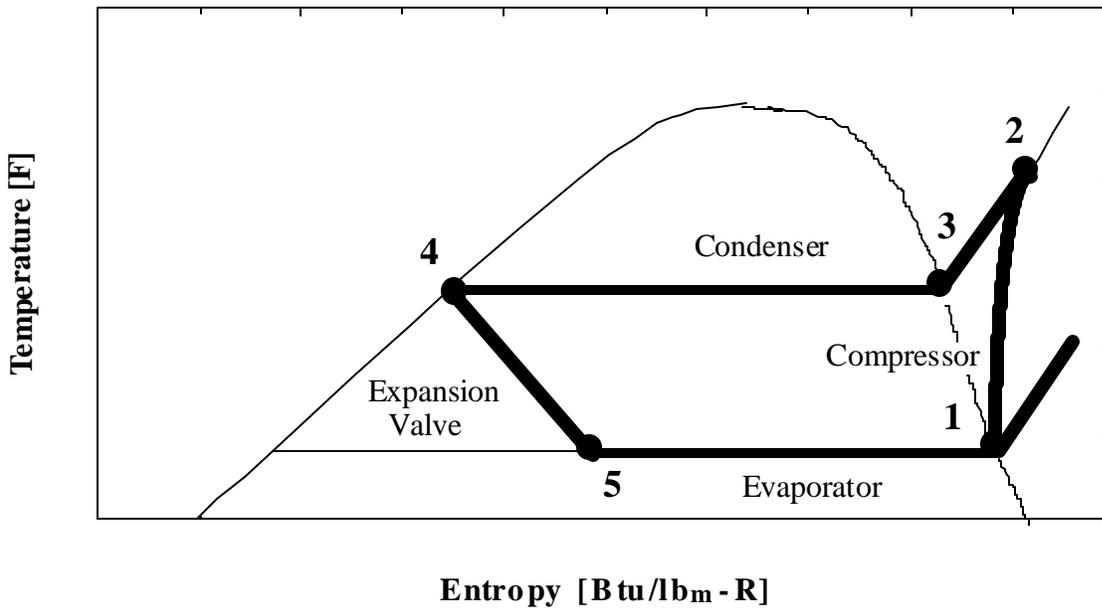
Figure 2.1 Temperature-entropy diagram of vapor compression cycle{ TC "Figure 2.1 Temperature-entropy diagram of vapor compression cycle" \l 8 }

The heat pump is able to either heat or cool a space by using a valve to reverse the flow direction of the refrigerant. During cooling, the heat pump evaporator cools and removes moisture from the indoor air stream, and the condenser, located outdoors, rejects heat to the environment. For heating, the "evaporator" is located outside, and absorbs heat from the environment, while the condenser discharges heat to the indoor air stream. Both of these processes are shown in Figure 2.2.
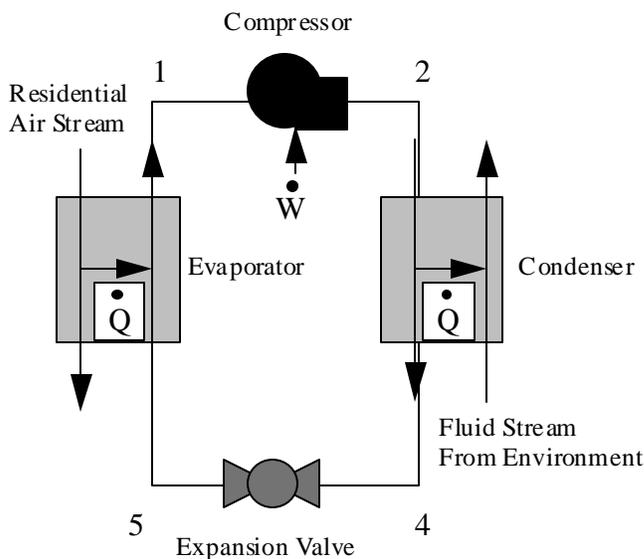
Cooling Mode **Heating Mode**
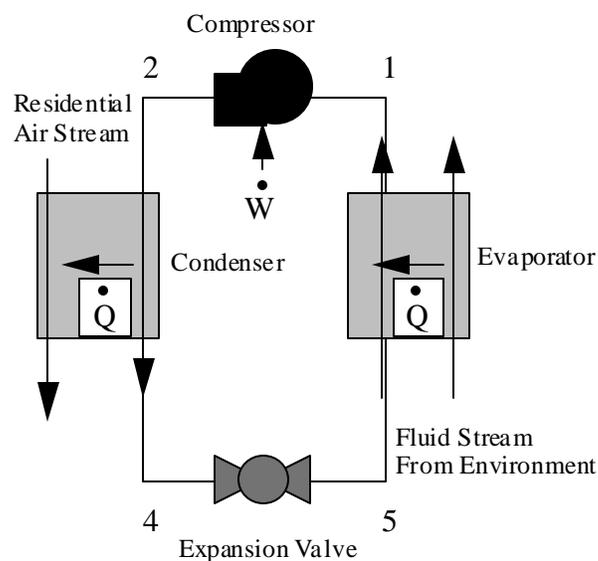
Figure 2.2 Cooling and heating cycles for a heat pump**{** TC "Figure 2.2 Cooling and heating cycles for a heat pump" \l 8 **}**

The outdoor condenser/evaporator can be either an air to refrigerant heat exchanger or a water-to-refrigerant heat exchanger. Air source heat pumps (ASHP) exchange heat with the environment by circulating ambient air through an air-to-refrigerant heat exchanger. Alternatively, water source heat pumps (WSHPs) and ground source heat pumps (GSHPs) transfer heat to the environment with a water-to-refrigerant heat exchanger. ASHPs and GSHPs are the two types of heat pumps used in residential applications.

2.2 Advantages of Ground Coupled Heat Exchangers{ TC **"2.2 Advantages of Ground Coupled Heat Exchangers" \l 2 }**

There are several advantages that GSHPs have over ASHPs. The first is the replacement of an outdoor fan with a fluid circulating pump. This may reduce power and it enables the heat pump, except for the ground coupled heat exchanger, to be completely contained indoors. Minimizing the

contact of equipment with the environment can prolong its life expectancy and GSHPs usually last longer than ASHPs. Another advantage is the water-to-refrigerant heat exchanger. Water has better heat transfer properties than air and this improves the performance of the heat pump. GSHPs can also utilize the benefits of a desuperheater, which is a device that will use the superheated refrigerant (state 2 to state 3) and produce domestic hot water. During the cooling season this heat would be wasted by an ASHP.

Other advantages are the result of the GSHPs' use of the earth as a heat source/sink. The ambient air temperature changes significantly throughout the year, while the temperature of the ground is relatively constant. Since the ground temperature remains closer to the temperature suitable for human comfort the heat pump performance (COP) remains high throughout the year. A related positive effect is that the summer peak demand is reduced compared to traditional air conditioners due to the higher COP in the summer. ASHPs do not provide this savings because they are operating with the same external fluid (ambient air) as traditional air conditioners. The final advantage is the elimination of the defrost cycle. When the air temperature drops below freezing a defrost cycle is required to prevent frost build up on the evaporator surface. The defrost cycle reduces the performance of ASHPs considerably, and ASHPs are not usually run when the ambient air temperature drops below 20 F.

2.3 Classification of Geothermal Heat Pumps{ TC "2.3 Classification of Geothermal Heat Pumps" \l 2 }

There are many types of GSHPs and they are classified by the type of ground coupled heat exchanger they employ. The two main groups of heat exchangers are open loop and closed loop. Open loop systems consist of open ended tubes that extract water from the environment, use it in the water to refrigerant heat exchanger, and then discharge it back to the environment. These heat exchangers obtain water from a well or river, and then release the water at the surface, into a river,

or into a well. Closed loop heat exchangers circulate the heat exchange fluid through tubes buried in the ground. The heat exchange fluid is first used in the heat pump's water to refrigerant heat exchanger, it then leaves the heat pump, circulates through the tubes and finally returns to the heat pump and is reused in the water to refrigerant heat exchanger. Closed loop heat exchangers can be subdivided into pond, horizontal, and vertical loops. Pond loops are tubes in the formation of a coil or slinky that are placed in the bottom of a nearby pond, river, or lake. Horizontal loops can be slinky, series layered, or parallel layered, and these are buried in horizontal trenches that are typically 3-8 feet deep. Since a large amount of space is required several tubes are often placed in a single trench. The final type of heat exchanger is the vertical heat exchanger, which is the subject of the thesis. These can be either u-tubes, tube-in-tube, or slinky and several are placed in parallel or series. Vertical heat exchangers are expensive, but are commonly used instead of horizontal units because they are easier to install in the available space.

The most common type of vertical ground coupled heat exchanger is the u-tube, usually consisting of a plastic tube 1-2 inches in diameter with a bend in the middle that reverses the direction of fluid flow. The tube is buried vertically in the ground and the heat exchanger fluid travels through it exchanging heat with the ground. An example of a two u-tube system in series is shown in Figure 2.3.
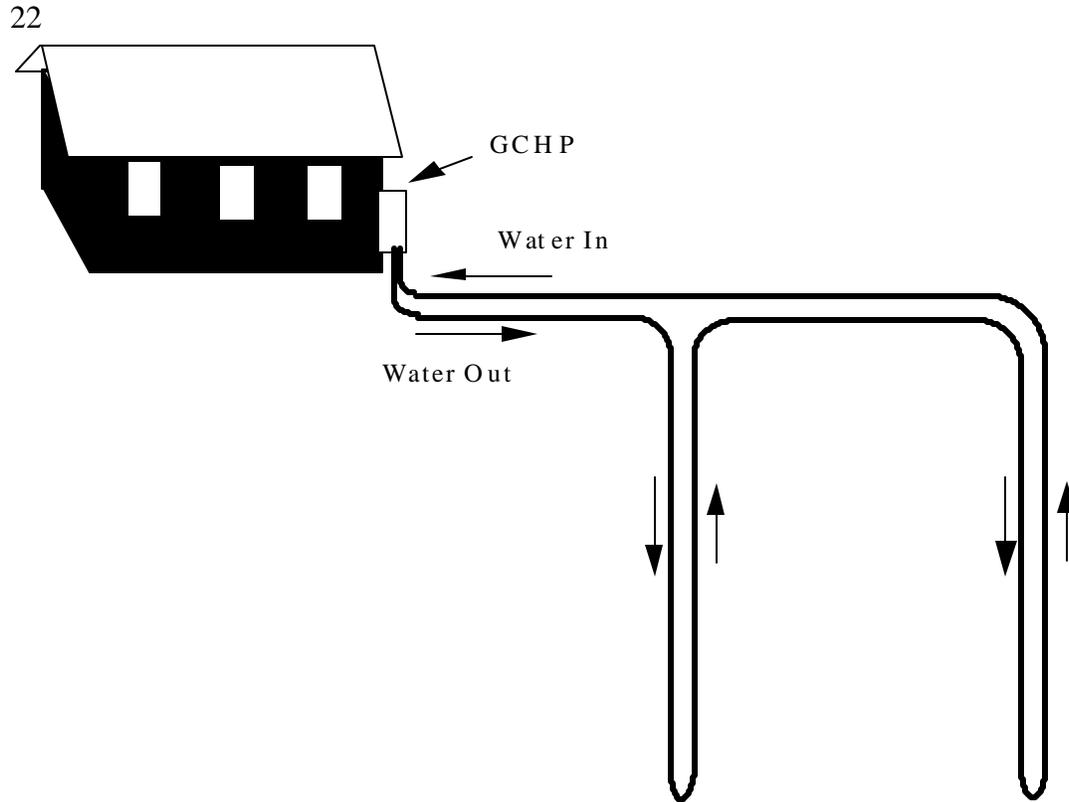
Figure 2.3 Example of two u-tubes in series**{** TC "Figure 2.3 Example of two u-tubes in series" \l 8 **}**

The two components for a GSHP are the heat pump and ground coupled heat exchanger. Modeling the operation of heat pump is well understood, but the modeling of a u-tube ground coupled heat exchanger is not Therefore, the remainder of this thesis focuses on the operation of the ground coupled heat exchanger.

2.4 Discussion of U-Tube Heat Exchangers**{** TC **"2.4 Discussion of U-Tube Heat Exchangers" \l 2 }**

A typical vertical ground coupled heat exchanger is made by drilling a hole in the ground, inserting a u-tube, and then refilling the space around the tubes with a new material referred to as grout. The grout can be a variety of materials ranging from concrete to sand. The heat exchanger fluid is

commonly water or a glycol solution. The fluid exchanges heat with the refrigerant in the heat pump, then exits the heat pump and enters the u-tube. The fluid flows down to the bottom of the tube, reverses direction, and returns to the heat pump. The fluid exchanges heat with the ground as it circulates through the u-tube, resulting in two tubes, also called legs, that contain fluid at different temperatures. One leg will be referred to as the "hot" tube and one will be referred to as the "cold" tube as illustrated in Figure 2.4.
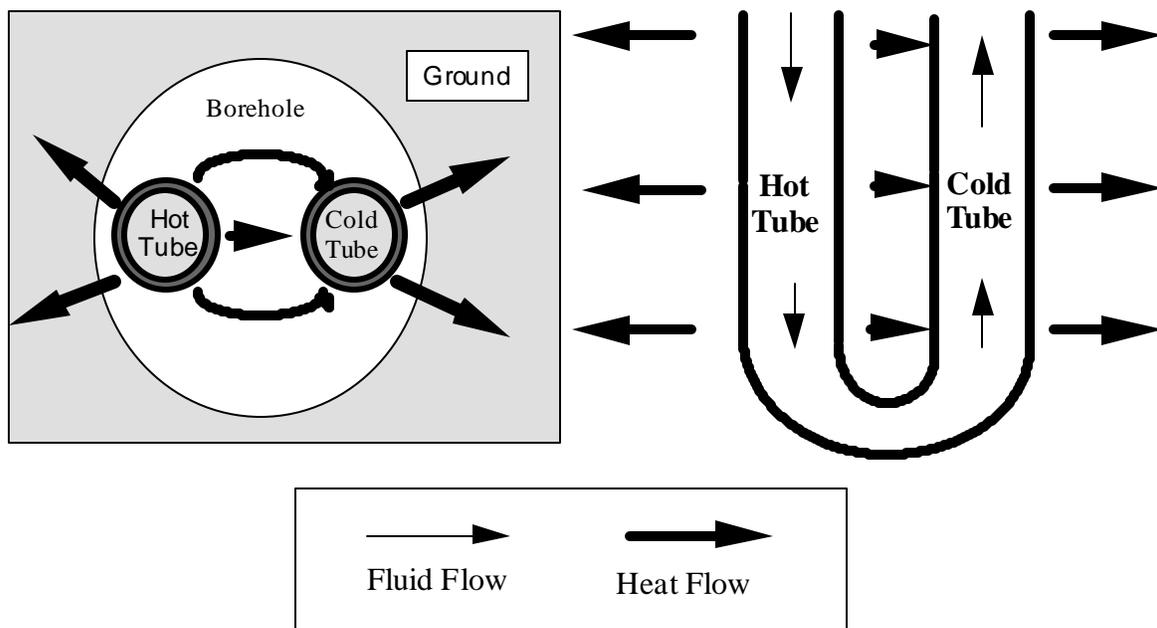
Figure 2.4 Example of fluid flow through a u-tube heat exchanger during cooling**{** TC "Figure 2.4 Example of fluid flow through a u-tube heat exchanger during cooling" \l 8 **}**

The geometry of a u-tube heat exchanger presents a unique heat transfer situation. The two tubes transfer heat to the ground, and also exchange heat with each other. The heat transfer between the tubes is commonly referred to as thermal interference or thermal coupling. Thermal coupling reduces the amount of energy transferred to the ground because the average temperature difference between the two tubes and the ground is less than if the thermal coupling did not occur. As the amount of heat transfer between the tubes increases the heat transfer to the ground decreases.

The effect of thermal coupling was investigated previously as a phenomena present in animals

(Mitchell and Myers 1968). This study is not directly applicable to the u-tube heat exchanger

system because the ground conductance (UA) is not the same as $(UA)_o$ in equation 2.1. This is due

to the change in the ground conductance with time as the ground is either heated or cooled. The

temperature of the ground is also a function of depth and time of year, and in this study the

temperature boundary is considered constant. Although this study is not directly applicable to u-

tube heat exchangers, it does provide insight into the significance of thermal coupling.

The thermal coupling between the blood in the artery and the blood in the vein of a heron leg acts as

an insulator and enables the heron to withstand the cold temperatures of a pond or lake. An

analytical solution to the heat transfer problem was obtained, and the effects of various parameters

on the temperature distribution of the blood analyzed. The geometry of a heron leg is similar to a u-

tube and equations 2.1 and 2.2 show the appropriate energy balance.

$$\dot{m}c\frac{dT_h}{dx} + (UA)_i \cdot (T_h - T_c) + (UA)_o \cdot (T_h - T_\infty) = 0 \qquad (2.1)$$

$$\dot{m}c\frac{dT_c}{dx} + (UA)_i \cdot (T_h - T_c) + (UA)_o \cdot (T_c - T_\infty) = 0 \qquad (2.2)$$

In these equations T_c and T_h represent the fluid temperature in the cold and hot tubes respectively.

T_∞ is the ground temperature at the farfield radius. For this analysis, the farfield radius is

considered the radial distance from the tubes at which the ground temperature is constant. The fluid

flow rate and specific heat are given the notation $\dot{m}c$. $(UA)_o$ represents both the conductances

from each leg of the u-tube to the ground located at the farfield radius. Since the tubes are small (1"

diameter) relative to their distance from the constant temperature boundary (8 ft) these are virtually

identical. The amount of coupling between the tubes is defined as $(UA)_i$. The equations are solved

with the boundary conditions of $T_h=T_o$ (inlet fluid temperature) at $x = 0$, and $T_h = T_v$ at x=L. The

problem is rewritten in non dimensional form, and the solution is given in equations 2.3 through 2.8.

$$\frac{(T_h - T_\infty)}{(T_o - T_\infty)} = \left| \frac{B \cosh A(1-\xi) + \sinh A(1-\xi)}{B \cosh A + \sinh A} \right| \tag{2.3}$$

$$\frac{(T_c - T_\infty)}{(T_o - T_\infty)} = \left| \frac{B \cosh A(1-\xi) - \sinh A(1-\xi)}{B \cosh A + \sinh A} \right| \tag{2.4}$$

$$A = N_o \sqrt{1 + 2\left(\frac{N_i}{N_o}\right)} \tag{2.5}$$

$$B = \sqrt{1 + 2\left(\frac{N_i}{N_o}\right)} \tag{2.6}$$

$$N_o = \frac{(UA)_o \cdot L}{\acute{m}c} \tag{2.7}$$

$$N_i = \frac{(UA)_i \cdot L}{\acute{m}c} \tag{2.8}$$

$$\xi = \frac{x}{L} \tag{2.9}$$

Figures 2.5 and 2.6 show the effects of excessive heat transfer between the two tubes. Figure 2.5 shows that as the coupling increases the net heat transfer to the ground decreases drastically. For large ratios of $(UA)_i$ to $(UA)_o$ the heat transfer approaches zero, and the value of $(UA)_o$ becomes almost irrelevant.

Figure 2.6 shows the effect of thermal coupling on the temperatures of the fluid in the tubes. The coordinate x is equal to 1 at the bottom of the u-tube and 0 at the top of the u-tube. It can be seen that the temperature of the fluid reaches the temperature of the surroundings more rapidly as $(UA)_i$

increases. When $(UA)_i$ becomes infinity the fluid temperature immediately reaches the surrounding

temperature, and the amount of heat transferred to the ground becomes zero.
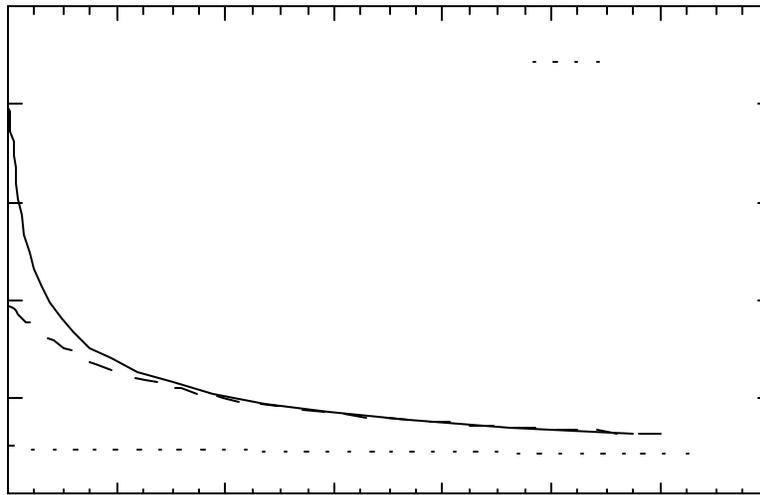
Figure 2.5. The effect of interference on the heat transfer to the ground{ TC "Figure 2.5. The effect of interference on the heat transfer to the ground" \l 8 }

Bottom of
the U-Tube

Top of
the U-Tube
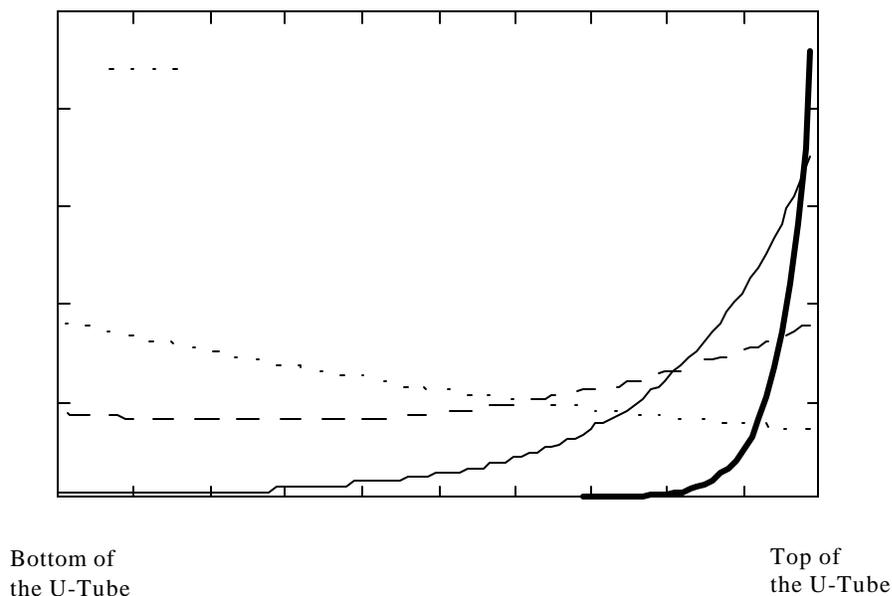
Figure 2.6 Temperature distribution along the cold tube as the coupling increases**{** TC "Figure 2.6 Temperature distribution along the cold tube as the coupling increases" \l 8 **}**

An infinite value for $(UA)_i$ may seem extreme, but it emphasizes the importance of correctly modeling a u-tube heat exchanger, especially as it pertains to the grout material and the separation between the legs of the tubes. With the great expense of installing the ground coupled heat exchangers even slight errors in modeling could result in significant economic losses.

2.5 Line Source Theory**{** TC "2.5 Line Source Theory" \l 2 **}**

Currently, the most common methods for modeling vertical u-tube heat exchangers employ the line source theory. This solution assumes that a tube is buried in a cylinder of soil. The soil located along the edge of this cylinder is not affected by the heat exchanger's absorption or rejection of energy. This distance was discussed previously as a constant temperature boundary and called the farfield radius. With the line source theory, the farfield radius must increase with the operation time of the heat exchanger. This occurs because the soil can not replenish/dissipate the energy as fast as

the heat exchanger removes/supplies it. The effectiveness of the heat exchanger declines because

the temperature difference between the tube and the soil will decrease. Finite difference and finite

element models often use a constant temperature boundary far from the u-tube. This prevents the

farfield radius from expanding beyond this distance, and the result is an artificially high heat transfer

with the ground (Giardina 1995).

