1 of 1

STD
SAFETY TECHNOLOGY DEPARTMENT

WSRC-TR-93-528

Keywords: Saltstone
Z-Area

Retention: Permanent

**Heat Transfer Modelling of the
Saltstone Pouring and Curing Process (U)**

Task Number: 93-016-0

By

M. A. Shadday Jr.

ISSUED: November 1993

Authorized Derivative Classifier

SRTC

**SAVANNAH RIVER TECHNOLOGY CENTER, AIKEN, SC 29808
Westinghouse Savannah River Company
Prepared for the U. S. Department of Energy under
Contract DE-AC09-89SR18035**

MASTER

DISTRIBUTION OF THIS DOCUMENT IS UNLIMITED

875

PROJECT: Defense Waste Storage
DOCUMENT: WSRC-TR-93-528
TITLE: Heat Transfer Modelling of the Saltstone
Pouring and Curing Process (U)
TASK NUMBER: 93-016-0
TASK TITLE: Saltstone Heat Transfer Modelling

APPROVALS

G. J. Hutchens, Technical reviewer

Date: 11-2-93

J. W. Jerrell, Technical reviewer

Date: 11-2-93

J. R. Pelfrey, Manager (EM&SG)

Date: 11-5-93

M. R. Buckner, Manager (ATS)

Date: 11-5-93

Table of Contents

Nomenclature	1
Introduction and Summary	3
Description of the Saltstone Disposal Vaults	3
Model Description	5
Results	12
Conclusions	26
References	27
Appendix A	Conduction Model Finite-Difference Equations 29
Appendix B	Moving Boundary Analytical Solution 35
Appendix C	Saltstone Surface Heat Transfer 37
Appendix D	gndtmp.f Code Listing 43
Appendix E	Pour Models Code Listings 47

Nomenclature

A	Area
C_p	Specific heat
g	Gravitational acceleration
h, h_{cv}	Convection heat transfer coefficient
h_{nc}	Natural convection heat transfer coefficient
h_{rad}	Thermal radiation heat transfer coefficient
k	Thermal conductivity
k_{eq}	Equivalent thermal conductivity for two adjacent dissimilar materials
m	Mass
$\dot{m}$	Mass flow rate
Nu	Nusselt number
Nu_L	Average Nusselt number over the length of a flat plate
Pr	Prandtl number
Q	Conduction heat transfer rate
q^*	Internal heat generation rate per unit mass
q_{cond}	Conduction heat transfer rate
q_{rad}	Thermal radiation exchange between the saltstone and the roof
q_{sky}	Thermal radiation exchange with the air
q_t	Solar radiation
Ra	Rayleigh number
Re_L	Reynolds number for flow over a flat plate with a length L
T	Temperature
T_{air}	Air temperature
T_b	Temperature of the solid node adjacent to the fill node
T_f	Free stream temperature for convection heat transfer

T_m	Boundary temperature between two nodes with dissimilar materials
T_o	Saltstone pour temperature
T_{rf}	Roof temperature
T_{sky}	Effective air temperature for thermal radiation exchange with a surface
T_{ss}	Saltstone surface temperature
T_{top}	Temperature of the fill node
t	Time
Δt	Time step
z	Rate of change of depth
Δz	Numerical model mesh size
α	Thermal diffusivity
α_{sol}	Solar absorptivity
β	Bulk coefficient of thermal expansion
ϵ_{rf}	Emissivity of the roof
ϵ_{ss}	Emissivity of the saltstone
ν	Kinematic viscosity
ρ	Density
σ	Stefan-Boltzmann constant

Introduction and Summary

A byproduct of the in tank precipitation, ITP, process will be 25 million gallons of low-level salt solution. This salt solution will be mixed with cement and a flyash/slag mixture and solidified in surface vaults in the Z-area Saltstone Facility. The curing process of saltstone involves exothermic reactions, and there is a maximum temperature limit of 90°C for the curing saltstone. If this temperature limit is exceeded, the physical properties of the saltstone can be degraded. A heat transfer model of the saltstone pouring and curing process has been developed that predicts transient temperature distributions in the curing saltstone. The purpose of this model is to predict peak temperatures as functions of the several independent variables in this process: pour temperature, the pour schedule, and seasonal variations in the ambient temperature.

The peak temperature of the saltstone is very sensitive to the internal heat generation that accompanies the curing process. Most of the energy is released over a short period of several hours, and the balance is released slowly over a period of time that can be in excess of a month. This long term low level internal heat generation is difficult to measure in laboratory calorimetry tests, and it can significantly influence the peak temperature in the saltstone. Due to the low thermal conductivity of the saltstone, the central region of the poured saltstone will essentially heat up adiabatically. The time dependence of the internal heat generation rate was determined from an analysis of the 1991 pilot pour test, (Shadday, 1993). With a pour schedule of eight hours a day and five days a week in the summer, the model predicts that the saltstone will have a peak temperature of 98°C with a pour temperature of 45°C, and a peak temperature of 88°C with a pour temperature of 30°C. With a pour schedule of three days a week, the peak temperature will be 88°C with a pour temperature of 45°C, and 80°C with a pour temperature of 30°C.

Description of the Saltstone Disposal Vaults

The ITP process will produce a stream of decontaminated salt solution as a byproduct of the high-level waste processing. This salt solution will be disposed of in the form of saltstone in above ground vaults. Figure 1 is a schematic of the saltstone pouring process. In the saltstone grouting facility, the salt solution will be mixed with furnace waste, flyash and slag, and cement to form a grout. This grout will be poured into concrete vaults and solidified to form saltstone. When a vault is filled, it will be covered with a waterproofed concrete cap, to form a permanent storage site.

The vaults consist of cells that are one hundred feet square and twenty-five feet deep. The grout, hereafter termed saltstone, will be poured into the cells in a steady-flow manner with a maximum flow rate of 1362 ft³/hr. At this flow rate, a cell will fill at the rate of 1.6 in/hr. Pouring in a cell will continue, with periodic interruptions, until the cell is full, and the depth of the saltstone is twenty-five ft. The pour schedule for a cell, "x" hours per day and "y" days per week, will be determined in part by the results of this analysis. An important constraint is that the curing saltstone temperature cannot exceed 90°C.

Figure 1: Schematic of the saltstone pouring process

Figure 2 is a drawing of a vertical cross section of a cell. The cell bottom is a 2.5 ft thick concrete pad that is at ground level, and it rests on a 1 ft thick layer of gravel and a 2 ft thick layer of clay. While pouring is in progress, a cell will be covered with a corrugated metal roof that is painted cream-colored and is not insulated. The cells are not forced ventilated. There is a gap between the underside of the roof and the top of the cell wall, but there are no ventilation openings in the roof, so there are no obvious paths for natural convection induced ventilation of the cells. It is reasonable to assume that the air temperature in a cell can vary significantly from the ambient air temperature.

Figure 2: Vertical cross section of a saltstone vault cell

Model Description

A transient heat transfer model of the saltstone pouring and curing process has been developed. The purpose of this model is to predict the peak temperature of the curing saltstone and identify pouring parameters such that this temperature does not exceed the technical standard of 90°C, (Wilhite, 1988). The saltstone heats up due to exothermic reactions during the curing process, and there will be a net heat loss through all of the boundaries. The peak temperature will therefore occur in the central region of the saltstone slab, which will be 100 ft square and at the maximum 25 ft thick. Saltstone has a very low thermal conductivity, so the influence of side wall heat transfer losses will be confined to the vicinity of the walls. The heat transfer in the central region of the saltstone will be essentially one-dimensional in the vertical direction, and this considerably simplifies the modelling. The model predicts the vertical temperature distributions in the saltstone during both the pouring phase and the period after the cell is filled. Input variables are: the pour schedule, the pour temperature, and the time of year that the pour takes place.

This is a one-dimensional finite-difference heat conduction model with a variable mesh, and a moving boundary at the top. The model uses the Crank-Nicolson numerical scheme, and the Thomas algorithm, (Anderson et. al., 1984), to solve the tridiagonal system of equations. The computational domain extends approximately eight meters into the ground. This is deep enough such that the lower boundary can be treated as insulated. The boundary condition at the surface of the saltstone accounts for natural convection and thermal radiation heat transfer with the metal roof. The equilibrium roof temperature is separately calculated as a function of: solar insolation, thermal radiation exchange with the sky, convection heat transfer with the surrounding air, and heat exchange with the saltstone. During the pouring phase, an analytical solution is used for the top control volume of the computational mesh that contains the moving boundary. The model starts

with the initial pouring of saltstone on the concrete floor of a cell, and the solution marches forward in time with timesteps of one hour. The mesh width for the saltstone is the thickness of a layer of saltstone that is poured in one hour, .0415 m. A variable width mesh is used for the concrete, gravel, and clay layers, and for the earth beneath the cell.