In its original form, the line source theory assumes that the buried tube has a uniform heat flux along

its entire surface, thus disregarding temperature gradients in the axial direction (Hart 1986). This

spreads the load evenly over the entire tube length resulting in the inlet and outlet fluid experiencing

the same load. This assumption underestimates the amount of heat transfer to the ground because in

reality the load changes with length. Figure 2.7 shows the difference between a single section and a

tube split into two sections. For a tube being heated, the soil temperature around the single section

decreases evenly. If the tube is divided into two axial sections the first section experiences a greater

load than the second. This brings the soil around the first section to a lower temperature than the

second section. The second section of the two section model now has higher soil temperatures

compared to the single section tube. This results in a higher fluid temperature for the two section

model. To help correct this problem several models using the line source theory have divided the

tubes into several sections and then applied a separate solution to each section.
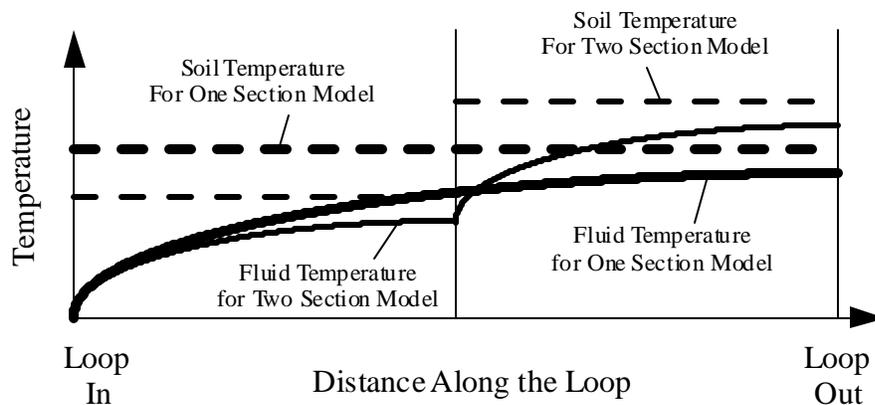
Figure 2.7 Fluid temperature profiles for one and two section loop models{ TC "Figure 2.7 Fluid

temperature profiles for one and two section loop models" \l 8 **}**

Chapter 3

Modeling a U-Tube Heat Exchanger**{** TC "Modeling a U-Tube Heat Exchanger" \l 1 **}**

This chapter discusses concepts for modeling a ground coupled heat exchanger. First, the properties of the materials and the effects of ground heat transfer are discussed. The chapter concludes with brief summaries of the early attempts to model the heat exchangers. This is intended to assist the future research in this field.

3.1 Thermal Properties and Behavior of Soil{ TC **"3.1 Thermal Properties and Behavior of Soil" \l 2 }**

In order to accurately simulate the use of a ground coupled heat exchanger the phenomena associated with ground heat transfer must be understood, and the thermal properties of the materials known. When the ground is used as a heat source/sink two things happen that may significantly affect the performance of the heat exchanger. Moisture migration occurs as the temperature gradient in the ground increases, and in colder climates, the ground freezes. Although these aspects of ground heat transfer are not investigated in this thesis it is important to understand their effects.

Moisture migration occurs during both heating and cooling seasons, but its effects are most important when cooling. In cooling season heat is rejected to the ground, resulting in higher temperatures around the tubes. The ground moisture moves in the direction of hot to cold temperatures, and this dries the soil around the tubes. Heat transfer to the ground decreases

because the thermal conductivity of dry soil is much less than damp or saturated soil (Table 3.2), and the soil will shrink as it dries forming air gaps. The thermal conductivity of these air gaps (0.015 Btu/hr-ft-F) is far lower than dry soil and acts as an insulator between the fluid and the ground. This adversely affects the exit temperature of the heat exchanger fluid and causes the performance of the heat pump to decline. Freezing is beneficial to ground coupled heat exchangers because the thermal properties of frozen soil are more favorable for heat transfer than unfrozen soil. Freezing occurs near the surface and so its effects are not as important for vertical heat exchangers.

The u-tube material is not chosen solely on the basis of its heat transfer properties, but also with the concern of resistance to wear, expense, and ease of installation. The International Ground Source Heat Pump Association (OSU 1988) provided standards which described acceptable tube material. The two most common materials used for the u-tubes are made of polyethylene or polybutylene (OSU 1988). Polyethylene has a thermal conductivity of 0.226 Btu/hr-ft-F and was used for all the simulations.

The grout material is very important when considering the performance of a geothermal heat pump. This material should have a large conductivity to increase the amount of heat transfer to the ground, but if the conductivity is too high the amount of interference between the two legs of the u-tube increases and the ground heat transfer decreases as described in section 2.4. The conductivity of the most common types of grout material are listed in table 3.1 (Kavanaugh).

32

Table 3.1 Conductivity (Btu/hr-ft-F) of typical grouts/backfills**{** TC "Table 3.1 Conductivity

(Btu/hr-ft-F) of typical grouts/backfills" \l 9 **}**

Grouts without Additives	k (Btu/hr-ft-F)	Grout with Additives	k (Btu/hr-ft-F)
20 % Bentonite	0.38-0.49	20% Bentonite - 40% Quartzite	0.80
30% Bentonite	0.40-0.50	30% Bentonite - 40% Quartzite	0.70-0.75
Cement Mortar	0.40-0.45	30% Bentonite - 30% Iron Ore	0.45
Concrete @ 130/150 lb/ft^2	0.60/0.80	60 % Quartzite - Flowable Fill	1.07
Concrete (50% quartz sand)	1.10-1.70	(Cement+Fly+Ash+Sand)	

The type of soil as well as its moisture content significantly affect the soil's thermal properties. There

are hundreds of soil classifications used for different soils found in the United States. These are

combined into five categories for modeling purposes, and are shown in Table 3.2 (OSU 1988).

The property of the soil can also vary with depth, and as the model becomes more sophisticated this

variation can be taken into consideration.

Table 3.2 Thermal properties of soil{ TC "Table 3.2 Thermal properties of soil" \l 9 }

	Thermal Conductivity (Btu/hr-ft-F)	Density (lbm/ft^3)	Specific Heat (Btu/lbm-F)
Heavy Soil Saturated	1.40	200	0.20
Heavy Soil Damp	0.75	131	0.23
Heavy Soil Dry	0.50	125	0.20
Light Soil Damp	0.50	100	0.25
Light Soil Dry	0.20	90	0.20

The temperature of the undisturbed soil is a function of depth and time of year. A function derived by Kusuda in 1965 estimates the seasonal variation of the ground temperature with depth. This is given as equation 3.1.

$$T(Z_{depth}, t_{year}) = T_{mean} - T_{amp} \cdot \exp^{Z_{depth}\left(\frac{\pi}{365 \cdot \alpha_s}\right)^{0.5}} \bullet$$

$$\cos\left\{\frac{2 \cdot \pi}{365}\left[t_{year} - t_{shift} - \frac{Z_{depth}}{2}\left(\frac{365}{\pi \cdot \alpha_s}\right)^{0.5}\right]\right\}$$

(3.1)

T_{mean} is the mean value of the ground temperature for the entire year. T_{amp} is the amplitude of the ground temperature at the surface over the course of the year. The surface temperature drops to a value of T_{mean} - T_{amp} during the winter, and rises to a temperature of T_{mean} + T_{amp} during the summer. Both of these values are dependent on the geographical location, and both are measured in Fahrenheit. The parameter t_{shift} is the difference between the beginning of the calendar year and the time at which the minimum ground temperature occurs.

The Kusuda equation is represented graphically in Figure 3.1.
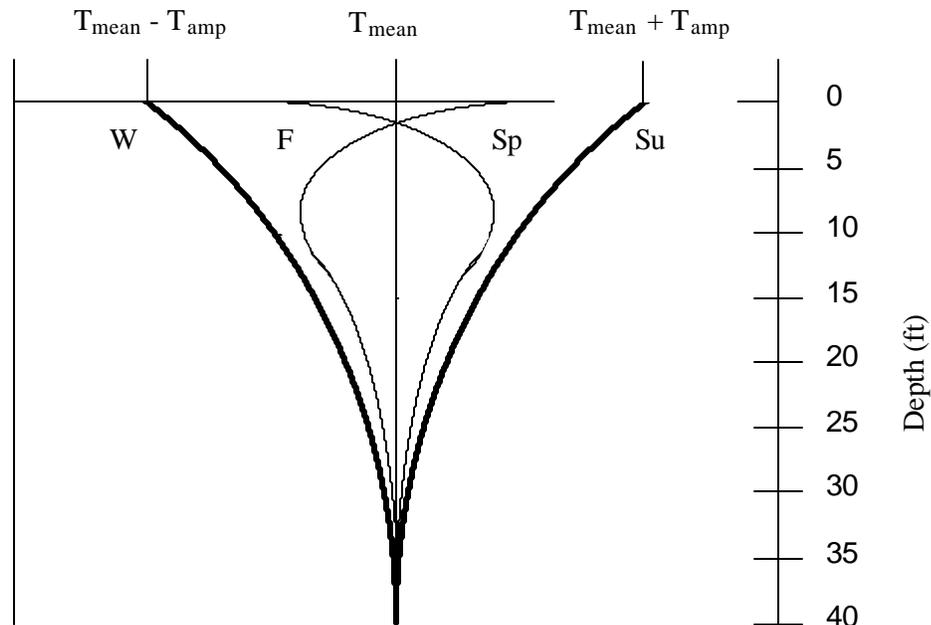
Figure 3.1 Kusuda Relationship for Temperature Distribution in Ground{ TC "Figure 3.1 Kusuda Relationship for Temperature Distribution in Ground" \l 8 }

The curve W shows the ground temperatures at their minimum during the winter, and the line Su shows the ground temperatures at their maximum during the summer. The ground temperature will fall between these two curves at any times other than these extremes. The deeper the soil the slower the ground temperature changes. This is shown by the two curves F and Sp which represent times in the spring and summer of the year.

3.2 Thermal System of a Vertical U-Tube Heat Exchanger{ TC "3.2 Thermal System of a Vertical U-Tube Heat Exchanger" \l 2 }

The thermal system of a u-tube heat exchanger consists of: heat transfer fluid, u-tubes, bore hole grout, and the surrounding ground. The volume of the ground that is affected by the u-tube heat

exchanger is often referred to as the "ground storage volume". For modeling purposes, the ground storage volume can be considered a cylinder with a height equal to the depth of the bore hole. The radius of the cylinder is large enough to ensure that the soil at this distance is not significantly affected by the u-tubes at the center. As indicated, this distance is referred to as the farfield radius, and varies with configurations. The temperature of the ground at the farfield radius is considered to be a function of depth using the relationship of Kusuda.

When modeling ground coupled heat exchangers the ground storage volume is divided axially into sections, and each section represents the thermal system at a specific depth as shown in Figures 3.3, 3.5 and 4.1. The heat transfer is symmetrical about a vertical plane passing through the center of the tubes. Consequently, it is only necessary to model one half of the ground storage volume and its associated tubes, fluid, and grout. A two dimensional mesh represents the fluid, tubes, grout, and soil at one axial section as shown in Figures 3.2, 3.4 and 4.2. These can then be used at each axial section to create a three dimensional model.

3.3 Finite Element Models{ TC "3.3 Finite Element Models" \l 2 }

The first models created to simulate vertical u-tube heat exchangers use finite element methods. This was done because it is much easier to model the two legs of the u-tube with a finite element mesh. The two tubes can be nearly circular, and connecting the nodes into a finite element mesh is straight forward.

The finite element model was created with the program FEHT, and it represents the fluid, tubes, grout material, and ground at one axial section. The fluid is considered lumped and either a constant temperature or constant heat flux source. The program enables the user to easily vary the temperature or heat flux of the fluid, as well as the material properties of the components. The geometry of the system (tube diameter, bore hole radius, etc.) is less flexible, and was unchanged for each simulation. Figure 3.2 shows a picture of the finite element mesh. Steady state and

transient simulations were used to investigate various conditions.
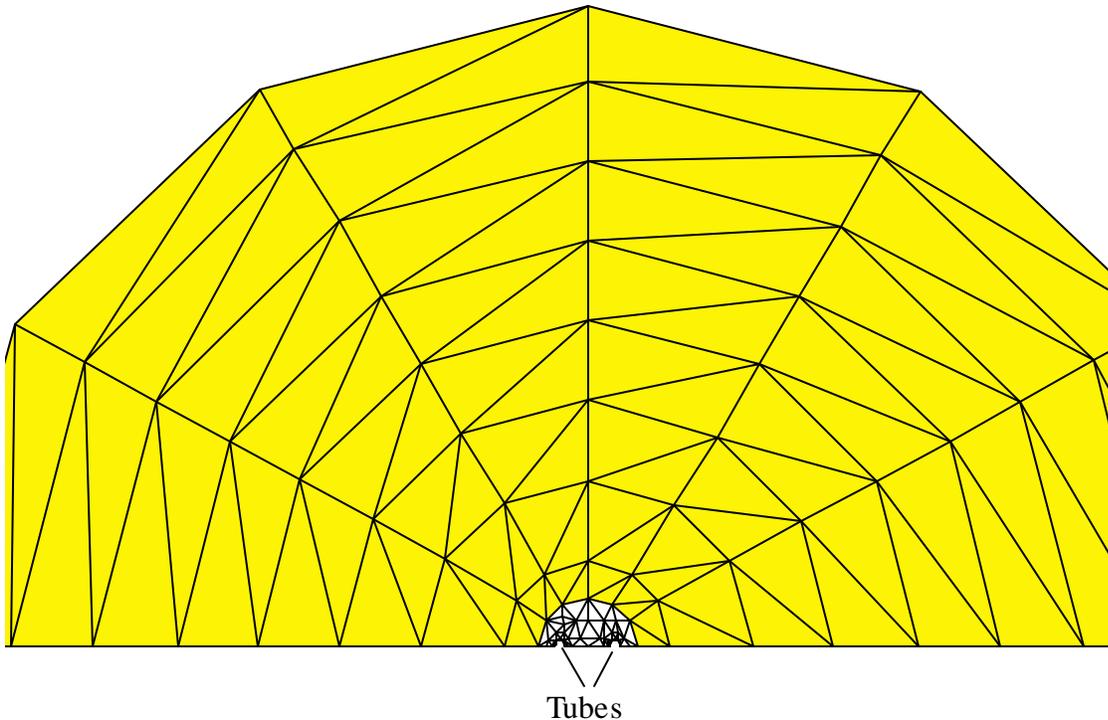
Figure 3.2 Two dimensional finite element mesh of ground coupled heat exchanger**{** TC "Figure 3.2 Two dimensional finite element mesh of ground coupled heat exchanger" \l 8 **}**

Initially, the model was used to gain insight concerning the magnitude of the coupling between the two tubes. The heat exchanger fluid was set to a constant temperature and an attempt was made to isolate the heat transfer due to the temperature difference between the two tubes. These attempts were unsuccessful because it was difficult to isolate this heat transfer. Since a three dimensional model using FEHT would be difficult to create, the model was only used to investigate assumptions, and provide a test for the final model described in chapter 4.

3.4 Finite Difference Model for Two Isolated Tubes{ TC **"3.4 Finite Difference Model for Two Isolated Tubes" \l 2 }**

The second model is a finite difference single tube approximation. A single tube of twice the length

of the bore hole is used to represent a u-tube as shown in Figure 3.3. The model approximates the two dimensional heat transfer in the ground as one dimensional, and did not allow axial conduction. The assumption of no axial conduction along the tube length is justified because of the large distances between the axial nodes relative to their temperature difference. The assumption of no axial conduction is extended to the ground below the heat exchanger and the ground and/or air above the heat exchanger. The ground temperature is calculated using a single tube approximation discussed in chapter 1, except an equivalent radius of twice the diameter is used, and thermal coupling is determined by equation 3.2. The temperatures calculated for the ground are then used at two locations for the single tube approximation as shown in Figure 3.3. The coupling between the tubes is approximated with a shape factor relationship for two cylinders in an infinite medium (Kreith) and is given by equation 3.1.

$$\text{Shape} = \frac{2 \cdot \pi}{\cosh^{-1}\left[\dfrac{\left(\frac{s}{r}\right)^2 - 2}{2 \cdot r}\right]} \tag{3.1}$$

The variable s is the center to center separation of the two legs of the u-tube, and the variable r is the u-tube radius. The heat transfer from the coupling is then included for the energy balance on the fluid nodes as expressed in equations 3.2. The capacitance of the tube and grout material was neglected. In equation 3.2a, R_{fs} is the resistance from the fluid to the ground, and consists of the resistance from the fluid to the tube wall, the tube resistance, and the resistance of the soil and grout. The parameter z' in equation 3.2b refers to the axial position of the fluid node that exchanges heat with the node at z. For example, in Figure 3.3 T(1,k) exchanges heat with T(6,k), therefore, in this case z=1 and z'=6.
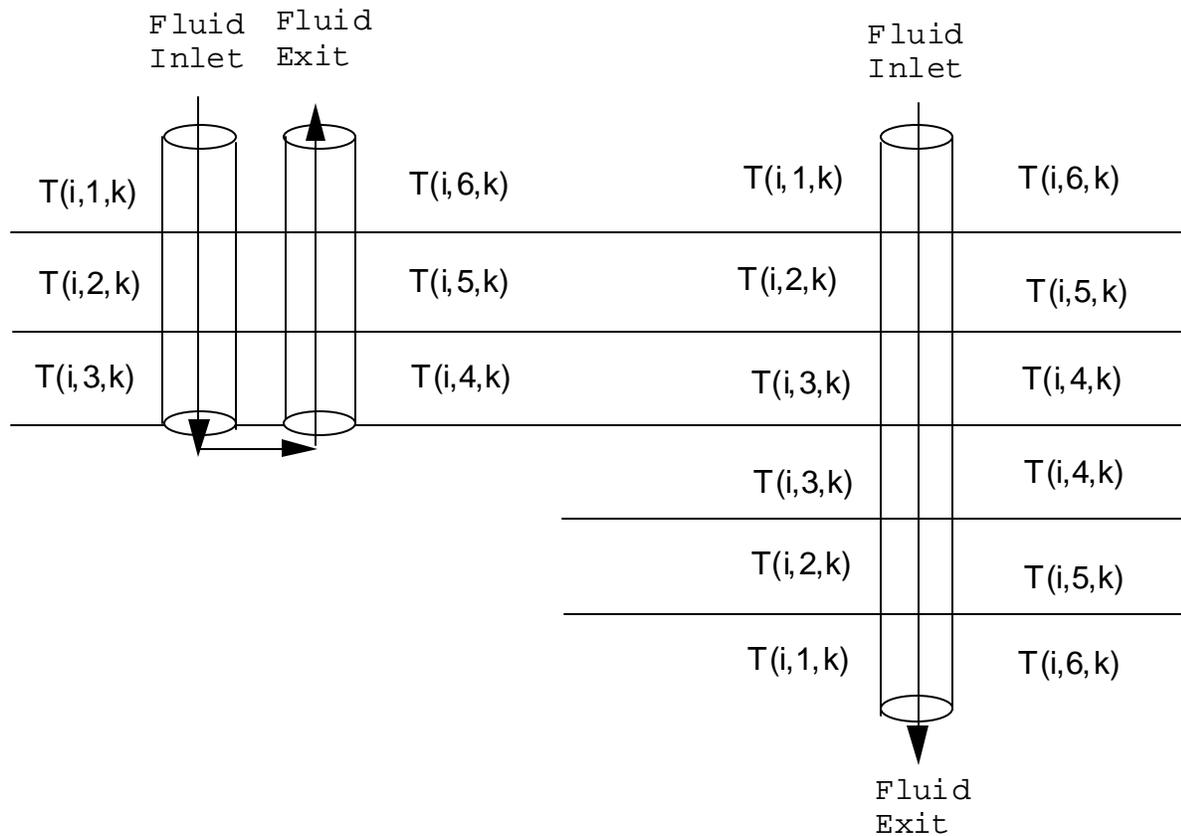
```
Fluid  Fluid                          Fluid
Inlet  Exit                           Inlet
```

T(i,1,k)   T(i,6,k)   T(i,1,k)   T(i,6,k)

T(i,2,k)   T(i,5,k)   T(i,2,k)   T(i,5,k)

T(i,3,k)   T(i,4,k)   T(i,3,k)   T(i,4,k)

                                           T(i,3,k)   T(i,4,k)

                                           T(i,2,k)   T(i,5,k)

                                           T(i,1,k)   T(i,6,k)

```
                                           Fluid
                                           Exit
```

Figure 3.3 Single tube approximation using the shape factor{ TC "Figure 3.3 Single tube approximation using the shape factor" \l 8 }

$$(\rho Vc)_f \frac{(T_f(z, k+1) - T_f(z, k))}{\Delta t} =$$

$$(\dot{m}c)_f (T_f(z-1,k) - T_f(z,k)) - (Q_{out} + Q_{coupling})$$

(3.2)

$$Q_{out} = \frac{(T_f(z,k) - T_s(i,z,k))}{R_{fs}}$$

(3.2a)

$$Q_{coupling} = (Shape \cdot k_s)(T_f(z,k) - T_f(z',k))$$

(3.2b)

This model performed the simulations rapidly, but using the shape factor overpredicted the amount of coupling. The model geometry is also limited by the shape factor because it becomes inaccurate

if L is less than 3 times r. Because of this and the single tube approximation required to calculate the ground temperature, this model is not accurate enough to be useful.

3.5 Rectangular Finite Difference Grids{ TC **"3.5 Rectangular Finite Difference Grids"** \l 2 }

This model simulates the thermal system of ground coupled u-tube heat exchangers as described in section 3.2. The model uses Cartesian coordinates at each axial section of the ground storage volume as shown in Figure 3.4. Each section represents the ground at a specific depth. The model allows heat transfer to occur in the x-y plane, but not axially. The assumption of no axial conduction is once again justified because of the large distances between the axial nodes relative to their temperature difference. The heat transfer above and below the heat exchanger was again neglected. Because of the complex geometry of a u-tube heat exchanger, the circular tubes are approximated with rectangles. The distance between the nodes along the x-axis (x) and the distance between nodes along the y-axis (y) are such that the perimeter of the rectangular and circular tubes are equal and the error in the cross sectional areas minimum. The rectangular grids are then used at each axial section to create a three dimensional model as shown in Figure 3.5. The fluid enters at the surface (z=1), travels through to the bottom of the grid (z=L), back up the return tube, and exits again at the top (z=1).