The governing equation for one-dimensional transient heat conduction, with internal heat generation, is equation 1. The finite-difference equations were derived with the control-volume approach. Figure 3 shows three interior nodes in a variable-width mesh, and equation 2 is the governing equation for the i^{th} node. The control volume to which the heat balance is applied is $\Delta V = A \Delta z_i$. The thermal conductivities are equivalent thermal conductivities for the cases where adjacent control volumes contain different materials. The last term in the equation accounts for internal heat generation.

$$\frac{\partial}{\partial z} \left(k \frac{\partial T}{\partial z} \right) + q^* = \rho C_p \frac{\partial T(z,t)}{\partial t} \quad (1)$$

Figure 3: A one-dimensional variable width mesh

$$\begin{aligned} (\rho C_p \Delta z)_i A \frac{T_i^{n+1} - T_i^n}{\Delta t} = & -k_{i-1,i} A \frac{\frac{1}{2}(T_{i-1}^{n+1} + T_{i-1}^n) - \frac{1}{2}(T_{i-1}^{n+1} + T_{i-1}^n)}{\frac{1}{2}(\Delta z_{i-1} + \Delta z_i)} \\ & + k_{i,i+1} A \frac{\frac{1}{2}(T_{i+1}^{n+1} + T_{i+1}^n) - \frac{1}{2}(T_i^{n+1} + T_i^n)}{\frac{1}{2}(\Delta z_i + \Delta z_{i+1})} + \rho_i \Delta z_i A q_i^* \end{aligned} \quad (2)$$

Similar equations are derived for the bottom insulated boundary and the top boundary during periods when pouring of the saltstone is not taking place. All of the finite-difference equations are derived and manipulated into the form that they are used by the code in appendix A.

The moving boundary is handled with an analytical solution. Figure 4 is a schematic of a liquid being poured on a heated surface at a constant flow rate. The heat transfer through the bottom surface is assumed to be constant, and Newton's law of cooling applies for heat transfer at the free surface. Equation 3 is the governing differential equation for this initial value problem, and with the initial condition that the mass is zero, equation 4 is the solution. This solution is incorporated directly into the tridiagonal system of finite-difference equations. The solution to equation 3 is developed in appendix B, and in the

appendix the solution is further manipulated into the form appropriate for the finite-difference model.

Figure 4: A schematic of a liquid being poured on a heated surface

$$C_p \frac{d}{dt}(mT) = \dot{m}C_p T_o + \dot{Q} - hA(T - T_f) \quad (3)$$

where $m(0) = 0$

$$T = \frac{T_o + \frac{\dot{Q}}{\dot{m}C_p} + \frac{hA}{\dot{m}C_p} T_f}{1 + \frac{hA}{\dot{m}C_p}} \quad (4)$$

Heat transfer at the surface of the saltstone is treated as heat transfer between two horizontal plates, the saltstone surface and the lower surface of the roof. The slope of the roof is neglected. Equation 5 is an expression for the natural convection heat transfer coefficient for two horizontal plates when the lower plate is hotter than the upper plate, (Hollands et al., 1975). If the upper plate is hotter, the air will stratify, and the heat transfer will be by conduction through the air layer. This mode of heat transfer is negligible in comparison with the other modes, and it is therefore neglected in this analysis. The enhancement of natural convection heat transfer by evaporation at the saltstone free surface and condensation on the under side of the roof was shown to be approximately two percent, (Shadday, Lab Notebook #2, 1993), and it is therefore also neglected. Thermal radiation heat transfer between the saltstone and the roof is important. The assumed emissivities of the saltstone and the roof are respectively 0.94 and 0.92, (Ozisik, 1985). The thermal radiation is calculated with an effective radiation heat transfer coefficient, equation 6. This allows all of the surface heat transfer to be calculated with Newton's law of cooling, equation 7.

$$h_{nc} = .0555k \left(\frac{\beta g \Delta T}{\nu \alpha} \right)^{1/3} \quad (5)$$

$$h_{rad} = \frac{\sigma(T_{ss} + T_{rf}) * (T_{ss}^2 + T_{rf}^2)}{\frac{1}{\epsilon_{ss}} + \frac{1}{\epsilon_{rf}} - 1} \quad (6)$$

$$q = (h_{rad} + h_{nc}) * (T_{ss} - T_{rf}) \quad (7)$$

The roof temperature is embedded in the surface boundary condition, and is therefore calculated separately. The thermal capacitance of the roof is neglected, so it is assumed to always be at thermal equilibrium. The roof is subjected to: heat transfer with the saltstone on the under side, and on the top convection heat transfer with air at the ambient temperature, thermal radiation exchange with the sky, and solar heating during the day. The solar absorptivity is assumed to be 0.3. The assumed solar radiation, as a function of the time of day, is shown in figure 5. Equation 8 is the expression for the solar radiation. This is typical for a sunny late spring or early fall day, (Duffie & Beckman, 1974). The annual change in the daily duration of sun light is not included in this model. The roof exchanges thermal radiation with the sky. The roof emissivity is assumed to be 0.92, the same as for the heat transfer with the saltstone. Equation 9 is the relation for the effective sky temperature for thermal radiation, (Duffie & Beckman, 1974). The convection heat transfer coefficient for heat transfer with the surrounding air is assumed to be 5.0 W/m²K. This is the approximate mean heat transfer coefficient for forced convection with a 5 mph wind over a 100 ft long flat plate. Writing a heat balance for the roof and simplifying results in a fourth order polynomial for the roof temperature, equation 10. This equation is solved with Newton's iterative method. Appendix C is a detailed description of the saltstone surface heat transfer.

$$q_t = 800 \cdot \sin\left(\frac{time - 7}{12} \pi\right) + 100 \quad (8)$$

$$T_{sky} = 0.0552 T_{air}^{1.5} \quad (9)$$

$$\left[\epsilon_{rf} \sigma + \frac{\sigma}{\frac{1}{\epsilon_{rf}} + \frac{1}{\epsilon_{ss}} - 1} \right] T_{rf}^4 + (h_{cv} + h_{nc}) T_{rf} - \left[\alpha_{sol} q_t + \epsilon_{rf} \sigma T_{sky}^4 + h_{cv} T_{air} + h_{nc} T_{ss} + \frac{\sigma T_{ss}^4}{\frac{1}{\epsilon_{rf}} + \frac{1}{\epsilon_{ss}} - 1} \right] = 0 \quad (10)$$

Figure 5: Solar radiation on the roof

The ground under the cell is also a heat sink for the saltstone. The initial temperature distribution in the ground is a function of the time of year that pouring commences. The code `gndtmp.f`, written in appendix D, is used to calculate the initial ground temperature distribution. This is a one-dimensional transient conduction model with the same mesh spacing as the bottom 38 nodes of the pour model. The ground temperature model is subjected to periodic natural convection boundary conditions and the annual variation of the average daily air temperature. The ground is modelled to a depth of approximately 8 meters. This is deep enough for the influence of the annual variation in surface temperature to not be detected. Figure 6 is a plot of periodic ground temperature distributions below a saltstone vault cell.

In this model, the internal heat generation rate is defined as a function of the elapsed time from pouring. The complexities of the reactions involved in the curing of saltstone require that the time dependence of the internal heat generation rate be determined empirically. The function used in this model was determined from an analysis of thermocouple data of a 1991 Z-area pilot pour test, (Shadday, 1993). The internal heat generation rate for the first forty hours of elapsed time was determined from a direct analysis of the thermocouple data and the assumption that the process was adiabatic. The subsequent slow rate of residual internal heat generation, that persisted for a month, was determined by benchmarking a transient heat transfer model of the pilot pour with the data and using the residual internal heat generation as an independent variable. Figure 7 shows the assumed internal heat generation rate for the saltstone for the first forty hours. There is a peak in the internal heat generation at approximately five hours. This is termed the second hydration peak, and the setting of the saltstone occurs in this spike, (Langton, 1986). Equations 11 through 20 are the expressions for the internal heat generation rate per unit mass, (W/kg).

Figure 6: Periodic ground temperature distributions below a saltstone vault cell

$$\dot{q} = 0.0 \rightarrow t \leq 2.5hr. \quad (11)$$

$$\dot{q} = \frac{C_p}{3600} \left[8.2 * \sin \left(1 - \frac{t-2.5}{5.5} \right) \pi \right] \rightarrow 2.5 \leq t \leq 7.5hr. \quad (12)$$

$$\dot{q} = \frac{C_p}{3600} [2.31 - .23875 * (t - 7.5)] \rightarrow 7.5 \leq t \leq 15.5hr. \quad (13)$$

$$\dot{q} = \frac{C_p}{3600} \left[.4 + 1.7 * \sin \left(1 - \frac{t-15.5}{1.5} \right) \pi \right] \rightarrow 15.5 \leq t \leq 17.0hr. \quad (14)$$

$$\dot{q} = \frac{C_p}{3600} [.3 * e^{-.036484 * (t-17.0)}] \rightarrow 17.0 \leq t \leq 45.0hr. \quad (15)$$

$$\dot{q} = \frac{C_p}{3600} [.10801 - .0006 * (t - 45.0)] \rightarrow 45.0 \leq t \leq 175.0hr. \quad (16)$$

$$\dot{q} = \frac{C_p}{3600} \left[.4 + 1.7 * \sin \left(1 - \frac{t-15.5}{1.5} \right) \pi \right] \rightarrow 15.5 \leq t \leq 17.0 \text{ hr.} \quad (17)$$

$$\dot{q} = \frac{C_p}{3600} \left[.02 + .01 * e^{-.065788 * (t-275.0)} \right] \rightarrow 275.0 \leq t \leq 450.0 \text{ hr.} \quad (18)$$

$$\dot{q} = \frac{C_p}{3600} \left[.015 + .005 * e^{-.065788 * (t-450.0)} \right] \rightarrow 450.0 \leq t \leq 800.0 \text{ hr} \quad (19)$$

$$\dot{q} = 0.0 \rightarrow t \geq 800.0 \text{ hr.} \quad (20)$$

Figure 7: Internal heat generation of saltstone as a function of elapsed time from pouring

Each nodal control volume in the model will have a different heat generation rate at a point in time because the elapsed time since pouring is different for each node. Figure 8 shows the internal heat generation rates for three nodes in the model, with a pour schedule of 8 hours per day and 5 days per week. Node #39 represents the saltstone poured in the first hour of the first day, node #50 is filled in the middle of the pouring period of the second day, and node #58 is filled in the middle of the pouring period of the third day. Node #58 is filled 24 hours after node #50, and this 24 hour delay is clearly seen in the time difference between the second hydration peaks of the two nodes.