Constant
Temperature
Boundary

Constant
Temperature
Boundary

y

x = 1

y = 1

x

Adiabatic Boundary

| □ | Nodal Area | ■ | Tube Nodes | ▬ | Farfield Radius |
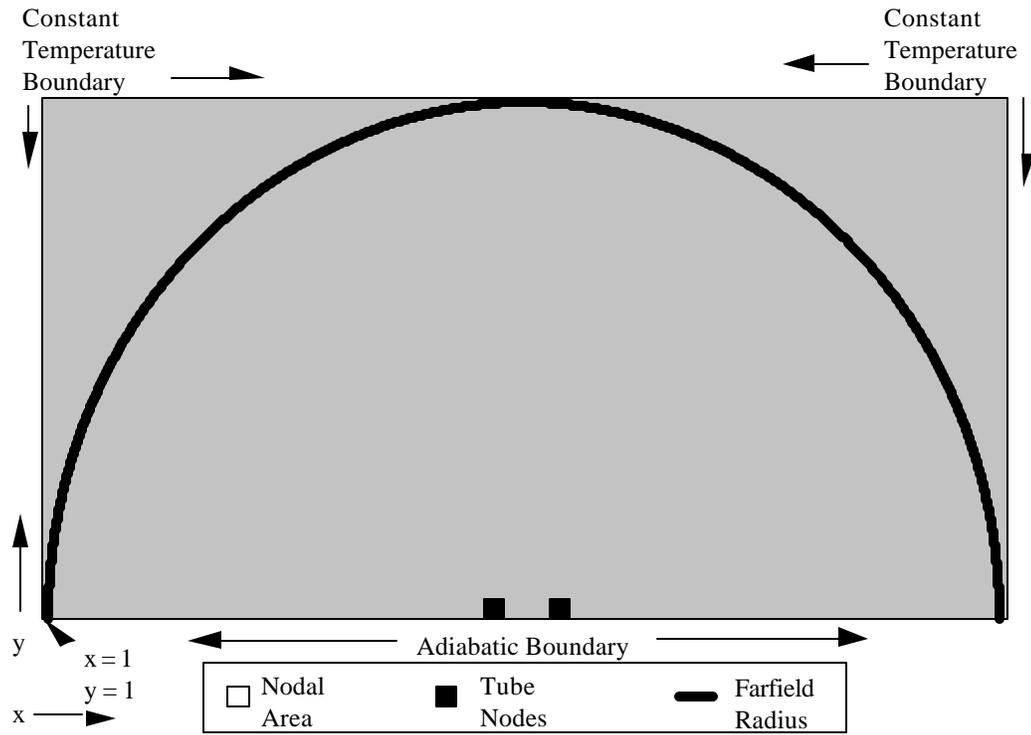
Figure 3.4 Rectangular mesh for u-tube model{ TC "Figure 3.4 Rectangular mesh for u-tube model" \l 8 }

Fluid
Inlet

Fluid
Exit

Section of
rectangle
Grid

^2Z

Z = 1 (Ground Surface)

Z = 2

●

●

●

●

Z = L (Bottom of Bore)
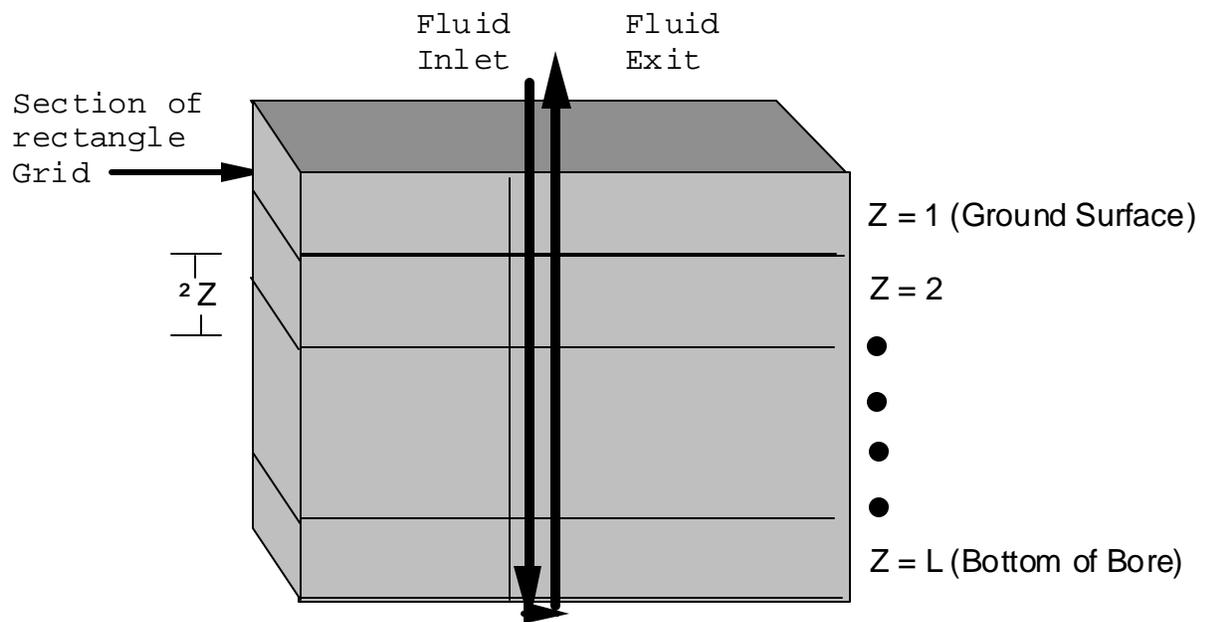
Figure 3.5   Rectangular grids are combined to make a three dimensional model{ TC "Figure 3.5

Rectangular grids are combined to make a three dimensional model" \l 8 **}**

This approach accurately modeled the system, but is extremely slow. An annual simulation would require approximately a week to complete. The model is slow because of the large number of nodes it used (approximately 20,000). Since typical u-tube diameters are 1" and the farfield radius is 8' this number of nodes is required for each axial section. The spacing needs to be small close to the tubes because of the large temperature gradients, but it can be increased as the radial distance from the tubes increases. Unfortunately, this is not easily implemented into a rectangular finite difference mesh.

An attempt to increase the speed of a rectangular finite difference mesh was made by employing the assumption that the heat transfer is one dimensional far from the u-tubes. Two dimensional heat transfer (in the x-y plane) is calculated near the tubes, but farther from the tubes the heat transfer is approximated as only radial. The distance from the tubes at which the heat transfer could be approximated as radial only was determined to be around 2 feet. This dramatically reduced the number of nodes, and increased the speed of the model. However, a more flexible model with fewer assumptions was created, and is described in Chapter 4.

Chapter 4

U-Tube Geothermal Heat Exchanger Model{ TC "U-Tube Geothermal Heat Exchanger Model" \l 1 }

This chapter gives a detailed description of the approach used to model a vertical u-tube ground coupled heat exchanger. It begins with a brief description of the finite difference numerical method, then describes the model. The chapter continues with the results for testing and validation of the method, and concludes by comparing it to the model created by the University of Lund.

4.1 Finite Difference Method{ TC "4.1 Finite Difference Method" \l 2 }

The final model employs a finite difference method (i.e. explicit euler) which is a numerical technique used to discretize heat transfer problems and transform the differential equations into a system of algebraic equations. The system is divided into points referred to as nodes, and each node represents a volume of material. The temperature of the volume is specified by its node. An entire system of nodes make up a finite difference grid or mesh. The energy transfer is driven by the temperature difference between the nodes, and obeys Fourier's law of heat transfer. An energy balance is applied to each node to obtain its temperature. Finite difference models simulate the transient response of a system by updating the temperature of the nodes with time, which is also divided into discrete steps. The temperatures of the nodes are updated after each time step, and the calculations continue until reaching the specified finishing time. There are two techniques to solving finite difference numerical equations. Explicit methods use the temperatures calculated at the

previous time step to obtain the new temperatures. These are constrained to a critical time step, and any time step larger than this causes the simulation to become unstable. The value for the time step is found by dividing the sum of the thermal capacity of the node by the sum of its surrounding thermal resistances as given in equation 4.1.

$$\Delta t_i = \frac{mass_i \cdot c_{p,i}}{\Sigma \frac{1}{R_i}} \tag{4.1}$$

Implicit methods use temperatures from the previous and new time step to obtain the new temperature of the fluid. Implicit methods do not have a time step constraint and are unconditionally stable. The numerical technique used in this model is the Explicit Euler method.

4.2 Modeling of U-Tube Heat Exchanger{ TC "4.2 Modeling of U-Tube Heat Exchanger" \l 2 }

The thermal system of a u-tube heat exchanger is described in section 3.1. The model is similar to the rectangular model described in section 3.4, but it uses cylindrical coordinates. The reason behind this grid geometry is to create a circular grid outside the bore hole. This allows for increasing the nodal spacing in the radial direction to minimize the total number of required. The ground storage volume is divided axially into sections, and each section represents the ground at a specific depth as shown in Figure 4.1. The model developed allows heat transfer in the ground to occur radially and circumferentially, but not axially. The assumption of no axial conduction along the tube length is justified because of the large distances between the axial nodes relative to their temperature difference. The assumption of no axial conduction is extended to the ground below the heat exchanger and the ground and/or air above the heat exchanger. A two dimensional cylindrical grid represents the fluid, tubes, grout, and soil at one axial section. Because of the complex geometry (two tubes in close proximity buried in a third material) of a u-tube heat exchanger the circular tubes

44

are approximated with non-circular cylindrical grid sections as shown in Figure 4.2. The node

positions are represented by the variables i in the radial direction, j in the azimuth direction, z in the

vertical direction, and k in time. The variables n and m are integers and their importance is

explained later (the values of m and n are 3 in Figure 4.2). The radial position of the tubes in the

model is always at the edge of the bore hole (i=m). The cylindrical grids are then used at each axial

section to create a three dimensional model, as shown in Figure 4.1. The fluid enters at the surface

(z = 1), travels through to the bottom of the grid (z = L), back up the return tube, and exits again at

the top (z = 1).

```
          Fluid      Fluid
          Inlet      Exit

Section of
Radial Grid
```

Z = 1 (Ground Surface)

^{2}Z

Z = 2

Z = L (Bottom of Bore)
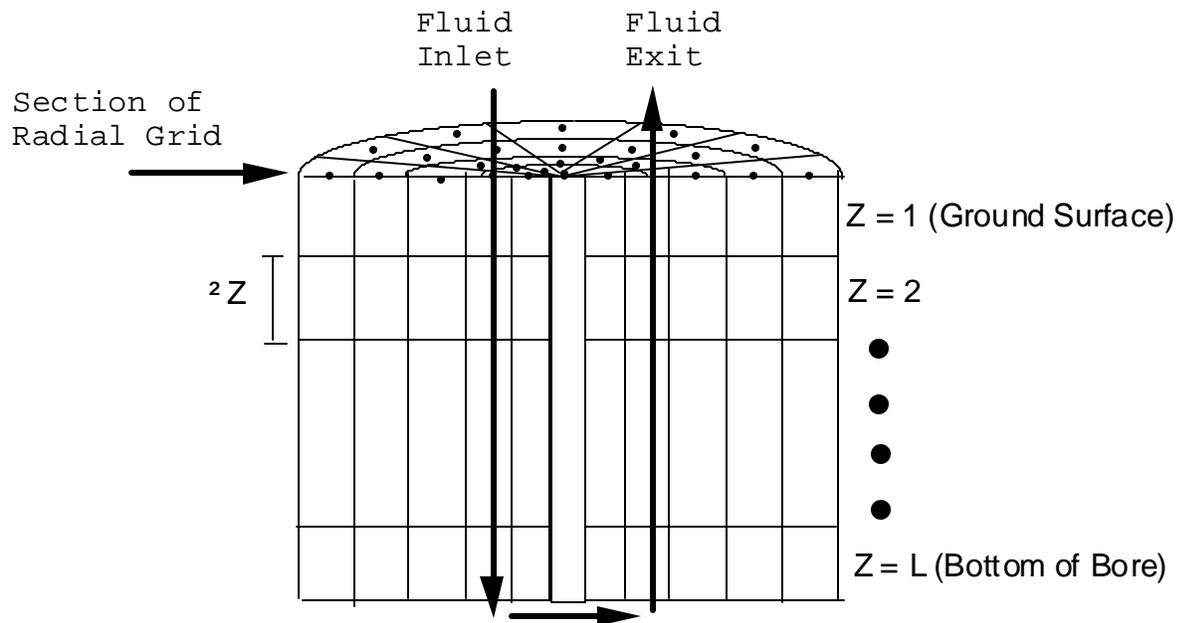
Figure 4.1. The cylindrical grids are combined to make a three dimensional model.{ TC "Figure
4.1. The cylindrical grids are combined to make a three dimensional model." \l 8 }

Figure 4.2. 2-D finite difference grid used to model system at one depth (m=n=3){ TC "Figure 4.2. 2-D finite difference grid used to model system at one depth (m=n=3)" \l 8 }
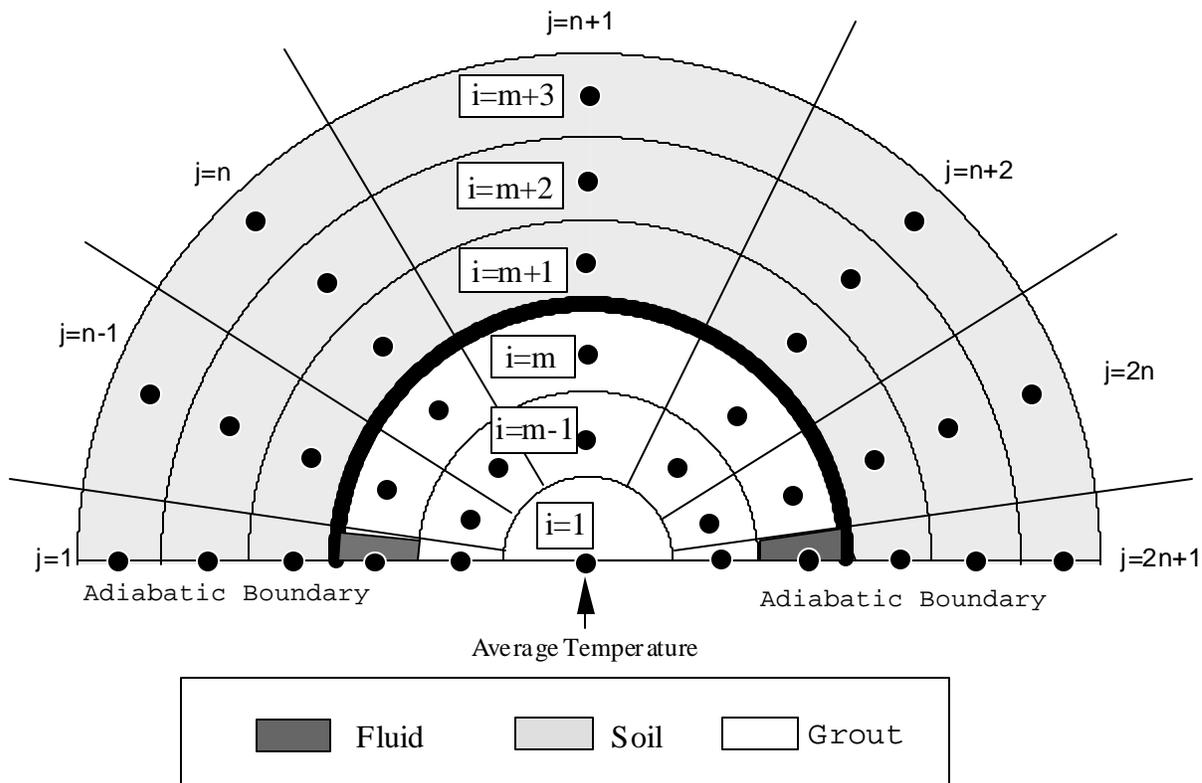
Figure 4.3 shows the resistance network for the soil and the grout nodes, except for the special nodes at j=n+1 and the nodes adjacent to the u-tube which are discussed later. The resistances between the nodes in the ground is given by equations 4.2 and 4.3.
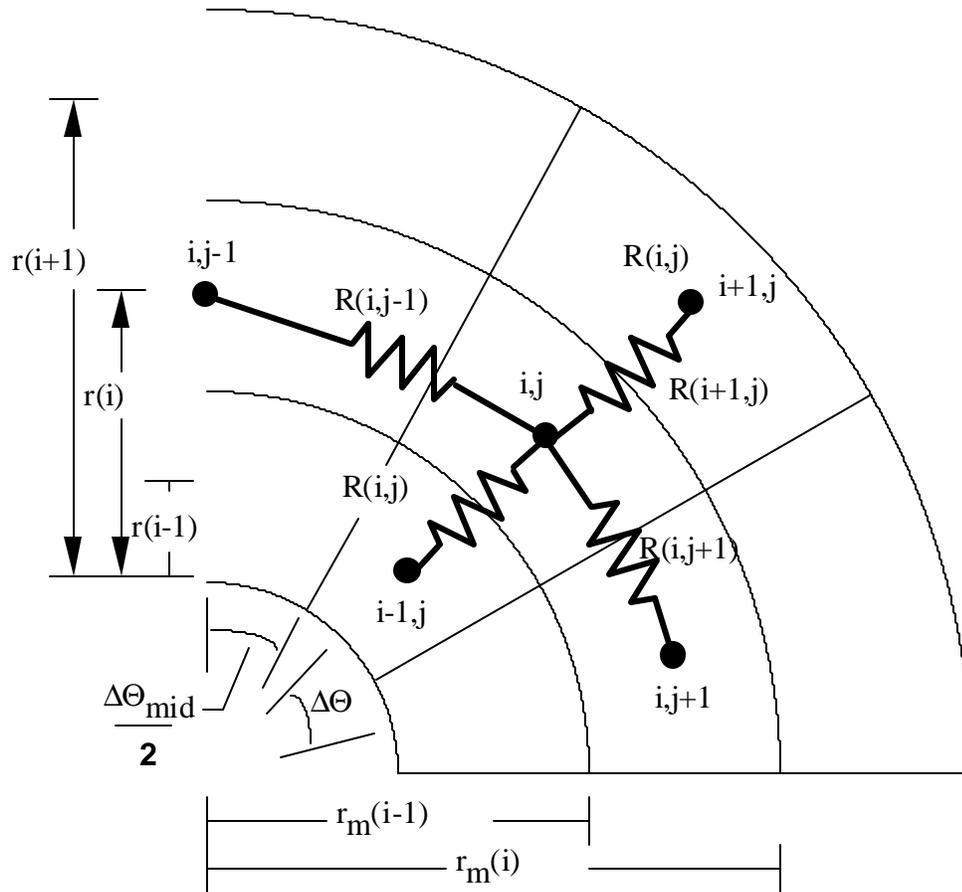
Figure 4.3 Resistance network for cylindrical mesh{ TC "Figure 4.3 Resistance network for cylindrical mesh" \l 8 }

$$R(i, j) = \frac{\ell n\left(\frac{r(i+1)}{r(i)}\right)}{k_s \cdot \Delta\theta \cdot \Delta Z} \tag{4.2}$$

$$R(i, j+1) = \frac{\Delta\theta \cdot r(i)}{k_s \cdot \Delta Z \cdot (r_m(i) - r_m(i-1))} \tag{4.3}$$

The resistance equations for the grout material are identical to those for the soil with k_S replaced with k_g. Figure 4.4 shows the resistance network for the grout to ground interface. Equation 4.4 is used to calculate the resistances between the nodes.

T(m+1,j,z,k)

r(m+1)

R_s

$r_m(m+1)$

R_g

r(m)

T(m,j,z,k)

☐ Soil   ☐ Grout
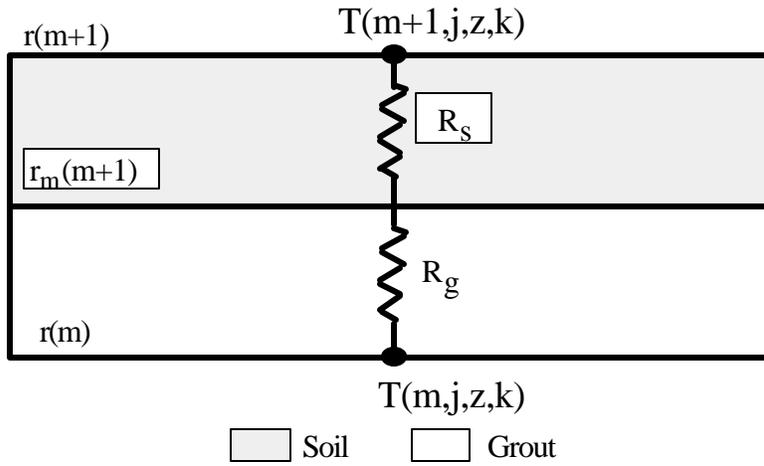
Figure 4.4 Resistance network for the soil to grout interface{ TC "Figure 4.4 Resistance network for the soil to grout interface" \l 8 }

$$R(m, j) = R_g + R_s \tag{4.4}$$

$$R_g = \frac{\ell n\left(\dfrac{r_m(m)}{r(m)}\right)}{k_g \cdot \Delta\theta \cdot \Delta Z} \tag{4.4a}$$

$$R_s = \frac{\ell n\left(\dfrac{r(m+1)}{r_m(m)}\right)}{k_s \cdot \Delta\theta \cdot \Delta Z} \tag{4.4b}$$

Equation 4.5 is an Euler type (i.e. explicit) finite difference energy balance on a soil node, and is used to update the soil node temperatures.

$$\left(\rho c_p V(i)\right)_s \frac{\left(T(i, j, z, k+1) - T(i, j, z, k)\right)}{\Delta t_2} =$$

$$\frac{\left(T(i-1, j, z, k) - T(i, j, z, k)\right)}{R(i-1, j)\cdot} + \frac{\left(T(i+1, j, z, k) - T(i, j, z, k)\right)}{R(i, j)} +$$

$$\frac{\left(T(i, j-1, z, k) - T(i, j, z, k)\right)}{R(i, j-1)} + \frac{\left(T(i, j+1, z, k) - T(i, j, z, k)\right)}{R(i, j+1)} \tag{4.5}$$

The capacitance in the grout is neglected and the grout node temperatures are calculated using equation 4.6.

$$0 = \frac{T(i-1,j,z,k) - T(i,j,z,k+1)}{R(i-1,j)} + \frac{T(i+1,j,z,k) - T(i,j,z,k)}{R(i,j)}$$

$$+ \frac{T(i,j+1,z,k) - T(i,j,z,k)}{R(i,j+1)} + \frac{T(i,j-1,z,k) - T(i,j,z,k)}{R(i,j-1)}$$

$$(4.6)$$

The radial spacing, R, and the angular spacing, $\Delta\theta$, are chosen such that the perimeter of the circular and the non circular tubes are equal and the difference in cross sectional areas is minimal. Since $\Delta\theta$ is a function of the tube size it will not necessarily be an integer divisor of 180. To account for this, the angular spacing between the nodes will be $\Delta\theta$, except at j=n+1 where the angular separation is $\left(\frac{\Delta\theta}{2} + \frac{\Delta\theta_{mid}}{2}\right)$. The value of $\Delta\theta_{mid}$ is calculated by first determining the number of nodes that will have an angular separation of $\Delta\theta$, this number is indexed as n, and then equation 4.7 is used to determine $\Delta\theta_{mid}$.

$$\Delta\theta_{mid} = 2 \cdot \left(\frac{\pi}{2} - \left(n - \frac{1}{2}\right) \cdot \Delta\theta\right)$$

$$(4.7)$$

Equations 4.2 and 4.4 are corrected by replacing $\Delta\theta$ with $\Delta\theta_{mid}$. Equation 4.3 requires the replacement of $\Delta\theta$ with $\left(\frac{\Delta\theta}{2} + \frac{\Delta\theta_{mid}}{2}\right)$, and equation 4.5 is corrected by calculating V(i) with $\Delta\theta_{mid}$ instead of $\Delta\theta$. The nodes along the adiabatic boundary (j=1 and j=2n+1) use $\Delta\theta$ spacing, but are only half the volume of the other nodes.