Figure 8: Internal heat generation rates for three saltstone nodes in the pour model

Appendix E contains a Fortran listing of the numerical model. The main programs are *saltst.f* for 5 days/week pour schedules and *xsalt.f* for the 3 days/week pour schedules. The subroutines are common to both main programs. The coding is extensively commented. The codes were run on a Macintosh II fx computer.

Results

A number of saltstone pouring scenarios were modelled to determine the peak temperature that would occur during the pouring and curing phases. The predicted peak temperatures varied between 70°C and 98°C with, as expected, the higher temperatures occurring during the summer. Summer and winter pours were modelled, with start pour dates of June 30 and December 31 for the cases in which the pouring took place five days a week. Figure 9 shows the saltstone depth versus elapsed time for the cases in which pouring is scheduled eight hours a day and five days a week. The vault cell is filled in four and a half weeks, and the summer and winter start pour dates have the saltstone pouring occurring during the hottest part of the summer and the coldest part of the winter. Figure 10 shows the initial vertical temperature distributions in the ground below the empty vault cells for the two seasonal pours.

Figure 9: Saltstone depth vs elapsed time, pouring 8 hours/day and 5 days/week

The base case with which all others are compared is: pouring scheduled 8 hours/day and 5 days/week, a start pour date of June 30, and a pour temperature of 30°C. Figure 11 shows vertical temperature profiles in the saltstone at the ends of the five day pouring weeks, for the first four weeks. There is a local temperature peak 8 inches below the surface. This is caused by the second hydration peaks of the saltstone poured on the last day of the week. Figure 12 shows vertical temperature distributions in the saltstone for several different weeks in the post pouring curing period. The peak temperature occurs at an elapsed time of 7 weeks. Figure 13 shows the daily peak saltstone temperatures versus elapsed time, for a total elapsed time of one year. The peak temperature is approximately 88°C at 7 weeks, and after one year the peak saltstone temperature is 54°C.

Figure 10: Ground temperature distributions on June 30 and Dec. 31

Figure 14 shows the influence of the pour temperature on the peak saltstone temperature, for 5 day/week summer pours. Raising the pour temperature from 30°C to 45°C raises the peak saltstone temperature by approximately 10° to 98°C. The threshold pour temperature, that results in a peak temperature of 90°C, is approximately 32°C. Figure 15 shows the influence of pour temperature on the peak saltstone temperature for 5 day/week winter pours. With a pour temperature of 30°C, the peak temperature is approximately 81°C, and raising the pour temperature by 15° to 45°C again raises the peak temperature by 10°C.

Figure 11: Vertical temperature profiles in the saltstone during the pouring phase, pour temperature 30°C, pouring 8 hours/day and 5 days/week

Figure 12: Vertical temperature profiles in the saltstone after the pouring phase is completed, pour temperature 30°C, 8 hours/day and 5 days/week

Figure 13: Saltstone daily peak temperature vs elapsed time

Figure 14: Saltstone daily peak temperature vs elapsed time and pour temp.

Figure 15: Saltstone daily peak temperature vs elapsed time and pour temp.

Pouring the saltstone 5 days/week in the summer, results in peak temperatures close to or exceeding 90°C. Pouring in two different cells on alternate days would reduce the rate at which an individual cell filled and still allow a high throughput to be maintained by the saltstone facility. Figure 16 shows the saltstone depth versus elapsed time for an 8 hour/day, 3 day/week pour schedule. This is a Monday/Wednesday/Friday or a Tuesday/Thursday/Saturday pour schedule. It takes seven and a half weeks to fill a cell. Figure 17 shows the daily peak temperatures for a summer pour. The start pour date has been moved back to June 15 because of the longer pour period. The peak temperature occurs in the tenth week. With a pour temperature of 30°C, the peak temperature is approximately 80°C, and raising the pour temperature to 45°C, raises the peak temperature to approximately 88°C. The oscillations in the daily peak temperature for the first three weeks are due to the saltstone cooling in the 40 hours that separate the end of one pour period and the start of the next pour period. After three weeks the saltstone slab is thick enough such that the temperature near the middle is not affected by the surface fluctuations, and the peak temperature occurs near the middle of the slab. Figure 18 shows the daily peak temperatures for winter pours. With a pour temperature of 30°C, the peak temperature is approximately 70°C, and with a pour temperature of 45°C, the peak temperature is approximately 78°C. Pouring on alternate days is apparently a viable solution to the high curing temperatures of the saltstone, though there is not a great amount of margin with respect to the 90°C temperature limit.

Figure 16: Saltstone depth vs elapsed time

Figure 17: Saltstone daily peak temperature vs elapsed time and pour temp.

Figure 18: Saltstone daily peak temperature vs elapsed time and pour temp.

The internal heat generation rate, as a function of elapsed time for the first forty hours, was determined from a direct analysis of the 1991 pilot pour test thermocouple data, (Shadday, 1993); essentially this was a calorimetry test. The balance of the assumed internal heat generation was indirectly inferred from a transient heat transfer analysis of the pilot pour test. To determine the importance of the residual internal heat generation that is assumed to persist for a month, I neglected the internal heat generation after 45 hours of elapsed time. Figure 19 shows the daily peak saltstone temperature versus elapsed time for the scenario with the modified internal heat generation rate. Also shown are results with the nominal internal heat generation rate. The case is for a summer pour with an 8 hour/day and 5 day/week pour schedule. Neglecting the residual internal heat generation lowers the predicted peak saltstone temperature by 16°C to 72°C . Figure 20 shows vertical temperature profiles in the saltstone at the ends of the five day pouring weeks, for the first four weeks. The peak temperature occurs just below the surface, and it is due to the second hydration peak of the saltstone poured on the last day of the week. The saltstone near the surface cools over the weekend, when pouring is interrupted, and this accounts for the oscillations in the daily peak temperatures for the first five weeks. The residual internal heat generation is very important in predicting the peak temperature of the curing saltstone. For the assumed internal heat generation relation that was used in this analysis, 67% of the energy was released in the first 40 hours. Neglecting the residual internal heat generation results in underestimating the exothermic energy by 33%.

Figure 19: Saltstone daily peak temperature vs elapsed time. The modified internal heat generation all occurs within 45 hours of pouring.

Figure 20: Vertical temperature profiles in the saltstone during the pouring phase, pour temperature 30°C, pouring 8 hours/day and 5 days/week. All of the exothermic reactions take place within 45 hours of pouring.

There is a large variation in the time delay or dormant period between pouring of the saltstone and the onset of the second peak of hydration, (Hay, 1991 & Langton, 1986). For this analysis, the second hydration peak was assumed to occur approximately 5 hours after pouring. This is at the low end for construction cement, and is earlier than most of the second hydration peaks of saltstone mixtures tested with calorimeters. To determine if the timing of the second hydration peak affects the peak temperature of the curing saltstone, I delayed the onset by 20 hours. Figure 21 is a plot of the delayed internal heat generation rate as a function of elapsed time. The time dependence of the internal heat generation rate is the same as that shown in figure 7 except for the 20 hour time delay. The total amount of energy released remains the same. Figure 22 is a plot of the daily peak temperatures versus elapsed time for both the nominal case and the case with the delayed internal heat generation. In both cases the pour parameters are; a pour temperature of 30°C, a pour schedule of 8 hours/day 5 days/week, and a start pour date of June 30. A 20 hour delay in the second hydration peak increases the peak temperature by 6°C. Figure 23 shows vertical temperature distributions in the saltstone at the ends of the five day pouring weeks, for the first four weeks. The surface temperatures are much lower than in the nominal case, figure 11. Heat transfer losses through the surface are lower, and consequently the mean and peak saltstone temperatures are higher.

Figure 21: Internal heat generation of saltstone as a function of elapsed time from pouring. The internal heat generation has been delayed 20 hours from the nominal case.

Figure 22: Saltstone daily peak temperature vs elapsed time. The internal heat generation is delayed 20 hours.

Figure 23: Vertical temperature profiles in the saltstone during the pouring phase, pour temperature 30°C, pouring 8 hours/day and 5 days/week. The exothermic reactions are delayed 20 hours.

Conclusions

A heat transfer model of the saltstone pouring process has been developed that is a useful tool for predicting transient temperature distributions in the saltstone. The predicted peak temperatures are close to or exceed the technical specification of 90°C. Pouring every other day does not result in a great deal of margin between the peak temperature and the technical specification. The residual internal heat generation is significant, and the most important conclusion to be drawn from this analysis is that it is very important to accurately know the rate of internal heat generation. If laboratory calorimetry tests do not have the resolution to measure the low level residual internal heat generation rates that can persist for periods in excess of a month, experiments to measure this residual energy release rate must be devised and run. Tests similar to the 1991 pilot pour, (Shadday, 1993), are a promising technique, as long as the boundary conditions are well known. With reliable data on the internal heat generation, the transient model herein described can be used with confidence.

References

- Anderson, D. A., Tannehill, J. C., Pletcher, R. H., 1984, Computational Fluid Mechanics and Heat Transfer, Hemisphere Pub. Corp.
- Duffie, J. A., Beckman, W. A., 1974, Solar Energy Thermal Processes, John Wiley & Sons.
- Hay, M. S., 1991, "Adiabatic Temperature Rise of Curing Saltstone: A Comparative Study of Hanford and SRS Materials (U)", WSRC-RP-91-0725, October 25, 1991
- Hollands, K. G. T., Raithby, G. D., Konicek, L., 1975, "Correlation Equations for Free Convection Heat Transfer in Horizontal Layers of Air and Water", *Int. J. of Heat & Mass Transfer*, vol. 18, pp 879-884.
- Langton, C. A., 1986, "Thermal Properties of Saltstone PSU Progress Report", DPST-86-460, May 28, 1986.
- Ozisik, M. N., 1985, Heat Transfer a Basic Approach, McGraw-Hill Book Company.
- Shadday, M. A., 1993, Laboratory Notebook #1, WSRC-NB-93-59.
- Shadday, M. A., 1993, Laboratory Notebook #2, WSRC-NB-93-159.
- Shadday, M. A., 1993, "Heat Transfer Model of the February 1991 Saltstone Pilot Pour Test (U)", WSRC-TR-93-315, June 1993.
- Wilhite, E. L., 1988, "Technical Standard for the Saltstone Facility", DPSTS-210-Z, July 7, 1988.

This Page Intentionally Left Blank

Appendix A Conduction Model Finite-Difference Equations

The saltstone heat transfer pour model is a one-dimensional conduction model with a variable mesh, variable properties, internal heat generation, and periodically a moving boundary. The finite-difference equations are derived by applying a time dependent energy balance, equation a1, to each of the nodal control volumes in the computational mesh.

$$\boxed{\text{Rate of increase of internal energy}} = \boxed{\text{Net heat transfer into the control volume}} + \boxed{\text{Internal heat generation}} \quad (\text{a1})$$

Figure A1 is a schematic of an interior nodal control volume and the two adjacent nodal control volumes. In the general case, the three control volumes can contain different materials with different thermodynamic properties. Conduction between adjacent nodes with different thermal conductivities is expressed in terms of an equivalent thermal conductivity that is a function of the two thermal conductivities and the mesh widths. Equation a2 is an expression for the conduction between the i and $i-1$ nodes. T_m is the temperature at the boundary between the two control volumes. The last two expressions in equation a2 can be solved for T_m , and the result is equation a3. The equivalent thermal conductivity, equation a4, can be solved for by substituting equation a3 into equation a2.

Figure A1: Schematic of an interior node

$$q_{cond} = -k_{eq}A \frac{T_i - T_{i-1}}{1/2(\Delta z_i + \Delta z_{i-1})} = -k_{i-1}A \frac{T_m - T_{i-1}}{1/2(\Delta z_{i-1})} = -k_iA \frac{T_i - T_m}{1/2(\Delta z_i)} \quad (\text{a2})$$

$$T_m = \frac{\Delta z_{i-1}k_iT_i + \Delta z_i k_{i-1}T_{i-1}}{\Delta z_i k_{i-1} + \Delta z_{i-1}k_i} \quad (\text{a3})$$

$$k_{eq} = \frac{k_i k_{i-1}(\Delta z_i + \Delta z_{i-1})}{\Delta z_i k_{i-1} + \Delta z_{i-1}k_i} \quad (\text{a4})$$

Application of the energy balance to the i^{th} node, with the Crank-Nicolson scheme, results in equation a5. This equation is implicit and the temperatures on the right side of the equation are the arithmetic means of the new and old values.

$$\begin{aligned}
(\rho C_p \Delta z)_i A \frac{T_i^{n+1} - T_i^n}{\Delta t} = & -k_{i-1,i} A \frac{\frac{1}{2}(T_i^{n+1} + T_i^n) - \frac{1}{2}(T_{i-1}^{n+1} + T_{i-1}^n)}{\frac{1}{2}(\Delta z_{i-1} + \Delta z_i)} \\
& + k_{i,i+1} A \frac{\frac{1}{2}(T_{i+1}^{n+1} + T_{i+1}^n) - \frac{1}{2}(T_i^{n+1} + T_i^n)}{\frac{1}{2}(\Delta z_i + \Delta z_{i+1})} + \rho_i \Delta z_i A q_i^*
\end{aligned} \tag{a5}$$

Appropriate expressions for the equivalent thermal conductivities are substituted into equation a5, and it is put into the form where all of the terms that contain temperatures at the new time level are placed on the left side, and the rest of the terms are placed on the right side. The result is equation a6, the final form of the equation for the interior nodes.

$$\begin{aligned}
& - \frac{k_{i-1} k_i \Delta t}{(\rho C_p \Delta z)_i (\Delta z_{i-1} k_i + \Delta z_i k_{i-1})} T_{i-1}^{n+1} + \left[1 + \frac{k_i \Delta t}{(\rho C_p \Delta z)_i} \left(\frac{k_{i-1}}{\Delta z_{i-1} k_i + \Delta z_i k_{i-1}} + \frac{k_{i+1}}{\Delta z_{i+1} k_i + \Delta z_{i+1} k_i} \right) \right] T_i^{n+1} \\
& - \frac{k_{i+1} k_i \Delta t}{(\rho C_p \Delta z)_i (\Delta z_{i+1} k_i + \Delta z_i k_{i+1})} T_{i+1}^{n+1} = T_i^n - \frac{k_{i-1} k_i \Delta t}{(\rho C_p \Delta z)_i (\Delta z_{i-1} k_i + \Delta z_i k_{i-1})} (T_i^n - T_{i-1}^n) \\
& + \frac{k_{i+1} k_i \Delta t}{(\rho C_p \Delta z)_i (\Delta z_{i+1} k_i + \Delta z_i k_{i+1})} (T_{i+1}^n - T_i^n) + \frac{q_i^* \Delta t}{(C_p)_i}
\end{aligned} \tag{a6}$$

Figure A2 is a schematic of the bottom nodal control volume. The bottom boundary is assumed to be insulated, so heat transfer occurs only between nodes #1 and #2. Equation a7 is the heat balance applied to node #1. The internal heat generation for this node is zero, but the term is included for generality. Equation a7 is simplified in the same manner as equation a5 to obtain equation a8, the final form for the bottom nodal equation.

Figure A2: Schematic of the bottom node

$$(\rho C_p \Delta z)_1 A \frac{T_1^{n+1} - T_1^n}{\Delta t} = k_{1,2} A \frac{\frac{1}{2}(T_2^{n+1} + T_2^n) - \frac{1}{2}(T_1^{n+1} + T_1^n)}{\frac{1}{2}(\Delta z_1 + \Delta z_2)} + (\rho \Delta z)_1 A q_1^* \tag{a7}$$

$$\left[1 + \frac{k_1 k_2 \Delta t}{(\rho C_p \Delta z)_1 (\Delta z_1 k_2 + \Delta z_2 k_1)} \right] T_1^{n+1} - \frac{k_1 k_2 \Delta t}{(\rho C_p \Delta z)_1 (\Delta z_1 k_2 + \Delta z_2 k_1)} T_2^{n+1} =$$

$$T_1^n + \frac{k_1 k_2 \Delta t}{(\rho C_p \Delta z)_1 (\Delta z_1 k_2 + \Delta z_2 k_1)} (T_2^n - T_1^n) + \frac{q_1^* \Delta t}{C_p}$$
(a8)

The treatment of the top node will depend on whether pouring is taking place, and during periods of pouring, an analytical solution is used for the top node temperature. The analytical solution is developed in appendix B. Equation a9 is the analytical solution with the base temperature expressed as the arithmetic mean of the new and old values.

$$T_{top}^{n+1} = \frac{T_o + \frac{k(T_b^{n+1} + T_b^n)}{\rho \dot{z} C_p \Delta z} + \frac{h T_{rf}}{\rho \dot{z} C_p}}{\left(1 + \frac{2k}{\rho \dot{z} C_p \Delta z} + \frac{h}{\rho \dot{z} C_p} \right)}$$
(a9)

note: $\dot{z} = \frac{\Delta z_{top}}{\Delta t}$

This analytical solution for the new temperature of the top node can be incorporated directly into the numerical solution. The top or fill node has the index npx and the adjacent solid node has the index npx. Equation a10 is the expression for the fill node temperature that is in a form consistent with the finite-difference equations. The heat transfer coefficient accounts for both natural convection and thermal radiation heat transfer. All of the surface losses must be expressed in the form of Newton's law of cooling.