As indicated, the tubes in the finite difference mesh are not circular, but approximated by non circular sections of the grid. The fluid to ground thermal resistance is determined with three resistance networks, R(m-1,1), R(m,1), and R(m,2), shown in Figure 4.5. Each network consists of the resistance from the fluid to the tube wall, the tube resistance, and the resistance of the soil or

grout. Equation 4.8 through 4.10 are used to determine R(m-1,1), R(m,1), and R(m,2).
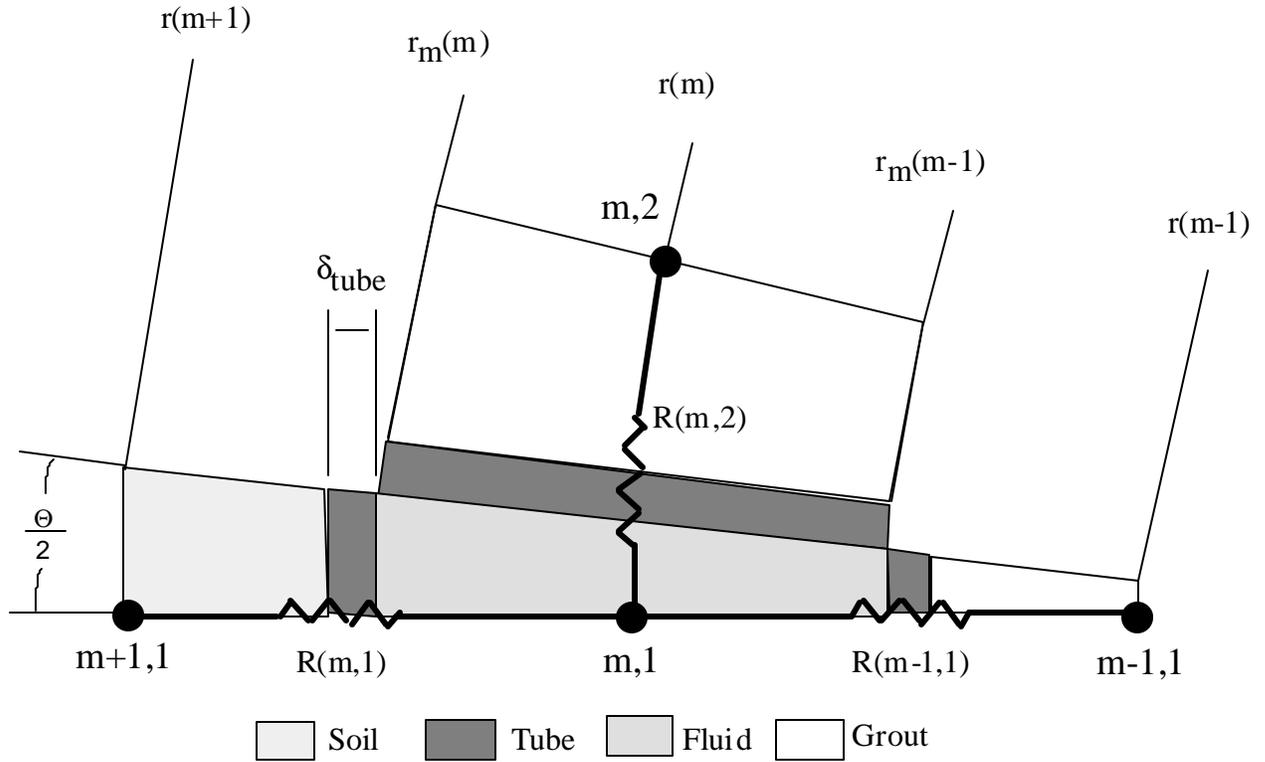
Figure 4.5 Non circular tube section.{ TC "Figure 4.5 Non circular tube section." \l 8 }

$$R(m-1,1) = R(m-1,1)_f + R(m-1,1)_t + R(m-1,1)_g \qquad (4.8)$$

$$R(m-1,1)_f = \frac{1}{h\Delta Z \dfrac{\Delta\theta}{2} r_m(m-1)} \qquad (4.8a)$$

$$R(m-1,1)_t = \frac{\ln\left(r_m(m-1)/(r_m(m-1)-\delta_t)\right)}{\left(\dfrac{\Delta\theta}{2}\right)k_t\Delta Z} \qquad (4.8b)$$

$$R(m-1,1)_g = \frac{\ln\left((r_m(m-1)-\delta_t)/r(m)\right)}{\left(\dfrac{\Delta\theta}{2}\right)k_g\Delta Z} \qquad (4.8c)$$

$$R(m,1) = R(m,1)_f + R(m,1)_t + R(m,1)_s \tag{4.9}$$

$$R(m,1)_f = \cfrac{1}{h\Delta Z \cfrac{\Delta\theta}{2} r_m(m)} \tag{4.9a}$$

$$R(m,1)_t = \cfrac{\ln\left(r_m(m)/(r_m(m) + \delta_t)\right)}{\left(\cfrac{\Delta\theta}{2}\right)k_t \Delta Z} \tag{4.9b}$$

$$R(m,1)_s = \cfrac{\ln\left((r(m+1) - \delta_t)/r_m(m)\right)}{\left(\cfrac{\Delta\theta}{2}\right)k_s \Delta Z} \tag{4.9c}$$

$$R(m,2) = R(m,2)_f + R(m,2)_t + R(m,2)_g \tag{4.10}$$

$$R(m,2)_f = \cfrac{1}{h\Delta Z(r_m(m) - r_m(m-1))} \tag{4.10a}$$

$$R(m,2)_t = \cfrac{\delta_t}{k_t \Delta Z(r_m(m) - r_m(m-1))} \tag{4.10b}$$

$$R(m,2)_g = \cfrac{\left(r(m)\left(\cfrac{\Delta\theta}{2}\right) - \delta_t\right)}{k_g \Delta Z(r_m(m) - r_m(m-1))} \tag{4.10c}$$

Since R and $\Delta\theta$ are calculated so the inside perimeters of the non circular and circular tubes are equal, their cross sectional areas will not be equal. Initially, the error in the cross sectional areas was minimized by using a program written in EES. It was discovered that the values of R that minimize the error are independent of the center to center leg spacing between the tubes as shown in Figure 4.6. Therefore, R can be calculated from a linear relationship as given by equation 4.11

$$R = 0.7854 \cdot (\text{U-Tube Inner Diameter}) \tag{4.11}$$

Since the cross sectional areas of the non circular and circular tubes are not equal, the flow area is

calculated using the flow area of a circular tube.
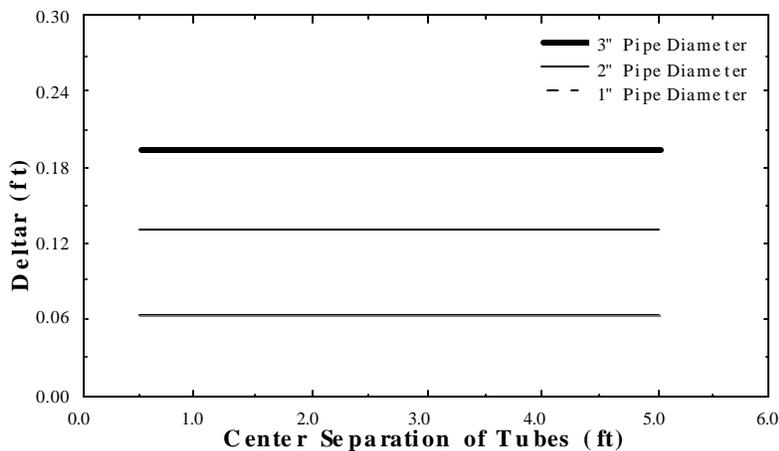
Figure 4.6 Independence of   R on center-center separation of tubes**{** TC "Figure 4.6
Independence of   R on center-center separation of tubes" \l 8 **}**

The equations for the heat transfer from the tubes to the ground are determined from an energy

balance on the fluid nodes as shown in Figure 4.7 and expressed in equations 4.12. The thermal

capacitance of the tube wall and grout are neglected. The energy balance is performed on both fluid

nodes; $T(m,1,z,k)$ and $T(m,2n+1,z,k)$.

$$\dot{m}c_p(T(m,1,z-1,k)-T_o)$$

ΔZ

Fluid Section
$(\rho Vc_p)_{fluid}$ • $T(m,1,z,k)$

$\bar{Q}_{out}$

$$\dot{m}c_p(T(m,1,z,k)-T_o)$$
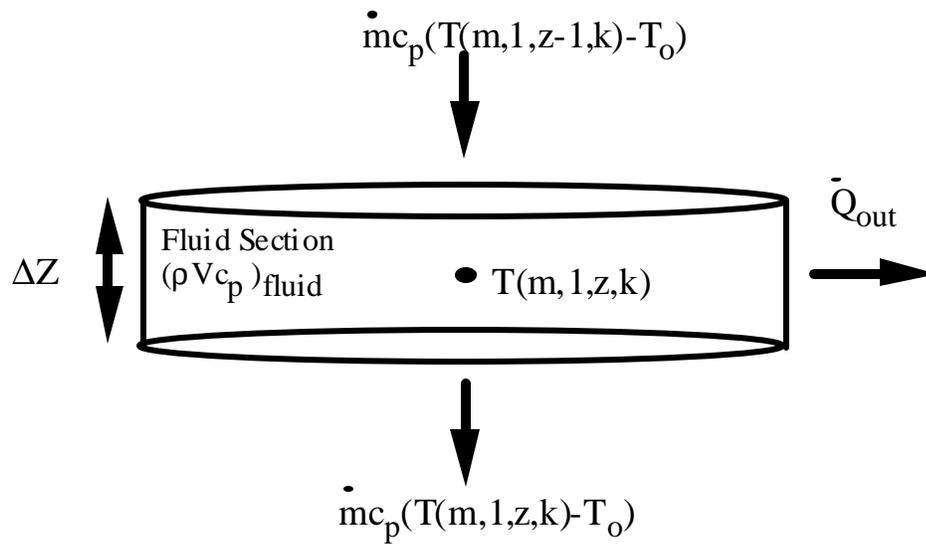
Figure 4.7 Energy balance on the fluid node{ TC "Figure 4.7 Energy balance on the fluid node" \l 8 }

$$T(m,1,z,k+1) = T(m,1,z,k) -$$

$$\frac{\dot{m}\Delta t_1}{(\rho V)_f}(T(m,1,z-1,k) - T(m,1,z,k)) - \frac{\Delta t}{(\rho Vc_p)_f}Q_{out} \tag{4.13}$$

$$Q_{out} = \frac{T(m,1,z,k) - T(m-1,1,z,k)}{R(m-1,1)} +$$

$$\frac{T(m,1,z,k) - T(m+1,1,z,k)}{R(m,1)} + \frac{T(m,1,z,k) - T(m,2,z,k)}{R(m,2)} \tag{4.13a}$$

Since the finite difference mesh is not symmetrical about the center the temperature of the node in the center of the mesh (i=1) is determined from the resistances between the center node and its adjacent nodes as shown in Figure 4.8. The resistance is determined from equations 4.14 and 4.15. Using these resistances, an energy balance on the center node will result in a central node temperature that is equal to the average temperature of its surrounding nodes. The nodes at T(m-1,1,z,k) and T(m-1,2n+1,z,k) use $\dfrac{\Delta\theta}{2}$ in place of $\Delta\theta$ in equation 4.15.
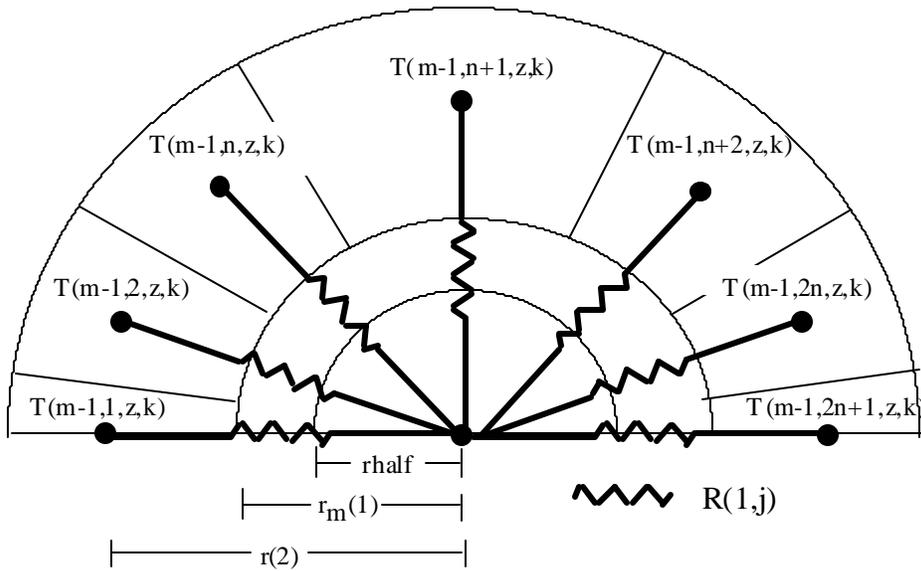
Figure 4.8 Resistances from the center node to its surrounding nodes**{ TC "Figure 4.8 Resistances from the center node to its surrounding nodes" \l 8 }**

$$\pi \cdot \text{rhalf}^2 = \pi\left(r_m(1)^2 - \text{rhalf}^2\right)$$ (4.14)

$$R(1,j) = \frac{\ell n\left(\dfrac{r(2)}{\text{rhalf}}\right)}{k_g \cdot \Delta\theta \cdot \Delta Z}$$ (4.15)

Equations 4.2 through 4.15 have been combined into a transient model for the ground coupled heat exchanger. The time required to run annual simulations is reduced by minimizing the number of nodes and splitting the system into two time domains.

The number of nodes in the system is dependent on the nodal spacing. The distance between the circumferential nodes is dependent on the system geometry, but the distance between axial nodes is 10 ft. The axial spacing can be increased or decreased depending on the level of accuracy desired. The radial spacing for the tube nodes and the nodes close to the tubes depends on the u-tube inner diameter as given by equation 4.11. This spacing is usually small (1 in) relative to the farfield radius (8 ft). The spacing needs to be small near the tubes because the temperature gradients are large.

54

As the radial distance from the tubes increases the temperature gradients decrease making it

possible to increase the radial spacing without loss in accuracy. Currently, the model starts with ΔR

at i=10 and increases it by multiplying it by a factor A. The new ΔR is then multiplied by A and this

continues until the final node at the farfield radius. Figure 4.9 illustrates this process. A program

was written in EES to calculate the appropriate value of A.
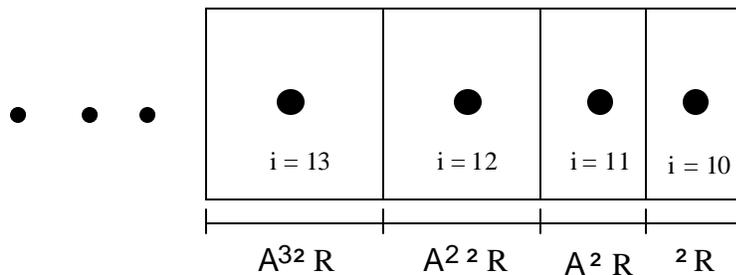
Figure 4.9 Increasing the separation distance between the nodes.**{ TC "Figure 4.9 Increasing the separation distance between the nodes." \l 8 }**

Two time steps are used, one for the fluid (Δt$_1$) and one for the ground (Δt$_2$). This is because the

critical time step for the finite difference mesh is not determined by equation 4.1 alone. It must also

correspond to the time it takes the fluid to travel through one tube section of length ΔZ (Figure 4.7).

For current practice, this time is often 15-30 seconds depending on the distance between the

sections. Initial testing of the model showed that the heat transfer in the ground usually occurs at a

much slower rate than this. The fluid temperature is solved at a short time step (Δt$_1$), and the

ground is solved at a larger time step (Δt$_2$). The value of int$_2$ is determined by dividing the ground

critical time step by fluid critical time step. This number is then truncated to ensure that int$_2$ is an

integer and that Δt$_2$ will still satisfy equation 3.1. The value for Δt$_2$ is then determined using

equation 4.16.

$$\Delta t_2 = int_2 \cdot \Delta t_1 \qquad\qquad (4.16)$$

For example, if t_1 was 15 seconds, and t_2 was 3 minutes then the fluid temperature could be calculated 12 times before the ground temperature was updated. The temperature of the ground would be determined from the fluid temperature calculated at the 12th time step. The fluid is then calculated 12 more times and then the ground temperature is again updated. The simulation will continue this way until finished. Using two time steps decreased the simulation time by about 80% over simulations in which the time steps were the same for both fluid and ground.

For the u-tube geometry specified in Table 1 the number of nodes radially, circumferentially, and axially were 17, 7, and 20, respectively. and the time steps for the fluid and ground were 20 seconds and 3 minutes respectively. The model is able to complete an annual simulation in approximately one half hour on a 166 Mhz Pentium computer.

The model is used with TRNSYS which is a program that combines individual components (i.e the heat exchanger, heat pump, building space) to simulate thermodynamic systems. The simulation time step for TRNSYS is defined by the user and is often much greater than the time steps discussed previously (1 hour is the suggested time step for TRNSYS). The heat exchanger model must operate below t_1 and t_2, so it will update the ground and fluid temperatures several times before returning an answer to TRNSYS. This enables the heat exchanger subroutine to operate at time steps below t_1, and t_2 while the TRNSYS simulation may run at any time step desired.

The number of times the fluid and ground temperatures are updated between TRNSYS calls is usually determined by t_1 and t_{TRNSYS}. The value of int$_1$ is calculated by dividing t_{TRNSYS} by t_1, truncating the resulting number, and then adding 1 to it. This ensures that t_1 is an integer division of t_{TRNSYS} as shown in equation 4.17.

$$\Delta t_1 = \frac{\Delta t_{TRNSYS}}{int_1} \qquad (4.17)$$

Next the subroutine determines the value of t_2 as shown in equation 4.16. The value for t_2 will not usually be an integer divisor of t_{TRNSYS}. Therefore, an initial t_2 is used for the first several ground temperature updates, but for the time step just before returning to TRNSYS it is recalculated to obtain a ground time step that will fit the TRNSYS time step. For this last time step the fluid temperatures are updated int_3 times before the ground is updated and the simulation returns to TRNSYS. Every grid temperature update results in a value for the exiting water temperature (EWT), but since only one EWT is required by the other components an average is calculated as shown in equation 4.18. For example, if $t_{TRNSYS} = 10$ minutes, $t_1 = 30$ seconds, and $t_2 = 3$ minutes the fluid temperature would be updated 6 times before the ground temperature was updated. The fluid temperature would then be calculated 12 more times and the ground temperature twice. On the final update, before returning to TRNSYS, t_2 is recalculated to be 1 minute. The fluid temperature is then calculated twice for this last time step. Since only one EWT is required for each time step the average of these 20 EWTs must be taken as shown in equation 4.18. In this equation $int_1 = 6$, $int_2 = 3$, and $int_3 = 2$.

$$EWT = \frac{EWT_1 + EWT_2 + \cdots + EWT_{int_1 \cdot int_2 + int_3}}{int_1 \cdot int_2 + int_3} \qquad (4.18)$$

4.3 Testing and Validation of the Model{ TC "4.3 Testing and Validation of the Model" \l 2 }

Several initial tests were run on the model to ensure it was working properly. Simulations with $k_s = 0$ resulted in no heat transfer to the ground, and simulations with $cp_f = 10^{12}$ resulted in an exit fluid temperature equal to the inlet temperature. Another test examined how the mesh behaved if 100 F water is supplied to the ground with a 70 F constant temperature boundary. The water initially flowed at 1.5 gpm and then was decreased to 0 gpm. After the flow was stopped every node in the grid became 70 F. When the flow rate of 1.5 gpm was not stopped the energy stored in the

ground was only 0.3 % different from the energy transferred from the fluid.

After these initial tests simulations were run with the conditions shown in table 4.5 and the effect of splitting the system into two time steps investigated. Figure 4.10 shows the effect of splitting the system in this manner. The variable int_2 refers to number of times the fluid temperatures are calculated before the ground temperatures are updated. It can be seen that the effects are most significant in the first hours of the simulation, but the solutions rapidly become identical.
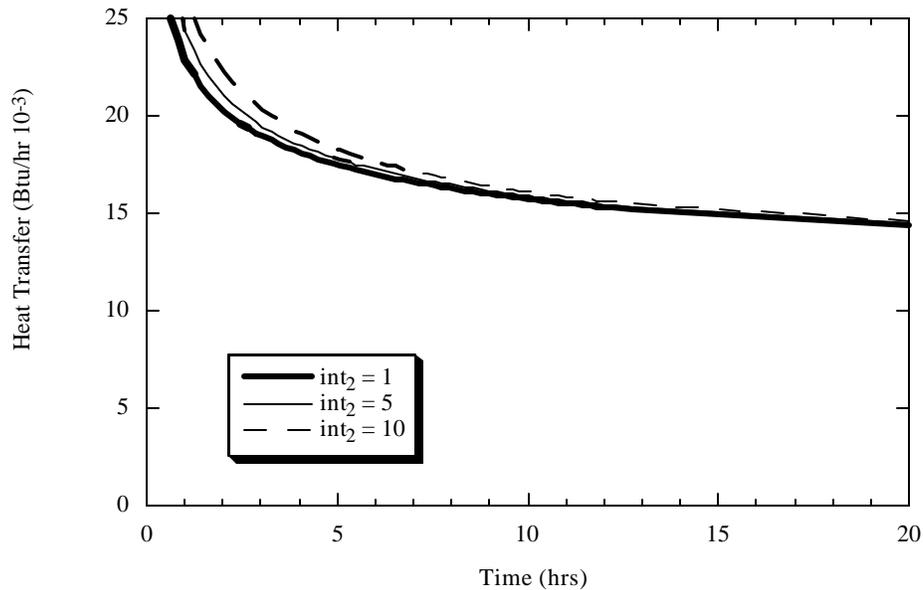
Figure 4.10 Effect of splitting the system into t_1 and t_2{ TC "Figure 4.10 Effect of splitting the system into t_1 and t_2" \l 8 }

Although splitting the system decreased the simulation time significantly the program was still dependent on the properties of the grout material. A large thermal conductivity of the grout would result in time steps much smaller than 15 seconds. Many models neglect the capacitance of the grout material, and it was decided to investigate the accuracy of this assumption. Figure 4.11 shows curves for the solutions to four different simulations. Each simulation was run with the same

conditions, except the soil capacitance was altered. The numbers in the legend of Figure 4.11 refer to the amount of soil that did not have capacitance (0.00 ft means all the capacitance in the soil was included). It can be seen that neglecting the capacitance near the tubes causes the model to slightly underpredict the heat transfer for the first few hours of simulation. Since the grout material is a small portion of the entire system it was concluded that the capacitance of this material could be neglected.
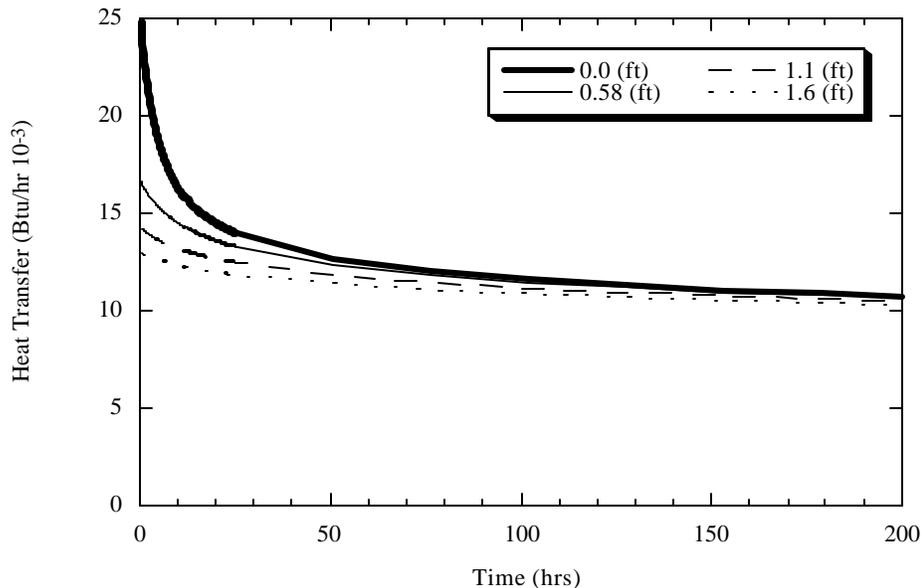
Figure 4.11 Neglecting the ground capacitance near the tube walls{ TC "Figure 4.11 Neglecting the ground capacitance near the tube walls" \l 8 }

The model was next compared to a finite difference model of a single circular tube. The cylindrical grid shown in Figure 4.2 was altered to allow calculation of the heat transfer from a single tube heat exchanger. The node that was originally used as the return tube of the u-tube was reprogrammed to be ground material. Therefore, the model would represent a single tube slightly off center (4" from the center of the grid). The model was run with the conditions shown in Table 4.1. For

comparison, the steady state solution was also calculated using heat exchanger effectiveness NTU relations. A large value of fluid specific heat (10^{12}) was used to maintain a constant fluid temperature at each depth. Although a farfield temperature of 0 F is not realistic it was used to provide large temperature gradients so the effect of various changes to the model could be easily observed.

Table 4.1 Single Tube Finite Difference Comparison{ TC "Table 4.1 Single Tube Finite Difference Comparison" \l 9 }

Farfield Radius	10 ft	c_s	0.2 Btu/lbm-F
Inner Tube Diameter	1.5 in	k_t	0.226 Btu/hr-ft-F
Bore hole Depth	200 ft	c_f	10^{12} Btu/lbm-F
k_s	0.75 Btu/hr-ft-F	Fluid Inlet Temp.	100 F
k_g	0.75 Btu/hr-ft-F	Flow Rate	3 gpm
ρ_s	131 lb/ft^3	Farfield Temperature	0 F

The results of these simulations are shown in Figure 4.12 along with the steady state solution obtained analytically. The heat transfer from the different models is plotted as a function of time. It can be seen that the results for the non-circular tube (labeled as geometry=1) are below the circular solution by about 5%. The reason for this difference was determined to be due to the different shapes of the tubes. The thermal resistances near the non circular tube are significantly different than those near the circular tube. Although the non circular model provides relatively good agreement an empirically determined geometry factor is used to improve the model.

60
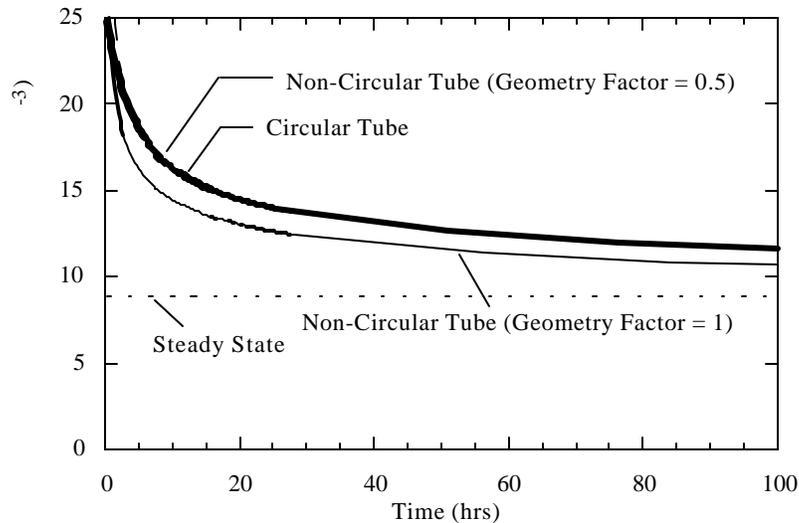
Figure 4.12 Comparison between a circular and non-circular tube{ TC "Figure 4.12 Comparison between a circular and non-circular tube" \l 8 }

The soil and grout components of the fluid to ground resistances (equations 6c, 7c, and 8c) are

multiplied by a geometry factor. Only the soil and grout components were altered because their

thermal resistance is less likely to vary than the convection resistance (equations 6a, 7a, and 8a) or

the tube resistance (equations 6b, 7b, and 8b). This is important because, in order to be useful, the

geometry factor needs to be independent of the configuration and material properties of the system.

The geometry factor was determined by matching the heat transfer from the non circular tube with

that of a circular tube under the conditions listed in Table 4.1. The results of the simulations showed

that the local geometry should be a value of 0.5. This simulation for a geometry factor of 0.5 is also

shown in Figure 4.12, and is virtually identical to that for a circular tube.

The steady state temperature contours of the simulations for the circular and the improved non

circular tubes were also compared, and are shown in Figure 4.13. There is excellent agreement

between the two models.
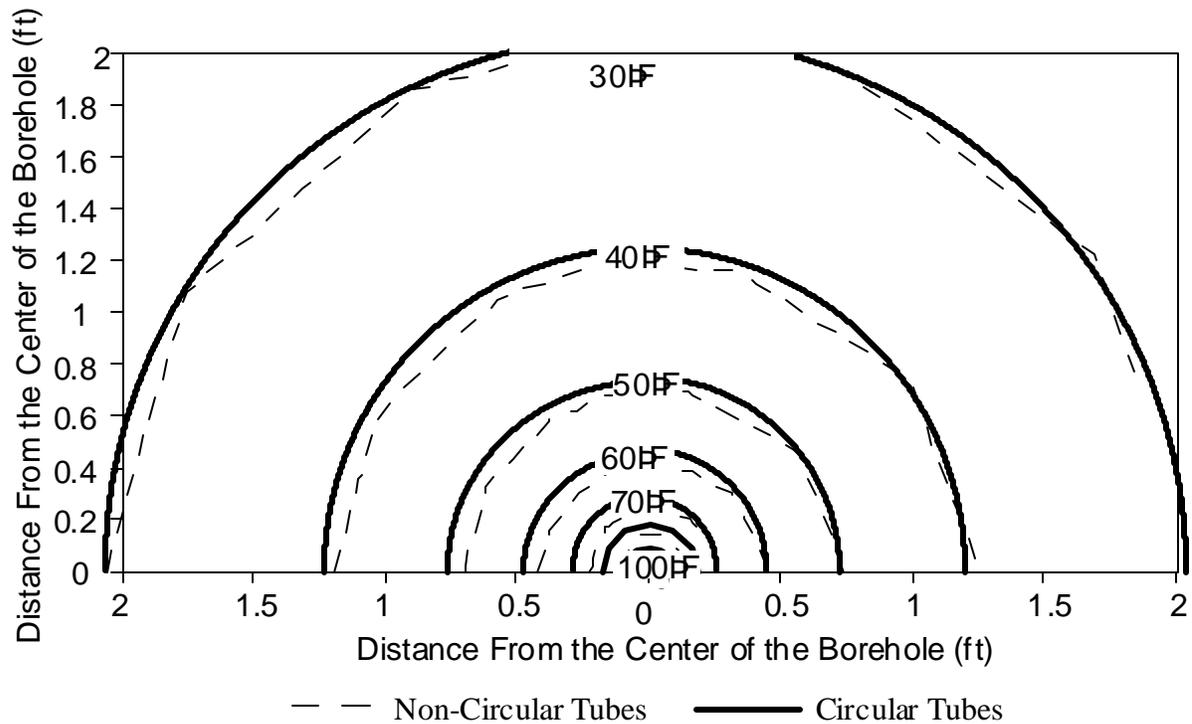
Figure 4.13 Comparison between one circular and one non-circular tube{ TC "Figure 4.13 Comparison between one circular and one non-circular tube" \l 8 }

It is possible that the geometry factor will be dependent on four variables: R, $\Delta\theta$, k_s, and k_g. After an appropriate geometry factor was determined for the initial case, several more simulations were run to estimate its dependency on these factors. It was found that the geometry factor was mainly dependent on the properties of the material, and showed little effect from the geometry of the system.

A second method for estimating the effect of the u-tube system on the geometry factor is to calculate it's largest and smallest values under reasonable conditions. The average of these values is used for several simulations and the error is determined. The diameter of u-tubes can range from 1/4" to 4", but the most common u-tube has a 1" diameter. Therefore, a good representation of the reasonable range of conditions was a minimum tube diameter of 1/2" and a maximum diameter of 2". The values for   R, $\Delta\theta$, k_s, k_g used to obtain the largest and smallest values for the geometry

factor are shown in Table 4.2. A geometry factor of 0.38 was determined and this results in an

accuracy within 3% for other simulations. Without the geometry factor the accuracy was within 8%.

Table 4.2. Values used to determine the largest and smallest geometry factor{ TC "Table 4.2.

Values used to determine the largest and smallest geometry factor" \l 9 }

Parameter	R	$\Delta\theta$	k_S	k_g
Small Value	0.033 ft	0.13 radians	0.2 Btu/hr-ft-F	0.2 Btu/hr-ft-F
Large Value	0.13 ft	0.52 radians	1.5 Btu/hr-ft-F	1.5 Btu/hr-ft-F

The final test was a u-tube simulation with realistic fluid specific heat. The model with non circular

tubes was run to steady state under the conditions shown in Table 4.3.

Table 4.3 Parameters for finite element and finite difference comparison{ TC "Table 4.3

Parameters for finite element and finite difference comparison" \l 9 }

Farfield Radius	8 ft	c_s	0.2 Btu/lbm-F
Inner Tube Diameter	1.3 in	k_t	0.75 Btu/hr-ft-F
Tube Separation	8 in	c_f	0.9981 Btu/lbm-F
Bore hole Depth	200 ft	Fluid Inlet Temp.	100 F
k_s	0.75 Btu/hr-ft-F	Flow Rate	3 gpm
k_g	0.75 Btu/hr-ft-F	Farfield Temperature	0 F
ρ_s	131 lb/ft^3		

The fluid temperatures at the surface (z=1) were then used as constant temperature sources in a

finite element heat transfer program (FEHT). The purpose of this comparison was to make sure the

model accurately simulated the two dimensional heat transfer of two tubes in an infinite medium.

The temperature contours of the ground surrounding the u-tube were compared and showed good agreement. Figure 4.14 shows a plot of the temperature contours for the two cases.
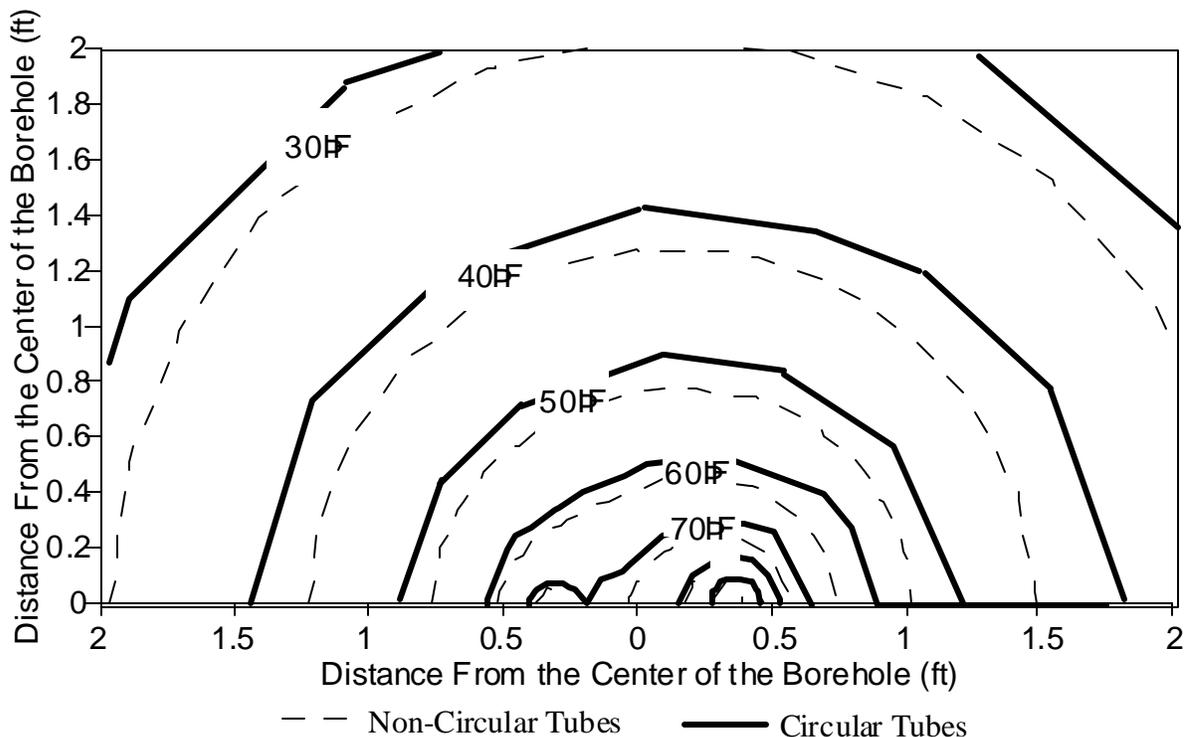
Figure 4.14 Contours for non circular and circular tubes (realistic fluid specific heat){ TC "Figure 4.14 Contours for non circular and circular tubes (realistic fluid specific heat)" \l 8 }

4.4 Comparison with the Fort Polk Installation{ TC "4.4 Comparison with the Fort Polk Installation" \l 2 }

The initial reason for creating a model to simulate u-tube heat exchangers was to estimate the performance of GSHPs that were installed in Fort Polk, Louisiana. Fort Polk is a military base that retrofitted 4005 residential housing units with geothermal heat pumps. There are 66 individual housing units (single, duplex ,fourplex,etc.). All of the units were designed with vertical closed loop u-tube heat exchangers. The majority of the bore holes were drilled to a depth of approximately 200 ft. The units were usually placed two u-tubes in series, and seventy five percent of the units

were equipped with desuperheaters. Some of theses units were extensively monitored so data was

available for comparison.

The model created at the University of Lund, described in Chapter 1, was used to estimate the

performance of the u-tube heat exchangers installed in Fort Polk. The model provided fairly

accurate simulations of the heat pumps operating. A comparison between these two models based

on the installation at Fort Polk is intended to provide further insight into the usefulness of the model.

The comparison is run based on the specifications given in Table 4.4 which are the basic conditions

at Fort Polk.

Table 4.4 Parameters used to compare with the University of Lund model.{ TC "Table 4.4

Parameters used to compare with the University of Lund model." \l 9 }

T_{inlet}	100 F	t_{shift}	32 days
T_{mean}	69 F	c_f	0.998 Btu/lbm-F
T_{amp}	17 F	mu_f	2.07 lbm/ft-hr
k_s	1.4 Btu/hr-ft-F	k_f	0.353 Btu/hr-ft-F
c_s	0.2 Btu/lbm-F	U-Tube Spacing	3 inches
ρ_s	200 lbm/ft^3	Bore hole Depth	200 feet
k_g	1.4 Btu/hr-ft-F	Inner Tube Diameter	1.1 inches
k_t	0.2427 Btu/hr-ft-F	Outer Tube Diameter	1.3 inches
ρ_f	62.4 lbm/ft^3	Flow Rate	1.5 gpm
Geometry Factor	0.38		

The Lund model determines the total heat transfer from the u-tubes with two calculations. The first

calculation determines the heat transfer to the ground near the tubes, and is referred to as the "local"

process. The local problem is solved in two steps. The first step determines an effective fluid to

bore hole resistance and then applies heat flux step pulses to obtain an average bore hole temperature. A finite difference mesh is then employed beyond the bore hole wall to calculate the heat transfer to the local storage volume. The volume of ground considered in this local process is specified by the user. The model's second calculation determines the heat rejected from the local ground volume to the ground surrounding it. The volume of the ground surrounding the local ground volume is determined by the model based on the ground heat exchanger configuration and the length of the simulation. The far edge of this volume is considered an adiabatic boundary.

The Lund model enables the user to insulate the top and side of the local storage volume, and this feature was utilized for the first comparison between the two models. The size of the local storage volumes for both the Lund and the finite difference models were set to equal 31,673 ft^3 (cylinder with a radius of approximately 7.1 feet and a height of 200 feet) and the edges of both boundaries insulated. The models were run with the conditions shown in Table 4, and comparisons were made between the fluid exit temperature and the heat transferred to the ground. After a one year simulation the fluid exit temperature of the Lund model was 98.2 F and the finite difference model provided a temperature of 98.6 F. The finite difference model showed that the u-tubes transferred 33.7 MBtu of heat to the ground and the Lund model provided slightly different answers depending on how the calculation was performed. If the heat transfer was determined by the temperature rise in the ground, the model showed that 32.8 MBtu were transferred to the ground. If an energy balance on the fluid nodes was integrated with time, the model provided an answer of 35.1 MBtu. These values are different from the finite difference model by 2.6% and 4.1% respectively. The main reason for this difference is the heat loss from the local storage volume in the Lund model. The Lund model does not include the option for insulation along the bottom of the storage volume. This amount of heat transfer was usually zero, but occasionally large values of heat transfer would appear. Even with this difference the agreement between the two models is good. The second comparison between the two models again used the conditions shown in Table 4, but this time the insulation along the side of both storage volumes was removed. The finite difference

model employed a constant temperature boundary at 20 feet from the u-tubes, while the Lund

model employed the adiabatic boundary as described previously. The results of this simulation also

provided nearly identical results with the Lund model and finite difference model showing fluid exit

temperatures of 90.8℉ and 90.5℉ respectively.

Chapter 5

Recommendations and Conclusions{ TC "Recommendations and Conclusions" \l 1 }

5.1 Conclusions{ TC "5.1 Conclusions" \l 2 }

The major accomplishment of this project is the design of a new model to simulate the unique heat transfer conditions present in a u-tube heat exchanger. The model employs a finite difference approach that is fundamental and provides the flexibility in modeling situations for which the line source approach can only approximate.

The non circular geometry provides solutions that are accurate to within 8%, but an empirically determined geometry factor applied to the resistances between the fluid and the ground can reduce the difference to 3%. The model has been validated for simple conditions, and a comparison with the model created at the University of Lund showed the model provides similar results.

Currently, the model is limited to simple conditions, but it is readily extended to include conditions that will make the model more useful to industry.

5.2 Recommendations for Future Work{ TC "5.2 Recommendations for Future Work" \l 2 }

The model presented in this thesis provides a rapid alternative method for simulating ground coupled u-tube heat exchangers. The model has proven effective for many conditions, but improvements on

several limitations would increase the model's usefulness for industry. These are summarized in the following paragraphs.

Although the center to center distance between the two legs of a u-tube should be maximized to enhance the heat exchanger performance, the importance of this is considered secondary in industry. The reason for this is the difficulty of installation, and in reality the u-tube legs are often taped together to make the installation of the u-tubes easier. Therefore, a model should be able to model a system where the u-tube spacing is near zero. Currently, the spacing between the tubes in the finite difference model is limited to a minimum of three times the radius of the u-tube. This results in an angular distance between nodes of approximately 0.6 radians. The reason for this limitation is that the angular distance between nodes is dependent on the u-tube diameter and spacing. The model requires at least seven circumferential nodes to work properly. If there are fewer than seven nodes the equations need to be altered because some nodes are adjacent to the fluid nodes and the nodes at $j = n+1$.

Currently, the model places the two legs of the u-tube at the edge of the bore holes, but since tubes are often taped together this is not realistic. An improved model would make the nodes at $i=m+1$ grout material and allow the radial spacing at this node to vary, similar to the angular distance $\Delta\theta_{mid}$ described previously. This would enable the model to make the bore hole larger than the tube spacing. The model should include the possibility of this distance equaling approximately zero, which is the current program geometry. Since the capacitance of the grout material is neglected the critical time step is not a limiting factor.

Another limitation comes from the angular length of $\Delta\theta_{mid}$ which is now determined from equation 4.7. The model will work slowly in TRNSYS if the ground critical time step becomes far less than the fluid critical time step. For some conditions the value for $\Delta\theta_{mid}$ is very small. The smaller $\Delta\theta_{mid}$, the slower the program runs, and if this value should become zero the program will not function properly.

As described previously, a local geometry factor was included to improve the accuracy of the model. This factor was determined to be dependent on the material properties of the ground and grout, but not as much on the geometry of the system. If the geometry factor is to be used in the future its dependency on the material properties of the system should be investigated more thoroughly.

Also discussed earlier was the use of an escalation factor on the radial distance between nodes. This factor is currently found by using a program in EES. This program is given in Appendix B along with a brief description of how to use it. It would be better to incorporate this directly into the program, but because of the unknown position and size of the tubes this is not straight forward.

Currently, the model uses 17 nodes in the radial direction. This includes 10 nodes with a radial spacing equal to   R as specified by equation 4.11. This was done to enable the model to simulate a variety of u-tube separation distances. However, it may not be necessary to use this many nodes in the radial direction, and decreasing this value could significantly increase the speed of the model.

The current model represents a very simplified case of vertical u-tube heat exchangers. In order to be more useful, more complicated system should be available. Commonly, u-tubes in different bore holes are placed closer together than the previously defined farfield radius. This would result in thermal interference between the bore holes, but in this model that is not taken into consideration. Also, because of high installation costs several u-tubes are often placed in the same bore hole. However, this is option not available. Finally, the model could be extended to include the effects of soil property variation with depth and moisture migration.

Appendix A

Additions to the TRNSYS type.

Several additions to the model have been completed, and a brief explanation of these changes is given in this appendix.

The node at $i = m + 1$ was altered to account for grout material beyond the edge of the u-tube. This enables the model to make the bore hole larger than the tube spacing as discussed in chapter 5 of the thesis.

The user is also able to specify four additional variables: the number of nodes used in the vertical and horizontal direction, the farfield radius, and the boundary conditions at the farfield radius (either adiabatic or constant temperature).

Finally, the program no longer uses the radial spacing described in chapter 4. Now, the program begins to increase the radial spacing, R, at the first ground node ($i = m+2$), and continues to increase R to the farfield radius specified. Equation A.1 gives the relationship between the nodes.

$$\frac{r(i+1)}{r(i)} = \left(\frac{r_{ff}}{r(m+2)}\right)^{\frac{1}{nodes}}$$ (A.1)

This is a simple relationship between the nodes that eliminates the need for the EES program.

Appendix B

TRNSYS code for the ground coupled heat exchanger model.

      SUBROUTINE TYPE75(TIME,XIN,OUT,T,DTDT,PAR,INFO,ICNTRL,*)

c
****
****
c
****
****
c This is an explicit finite difference method for a ground source heat exchanger.
c The tubes are approximated as non-circular tubes in a radial grid. The grout and
c tubes capacitances are neglected.
c
c The definition of all parameters used in the program are listed below:
c
c A: Escalation factor for the radial distance between nodes in mesh
c   This factor is started at the 11th node, and the number is found
c   from an EES program.
c
c atube1:   Heat transfer area of the tube to ground for node told(m-1,c,z) and told(m,c,z) (ft^3).
c
c atube2:   Heat transfer area of the tube to ground for node told(m-1,c,z) and told(m,c,z)
(ft^3).
c
c atube3:   Heat transfer area of the tube to ground for node told(m-1,c,z) and told(m,c,z)
(ft^3).
c
c c: The current circumferential position of node in calculation.
c
c check: Constant used to determine if u-tube geometry is possible.
c
c cpfluid: Specific heat of the heat exchanger fluid used in u-tube (Btu/lbm-F).
c
c cpground: Specific Heat of the ground (Btu/lbm-F)
c
c deltar: Radial distance between the radial midpoints of nodes (ft).
c
c deltatheta:Circumferential distance between nodes (radians).
c
c deltathetamid: Circumferentail distance between node at theta = 90 and nodes adjacent to it
(radians).
c
c deltat1:   Time step for the fluid nodes (hr).
c
c deltat2:   Time step for the ground/grout nodes (hr: integer multiple of deltat1).

```
c
c deltaz:        Axial distance (depth) between nodes (ft).
c
c depth:               The depth of the borehole for the u-tube (ft).
c
c dtubei:        Inside diameter of the u-tube (ft).
c
c dtubeo:        Outside diameter of the u-tube (ft).
c
c fluidcap1:     Constant containing values for finite difference equations for fluid (dimensionless).
c
c fluidcap2: Constant containing values for finite difference equations for fluid (hr-F/Btu).
c
c fp:            Constant used to calculate the fluid to tube heat transfer coeficient.
c
c galfluid:      Flowrate for heat exchanger fluid (gallons/minute).
c
c grndcap:       Array containing values for finite difference equations for ground (hr-F/Btu).
c
c grndcapmid:Array containing values for finite difference equations for ground at theta = 90 (hr-
F/Btu).
c
c htube1:        Heat transfer coefficient for fluid to tube walls (Btu/hr-ft^2-F).
c
c kfluid:        Conductivity of the heat exchanger fluid used in u-tube (Btu/hr-ft-F).
c
c kgrnd:               Conductivity of the ground (Btu/hr-ft-F)
c
c kgrout:        Conductivity of the grout material in the borehole (Btu/hr-ft-F)
c
c ktube:     Conductivity of the u-tube walls (Btu/hr-ft-F)
c
c ltube:               Thickness of u-tube wall (ft).
c
c m:                   The radial position of the tube node for the prescribed geometry.
c
c mufluid:       Dynamic Viscosity of heat exchanger fluid used in u-tube (lbf-s/ft^2).
c
c n:                   The number of nodes in circumferential direction.  Starting with the
c              node along the adiabatic boundary (theta=0 degrees) up to the node adjacent
c              to the node at theta = 90 degrees.
c
c nr:            The total number of radial nodes in the finite difference mesh (constant).
c
```

74

c ns: The number of times the loop nt is run.
c   (nt*ns*nv*deltat1 = total time of simulation)
c
c nt: The number of calculations before the answer is saved.
c
c nu: Nusselt number used to calculate fluid to tube heat transfer coeficient.
c
c nv: The number of calculations performed by on the fluid nodes before the
c   ground/grout node temperatures are updated.
c
c nz: The total number of axial nodes in the finite difference mesh (constant).
c
c pi: Mathematical constant of pi.
c
c pr: Prandtl number used to calculate fluid to tube heat transfer coeficient.
c
c qout.dat: Contains data for the heat transfer output for exchanger (Btu/hr).
c
c qoutdown: The heat transfer form the tube to the ground travelling down the tube (Btu/hr).
c
c qouttotal: Total amount of heat transfer from the tubes to the ground.
c
c qoutup: The heat transfer from the tube to the ground traveling up the tube (Btu/hr).
c
c r: The current radial postion of node in calculation.
c
c rad: The radial position of the nodes in the mesh (ft).
c
c radmid: The radial position of the midpoint between nodes in the mesh (ft).
c
c radmidchk: Constant used to determine the position of the tube node.
c
c radtubei: Inner radius of the u-tube (ft).
c
c radtubeo: Outer radius of the u-tube (ft).
c
c re: Reynolds number used to calculate fluid to tube heat transfer coeficient.
c
c rescirc: Circumferential resistance between nodes in mesh.
c
c rescircmid:Circumferential resistance between nodes in mesh at theta = 90.
c
c resrad: Radial resistance between nodes in mesh.
c

c resradmid: Radial resistance between nodes in mesh at theta = 90.

c

c restubegrnd1: Resistance of the tube to ground for node told(m-1,c,z) and told(m,c,z) (hr-F/Btu).

c

c restubegrnd2: Resistance of the tube to ground for node told(m+1,c,z) and told(m,c,z) (hr-F/Btu).

c

c restubegrnd3: Resistance of the tube to ground for node told(m,c+1,z) and told(m,c,z) (hr-F/Btu).

c

c rhofluid: Density of the heat exchanger fluid used in u-tube (lbm/ft^3).

c

c rhogrnd: Density of the ground (lbm/ft^3)

c

c s: The current calculation in the ns loop.

c

c shape: Shape factor used to increase local heat transfer to ground.

c

c u: The current calculation in the nt loop.

c

c tcrit1:   Critical time step for the ground not at theta = 90 (hr).

c

c tcrit2:   Critical time step for the ground at theta = 90 (hr).

c

c tcrit3:   Critical time step for the ground next to the tube (hr).