$$-\left[\frac{k_{npm} \Delta t}{(\rho \Delta z C_p)_{npx} \Delta z_{npm}} \right] T_{npm}^{n+1} + \left(1 + \frac{2k_{npm} \Delta t}{(\rho \Delta z C_p)_{npx} \Delta z_{npm}} + \frac{h \Delta t}{(\rho \Delta z C_p)_{npx}} \right) T_{npx}^{n+1} =$$

$$T_o + \left[\frac{k_{npm} \Delta t}{(\rho \Delta z C_p)_{npx} \Delta z_{npm}} \right] T_{npm}^n + \frac{h \Delta t T_{rf}}{(\rho \Delta z C_p)_{npx}}$$
(a10)

During periods when pouring is not taking place, the top boundary is treated conventionally with a finite-difference equation. Figure A3 is a schematic of the top node during the non-pouring phase of the operation. The node is in the center of the control volume rather than on the boundary, because at some later time this node will become an interior node. The heat transfer coefficient has to be modified because the nodal temperature is not the surface temperature. The equivalent heat transfer coefficient is derived in the same manner as the equivalent thermal conductivity for two adjacent nodes with different substances. Equations a11 through a13 outline the derivation.

Figure A3: Schematic of the top node during the non-pouring phase

$$h_{eq}A(T_{rf} - T_{npx}) = kA \frac{T_s - T_{npx}}{\Delta z/2} = hA(T_{rf} - T_s) \quad (a11)$$

$$T_s = \frac{hT_{rf} + \frac{2k}{\Delta z}T_{npx}}{h + \frac{2k}{\Delta z}} \quad (a12)$$

$$h_{eq} = \frac{h \frac{2k}{\Delta z}}{h + \frac{2k}{\Delta z}} \quad (a13)$$

Equation a14 is the heat balance on the top node. Substituting in the appropriate expression for the equivalent thermal conductivity for the top two nodes and simplifying yields equation a15, the final form of the top node finite-difference equation.

$$\begin{aligned} (\rho C_p \Delta z)_{npx} A \frac{T_{npx}^{n+1} - T_{npx}^n}{\Delta t} = & -k_{npx, npx} A \frac{\frac{1}{2}(T_{npx}^{n+1} + T_{npx}^n) - \frac{1}{2}(T_{npm}^{n+1} + T_{npm}^n)}{\frac{1}{2}(\Delta z_{npm} + \Delta z_{npx})} \\ & + h_{eq}A \left[T_{rf} - \frac{1}{2}(T_{npx}^{n+1} + T_{npx}^n) \right] + (\rho \Delta z)_{npx} A q_{npx}^* \end{aligned} \quad (a14)$$

$$\begin{aligned}
& - \frac{k_{npm}k_{npx}\Delta t}{(\rho C_p \Delta z)_{npx} (\Delta z_{npm}k_{npx} + \Delta z_{npx}k_{npm})} T_{npm}^{n+1} + \\
& \left[1 + \frac{\Delta t}{(\rho C_p \Delta z)_{npx}} \left(\frac{k_{npm}k_{npx}}{(\Delta z_{npm}k_{npx} + \Delta z_{npx}k_{npm})} + \frac{h_{eq}}{2} \right) \right] T_{npx}^{n+1} = \\
& \left[1 - \frac{k_{npm}k_{npx}\Delta t}{(\rho C_p \Delta z)_{npx} (\Delta z_{npm}k_{npx} + \Delta z_{npx}k_{npm})} - \frac{h_{eq}\Delta t}{2(\rho C_p \Delta z)_{npx}} \right] T_{npx}^n + \\
& \frac{k_{npm}k_{npx}\Delta t}{(\rho C_p \Delta z)_{npx} (\Delta z_{npm}k_{npx} + \Delta z_{npx}k_{npm})} T_{npm}^n + \frac{h_{eq}\Delta t}{(\rho C_p \Delta z)_{npx}} T_{rf} + \frac{q_{npx}^*\Delta t}{C_{p,npx}}
\end{aligned} \tag{a15}$$

The finite-difference equations form a tridiagonal system of equations, equation a16. There is a tridiagonal coefficient matrix that is multiplied by the vector of the new temperatures. The right side of equation a16 is a vector formed by the right sides of the finite-difference equations. This system of equations is solved directly with the Thomas algorithm.

$$\begin{bmatrix} d_1 & c_1 & 0 & 0 & 0 \\ a_1 & d_2 & c_2 & 0 & 0 \\ 0 & \ddots & \ddots & \ddots & 0 \\ 0 & 0 & a_{npm-1} & d_{npm} & c_{npm} \\ 0 & 0 & 0 & a_{npm} & d_{npx} \end{bmatrix} \begin{bmatrix} T_1^{n+1} \\ T_2^{n+1} \\ \vdots \\ T_{npm}^{n+1} \\ T_{npx}^{n+1} \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_{npm} \\ b_{npx} \end{bmatrix} \tag{a16}$$

This Page Intentionally Left Blank

Appendix B Moving Boundary Analytical Solution

When saltstone pouring is taking place, the top control volume in the computational mesh has a variable mass and also a moving boundary. This complicates the writing of a finite-difference equation for this node. Fortunately the temperature of this node can be solved for analytically. Consider the problem of a liquid being poured onto an initially dry solid surface at a constant rate. The liquid is cooled by convection at the free surface and subject to conduction heat transfer through the solid/liquid interface. Figure B1 is a schematic of this problem, and equation b1 is the governing differential equation.

Figure B1: Schematic of the fill node

$$C_p \frac{d}{dt}(mT_{top}) = \dot{m}C_p T_o - \frac{2kA}{\Delta z}(T_{top} - T_b) - hA(T_{top} - T_{rf}) \quad (b1)$$

where: $m(0) = 0 \quad \therefore m = \dot{m}t$

Expand the differential term in equation b1 and simplify.

$$\frac{dT_{top}}{dt} + \left(1 + \frac{2kA}{\dot{m}C_p \Delta z} + \frac{hA}{\dot{m}C_p}\right) \frac{T_{top}}{t} = \frac{T_o}{t} + \left(\frac{2kA}{\dot{m}C_p \Delta z}\right) \frac{T_b}{t} + \left(\frac{hA}{\dot{m}C_p}\right) \frac{T_{rf}}{t} \quad (b2)$$

Equation b2 has the following form, and equations of this class can be solved directly with an integrating factor that is defined by equation b4. Equation b5 is the integrating factor for equation b2, and equation b6 is the right side of equation b2 multiplied by the integrating factor and integrated.

$$\frac{dT_{top}}{dt} + P(t)T_{top} = Q(t) \quad (b3)$$

$$\phi(t) = e^{\int P(t)dt} \quad (b4)$$

$$\phi(t) = t^{\left(1 + \frac{2kA}{\dot{m}C_p\Delta z} + \frac{hA}{\dot{m}C_p}\right)} \quad (\text{b5})$$

$$\int \phi(t)Q(t)dt = \left[T_o + \frac{2kAT_b}{\dot{m}C_p\Delta z} + \frac{hAT_{rf}}{\dot{m}C_p} \right] \frac{t^{\left(1 + \frac{2kA}{\dot{m}C_p\Delta z} + \frac{hA}{\dot{m}C_p}\right)}}{\left(1 + \frac{2kA}{\dot{m}C_p\Delta z} + \frac{hA}{\dot{m}C_p}\right)} \quad (\text{b6})$$

The integrated differential equation has the form of equation b7, and the solution is equation b8. This is equivalent to equation 4 in the main part of this document.

$$\phi(t)T_{top} = \int \phi(t)Q(t)dt \quad (\text{b7})$$

$$T_{top} = \frac{T_o + \frac{2kAT_b}{\dot{m}C_p\Delta z} + \frac{hAT_{rf}}{\dot{m}C_p}}{\left(1 + \frac{2kA}{\dot{m}C_p\Delta z} + \frac{hA}{\dot{m}C_p}\right)} \quad (\text{b8})$$

The area can be removed from the solution with the following identity for the mass flow rate, equation b9, and the final solution to the differential equation is equation b10.

$$\dot{m} = \rho A \dot{z} \quad (\text{b9})$$

$$T_{top} = \frac{T_o + \frac{2kT_b}{\rho \dot{z} C_p \Delta z} + \frac{hT_{rf}}{\rho \dot{z} C_p}}{\left(1 + \frac{2k}{\rho \dot{z} C_p \Delta z} + \frac{h}{\rho \dot{z} C_p}\right)} \quad (\text{b10})$$

Appendix C Saltstone Surface Heat Transfer

The saltstone vaults are covered with corrugated metal roofs with a shallow slope. In this analysis, the slope of the roof is neglected, and the saltstone and the roof are treated as parallel plates. Figure C1 is a schematic of the surface heat transfer of the saltstone. The heat transfer between the roof and the saltstone is by natural convection and thermal radiation. In the numerical model all of the surface heat transfer is defined in terms of Newton's law of cooling, so a composite heat transfer coefficient is used that includes both natural convection and radiation. The temperature sink for surface heat transfer from the saltstone is the roof. For each timestep of the numerical model, the roof temperature is calculated from a steady-state heat balance. The roof is subjected to heat transfer with the saltstone on the lower surface, and thermal radiation and convection with the air on the upper surface. The thermal radiation on the upper surface is infrared exchange with the sky and direct solar radiation during the day.