c

c tcrit4:   Critical time step for the fluid nodes (hr)

c

c temp.dat: Contains temperature profiles for u-tube exchanger (F).

c

c tff: The temperature of undisturbed ground at the location of heat exchanger installation (F).

c

c tinlet: Temperature of the fluid entering the ground coupled heat exchanger (F).

c

c tnew: The temperature of the node (ground/grout/fluid) calculated at current time step (F).

c

c told: The temperature of the node (ground/grout/fluid) calculated at previous time step (F).

c

c tubesep: Center to center separation between tubes in u-tube (ft).

c

c v: The current calculation in the nv loop.

c

```
c  vfluid:      Volume of fluid contained in the tube (ft^3).
c
c  vol:         Volume of ground/grout represented by the node at that position.
c
c  volmid:      Volume of ground/grout represented by node at theta = 90 (ft^3).
c
c  wmflow:      Flow rate of heat exchanger fluid through tube (lbm/hr).
c
c  wspeed:      Velocity of heat exchanger fluid through tube (ft/hr).
c
c  x:                   Used to change cylindrical coordinates to cartesian coordiantes.
c
c  y:                   Used to change cylindrical coordinates to cartesian coordiantes.
c
c  z:                   The current axial position of the node in calculation.
c
c
************************************************************************
********
c
************************************************************************
********

c     IMPLICIT NONE

C
************************************************************************
********
C  VARIABLES FOR TRNSYS
C
************************************************************************
********
      INTEGER ICNTRL,INFO
      INTEGER N_IN,N_OUT,N_PARM
      INTEGER NSTORE,IAV,NUMSTR
      INTEGER ISTORE
      INTEGER IWARN
      INTEGER LUR,LUW,IFORM,LUK

      PARAMETER (N_IN=2,N_OUT=4,N_PARM=22)
      INCLUDE '\trnwin\kernal\param.inc'

      REAL TIME,T,DTDT,PAR,S
      REAL TIMEO,TFINAL,DELT
```

```
      REAL*8 XIN,OUT

      CHARACTER*3 YCHECK(N_IN),OCHECK(N_OUT)
      DIMENSION XIN(N_IN),PAR(N_PARM),OUT(N_OUT),INFO(15)
      DIMENSION S(NUMSTR)

      COMMON /LUNITS/ LUR,LUW,IFORM,LUK
      COMMON /STORE/ NSTORE,IAV,S
      COMMON /SIM/ TIMEO,TFINAL,DELT,IWARN

      DATA YCHECK /'TE2','VF7'/
      DATA OCHECK /'TE2','VF7','TE2','TE2'/

C
***********************************************************************
********
C  Variables used for the subroutine
C
***********************************************************************
********

c  Variables used in do loops
      integer   r,z,n,c,m,tms2,tms3,tms4,
     .          radnodes,boundary

c  Variables for number of nodes
      integer   nr,nz,nmax,times1,times2,times3,times4,nfluid2,nfluid3,
     .          nzmax,nrmax

c  Set the number of nodes nx,ny,nzmax, and nt
c  Note:  nz must be equal to the number of nodes in the z direction+1 because z=1 is inlet fluid
temp
c       nr is greater than the nodes in the actual ground storage volume.
      parameter (nrmax=20,nzmax=30,nmax=20,nfluid2=30,nfluid3=1000)

c  Define Array Variables
      real*8 qoutup(nzmax),qoutdown(nzmax),rad(nrmax),resrad(nrmax),
     .       rescirc(nrmax),vol(nrmax),qouttotal(0:nzmax)
     .       ,grndcap(nrmax),resradmid(nrmax),vert(nzmax),
     .       rescircmid(nrmax),volmid(nrmax),grndcapmid(nrmax),
     .       deltar(nrmax),radmid(nrmax)

      real*8 told(nrmax,nmax,0:nzmax),tnew(nrmax,nmax,0:nzmax)
     .       ,tff(nzmax),tfluid(nfluid2,nfluid3),qout(nfluid2,nfluid3)
```

```
c  Variables used for the distance between nodes.
      real*8 deltaz,deltatheta,deltathetamid

c  Variables for thermal properties of ground and fluid.
      real*8 rhogrnd,cpground,cpfluid,kgrnd,kgrout,rhofluid,galfluid,
     .      fluidcap1,fluidcap2,kfluid,mufluid,alphagrnd,tfluidavg,
     .      qoutavg

c  Variables used for initial temperature and constants.
      real*8 pi,tinlet,shape,A,dr,radmidchk,depth,re
     .      ,fp,pr,nu,nu1,nu2,numer,rhalf,tmean,tamp,tshift

c  Variables used to determine the critical time step.
      real*8 tcrit1,tcrit2,tcrit4,tcrit5,deltat1,deltat2

c  Variables used for ground to tube resistance calculations.
      real*8 restubegrnd1,restubegrnd2,restubegrnd3,htube1,
     .      wspeed,wmflow,vfluid,ktube,radtubei,radtubeo,dtubeo,
     .      dtubei,tubesep,ltube,atube1,atube2,atube3,check,
     .      htube2,restubegrnd4,restubegrnd5,restubegrnd6,tgravg,
     .      vt,grvol,radbore,radff,geofactor

c
********************************************************************************
********
C  INPUTS TO THE PROBLEM
c
********************************************************************************
********
      tinlet = XIN(1)
      galfluid = XIN(2)

C
********************************************************************************
********
C  CHECK NUMBER OF PARAMETERS,INPUTS ON FIRST CALL
C
********************************************************************************
********
      IF(INFO(7).LE.-1) THEN

C  SET INITIAL CONDITIONS
      INFO(6) = N_OUT
```

```
      INFO(9) = 1
      INFO(10) = 15000
      CALL TYPECK(1,INFO,N_IN,N_PARM,0)
      CALL RCHECK(INFO,YCHECK,OCHECK)



C
***********************************************************************
********
C  Set the parameters on the first call.
C
***********************************************************************
********
      tubesep = DBLE(PAR(1))
      dtubei = DBLE(PAR(2))
      dtubeo = DBLE(PAR(3))
      depth = DBLE(PAR(4))
      kgrnd = DBLE(PAR(5))
      rhogrnd = DBLE(PAR(6))
      cpground = DBLE(PAR(7))
      kgrout = DBLE(PAR(8))
      ktube = DBLE(PAR(9))
      rhofluid = DBLE(PAR(10))
      cpfluid = DBLE(PAR(11))
      kfluid = DBLE(PAR(12))
      mufluid = DBLE(PAR(13))
      tmean = DBLE(PAR(14))
      tamp = DBLE(PAR(15))
      tshift = DBLE(PAR(16))
      radbore = DBLE(PAR(17))
      radff = DBLE(PAR(18))
      nz = INT(PAR(19)+0.1d0)
      radnodes = INT(PAR(20)+0.1d0)
      boundary = INT(PAR(21)+0.1d0)
      geofactor = DBLE(PAR(22))



c
***********************************************************************
********
c     This section of the program determines the grid geometry
c     (size and nodal distances,etc.) and the resistnaces between nodes.
c
***********************************************************************
```

****

```fortran
c  ----------------------------------------------------------------------------------------
c  Set constant pi
c  ----------------------------------------------------------------------------------------
      pi = 4d0*atan(1d0)


c
********************************************************************************
********
c  Input initial conditions of system
c
********************************************************************************
********
      ISTORE = INFO(10)


c  ----------------------------------------------------------------------------------------
c  Input Farfield Temperature Relationship
c  ----------------------------------------------------------------------------------------


c  ----------------------------------------------------------------------------------------
c  Recalculate some tube geometry.
c  ----------------------------------------------------------------------------------------
      radtubeo = dtubeo/2d0
      radtubei = dtubei/2d0
      ltube = radtubeo-radtubei


c  ----------------------------------------------------------------------------------------
c  Determine the proper radial distance between nodes (from EES curve fit).
c  ----------------------------------------------------------------------------------------
      deltar(1) = 0.7853997d0*dtubei
      deltar(2) = 0.7853997d0*dtubei


c  ----------------------------------------------------------------------------------------
c  Set node spacing for deltaz the vertical distance between nodes.
c  ----------------------------------------------------------------------------------------
      deltaz = depth/(nz)

      if (nz.gt.nzmax) then
       WRITE (LUW,*) "Decrease the number of vertical nodes to",nzmax
      .,"or increase the parameters nzmax and nfluid2 to", nz
       CALL MYSTOP(1001)
       RETURN 1
       end if
```

```
c  ------------------------------------------------------------------------------------------
c  Determine the position of the tube nodes.
c  m is the number that is the position of the tube nodes.
c  m can not be greater than 10.
c  If m=1 then there is not enough room to place the u-tubes in the borehole specified.
c  If m=2 then the tubes are adjacent to the center node which will not work, and then
c       the length of deltar(2) is reduced until m=3.
c  ------------------------------------------------------------------------------------------
      dr = deltar(1)
  10  radmidchk = tubesep/2d0 - deltar(1)/2d0
      m = 1
      do while (radmidchk.gt.0d0)
       m=m+1
       radmidchk = radmidchk - dr
      end do


c  ------------------------------------------------------------------------------------------
c  Make sure the geometry of the system is possible.
c  ------------------------------------------------------------------------------------------
      if (m.le.1) Then
       WRITE (LUW,*) "This system is not possible."
       WRITE (LUW,*) "The tube is too big for the borehole."
       CALL MYSTOP(1001)
       RETURN 1
      end if


c  ------------------------------------------------------------------------------------------
c  Make sure there is a node between the tubes and the center node.
c  ------------------------------------------------------------------------------------------
      If (m.le.2) Then
       deltar(2)=dr/2d0
       dr=dr/2d0
       goto 10
      end if


c  ------------------------------------------------------------------------------------------
c  Set deltar for the borehole nodes (deltar(1) holds the value only it is not used).
c  ------------------------------------------------------------------------------------------
      do r = 3,m
       deltar(r) = deltar(1)
      end do


c  ------------------------------------------------------------------------------------------
```

82
```
c  Set the position of the tube nodes (r=m).
c  -----------------------------------------------------------------------------------
      rad(m) = tubesep/2d0


c  -----------------------------------------------------------------------------------
c  Set the position of the midpoints between nodes (r=m).
c  -----------------------------------------------------------------------------------
      radmid(m) = rad(m) + deltar(m)/2d0


c  -----------------------------------------------------------------------------------
c  Set the position of the midpoints between nodes at the borehole edge (r=m+1).
c  -----------------------------------------------------------------------------------
      radmid(m+1) = radbore


c  -----------------------------------------------------------------------------------
c  Set the position of the midpoints between nodes at the borehole edge (r=m+1).
c  -----------------------------------------------------------------------------------
      rad(m+1) = (radmid(m+1)+(radmid(m)+ltube))/2d0


c  -----------------------------------------------------------------------------------
c  Set the position of the midpoints between nodes (r<m).
c  -----------------------------------------------------------------------------------
      If (radmid(m+1).le.radmid(m)) Then
        WRITE (LUW,*) "This system is not possible."
        WRITE (LUW,*) "The tube is too big for the borehole."
        CALL MYSTOP(1001)
        RETURN 1
      end if


c  -----------------------------------------------------------------------------------
c  Set the position of the midpoints between nodes (r<m).
c  -----------------------------------------------------------------------------------
      do r = m-1,1,-1
        radmid(r)=radmid(r+1)-deltar(r+1)
      end do


c  -----------------------------------------------------------------------------------
c  Determine the radial position of the nodes (r<m, r=m is tube node).
c  -----------------------------------------------------------------------------------
      rad(m-1)=((radmid(m-1)-ltube)+radmid(m-2))/2

      do r=2,m-2,1
        rad(r) = (radmid(r)+radmid(r-1))/2d0
      end do
```

```
c  --------------------------------------------------------------------------------
c  Set the radial position for the ground nodes for constant temperature boundary.
c  --------------------------------------------------------------------------------
      If (boundary.ne.1) then
      nr = m+1+radnodes
      if (nr.gt.nrmax) then
      WRITE (LUW,*) "Decrease the number of radial nodes to",nrmax-(m+1)
      WRITE (LUW,*) "or increase the parameter nrmax to", nr
      CALL MYSTOP(1001)
      RETURN 1
      end if
      rad(nr) = radff
      A = (rad(nr)/radmid(m+1))**(1d0/radnodes)
      rad(m+2) = A*radmid(m+1)
      do r = m+2,nr-2,1
        rad(r+1) = A*rad(r)
      end do
c  Set the position of the midpoints between nodes (m+1<r).
      do r = m+2,nr-1,1
        radmid(r) = (rad(r+1)+rad(r))/2d0
      end do
c  Reposition the nodes to be in the middle of area
      do r = m+2,nr-1,1
        rad(r) = (radmid(r)+radmid(r-1))/2d0
      end do
      end if


c  --------------------------------------------------------------------------------
c  Set the radial position for the ground nodes for adiabatic boundary.
c  --------------------------------------------------------------------------------
      If (boundary.eq.1) then
      nr = m+2+radnodes
      if (nr.gt.nrmax) then
      WRITE (LUW,*) "Decrease the number of radial nodes to",nrmax-(m+1)
      WRITE (LUW,*) "or increase the parameter nrmax to", nr
      CALL MYSTOP(1001)
      RETURN 1
      end if
      radmid(nr-1) = radff
      A = (radmid(nr-1)/radmid(m+1))**(1d0/radnodes)
      do r = m+1,nr-3,1
        radmid(r+1) = A*radmid(r)
      end do
```

```
c  Set the position of the midpoints between nodes (m+1<r).
      do r = m+2,nr-1,1
         rad(r) = (radmid(r)+radmid(r-1))/2d0
      end do
      rad(nr) = rad(nr-1)+1d0
      end if


c  ------------------------------------------------------------------------------------------
c  Another check to see if the system geometry is possible.
c  If this is true then the tubes overlap due to the thickness of the tubes.
c  ------------------------------------------------------------------------------------------
      If (radmid(m-1).le.0d0) Then
        WRITE (LUW,*) "This system is not possible."
        WRITE (LUW,*) "The tube is too thick for the borehole spacing."
        CALL MYSTOP(1001)
        RETURN 1
      end if


c  ------------------------------------------------------------------------------------------
c  Angular separation between all nodes except at c=n+1.
c  ------------------------------------------------------------------------------------------
      deltatheta = (pi*dtubei-2*deltar(1))/(radmid(m) + radmid(m-1))


c  ------------------------------------------------------------------------------------------
c  Angular distance occupied by the node at theta = 90 degrees.
c  n = the number of nodal points from the adiabatic boundary to the node adjacent to
c      the node at 90 degrees.
c  2*n+1 would be equal to the total number of nodes.
c  ------------------------------------------------------------------------------------------
      n = 1
      check = (pi/2d0)-(dfloat(n)-0.5d0)*deltatheta
      do while (check.gt.deltatheta)
        n=n+1
        check = (pi/2d0)-(dfloat(n)-0.5d0)*deltatheta
      end do

      deltathetamid = 2d0*((pi/2d0)-(dfloat(n)-0.5d0)*deltatheta)


c  ------------------------------------------------------------------------------------------
c  Another check to see if the program will model the system correctly.
c  If this is true then the geometry is possible, but this program will not accurately model it.
c  ------------------------------------------------------------------------------------------
      If (n.lt.3) Then
      WRITE (LUW,*)"This program will not model this system correctly."
```

```fortran
      WRITE (LUW,*)"The Angular distance between tubes is too great."
       CALL MYSTOP(1001)
       RETURN 1
      end if


c  ------------------------------------------------------------------------------------
c  Make sure nmax is large enough to contain all circumferential nodes.
c  ------------------------------------------------------------------------------------
      If (2*n+1.gt.nmax) Then
      WRITE (LUW,*)"This program will not model this system correctly."
      WRITE (LUW,*) "The parameter nmax must be increased to",2*n+1
      CALL MYSTOP(1001)
      RETURN 1
      end if

      if (nz*nr*(2*n+1)+8*nr+6.gt.15000) then
       WRITE (LUW,*) "Increase the size of the S-Array."
       CALL MYSTOP(1001)
       RETURN 1
      end if


c  ------------------------------------------------------------------------------------
c  Volume contained by each node
c  ------------------------------------------------------------------------------------
      do r = 2,nr-1,1
       vol(r) = (deltatheta/2d0)*(deltaz)
     .       * (radmid(r)**2d0
     .       - radmid(r-1)**2d0)

       volmid(r) = (deltathetamid/2d0)*(deltaz)
     .       * (radmid(r)**2d0
     .       - radmid(r-1)**2d0)

      end do




c  ------------------------------------------------------------------------------------
c  Volume contained by the last node for the constant temperature case
c  ------------------------------------------------------------------------------------
      If (boundary.ne.1) then
       vol(nr) = (deltatheta/2d0)*(deltaz)
     .       * (radff**2d0
     .       - radmid(nr-1)**2d0)
```

```fortran
      volmid(nr) = (deltathetamid/2d0)*(deltaz)
     .        * (radff**2d0
     .        -  radmid(nr-1)**2d0)
      end if


c -----------------------------------------------------------------------------------------
c  Borehole grout resistance in radial direction (r<m).
c -----------------------------------------------------------------------------------------
      do r = 2,m,1
      resrad(r) = dlog(rad(r+1)/rad(r))
     .          / (deltatheta*kgrout*deltaz)

      resradmid(r) = dlog(rad(r+1)/rad(r))
     .          / (deltathetamid*kgrout*deltaz)
      end do


c -----------------------------------------------------------------------------------------
c  Ground resistance in radial direction (r>m).
c -----------------------------------------------------------------------------------------
      do r = m+2,nr-1,1
      resrad(r) = dlog(rad(r+1)/rad(r))
     .          / (deltatheta*kgrnd*deltaz)

      resradmid(r) = dlog(rad(r+1)/rad(r))
     .          / (deltathetamid*kgrnd*deltaz)
      end do


c -----------------------------------------------------------------------------------------
c  Resistance in radial direction at grout/ground interface (r=m+1).
c -----------------------------------------------------------------------------------------
      resrad(m+1) =dlog(radmid(m+1)/rad(m+1))/(deltatheta*kgrout*deltaz)
     .        + dlog(rad(m+2)/radmid(m+1))/(deltatheta*kgrnd*deltaz)

      resradmid(m+1)=dlog(radmid(m+1)/rad(m+1))
     .        /(deltathetamid*kgrout*deltaz)
     .        +dlog(rad(m+2)/radmid(m+1))
     .        /(deltathetamid*kgrnd*deltaz)



c -----------------------------------------------------------------------------------------
c  MAKE RESISTANCE INFINITE AT BOUNDARY.
c -----------------------------------------------------------------------------------------
      if (boundary.eq.1) then
```

```
        resrad(nr-1) = 1e12
        resradmid(nr-1) = 1e12
        end if


c  ------------------------------------------------------------------------------------------
c  The resistance at the center of the node is calculated.
c  ------------------------------------------------------------------------------------------
     rhalf = radmid(1)/sqrt(2d0)
     resrad(1)=dlog(rad(2)/rhalf)/(deltatheta*kgrout*deltaz)
     resradmid(1)=dlog(rad(2)/rhalf)/(deltathetamid*kgrout*deltaz)


c  ------------------------------------------------------------------------------------------
c  Borehole grout resistance in circumferential direction (r<m+2).
c  ------------------------------------------------------------------------------------------
     do r = 2,m+1,1
       rescirc(r)=deltatheta*rad(r)
     .            /(kgrout*(radmid(r)-radmid(r-1))*deltaz)

       rescircmid(r)=(deltathetamid/2d0+deltatheta/2d0)*rad(r)
     .               /(kgrout*(radmid(r)-radmid(r-1))*deltaz)
     end do


c  ------------------------------------------------------------------------------------------
c  Borehole grout resistance in circumferential direction (r>m+1).
c  ------------------------------------------------------------------------------------------
     do r = m+2,nr,1
       rescirc(r)=deltatheta*rad(r)
     .            /(kgrnd*(radmid(r)-radmid(r-1))*deltaz)

       rescircmid(r)=(deltathetamid/2d0+deltatheta/2d0)*rad(r)
     .               /(kgrnd*(radmid(r)-radmid(r-1))*deltaz)
     end do


c  ------------------------------------------------------------------------------------------
c  Volume of fluid in the tube.  This volume is an approximation because in order to keep
c  the perimeter of a non-circular tube consistent with a circular tube the cross sectional
c  areas will be different.  However, in order to have a consistent amount of energy in the
c  system the volume of fluid should be the same so the volume of fluid is calculated by
c  considering the tube to be circular.
c  ------------------------------------------------------------------------------------------
     vfluid = (pi/2d0)*((radtubei)**2d0)*deltaz


c  Insert initial values into the S-Array
```

```fortran
      S(ISTORE) = m
      S(ISTORE+1) = n
      S(ISTORE+2) = deltatheta
      S(ISTORE+3) = deltathetamid
      S(ISTORE+4) = deltaz
      S(ISTORE+5) = vfluid
      S(ISTORE+6) = pi

      do r = 1,nr,1
      S(ISTORE+6+r) = rad(r)
      S(ISTORE+6+nr+r) = radmid(r)
      S(ISTORE+6+2*nr+r) = vol(r)
      S(ISTORE+6+3*nr+r) = volmid(r)
      S(ISTORE+6+4*nr+r) = resrad(r)
      S(ISTORE+6+5*nr+r) = resradmid(r)
      S(ISTORE+6+6*nr+r) = rescirc(r)
      S(ISTORE+6+7*nr+r) = rescircmid(r)
      end do

      alphagrnd = kgrnd/(rhogrnd*cpground)
      deltaz = depth/(nz)
      do z = 1,nz,1
      vert(z) = deltaz*(dfloat(z)-0.5d0)
      tff(z)=tmean-tamp*dexp(-vert(z)*(pi/8766.0d0/alphagrnd)**0.5d0)
     .     *dcos(2d0*pi/8766.0d0*(TIME-(tshift*24.0d0)-vert(z)/2d0
     .     *(8766.0d0/pi/alphagrnd)**0.5d0))
      end do


      do z = 1,nz,1
       do c = 1,2*n+1,1
        do r = 1,nr,1
         S(ISTORE+6+8*nr+(z-1)*(2*n+1)*nr + nr*(c-1) + r) = tff(z)
        end do
       end do
      end do

        RETURN 1
      END IF

c
********************************************************************************
********
C  Tube to ground resistance,and time step change with time
```

```
c
***********************************************************************
********
c -----------------------------------------------------------------------------
c Calculate the heat transfer coefficient for the fluid to tube wall heat transfer.
c -----------------------------------------------------------------------------
c fluid Flow Rate
      wspeed = galfluid*231d0*60d0/(12d0**3d0)/(pi*((radtubei)**2d0))
      wmflow = (60d0*galfluid*231d0*rhofluid/(12d0**3d0))/2d0

      if (wspeed.lt.0.1d0) then
      nu = 4.36
      goto 20
      end if


      re = (wspeed*dtubei*rhofluid)/(mufluid)
      fp = (0.79d0*dlog(re)-1.64d0)**(-2d0)
      pr = cpfluid*mufluid/kfluid
      nu1 = ((fp/8d0)*(re-1000d0)*pr)
      nu2 = (1d0+12.7d0*((fp/8d0)**0.5d0)*((pr**(2d0/3d0))-1d0))
      nu=nu1/nu2

      If (re.lt.2300d0) Then
      nu = 4.36
      end if

  20  htube1 = kfluid*nu/dtubei


c -----------------------------------------------------------------------------
c Tube to ground resistance parameters
c -----------------------------------------------------------------------------
      shape = geofactor
      atube1 = deltaz*(deltatheta/2d0)*(radmid(m-1))
      atube2 = deltaz*(deltatheta/2d0)*(radmid(m))
      atube3 = (radmid(m)-radmid(m-1))*deltaz
      restubegrnd1 =  1d0/(htube1*atube1)
     .          + dlog(radmid(m-1)/(radmid(m-1)-ltube))
     .              /((deltatheta/2d0)*ktube*deltaz)
     .          + (dlog((radmid(m-1)-ltube)/rad(m-1))
     .              /((deltatheta/2d0)*kgrout*deltaz))*shape

      restubegrnd2 =  1d0/(htube1*atube2)
     .          + dlog((radmid(m)+ltube)/radmid(m))
```

```
.                    /((deltatheta/2d0)*ktube*deltaz)
.              + (dlog(rad(m+1)/(radmid(m)+ltube))
.                    /((deltatheta/2d0)*kgrout*deltaz))*shape

      restubegrnd3 = 1d0/(htube1*atube3)
.              +  ltube/(ktube*atube3)
.              + ((rad(m)*(deltatheta/2d0)-ltube)/
.               (kgrout*atube3))*shape
```

c Resistance Parameters for Travelling up the tube
```
      htube2 = htube1

      restubegrnd4 =  1d0/(htube2*atube1)
.              + dlog(radmid(m-1)/(radmid(m-1)-ltube))
.                    /((deltatheta/2d0)*ktube*deltaz)
.              + (dlog((radmid(m-1)-ltube)/rad(m-1))
.                    /((deltatheta/2d0)*kgrout*deltaz))*shape

      restubegrnd5 =  1d0/(htube2*atube2)
.              + dlog((radmid(m)+ltube)/radmid(m))
.                    /((deltatheta/2d0)*ktube*deltaz)
.              + (dlog(rad(m+1)/(radmid(m)+ltube))
.                    /((deltatheta/2d0)*kgrout*deltaz))*shape

      restubegrnd6 = 1d0/(htube2*atube3)
.              +  ltube/(ktube*atube3)
.              + ((rad(m)*(deltatheta/2d0)-ltube)/
.               (kgrout*atube3))*shape
```


c
****
****
c This section calculates the critical time step for the finite difference explicit
c euler solution technique for the problem. This is also the section where the split
c system is calculated. In other words the number of times the fluid temperature is
c calculated before the ground/grout temperature is updated is determined here.
c
****
****

c ---
c Critical time steps for the ground
c ---

```
      tcrit1 = rhogrnd*cpground*vol(m+2)
     .       / (1d0/resrad(m+1)+1d0/resrad(m+2)+2d0/rescirc(m+2))

      tcrit2 = rhogrnd*cpground*volmid(m+2)
     .       /(1d0/resradmid(m+1)+1d0/resradmid(m+2)+2d0/rescircmid(m+2))


c  ----------------------------------------------------------------------------------------
c  Critical Time Step for the fluid in the u-tube.
c  ----------------------------------------------------------------------------------------
      tcrit4 = (rhofluid*vfluid)/(1d0/(cpfluid*restubegrnd1)
     .       + 1d0/(cpfluid*restubegrnd2)
     .       + 1d0/(cpfluid*restubegrnd3) + wmflow)


c  ----------------------------------------------------------------------------------------
c  The critical time step for the fluid nodes needs to be an integer multiple of the TRNSYS
c  time step.
c  ----------------------------------------------------------------------------------------
      times1 = idint(DELT/tcrit4)+1
      deltat1 = DELT/dfloat(times1)


c  ----------------------------------------------------------------------------------------
c  If the ground critical time step is smaller than the fluid time step change times1.
c  ----------------------------------------------------------------------------------------
      tcrit5 = min(tcrit1,tcrit2)

      If (tcrit5.lt.deltat1) Then
       times1 = idint(DELT/tcrit5)+1
       deltat1 = DELT/dfloat(times1)
      end if


c  ----------------------------------------------------------------------------------------
c  Deltat2 is the time step used for the ground/grout nodes.
c  Times2 is the number of times the fluid nodes will be updated before the ground nodes.
c  ----------------------------------------------------------------------------------------
      times2 = idint(tcrit5/deltat1)

      if(times2.gt.nz-1) then
       times2 = nz
      end if

      if(times2.gt.times1) then
       times2 = times1
      end if
```

```
      deltat2 = dfloat(times2)*deltat1


c  ------------------------------------------------------------------------------------
c  Times3+1 is the number of times the ground nodes will be updated before going to TRNSYS.
c  ------------------------------------------------------------------------------------
      times3 = times1/times2

      if(times3.gt.nfluid3) then
      WRITE(LUW,*) "The program will not run properly."
      WRITE(LUW,*)"Increase the parameter nfluid3 to",times3
      WRITE(LUW,*)"or decrease the simulation time step."
      CALL MYSTOP(1001)
      RETURN 1
      end if


c  ------------------------------------------------------------------------------------
c  Times4 is the number of "extra" loops that need to be run to get the time steps on the
c  same time as TRNSYS.
c  ------------------------------------------------------------------------------------
      times4 = times1-times2*times3


c  ------------------------------------------------------------------------------------
c  Constants dependent on deltat1
c  ------------------------------------------------------------------------------------
      fluidcap1 = wmflow*deltat1/(rhofluid*vfluid)
      fluidcap2 = deltat1/(rhofluid*vfluid*cpfluid)


c  ------------------------------------------------------------------------------------
c  Constants dependent on deltat2
c  ------------------------------------------------------------------------------------
      do r = m+2,nr-1
      grndcap(r) = (deltat2)/((rhogrnd*cpground*vol(r)))
      grndcapmid(r) = (deltat2)/((rhogrnd*cpground*volmid(r)))
      end do



c
********************************************************************************
********
C  REINITIALIZE THE VALUES FROM THE S ARRAY
c
********************************************************************************
********
c  ------------------------------------------------------------------------------------
```

```
c  Update the farfield temperatures for the time of year.
c  ------------------------------------------------------------------------------------
      do z = 1,nz,1
      vert(z) = deltaz*(dfloat(z)-0.5d0)
      tff(z)=tmean-tamp*dexp(-vert(z)*(pi/8766.0d0/alphagrnd)**0.5d0)
    .       *dcos(2d0*pi/8766.0d0*(TIME-(tshift*24.0d0)-vert(z)/2d0
    .       *(8766.0d0/pi/alphagrnd)**0.5d0))
      end do

      ISTORE = INFO(10)

c  Take values from the S-Array
      m = S(ISTORE)
      n = S(ISTORE+1)
      deltatheta = S(ISTORE+2)
      deltathetamid = S(ISTORE+3)
      deltaz = S(ISTORE+4)
      vfluid = S(ISTORE+5)
      pi = S(ISTORE+6)

      do r = 1,nr,1
      rad(r) = S(ISTORE+6+r)
      radmid(r) = S(ISTORE+6+nr+r)
      vol(r) = S(ISTORE+6+2*nr+r)
      volmid(r) = S(ISTORE+6+3*nr+r)
      resrad(r) = S(ISTORE+6+4*nr+r)
      resradmid(r) = S(ISTORE+6+5*nr+r)
      rescirc(r) = S(ISTORE+6+6*nr+r)
      rescircmid(r) = S(ISTORE+6+7*nr+r)
      end do

      S(ISTORE+6+8*nr)=tinlet
      told(m,1,0) = S(ISTORE+6+8*nr)
      tnew(m,1,0) = S(ISTORE+6+8*nr)

      do z = 1,nz,1
       do c = 1,2*n+1,1
        do r = 1,nr,1
         told(r,c,z) = S(ISTORE+6+8*nr+(z-1)*(2*n+1)*nr + nr*(c-1) + r)
        end do
       end do
      end do
```

```
      do z = 1,nz,1
       do c = 1,2*n+1,1
         tnew(nr,c,z) = tff(z)
         told(nr,c,z) = tff(z)
       end do
      end do


c
********************************************************************
********
c  Begin the numerical calculations
c
********************************************************************
********

c  begin the times3 loop for time step deltat2
      do tms3 = 1,times3,1


c  ------------------------------------------------------------------------------------
c  Run the fluid nodes at the smallest time step.
c  ------------------------------------------------------------------------------------
c  begin the times2 loop for deltat1
      do tms2 = 1,times2,1

c  Initial Qout and Fluid inlet temperature.
      qouttotal(1) = 0d0

c  Energy Balance on nodes used for fluid flowing down tube
      do z = 1,nz,1
      qoutdown(z) = (told(m,1,z) - told(m-1,1,z))*(1d0/(restubegrnd1))
      .          + (told(m,1,z) - told(m+1,1,z))*(1d0/(restubegrnd2))
      .          + (told(m,1,z) - told(m,2,z))*(1d0/restubegrnd3)

      tnew(m,1,z) = told(m,1,z)
      .          - fluidcap2*qoutdown(z)
      .          + fluidcap1*(told(m,1,z-1)-told(m,1,z))
      end do

c  Energy Balance on nodes used for fluid flowing up tube
      do z = 1,nz-1,1
      qoutup(z)=
      .   (told(m,2*n+1,z)-told(m-1,2*n+1,z))*(1d0/(restubegrnd4))
      .   +(told(m,2*n+1,z)-told(m+1,2*n+1,z))*(1d0/(restubegrnd5))
```

```
    .  +(told(m,2*n+1,z)-told(m,2*n,z))*(1d0/restubegrnd6)

    tnew(m,2*n+1,z) = told(m,2*n+1,z)
    .           -fluidcap2*qoutup(z)
    .           +fluidcap1*(told(m,2*n+1,z+1)-told(m,2*n+1,z))
    end do

c  At the bottom of the u-tube the connecting fluid nodes are c=1 and c=2*n+1
    qoutup(nz) =
    .   (told(m,2*n+1,nz)-told(m-1,2*n+1,nz))*(1d0/(restubegrnd4))
    .  +(told(m,2*n+1,nz)-told(m+1,2*n+1,nz))*(1d0/(restubegrnd5))
    .  +(told(m,2*n+1,nz)-told(m,2*n+1-1,nz))*(1d0/restubegrnd6)

    tnew(m,2*n+1,nz) = told(m,2*n+1,nz)
    .           -(fluidcap2)*qoutup(nz)
    .           +(fluidcap1)*(told(m,1,nz)-told(m,2*n+1,nz))

c  Total Sum of Qout
    do z = 1,nz,1
    qouttotal(z)=qouttotal(z-1)+qoutdown(z)+qoutup(z)
    end do


c  ---------------------------------------------------------------------------------
c  ---------------------------------------------------------------------------------

c  Replace old fluid temp with new temperature.
    do z = 1,nz,1
    told(m,2*n+1,z) = tnew(m,2*n+1,z)
    told(m,1,z) = tnew(m,1,z)
    end do

    tfluid(tms2,tms3) = tnew(m,2*n+1,1)
    qout(tms2,tms3) = qouttotal(nz)

c  End of the times2 loop
    end do


c  ---------------------------------------------------------------------------------
c  Run the grout with deltat2 time step.
c  ---------------------------------------------------------------------------------
c  Energy Balance on the grout nodes
    do z = 1,nz,1
    do r = 2,m+1,1
    do c = 3,n-1,1
```

```
     tnew(r,c,z) = ( (1d0/resrad(r-1))*told(r-1,c,z)
  .          + (1d0/resrad(r))*told(r+1,c,z)
  .          + (1d0/rescirc(r))*told(r,c-1,z)
  .          + (1d0/rescirc(r))*told(r,c+1,z)  )
  . / ( 1d0/resrad(r-1)+1d0/resrad(r)+1d0/rescirc(r)+1d0/rescirc(r) )
     end do
     end do
     end do


     do z = 1,nz,1
     do c = n+3,2*n-1,1
     do r = 2,m+1,1
     tnew(r,c,z) = ( (1d0/resrad(r-1))*told(r-1,c,z)
  .          + (1d0/resrad(r))*told(r+1,c,z)
  .          + (1d0/rescirc(r))*told(r,c-1,z)
  .          + (1d0/rescirc(r))*told(r,c+1,z)  )
  . / ( 1d0/resrad(r-1)+1d0/resrad(r)+1d0/rescirc(r)+1d0/rescirc(r) )
     end do
     end do
     end do


c  Do calculations for r=2,m-1,c=2,c=2*n.
     do z=1,nz,1
     do r = 2,m-1,1
     tnew(r,2,z)=( (1d0/resrad(r-1))*told(r-1,2,z)
  .          +(1d0/resrad(r))*told(r+1,2,z)
  .          +(1d0/rescirc(r))*told(r,1,z)
  .          +(1d0/rescirc(r))*told(r,3,z) )
  . /(1d0/resrad(r-1)+1d0/resrad(r)+1d0/rescirc(r)+1d0/rescirc(r))


     tnew(r,2*n,z)=( (1d0/resrad(r-1))*told(r-1,2*n,z)
  .          +(1d0/resrad(r))*told(r+1,2*n,z)
  .          +(1d0/rescirc(r))*told(r,2*n-1,z)
  .          +(1d0/rescirc(r))*told(r,2*n+1,z) )
  . /(1d0/resrad(r-1)+1d0/resrad(r)+1d0/rescirc(r)+1d0/rescirc(r))
     end do
     end do

c  Do calculations for the edge of the grout r=m+1,c=2,c=2*n.
     do z=1,nz,1
     tnew(m+1,2,z)=( (1d0/resrad(m))*told(m,2,z)
  .          +(1d0/resrad(m+1))*told(m+2,2,z)
```

```
.        +(1d0/rescirc(m+1))*told(m+1,1,z)
.        +(1d0/rescirc(m+1))*told(m+1,3,z) )
./(1d0/resrad(m)+1d0/resrad(m+1)+1d0/rescirc(m+1)+1d0/rescirc(m+1))


 tnew(m+1,2*n,z)=( (1d0/resrad(m))*told(m,2*n,z)
.        +(1d0/resrad(m+1))*told(m+2,2*n,z)
.        +(1d0/rescirc(m+1))*told(m+1,2*n-1,z)
.        +(1d0/rescirc(m+1))*told(m+1,2*n+1,z) )
./(1d0/resrad(m)+1d0/resrad(m+1)+1d0/rescirc(m+1)+1d0/rescirc(m+1))
 end do

c  Nodes at or adjacent to theta=90 (n,n+1,n+2) must have deltatheta altered.
   do z = 1,nz,1
   do r = 2,m+1,1
   tnew(r,n,z) = ( (1d0/resrad(r-1))*told(r-1,n,z)
.          + (1d0/resrad(r))*told(r+1,n,z)
.          + (1d0/rescirc(r))*told(r,n-1,z)
.          + (1d0/rescircmid(r))*told(r,n+1,z)  )
 . /(1d0/resrad(r-1)+1d0/resrad(r)+1d0/rescirc(r)+1d0/rescircmid(r))
   end do
   end do


   do z = 1,nz,1
   do r = 2,m+1,1
   tnew(r,n+1,z) = ( (1d0/resradmid(r-1))*told(r-1,n+1,z)
.          + (1d0/resradmid(r))*told(r+1,n+1,z)
.          + (1d0/rescircmid(r))*told(r,n,z)
.          + (1d0/rescircmid(r))*told(r,n+2,z)  )
./(1d0/resradmid(r-1)+1d0/resradmid(r)+1d0/rescircmid(r)
 . +1d0/rescircmid(r))
   end do
   end do


   do z = 1,nz,1
   do r = 2,m+1,1
   tnew(r,n+2,z) = ( (1d0/resrad(r-1))*told(r-1,n+2,z)
.          + (1d0/resrad(r))*told(r+1,n+2,z)
.          + (1d0/rescircmid(r))*told(r,n+1,z)
.          + (1d0/rescirc(r))*told(r,n+3,z)  )
./(1d0/resrad(r-1)+1d0/resrad(r)+1d0/rescirc(r)+1d0/rescircmid(r))
   end do
   end do
```

```
c  Energy Balance on grout node adjacent to the tube(r=m,c=2).
      do z = 1,nz,1
      tnew(m,2,z)=( (1d0/resrad(m-1))*told(m-1,2,z)
    .          +(1d0/resrad(m))*told(m+1,2,z)
    .          +(1d0/restubegrnd3)*told(m,1,z)
    .          +(1d0/rescirc(m))*told(m,3,z) )
    . /(1d0/resrad(m-1)+1d0/resrad(m)+1d0/restubegrnd3+1d0/rescirc(m))
      end do


c  Energy Balance on grout node adjacent to the tube (r=m,c=2*n+1-1).
      do z = 1,nz,1
      tnew(m,2*n+1-1,z)=( (1d0/resrad(m-1))*told(m-1,2*n+1-1,z)
    .          +(1d0/resrad(m))*told(m+1,2*n+1-1,z)
    .          +(1d0/rescirc(m))*told(m,2*n+1-2,z)
    .          +(1d0/restubegrnd6)*told(m,2*n+1,z) )
    . /(1d0/resrad(m-1)+1d0/resrad(m)+1d0/restubegrnd6+1d0/rescirc(m))
      end do


c  Energy Balance on nodes along adiabatic boundary (c=1 and c=2*n+1)
      do z = 1,nz,1
      do r = 2,m-2,1
      tnew(r,1,z) = ( (1d0/(2d0*resrad(r-1)))*told(r-1,1,z)
    .          +(1d0/(2d0*resrad(r)))*told(r+1,1,z)
    .          +(1d0/rescirc(r))*told(r,2,z) )
    . /(1d0/(2d0*resrad(r-1))+1d0/(2d0*resrad(r))+1d0/rescirc(r))


      tnew(r,2*n+1,z) = ((1d0/(2d0*resrad(r-1)))*told(r-1,2*n+1,z)
    .          +(1d0/(2d0*resrad(r)))*told(r+1,2*n+1,z)
    .          +(1d0/rescirc(r))*told(r,2*n+1-1,z))
    . /(1d0/(2d0*resrad(r-1))+1d0/(2d0*resrad(r))+1d0/rescirc(r))
      end do
      end do



c  Energy Balance on nodes next to tube (r=m-1) along adiabatic boundary.
      do z = 1,nz,1
c  Energy Balance c=1 nodes
      tnew(m-1,1,z) = ( (1d0/restubegrnd1)*told(m,1,z)
    .          +(1d0/(2*resrad(m-2)))*told(m-2,1,z)
    .          +(1d0/rescirc(m-1))*told(m-1,2,z)   )
    . /(1d0/restubegrnd1+1d0/(2*resrad(m-2))+1d0/rescirc(m-1))
```

```
c  Energy Balance c=2*n+1 nodes
      tnew(m-1,2*n+1,z) = (  (1d0/restubegrnd4)*told(m,2*n+1,z)
     .           +(1d0/(2*resrad(m-2)))*told(m-2,2*n+1,z)
     .           +(1d0/rescirc(m-1))*told(m-1,2*n+1-1,z) )
     . /(1d0/restubegrnd4+1d0/(2*resrad(m-2))+1d0/rescirc(m-1))
      end do


c  Energy Balance on nodes next to tube (r=m+1) along adiabatic boundary.
      do z = 1,nz,1
c  Energy Balance c=1 nodes
      tnew(m+1,1,z) = (  (1d0/restubegrnd2)*told(m,1,z)
     .           +(1d0/(2d0*resrad(m+1)))*told(m+2,1,z)
     .           +(1d0/rescirc(m+1))*told(m+1,2,z)   )
     . /(1d0/restubegrnd2+1d0/(2d0*resrad(m+1))+1d0/rescirc(m+1))



c  Energy Balance c=2*n+1 nodes
      tnew(m+1,2*n+1,z) = (  (1d0/restubegrnd5)*told(m,2*n+1,z)
     .           +(1d0/(2d0*resrad(m+1)))*told(m+2,2*n+1,z)
     .           +(1d0/rescirc(m+1))*told(m+1,2*n+1-1,z) )
     . /(1d0/restubegrnd5+1d0/(2d0*resrad(m+1))+1d0/rescirc(m+1))
      end do


c  Energy Balance on the node at the center of the grid.
      do z = 1,nz,1
      numer = 0d0
      do c = 1,n
      numer = numer + (1d0/resrad(1))*told(2,c,z)
      end do

      do c = n+2,2*n+1
      numer = numer + (1d0/resrad(1))*told(2,c,z)
      end do

      numer = numer + (1d0/resradmid(1))*told(2,n,z)

      do c = 1,2*n+1
      tnew(1,c,z)=numer/((2d0*n)/resrad(1)+1d0/resradmid(1))
      end do

      end do
```

```
c  ------------------------------------------------------------------------------------
c  Run ground nodes at deltat2 time step.
c  ------------------------------------------------------------------------------------
      do z = 1,nz,1
      do r = m+2,nr-1,1
      do c = 2,n-1,1
      tnew(r,c,z) = told(r,c,z)
     . +(grndcap(r)/resrad(r-1))*(told(r-1,c,z)-told(r,c,z))
     . +(grndcap(r)/resrad(r))*(told(r+1,c,z)-told(r,c,z))
     . +(grndcap(r)/rescirc(r))*(told(r,c-1,z)-told(r,c,z))
     . +(grndcap(r)/rescirc(r))*(told(r,c+1,z)-told(r,c,z))
      end do
      end do
      end do

      do z = 1,nz,1
      do r = m+2,nr-1,1
      do c = n+3,2*n,1
      tnew(r,c,z) = told(r,c,z)
     . +(grndcap(r)/resrad(r-1))*(told(r-1,c,z)-told(r,c,z))
     . +(grndcap(r)/resrad(r))*(told(r+1,c,z)-told(r,c,z))
     . +(grndcap(r)/rescirc(r))*(told(r,c-1,z)-told(r,c,z))
     . +(grndcap(r)/rescirc(r))*(told(r,c+1,z)-told(r,c,z))
      end do
      end do
      end do


c  At the center of the grid n,(n+1),(n+2) the deltatheta must be altered
      do z = 1,nz,1
      do r = m+2,nr-1,1
      tnew(r,n,z) = told(r,n,z)
     . +(grndcap(r)/resrad(r-1))*(told(r-1,n,z)-told(r,n,z))
     . +(grndcap(r)/resrad(r))*(told(r+1,n,z)-told(r,n,z))
     . +(grndcap(r)/rescirc(r))*(told(r,n-1,z)-told(r,n,z))
     . +(grndcap(r)/rescircmid(r))*(told(r,n+1,z)-told(r,n,z))
      end do
      end do

      do z = 1,nz,1
      do r = m+2,nr-1,1
      tnew(r,n+1,z) = told(r,n+1,z)
     . +(grndcapmid(r)/resradmid(r-1))*(told(r-1,n+1,z)-told(r,n+1,z))
     . +(grndcapmid(r)/resradmid(r))*(told(r+1,n+1,z)-told(r,n+1,z))
```

```
      . +(grndcapmid(r)/rescircmid(r))*(told(r,n,z)-told(r,n+1,z))
      . +(grndcapmid(r)/rescircmid(r))*(told(r,n+2,z)-told(r,n+1,z))
       end do
       end do

       do z = 1,nz,1
       do r = m+2,nr-1,1
       tnew(r,n+2,z) = told(r,n+2,z)
      . +(grndcap(r)/resrad(r-1))*(told(r-1,n+2,z)-told(r,n+2,z))
      . +(grndcap(r)/resrad(r))*(told(r+1,n+2,z)-told(r,n+2,z))
      . +(grndcap(r)/rescircmid(r))*(told(r,n+1,z)-told(r,n+2,z))
      . +(grndcap(r)/rescirc(r))*(told(r,n+3,z)-told(r,n+2,z))
       end do
       end do

c  Energy Balance on nodes along adiabatic boundary (c=1 and c=2*n+1)
       do z = 1,nz,1
       do r = m+2,nr-1,1
       tnew(r,1,z) = told(r,1,z)
      . +(grndcap(r)/resrad(r-1))* (told(r-1,1,z)-told(r,1,z))
      . +(grndcap(r)/resrad(r))*(told(r+1,1,z)-told(r,1,z))
      . +(2d0*grndcap(r)/rescirc(r))*(told(r,2,z)-told(r,1,z))

       tnew(r,2*n+1,z) = told(r,2*n+1,z)
      . +(grndcap(r)/resrad(r-1))* (told(r-1,2*n+1,z)-told(r,2*n+1,z))
      . +(grndcap(r)/resrad(r))*(told(r+1,2*n+1,z)-told(r,2*n+1,z))
      . +(2d0*grndcap(r)/rescirc(r))*(told(r,2*n+1-1,z)-told(r,2*n+1,z))
       end do
       end do

c  Replace old values with new values
       do z = 1,nz,1
        do c = 1,2*n+1,1
         do r = 1,nr,1
          told(r,c,z) = tnew(r,c,z)
         end do
        end do
       end do

c  End of the times3 loop
       end do

C
************************************************************************
```

```
********
C  *******Perform the calculations one more time for
times4***************************
C
*************************************************************************
********

c  Recalculate the ground time step
     deltat2 = dfloat(times4)*deltat1


c  -----------------------------------------------------------------------------------
c  Constants dependent on deltat2
c  -----------------------------------------------------------------------------------
     do r = m+2,nr-1
     grndcap(r) = (deltat2)/((rhogrnd*cpground*vol(r)))
     grndcapmid(r) = (deltat2)/((rhogrnd*cpground*volmid(r)))
     end do


c  -----------------------------------------------------------------------------------
c  Run the fluid nodes at the smallest time step.
c  -----------------------------------------------------------------------------------
c  begin the times4 loop for deltat1
     do tms4 = 1,times4,1

c  Initial Qout and Fluid inlet temperature.
     qouttotal(1) = 0d0

c  Energy Balance on nodes used for fluid flowing down tube
     do z = 1,nz,1
     qoutdown(z) = (told(m,1,z) - told(m-1,1,z))*(1d0/(restubegrnd1))
     .         + (told(m,1,z) - told(m+1,1,z))*(1d0/(restubegrnd2))
     .         + (told(m,1,z) - told(m,2,z))*(1d0/restubegrnd3)

     tnew(m,1,z) = told(m,1,z)
     .         - fluidcap2*qoutdown(z)
     .         + fluidcap1*(told(m,1,z-1)-told(m,1,z))
     end do

c  Energy Balance on nodes used for fluid flowing up tube
     do z = 1,nz-1,1
     qoutup(z)=
     .   (told(m,2*n+1,z)-told(m-1,2*n+1,z))*(1d0/(restubegrnd4))
     .   +(told(m,2*n+1,z)-told(m+1,2*n+1,z))*(1d0/(restubegrnd5))
     .   +(told(m,2*n+1,z)-told(m,2*n,z))*(1d0/restubegrnd6)
```

```
      tnew(m,2*n+1,z) = told(m,2*n+1,z)
    .          -fluidcap2*qoutup(z)
    .          +fluidcap1*(told(m,2*n+1,z+1)-told(m,2*n+1,z))
      end do


c  At the bottom of the u-tube the connecting fluid nodes are c=1 and c=2*n+1
      qoutup(nz) =
    .   (told(m,2*n+1,nz)-told(m-1,2*n+1,nz))*(1d0/(restubegrnd4))
    .  +(told(m,2*n+1,nz)-told(m+1,2*n+1,nz))*(1d0/(restubegrnd5))
    .  +(told(m,2*n+1,nz)-told(m,2*n+1-1,nz))*(1d0/restubegrnd6)

      tnew(m,2*n+1,nz) = told(m,2*n+1,nz)
    .            -(fluidcap2)*qoutup(nz)
    .            +(fluidcap1)*(told(m,1,nz)-told(m,2*n+1,nz))

c  Total Sum of Qout
      do z = 1,nz,1
      qouttotal(z)=qouttotal(z-1)+qoutdown(z)+qoutup(z)
      end do
c  -------------------------------------------------------------------------------------
c  -------------------------------------------------------------------------------------


c  Replace old fluid temp with new temperature.
      do z = 1,nz,1
      told(m,2*n+1,z) = tnew(m,2*n+1,z)
      told(m,1,z) = tnew(m,1,z)
      end do

      tfluid(tms4,times3+1) = tnew(m,2*n+1,1)
      qout(tms4,times3+1) = qouttotal(nz)


c  End of the times4 loop
      end do


c  -------------------------------------------------------------------------------------
c  Run the grout with deltat2 time step.
c  -------------------------------------------------------------------------------------
c  Energy Balance on the grout nodes
      do z = 1,nz,1
      do r = 2,m+1,1
      do c = 3,n-1,1
      tnew(r,c,z) = ( (1d0/resrad(r-1))*told(r-1,c,z)
    .          + (1d0/resrad(r))*told(r+1,c,z)
```

```fortran
     .          + (1d0/rescirc(r))*told(r,c-1,z)
     .          + (1d0/rescirc(r))*told(r,c+1,z)  )
     . / ( 1d0/resrad(r-1)+1d0/resrad(r)+1d0/rescirc(r)+1d0/rescirc(r) )
      end do
      end do
      end do

      do z = 1,nz,1
      do c = n+3,2*n-1,1
      do r = 2,m+1,1
      tnew(r,c,z) = ( (1d0/resrad(r-1))*told(r-1,c,z)
     .          + (1d0/resrad(r))*told(r+1,c,z)
     .          + (1d0/rescirc(r))*told(r,c-1,z)
     .          + (1d0/rescirc(r))*told(r,c+1,z)  )
     . / ( 1d0/resrad(r-1)+1d0/resrad(r)+1d0/rescirc(r)+1d0/rescirc(r) )
      end do
      end do
      end do


c  Do calculations for r=2,m-1,c=2,c=2*n.
      do z=1,nz,1
      do r = 2,m-1,1
      tnew(r,2,z)=( (1d0/resrad(r-1))*told(r-1,2,z)
     .          +(1d0/resrad(r))*told(r+1,2,z)
     .          +(1d0/rescirc(r))*told(r,1,z)
     .          +(1d0/rescirc(r))*told(r,3,z) )
     . /(1d0/resrad(r-1)+1d0/resrad(r)+1d0/rescirc(r)+1d0/rescirc(r))


      tnew(r,2*n,z)=( (1d0/resrad(r-1))*told(r-1,2*n,z)
     .          +(1d0/resrad(r))*told(r+1,2*n,z)
     .          +(1d0/rescirc(r))*told(r,2*n-1,z)
     .          +(1d0/rescirc(r))*told(r,2*n+1,z) )
     . /(1d0/resrad(r-1)+1d0/resrad(r)+1d0/rescirc(r)+1d0/rescirc(r))
      end do
      end do

c  Do calculations for the edge of the grout r=m+1,c=2,c=2*n.
      do z=1,nz,1
      tnew(m+1,2,z)=( (1d0/resrad(m))*told(m,2,z)
     .          +(1d0/resrad(m+1))*told(m+2,2,z)
     .          +(1d0/rescirc(m+1))*told(m+1,1,z)
     .          +(1d0/rescirc(m+1))*told(m+1,3,z) )
```

```
     ./(1d0/resrad(m)+1d0/resrad(m+1)+1d0/rescirc(m+1)+1d0/rescirc(m+1))


     tnew(m+1,2*n,z)=( (1d0/resrad(m))*told(m,2*n,z)
   .        +(1d0/resrad(m+1))*told(m+2,2*n,z)
   .        +(1d0/rescirc(m+1))*told(m+1,2*n-1,z)
   .        +(1d0/rescirc(m+1))*told(m+1,2*n+1,z) )
   ./(1d0/resrad(m)+1d0/resrad(m+1)+1d0/rescirc(m+1)+1d0/rescirc(m+1))
     end do

c  Nodes at or adjacent to theta=90 (n,n+1,n+2) must have deltatheta altered.
     do z = 1,nz,1
     do r = 2,m+1,1
     tnew(r,n,z) = ( (1d0/resrad(r-1))*told(r-1,n,z)
   .        + (1d0/resrad(r))*told(r+1,n,z)
   .        + (1d0/rescirc(r))*told(r,n-1,z)
   .        + (1d0/rescircmid(r))*told(r,n+1,z)  )
   . /(1d0/resrad(r-1)+1d0/resrad(r)+1d0/rescirc(r)+1d0/rescircmid(r))
     end do
     end do


     do z = 1,nz,1
     do r = 2,m+1,1
     tnew(r,n+1,z) = ( (1d0/resradmid(r-1))*told(r-1,n+1,z)
   .        + (1d0/resradmid(r))*told(r+1,n+1,z)
   .        + (1d0/rescircmid(r))*told(r,n,z)
   .        + (1d0/rescircmid(r))*told(r,n+2,z)  )
   ./(1d0/resradmid(r-1)+1d0/resradmid(r)+1d0/rescircmid(r)
   . +1d0/rescircmid(r))
     end do
     end do


     do z = 1,nz,1
     do r = 2,m+1,1
     tnew(r,n+2,z) = ( (1d0/resrad(r-1))*told(r-1,n+2,z)
   .        + (1d0/resrad(r))*told(r+1,n+2,z)
   .        + (1d0/rescircmid(r))*told(r,n+1,z)
   .        + (1d0/rescirc(r))*told(r,n+3,z)  )
   ./(1d0/resrad(r-1)+1d0/resrad(r)+1d0/rescirc(r)+1d0/rescircmid(r))
     end do
     end do

c Energy Balance on grout node adjacent to the tube(r=m,c=2).
     do z = 1,nz,1
```

```
    tnew(m,2,z)=( (1d0/resrad(m-1))*told(m-1,2,z)
    .        +(1d0/resrad(m))*told(m+1,2,z)
    .        +(1d0/restubegrnd3)*told(m,1,z)
    .        +(1d0/rescirc(m))*told(m,3,z) )
    . /(1d0/resrad(m-1)+1d0/resrad(m)+1d0/restubegrnd3+1d0/rescirc(m))
    end do

c  Energy Balance on grout node adjacent to the tube (r=m,c=2*n+1-1).
    do z = 1,nz,1
    tnew(m,2*n+1-1,z)=( (1d0/resrad(m-1))*told(m-1,2*n+1-1,z)
    .        +(1d0/resrad(m))*told(m+1,2*n+1-1,z)
    .        +(1d0/rescirc(m))*told(m,2*n+1-2,z)
    .        +(1d0/restubegrnd6)*told(m,2*n+1,z) )
    . /(1d0/resrad(m-1)+1d0/resrad(m)+1d0/restubegrnd6+1d0/rescirc(m))
    end do

c  Energy Balance on nodes along adiabatic boundary (c=1 and c=2*n+1)
    do z = 1,nz,1
    do r = 2,m-2,1
    tnew(r,1,z) = ( (1d0/(2d0*resrad(r-1)))*told(r-1,1,z)
    .        +(1d0/(2d0*resrad(r)))*told(r+1,1,z)
    .        +(1d0/rescirc(r))*told(r,2,z) )
    . /(1d0/(2d0*resrad(r-1))+1d0/(2d0*resrad(r))+1d0/rescirc(r))


    tnew(r,2*n+1,z) = ((1d0/(2d0*resrad(r-1)))*told(r-1,2*n+1,z)
    .        +(1d0/(2d0*resrad(r)))*told(r+1,2*n+1,z)
    .        +(1d0/rescirc(r))*told(r,2*n+1-1,z))
    . /(1d0/(2d0*resrad(r-1))+1d0/(2d0*resrad(r))+1d0/rescirc(r))
    end do
    end do



c  Energy Balance on nodes next to tube (r=m-1) along adiabatic boundary.
    do z = 1,nz,1
c  Energy Balance c=1 nodes
    tnew(m-1,1,z) = ( (1d0/restubegrnd1)*told(m,1,z)
    .        +(1d0/(2*resrad(m-2)))*told(m-2,1,z)
    .        +(1d0/rescirc(m-1))*told(m-1,2,z)   )
    . /(1d0/restubegrnd1+1d0/(2*resrad(m-2))+1d0/rescirc(m-1))


c  Energy Balance c=2*n+1 nodes
```

```
      tnew(m-1,2*n+1,z) = (  (1d0/restubegrnd4)*told(m,2*n+1,z)
     .          +(1d0/(2*resrad(m-2)))*told(m-2,2*n+1,z)
     .          +(1d0/rescirc(m-1))*told(m-1,2*n+1-1,z) )
     . /(1d0/restubegrnd4+1d0/(2*resrad(m-2))+1d0/rescirc(m-1))
      end do

c  Energy Balance on nodes next to tube (r=m+1) along adiabatic boundary.
      do z = 1,nz,1
c  Energy Balance c=1 nodes
      tnew(m+1,1,z) = (  (1d0/restubegrnd2)*told(m,1,z)
     .          +(1d0/(2d0*resrad(m+1)))*told(m+2,1,z)
     .          +(1d0/rescirc(m+1))*told(m+1,2,z)   )
     . /(1d0/restubegrnd2+1d0/(2d0*resrad(m+1))+1d0/rescirc(m+1))


c  Energy Balance c=2*n+1 nodes
      tnew(m+1,2*n+1,z) = (  (1d0/restubegrnd5)*told(m,2*n+1,z)
     .          +(1d0/(2d0*resrad(m+1)))*told(m+2,2*n+1,z)
     .          +(1d0/rescirc(m+1))*told(m+1,2*n+1-1,z) )
     . /(1d0/restubegrnd5+1d0/(2d0*resrad(m+1))+1d0/rescirc(m+1))
      end do

c  Energy Balance on the node at the center of the grid.
      do z = 1,nz,1
      numer = 0d0
      do c = 1,n
      numer = numer + (1d0/resrad(1))*told(2,c,z)
      end do

      do c = n+2,2*n+1
      numer = numer + (1d0/resrad(1))*told(2,c,z)
      end do

      numer = numer + (1d0/resradmid(1))*told(2,n,z)

      do c = 1,2*n+1
      tnew(1,c,z)=numer/((2d0*n)/resrad(1)+1d0/resradmid(1))
      end do

      end do



c  -----------------------------------------------------------------------
c  Run ground nodes at deltat2 time step.
```

```
c  ------------------------------------------------------------------------------------
      do z = 1,nz,1
      do r = m+2,nr-1,1
      do c = 2,n-1,1
      tnew(r,c,z) = told(r,c,z)
     . +(grndcap(r)/resrad(r-1))*(told(r-1,c,z)-told(r,c,z))
     . +(grndcap(r)/resrad(r))*(told(r+1,c,z)-told(r,c,z))
     . +(grndcap(r)/rescirc(r))*(told(r,c-1,z)-told(r,c,z))
     . +(grndcap(r)/rescirc(r))*(told(r,c+1,z)-told(r,c,z))
      end do
      end do
      end do

      do z = 1,nz,1
      do r = m+2,nr-1,1
      do c = n+3,2*n,1
      tnew(r,c,z) = told(r,c,z)
     . +(grndcap(r)/resrad(r-1))*(told(r-1,c,z)-told(r,c,z))
     . +(grndcap(r)/resrad(r))*(told(r+1,c,z)-told(r,c,z))
     . +(grndcap(r)/rescirc(r))*(told(r,c-1,z)-told(r,c,z))
     . +(grndcap(r)/rescirc(r))*(told(r,c+1,z)-told(r,c,z))
      end do
      end do
      end do


c  At the center of the grid n,(n+1),(n+2) the deltatheta must be altered
      do z = 1,nz,1
      do r = m+2,nr-1,1
      tnew(r,n,z) = told(r,n,z)
     . +(grndcap(r)/resrad(r-1))*(told(r-1,n,z)-told(r,n,z))
     . +(grndcap(r)/resrad(r))*(told(r+1,n,z)-told(r,n,z))
     . +(grndcap(r)/rescirc(r))*(told(r,n-1,z)-told(r,n,z))
     . +(grndcap(r)/rescircmid(r))*(told(r,n+1,z)-told(r,n,z))
      end do
      end do

      do z = 1,nz,1
      do r = m+2,nr-1,1
      tnew(r,n+1,z) = told(r,n+1,z)
     . +(grndcapmid(r)/resradmid(r-1))*(told(r-1,n+1,z)-told(r,n+1,z))
     . +(grndcapmid(r)/resradmid(r))*(told(r+1,n+1,z)-told(r,n+1,z))
     . +(grndcapmid(r)/rescircmid(r))*(told(r,n,z)-told(r,n+1,z))
     . +(grndcapmid(r)/rescircmid(r))*(told(r,n+2,z)-told(r,n+1,z))
```

```
      end do
      end do


      do z = 1,nz,1
      do r = m+2,nr-1,1
      tnew(r,n+2,z) = told(r,n+2,z)
     . +(grndcap(r)/resrad(r-1))*(told(r-1,n+2,z)-told(r,n+2,z))
     . +(grndcap(r)/resrad(r))*(told(r+1,n+2,z)-told(r,n+2,z))
     . +(grndcap(r)/rescircmid(r))*(told(r,n+1,z)-told(r,n+2,z))
     . +(grndcap(r)/rescirc(r))*(told(r,n+3,z)-told(r,n+2,z))
      end do
      end do


c  Energy Balance on nodes along adiabatic boundary (c=1 and c=2*n+1)
      do z = 1,nz,1
      do r = m+2,nr-1,1
      tnew(r,1,z) = told(r,1,z)
     . +(grndcap(r)/resrad(r-1))* (told(r-1,1,z)-told(r,1,z))
     . +(grndcap(r)/resrad(r))*(told(r+1,1,z)-told(r,1,z))
     . +(2d0*grndcap(r)/rescirc(r))*(told(r,2,z)-told(r,1,z))

      tnew(r,2*n+1,z) = told(r,2*n+1,z)
     . +(grndcap(r)/resrad(r-1))* (told(r-1,2*n+1,z)-told(r,2*n+1,z))
     . +(grndcap(r)/resrad(r))*(told(r+1,2*n+1,z)-told(r,2*n+1,z))
     . +(2d0*grndcap(r)/rescirc(r))*(told(r,2*n+1-1,z)-told(r,2*n+1,z))
      end do
      end do


c  Calculate the average value for the fluid temperature.
      tfluidavg = 0d0
      qoutavg = 0d0
      do tms3 = 1,times3,1
       do tms2 = 1,times2,1
        tfluidavg = tfluidavg + tfluid(tms2,tms3)
        qoutavg = qoutavg + qout(tms2,tms3)
       end do
      end do

      do tms4 = 1,times4,1
       tfluidavg = tfluidavg + tfluid(tms4,times3+1)
       qoutavg = qoutavg + qout(tms4,times3+1)
      end do

      tfluidavg = tfluidavg/(times2*times3+times4)
```

```
      qoutavg = qoutavg/(times2*times3+times4)


c  Store the temperatures in the S-array before going back to TRNSYS
c     ISTORE = INFO(10)
      do z = 1,nz,1
       do c = 1,2*n+1,1
        do r = 1,nr,1
         S(ISTORE+6+8*nr+(z-1)*(2*n+1)*nr + nr*(c-1) + r) = tnew(r,c,z)
        end do
       end do
      end do



c  calculate the average storage temperature

      vt = 0d0
      do z = 1,nz,1
       do r = m+2,nr,1
        vt = vt + (vol(r)/2)*tnew(r,1,z)
        vt = vt + (vol(r)/2)*tnew(r,2*n+1,z)
       end do
      end do

      do z = 1,nz,1
       do c = 2,n,1
        do r = m+2,nr,1
         vt = vt + vol(r)*tnew(r,c,z)
        end do
       end do
      end do

      do z = 1,nz,1
       do c = n+2,2*n,1
        do r = m+2,nr,1
         vt = vt + vol(r)*tnew(r,c,z)
        end do
       end do
      end do

      do z = 1,nz,1
       do r = m+2,nr,1
        vt = vt + volmid(r)*tnew(r,n+1,z)
       end do
```

```
      end do

      if (boundary.ne.1) then
      grvol = (depth*pi)*(rad(nr))**2 - pi*depth*radmid(m+1)**2
      end if

      if (boundary.eq.1) then
      grvol = (depth*pi)*(radmid(nr-1))**2 - pi*depth*radmid(m+1)**2
      end if

      tgravg = vt/(grvol/2)


c    IF (TIME.GE.TFINAL) THEN
c      do z = 1,nz,40
c        write(LUW,900) (tnew(r,7,z),r=17,2,-1),told(1,1,z),
c    .              (tnew(r,2*n+1-6,z),r=2,17,1)
c      write(LUW,900) (tnew(r,6,z),r=17,2,-1),told(1,1,z),
c    .              (tnew(r,2*n+1-5,z),r=2,17,1)
c      write(LUW,900) (tnew(r,5,z),r=17,2,-1),told(1,1,z),
c    .              (tnew(r,2*n+1-4,z),r=2,17,1)
c      write(LUW,900) (tnew(r,4,z),r=7,2,-1),told(1,1,z),
c    .              (tnew(r,2*n+1-3,z),r=2,7,1)
c      write(LUW,900) (tnew(r,3,z),r=7,2,-1),told(1,1,z),
c    .              (tnew(r,2*n+1-2,z),r=2,7,1)
c      write(LUW,900) (tnew(r,2,z),r=7,2,-1),told(1,1,z),
c    .              (tnew(r,2*n+1-1,z),r=2,7,1)
c       write(LUW,900) (tnew(r,1,z),r=7,2,-1),told(1,1,z),
c    .              (tnew(r,2*n+1,z),r=2,7,1)
c        write(LUW,*) "     "
c     end do
c    ENDIF
c 900  format(1x,17f7.2,1x,f7.2,1x,17f7.2)

c      if(tms3.gt.9) then
c     CALL MYSTOP(1001)
c     RETURN 1
c      end if
C
****************************************************************************
********
C **********OUTPUTS TO THE
PROBLEM*********************************************
C
```

```
***********************************************************************
********
```

```
        OUT(1) = galfluid
        OUT(2) = tfluidavg
        OUT(3) = qoutavg
        OUT(4) = tgravg

        RETURN 1
        end
```