Figure C1: Schematic of the saltstone surface heat transfer

Equation c1 is a relation for the mean Nusselt number for turbulent flow over a flat plate, (Ozisik, 1985). Air flowing with a free stream velocity of 5 mph over a 100 ft plate has a heat transfer of approximately 5 W/m²K, and this is the assumed value of h_{cv} in the pour model.

$$\bar{Nu}_L = 0.036 Pr^{.43} (Re_L^8 - 9200) \Rightarrow Re_L \geq 2 \times 10^5 \quad (c1)$$

Equation c2 is the relation for the infrared radiation exchange between the roof and the sky. The assumed view factor for this radiation exchange is one. Equation c3 is the relation for the effective sky temperature, (Duffie & Beckman, 1974). This radiation exchange occurs both during the daylight hours and also at night.

$$q_{sky} = \epsilon_{rf} \sigma (T_{rf}^4 - T_{sky}^4) \quad (c2)$$

$$T_{sky} = 0.0552 T_{air}^{1.5} \quad (c3)$$

During the day, the roof is heated directly by solar radiation, which is a strong function of the cloud cover. The greatest heat load on the roof occurs on sunny days, and therefore the assumption that all days are sunny is conservative with respect to calculating the peak curing temperature of the saltstone. Equation c4 is the expression for the solar radiation as a function of the time of day, and the results are plotted in figure C2. Also shown in figure C2 is data for solar radiation on a flat surface, (Ozisik, 1985). This is typical for a sunny late spring or early fall day. The annual change in the daily duration of sun light is not included in this model. The assumed solar absorptivity is 0.3. This is the tabulated value for light cream colored oil base paint, (Ozisik, 1985). This differs considerably from the assumed emissivity for the roof of 0.92. The emissivity is applicable for longer wave length infrared radiation.

$$q_t = 800 \cdot \sin\left(\frac{time - 7}{12} \pi\right) + 100 \quad (c4)$$

Figure C2: Total solar radiation on a horizontal surface versus time on a clear day

The underside of the roof is subjected to both both natural convection and thermal radiation heat transfer with the surface of the saltstone. Equation c5 is the relation for radiation heat transfer between two infinite parallel plates. The saltstone is assumed to have an emissivity of 0.94. This is the tabulated value for rough concrete, (Ozisik, 1985). With these high emissivities for the saltstone and the roof, thermal radiation is an important heat transfer mechanism. The radiation heat transfer is approximately four times that due to natural convection.

$$q_{rad} = \frac{\sigma(T_{ss}^4 - T_{rf}^4)}{\frac{1}{\epsilon_{ss}} + \frac{1}{\epsilon_{rf}} - 1} \quad (c5)$$

Natural convection between two horizontal parallel plates, when the lower plate is hotter than upper plate, is governed by equation c6, (Hollands et al., 1975). While the Nusselt number is a function of the plate spacing, the heat transfer coefficient is not, equation c7.

$$Nu = \frac{hD}{k} = .0555 Ra^{1/3} \Rightarrow Ra = \frac{\beta g \Delta T D^3}{\nu \alpha} \quad (c6)$$

$$h_{nc} = .0555 k \left(\frac{\beta g \Delta T}{\nu \alpha} \right)^{1/3} \quad (c7)$$

Equations c1 through c7 define the various heat transfer mechanisms to which the roof is subjected. Equation c8 is a steady-state heat balance applied to the roof. The independent variable in this equation is the roof temperature, and equation c9 is the heat balanced rearranged as a fourth order polynomial of the roof temperature. Equation c9 is solved every timestep, by Newton's iterative method, for the roof temperature.

$$\alpha_{sol} q_t + \epsilon_{rf} \sigma (T_{sky}^4 - T_{rf}^4) - h_{cv} (T_{rf} - T_{air}) + h_{nc} (T_{salt} - T_{rf}) + \frac{\sigma (T_{ss}^4 - T_{rf}^4)}{\frac{1}{\epsilon_{ss}} + \frac{1}{\epsilon_{rf}} - 1} = 0 \quad (c8)$$

$$\left[\epsilon_{rf} \sigma + \frac{\sigma}{\frac{1}{\epsilon_{rf}} + \frac{1}{\epsilon_{ss}} - 1} \right] T_{rf}^4 + (h_{cv} + h_{nc}) T_{rf} - \left[\alpha_{sol} q_t + \epsilon_{rf} \sigma T_{sky}^4 + h_{cv} T_{air} + h_{nc} T_{ss} + \frac{\sigma T_{ss}^4}{\frac{1}{\epsilon_{rf}} + \frac{1}{\epsilon_{ss}} - 1} \right] = 0 \quad (c9)$$

The surface boundary condition for the saltstone conduction calculation is expressed in terms of Newton's law of cooling. This requires that an effective heat transfer coefficient for the thermal radiation between the saltstone and the roof be derived, equation c11. The natural convection and radiation heat transfer coefficients are summed to calculate the total heat transfer between the saltstone and the roof, equation c12.

$$q_{rad} = \frac{\sigma (T_{ss}^4 - T_{rf}^4)}{\frac{1}{\epsilon_{ss}} + \frac{1}{\epsilon_{rf}} - 1} = h_{rad} (T_{ss} - T_{rf}) \quad (c10)$$

$$h_{rad} = \frac{\sigma (T_{ss} + T_{rf}) (T_{ss}^2 + T_{rf}^2)}{\frac{1}{\epsilon_{ss}} + \frac{1}{\epsilon_{rf}} - 1} \quad (c11)$$

$$q = (h_{rad} + h_{nc}) * (T_{ss} - T_{rf}) \quad (c12)$$

The daily average temperature is assumed to vary sinusoidally $\pm 20^\circ\text{F}$ about an annual mean temperature of 65°F . The minimum daily average temperature is assumed to occur on January 15 and the maximum on July 15. Equation c13 is the relation for the annual variation in the daily average temperature. The diurnal temperature variation is assumed to be $\pm 12^\circ\text{F}$ about the daily average temperature, equation c14. Figure C3 is a plot of the assumed variation of the air temperature for six months, starting on January 15. This plot shows both the annual variation of the daily average temperature and the superimposed daily variation. The variable *#day* in equation c13 is the number of days from January 15.

$$T_{avg} = 291.483 - 11.111 * \cos\left(\frac{2\pi \#day}{365}\right) \Rightarrow 0 \leq \#day \leq 365 \quad (c13)$$

$$T_{air} = T_{avg} - 6.666 * \cos\left(\frac{\pi \#hr}{12}\right) \Rightarrow 0 \leq \#hr \leq 24 \quad (c14)$$

Figure C3: Assumed annual variation in air temperature, starting on January 15.

This Page Intentionally Left Blank

Appendix D gndtmp.f Code Listing

```

      program gndtmp
c
c *****
c   This program calculates the vertical temperature distribution
c   in the ground below a saltstone vault. The temperature
c   distribution is due to the annual variation in the air temperature.
c *****
c
      dimension z(38),zk(38),rho(38),cp(38),t(38),tn(38),a(38),b(38)
      dimension c(38),d(38),x(38),depth(38)
c
c   gtmp.dat is the output file with the depth and temperature results.
c   It is used to plot a temperature profile in the ground.
c   gint.dat is the input data file for the saltst.f program. It has
c   the mesh widths, z(i), and the nodal temperatures.
c
      open (unit=10,file='gtmp.dat',status='new')
      open (unit=11,file='gint.dat',status='new')
      imax=38
      im=imax-1
c
c   These are the physical properties of the layers below the vault:
c
c      rhocn      density of concrete (kg/m^3)
c      cpcn      specific heat of concrete (J/kg K)
c      zkcncn    thermal conductivity of concrete (W/m K)
c
c      rhogr      density of gravel (kg/m^3)
c      cpgr      specific heat of gravel (J/kg K)
c      zkgr      thermal conductivity of gravel (W/m K)
c
c      rhocl      density of clay (kg/m^3)
c      cpcl      specific heat of clay (J/kg K)
c      zkcl      thermal conductivity of clay (W/m K)
c
c      rhodt      density of dirt (kg/m^3)
c      cpdt      specific heat of dirt (J/kg K)
c      zkdt      thermal conductivity of dirt (W/m K)
c
      rhocn=2307.0
      cpcn=753.0
      zkcncn=1.4
      rhogr=2050.0
      cpgr=1840.0
      zkgr=.52
      rhocl=1457.7
      cpcl=880.0
      zkcl=1.3
      rhodt=1500.0
      cpdt=1842.0
      zkdt=.35
c
      do 25 i=1,imax
         rho(i)=rhodt
         cp(i)=cpdt
         zk(i)=zkdt
25      continue
      do 26 i=17,imax
         rho(i)=rhocl
         cp(i)=cpcl
         zk(i)=zkcl
26      continue

```



```

do 27 i=25,imax
  rho(i)=rhogr
  cp(i)=cpgr
  zk(i)=zkgr
27 continue
do 28 i=29,imax
  rho(i)=rhocn
  cp(i)=cpcn
  zk(i)=zkcen
28 continue
c
c   These two loops input the values for the mesh widths
c   z(i) = mesh width for the ith node
c
do 30 i=1,imax
  z(i)=.0762
30 continue
do 31 i=15,1,-1
  z(i)=1.2*z(i+1)
31 continue
c
c   depth(i) is the depth below the vault bottom of the ith node.
c
depth(imax)=z(imax)/2.0
do 35 i=im,1,-1
  depth(i)=depth(i+1)+(z(i+1)+z(i))/2.0
35 continue
c
c   tbar is the annual average air temperature. It is also the
c   assumed initial temperature of the ground for this calculation.
c
tbar=18.3333+273.15
do 40 i=1,imax
  t(i)=tbar
  tn(i)=tbar
40 continue
c
c   dt is the timestep in seconds. The timestep is one day.
c
dt=86400.0
dtd=dt/86400.0
day=0.0
cl=.017214206
c
c   The date for which the temperature distribution is determined
c   is a function of the number of iterations that code goes through.
c   The code goes first goes through one complete annual cycle starting
c   on January 15. The maximum number of iterations is x*365 plus the
c   number of days from Jan. 15 to the desired date. This number is
c   the maximum value of j in the following "do" statement.
c
c   *****
c   do 1000 j=1,9291
c   *****
c   day=day+dtd
c   dayt=day
c
c   The daily average air temperature is calculated by # 400.
c
if (dayt .le. 365.0) go to 400
  dayt=dayt-365.0
400 tair=tbar-11.11111*cos(cl*dayt)
c
c   Elements of the tridiagonal matrix are calculated

```

```

c
  do 425 i=1,im
    a(i)=-((2.0*zk(i)*zk(i+1)*dt)/
    &      ((rho(i+1)*cp(i+1)*z(i+1))*
    &      (z(i)*zk(i+1)+z(i+1)*zk(i))))
    b(i)=t(i)
    c(i)=-((2.0*zk(i)*zk(i+1)*dt)/
    &      ((rho(i)*cp(i)*z(i))*
    &      (z(i)*zk(i+1)+z(i+1)*zk(i))))
425  continue
c
c   The natural convection heat transfer coefficient is calculated.
c   The logic determines the correct correlation depending on whether
c   the air is heating or cooling the surface.
c
    if (tair .lt. t(imax))go to 430
    delt=tair-t(imax)
    h=.59*((delt/7.62)**.25)
    go to 431
430  delt=t(imax)-tair
    h=1.52*(delt**.33333)
431  heq=(2.0*h*zk(imax))/((h*z(imax))+(2.0*zk(imax)))
    b(imax)=t(imax)+((heq*dt*tair)/(rho(imax)*cp(imax)*z(imax)))
    d(1)=1.0+((2.0*zk(1)*zk(2)*dt)/(rho(1)*cp(1)*z(1)*
    &      (z(1)*zk(2)+z(2)*zk(1))))
    do 426 i=2,im
      d(i)=1.0+((2.0*dt)/(rho(i)*cp(i)*z(i)))*
    &      ((zk(i)*zk(i+1))/(z(i)*zk(i+1)+z(i+1)*zk(i))+
    &      (zk(i-1)*zk(i))/(z(i-1)*zk(i)+z(i)*zk(i-1)))
426  continue
    d(imax)=1.0+(dt/(rho(imax)*cp(imax)*z(imax)))*
    &      (heq+(2.0*zk(im)*zk(imax))/(
    &      (z(im)*zk(imax)+z(imax)*zk(im))))
c
c   subroutine tri solves a tridiagonal system of equations
c   with the Thompson algorithm.  n is the number of equations.
c
    n=imax
    call tri(n,a,d,c,b,x)
    do 450 i=1,imax
      tn(i)=x(i)
450  continue
    do 550 i=1,imax
      t(i)=tn(i)
550  continue
c
c   this is the end of the computational loop
c
1000 continue
c
c   output:
c
    do 1050 i=imax,1,-1
      t(i)=t(i)-273.15
      write (10,2000)depth(i),t(i)
1050 continue
    do 1060 i=1,imax
      write (11,2000)z(i),tn(i)
1060 continue
2000 format (1x,2g18.6)
    stop
    end

```

This Page Intentionally Left Blank

Appendix E Pour Models Code Listings

```

      program saltst
C
C *****
C      This program calculates the transient vertical temperature distribution
C      in saltstone as it is poured into a vault. This is a one-dimensional
C      conduction code with a moving boundary. Inputs to the code are the
C      ground temperature distribution below the vault, the pour temperature,
C      and the pour schedule.
C
C      Read file:   gint.dat      (initial ground temp. distribution)
C      Subroutines: surfht.f      (surface heat transfer)
C                  airtmp.f      (air temperature)
C                  newton.f      (newton iteration solver)
C                  pour.f        (heat transfer while pouring)
C                  nopour.f      (heat transfer while not pouring)
C                  tri.f         (tri-diagonal matrix solver)
C                  intht.f       (saltstone internal heat generation)
C *****
C
C      dimension telap(222)
C      common t(222),z(222),rho(222),cp(222),zk(222),qstar(222)
C      open (unit=10,file='gint.dat',status='old')
C      open (unit=11,file='sspktp.dat',status='new')
C      open (unit=12,file='ssvtpds.dat',status='new')
C      open (unit=13,file='ssz/tim.dat',status='new')
C      open (unit=14,file='sshtls.dat',status='new')
C      open (unit=15,file='specal.dat',status='new')
C
C *****
C      *   INPUTS:
C      *
C      *      tpour      saltstone pour temperature (K)
C      *      nhpour     number of hours of pouring each day
C      *      dt         time step size (sec)
C      *      wkmax      number of weeks that the code runs
C      *      dayt       start day (Jan 15 + "dayt" days)
C      *      hrday      hour of the day that pouring starts -1
C      *
C *****
C
C      tpour=303.15
C      nhpour=8
C      nhpp=nhpour+1
C      dt=3600.0
C      wkmax=10.0
C      dayt=166.0
C      hrday=8.0
C
C *****
C      *   OUTPUT FILES:
C      *
C      *      sspktp.dat  peak temp vs time
C      *      ssvtpds.dat weekly vertical tmp distribution
C      *      ssz/tim.dat saltstone depth vs time
C      *      sshtls.dat  surface & bottom heat transfer losses
C      *      specal.dat  special output not of general interest
C *****
C
C      write (11,400)tpour
C      srtday=dayt+15.0
C      write (11,401)srtday
C      write (11,402)nhpour
C      daymax=wkmax*7.0

```

```

write (11,206)
write (13,1300)
dday=0.0
depth=0.0
write (13,207)dday,depth
c
c These interger variables define the computational mesh.
c The first 40 nodes model the ground under the vault with a
c variable mesh. The balance of the nodes, (41 to 224), model the
c poured saltstone, and this region has a uniform mesh.
c
  ibot=38
  ibotp=ibot+1
  imax=222
c
c This loop reads input file 'gint.dat'. This file contains the mesh
c widths and temperatures for the 40 nodes in the ground below the vault.
c
  do 10 i=1,ibot
    read (10,2000)z(i),t(i)
10  continue
c
c This loop inputs the mesh widths for the pouring saltstone. The mesh
c width is a function of the pour rate and the timestep size.
c delta"z"=.0415 m corresponds to dt=1 hour and pour rate of 1362 ft^3/hr.
c
  do 11 i=ibotp,imax
    z(i)=.0415
11  continue
c
c These are the physical properties:
c
c   rhocn density of concrete (kg/m^3)
c   cpcn specific heat of concrete (J/kg K)
c   zkcن thermal conductivity of concrete (W/m K)
c
c   rhogr density of gravel (kg/m^3)
c   cpgr specific heat of gravel (J/kg K)
c   zkgr thermal conductivity of gravel (W/m K)
c
c   rhocl density of clay (kg/m^3)
c   cpcl specific heat of clay (J/kg K)
c   zkcl thermal conductivity of clay (W/m K)
c
c   rhodt density of dirt (kg/m^3)
c   cpdt specific heat of dirt (J/kg K)
c   zkdt thermal conductivity of dirt (W/m K)
c
c   rhoss density of saltstone (kg/m^3)
c   cpss specific heat of saltstone (J/kg K)
c   zkss thermal conductivity of saltstone (W/m K)
c
rhocn=2307.0
cpcn=753.0
zkcن=1.4
rhogr=2050.0
cpgr=1840.0
zkgr=.52
rhocl=1457.7
cpcl=880.0
zkcl=1.3
rhodt=1500.0
cpdt=1842.0
zkdt=.35