REFERENCES

Bennet J., Claesson J., Hellsrom G., 1987, Multipole Method to Compute the Conductive Heat Flows to and Between Pipes in a Cylinder, Notes on Heat Transfer 2-1987, Depts. of Building Physics and Mathmatical Physics, Lund Institute of Technology, Box 118, S-221 00 Lund, Sweden.

Bose, J.E. ,1984, Closed-loop ground-coupled heat pump design manual, Stillwater, OK: Oklahoma State University, Engineering Technology Exstension.

Cengel, Yunus A. and Boles, Michael A.,1989, Thermodynamics an Engineering Approach, McGraw-Hill, Inc., New York, NY.

Claesson J., and Dunard, A.,1983, Heat Extraction from the Ground by Horizontal Pipes - A Mathematical Analysis, Swedish Council for Building Research, Document DI.

Deerman, J.D. and S.P. Kavanaugh, 1991, Simulation of Vertical U-tube Ground -Coupled Heat Pumps Using the Cylindrical Heat Source Solution, ASHRAE Transactions, Atlanta, Vol 97 part 1, pp 287-295.

Dobson, Monte K., O' Neal, Dennis L., Aldred, William, 1995, A Modified Analytical Method for Simulating Cyclic Operation of Vertical U-tube Ground-Coupled Heat Pumps, Solar Engineering 1995, ASME/JSEE/KSES International Solar Energy Conference, Vol 1, pp 69-76.

Ferziger, Joel H., 1981, Numerical Methods for Engineering Application, John Wiley and Sons, Inc. New York, NY.

Hellstrom, Goran, 1991, Ground Heat Storage: Thermal Analyses of Duct Storage Systems, Department of Mathematical Physics, University of Lund, Sweden.

114

Hughes, P.J., White, D. L., Huang, H.L., and Shonder, J. A., The Evaluation of a Community-Wide Residential Geothermal Heat Pump Retrofit at Fort Polk, LA, Oak Ridge National Laboratory

Incropera, Frank P. and DeWill, David P., 1990, Introduction to Heat Transfer, John Wiley and Sons, New York, NY.

Ingersoll, L.R., O.J. Zobel, and A.C. Ingersoll. 1954, Heat Conduction with Engineering, Geological, and Other Applications, New York: McGraw Hill.

Kavanaugh, S.P., 1984, Simulation and Experimental Verification of Vertical Ground Coupled Heat Pump Systems., Ph.D. dissertation, Oklahoma State University.

Kavanaugh, S.P.,1993, Ground Source Heat Pumps: Design of Geothermal Systems for Commerical and Institution Buildings, Oklahoma State University, Stillwater, OK.

Klein, S.A., et al., TRNSYS - A Transient System Simulation Program, version 14.1 Solar Energy Laboratory, University of Wisconsin-Madison, Madison, WI.

Klein, S.A., et al., FEHT - A Finite Element Heat Transfer Program, Solar Energy Laboratory, University of Wisconsin-Madison, Madison, WI.

Kusuda T., and Archenbach, P.R. "Earth Temperature and Thermal Diffusivity at Selected Stations in the United States", ASHRAE Trans., Vol. 71, Part 1, 1965.

Mei, V.C. and Fischer, S. K., 1984, A Theoretical and Experimental Analysis of Vertical Concentric-Tube Ground-Coupled Heat Exchangers, Oak Ridge National Laboratory, Contract-153.

Mitchell, John W. and Myers, Glen E., 1968, An Analytical Model of the Countercurrent Heat Exchanger Phenomena, Biophysical Journal, 1968, Vol. 8, No. 8, pp 897 -911.

Muraya, Norman K., 1994, Numerical Modelling of the Transient Thermal Interference of Vertical U-Tube Heat Exchangers, Texas A&M University.

Oklahoma State University,1989, CLGS - Closed-Loop/Ground-Source Heat Pump Design.

Oklahoma State University,1988, Closed-Loop/Geothermal Heat Pump Systems: Design and Installation Standards, Stillwater, OK: Prepared for the National Rural Electric Cooperative Association and distributed by the International Ground-Source Heat Pump Association.

Schaeffer, Martin, 1995, Computer Simulation and Experimental Analysis of Transport Phenomena During the Thermal Processing of Meat Emulsion Products, University of Wisconsin-Madison.

Thorton, Jeff, et al., 1996, Comparison of Ground Source Heat Pump Design Sizing Programs, Thermal Energy System Specialists, Madison, WI.