```

```

      rhoss=1797.63
      cpss=1302.06
      zkss=.76
c
      do 25 i=1,ibot
         rho(i)=rhodt
         cp(i)=cpdt
         zk(i)=zkdt
25      continue
         do 26 i=17,ibot
            rho(i)=rhocl
            cp(i)=cpcl
            zk(i)=zkcl
26      continue
         do 27 i=25,ibot
            rho(i)=rhogr
            cp(i)=cpgr
            zk(i)=zkgr
27      continue
         do 28 i=29,ibot
            rho(i)=rhocn
            cp(i)=cpcn
            zk(i)=zken
28      continue
         do 29 i=ibotp,imax
            rho(i)=rhoss
            cp(i)=cpss
            zk(i)=zkss
29      continue
         do 30 i=1,imax
            qstar(i)=0.0
            telap(i)=0.0
30      continue
         npx=ibot
         dz=z(ibotp)
         zdepth=0.0
         hour=0.0
         week=0.0
         day=1.0
         zkoz=2.0*zk(ibot)*zk(ibotp)/(z(ibot)*zk(ibotp)+z(ibotp)*zk(ibot))
         qtpsum=0.0
         qbtsum=0.0
c
c      *****
c      Start the computational loop
c      *****
c
      xxxx=.5
100     nday=1
         week=week+1.0
         if (week .gt. wkmax)go to 160
105     if (nday .gt. 5)go to 140
         if (npx .eq. imax)go to 140
c
c      This loop (do 120 ...) calculates the heat transfer during
c      the pouring phase of a day. nhpour is the number of hours
c      that the saltstone is poured a day.
c
110     do 120 nhr=1,nhpour
         tsurf=t(npx)
         hrday=hrday+1.0
         call surfht(dayt,hrday,tsurf,tr,heq,tair)
c      write (15,3000)xxxx,tsurf,tr,heq
         npx=npx+1

```

```

        call pour(npx,dt,tpour,tr,heq)
        zdepth=zdepth+dz
        telap(npx)=-dt/2.0
        do 115 i=ibotp,npx
            telap(i)=telap(i)+dt
            elapt=telap(i)
            ccp=cp(i)
c         call intht(elapt,dt,ccp,source)
            call inthtx(elapt,dt,ccp,source)
            qstar(i)=source
115        continue
c         write (15,208)xxxx,qstar(39),qstar(50),qstar(58)
            xxxx=xxxx+1.0
            hour=hour+1.0
            depth=zdepth/.30548
            dday=hour/24.0
            qbott=zkoz*(t(ibotp)-t(ibot))
            qtop=heq*(t(npx)-tr)
            write (14,4000)dday,qtop,qbott
            qtpsum=qtpsum+qtop*3600.0
            qbtsum=qbtsum+qbott*3600.0
120        continue
            write (13,207)dday,depth
c
c         This loop (do 130...) calculates the heat transfer during
c         the nonpouring phase of a day.
c
            do 130 nhr=nhpp,24
                tsurf=t(npx)
                if (hrday .lt. 24.0)then
                    hrday=hrday+1.0
                else
                    hrday=1.0
                    dayt=dayt+1.0
                end if
                call surfht(dayt,hrday,tsurf,tr,heq,tair)
c             write (15,3000)xxxx,tsurf,tr,heq
                call nopour(npx,dt,tr,heq,heff)
                do 125 i=ibotp,npx
                    telap(i)=telap(i)+dt
                    elapt=telap(i)
                    ccp=cp(i)
c                 call intht(elapt,dt,ccp,source)
                    call inthtx(elapt,dt,ccp,source)
                    qstar(i)=source
125                continue
c                 write (15,208)xxxx,qstar(39),qstar(50),qstar(58)
                    xxxx=xxxx+1.0
                    hour=hour+1.0
                    depth=zdepth/.30548
                    dday=hour/24.0
                    qbott=zkoz*(t(ibotp)-t(ibot))
                    qtop=heff*(t(npx)-tr)
                    write (14,4000)dday,qtop,qbott
                    qtpsum=qtpsum+qtop*3600.0
                    qbtsum=qbtsum+qbott*3600.0
130                continue
                    write (9,212)dday,daymax
                    write (13,207)dday,depth
c
c         This loop (133) calculates and reads out the peak temperature
c         once a day.
c
            tpeak=t(ibotp)

```

```

do 133 i=ibotp,npx
  if (t(i) .lt. tpeak)go to 133
    tpeak=t(i)
133 continue
tpeak=tpeak-273.15
write (11,207)dday,tpeak
nday=nday+1
day=day+1.0
if (nday .lt. 6)go to 134
  write (12,2500)week
  write (12,2501)
  deltaz=.0415
  axpos=-deltaz/2.0
c
c   This loop (1105) writes out the vertical temperature distribution
c   in the saltstone once a week during the pouring phase of the
c   operation. The output file is ssvtpds.dat
c
  do 1105 i=ibotp,npx
    axpos=axpos+deltaz
    tdegc=t(i)-273.15
    write (12,207)axpos,tdegc
1105 continue
134 go to 105
c
c   This loop (do 150...) calculates the heat transfer during
c   the days when pouring does not take place, such as weekends.
c
140 do 150 nhr=1,24
  tsurf=t(npix)
  if (hrday .lt. 24.0)then
    hrday=hrday+1.0
  else
    hrday=1.0
    dayt=dayt+1.0
  end if
  call surfht (dayt,hrday,tsurf,tr,heq,tair)
c  write (15,3000)xxxx,tsurf,tr,heq
  call nopour(npix,dt,tr,heq,heff)
  do 145 i=ibotp,npx
    telap(i)=telap(i)+dt
    elapt=telap(i)
    ccp=cp(i)
c    call intht (elapt,dt,ccp,source)
    call inthtx (elapt,dt,ccp,source)
    qstar(i)=source
145 continue
c    write (15,208)xxxx,qstar(39),qstar(50),qstar(58)
    xxxx=xxxx+1.0
    hour=hour+1.0
    depth=zdepth/.30548
    dday=hour/24.0
    qbott=zkoz*(t(ibotp)-t(ibot))
    qtop=heff*(t(npix)-tr)
    write (14,4000)dday,qtop,qbott
    qtpsum=qtpsum+qtop*3600.0
    qbtsum=qbtsum+qbott*3600.0
150 continue
  write (9,212)dday,daymax
  write (13,207)dday,depth
c
c   This loop (153) calculates and reads out the peak temperature
c   once a day.
c

```



```

        tpeak=t(ibotp)
        do 153 i=ibotp,npx
            if (t(i) .lt. tpeak)go to 153
                tpeak=t(i)
153    continue
        tpeak=tpeak-273.15
        write (11,207)dday,tpeak
        nday=nday+1
        day=day+1.0
        if (nday .gt. 7)go to 100
        go to 105
160    write (9,210)
        write (14,215)wkmax
        write (14,220)qtpsum
        write (14,225)qbtsum
        qltot=qtpsum+qbtsum
        qtrrat=qtpsum/qltot
        qbrat=qbtsum/qltot
        write (14,207)qtrrat,qbrat
c
c *****
c end of the computational loop
c *****
c
c The final vertical temperature distribution in the saltstone is
c written to file ssvtpds.dat
c
        week=week-1.0
        write (12,210)
        write (12,2500)week
        write (12,2501)
        deltaz=.0415
        axpos=-deltaz/2.0
        do 1100 i=ibotp,npx
            axpos=axpos+deltaz
            tdegc=t(i)-273.15
            write (12,207)axpos,tdegc
1100    continue
        pause
200    format (g15.6,i5,2g15.6,'pour')
205    format (g15.6,i5,2g15.6,'no pour')
206    format (1x,' DAY PEAk TEMP.')
207    format (2g16.8)
208    format (4g16.8)
210    format ('end')
212    format (1x,'day=',g14.6,' daymax=',g14.6)
215    format (1x,'week=',g14.6)
220    format (1x,'top heat trans loss=',g20.8,' J/m^2')
225    format (1x,'bottom heat trans loss=',g20.8,' J/m^2')
400    format (1x,'pour temp =',g14.6,' "K"')
401    format (1x,'start date =',g14.6)
402    format (1x,'pour schedule = ',i4,' hrs/day, 5 days/wk')
1300    format (1x,' day depth (ft)')
2000    format (1x,2g18.6)
2010    format (1x,i5,g18.6)
2500    format (1x,'week = ',g18.6)
2501    format (1x,' height (m) temp (K)')
3000    format (1x,4g16.8)
4000    format (1x,3g18.8)
        stop
        end

```

```

      program xsalt
c
c      This is the main program for the case where the saltstone is poured three
c      days per week, (Monday, Wednesday, & Friday)
c
c      *****
c      This program calculates the transient vertical temperature distribution
c      in saltstone as it is poured into a vault. This is a one-dimensional
c      conduction code with a moving boundary. Inputs to the code are the
c      ground temperature distribution below the vault, the pour temperature,
c      and the pour schedule.
c
c      Read file:      gint.dat      (initial ground temp. distribution)
c      Subroutines:    surfht.f      (surface heat transfer)
c                   airtmp.f      (air temperature)
c                   newton.f      (newton iteration solver)
c                   pour.f      (heat transfer while pouring)
c                   nopour.f      (heat transfer while not pouring)
c                   tri.f      (tri-diagonal matrix solver)
c                   intht.f      (saltstone internal heat generation)
c      *****
c
c      dimension telap(222)
c      common t(222),z(222),rho(222),cp(222),zk(222),qstar(222)
c      open (unit=10,file='gint.dat',status='old')
c      open (unit=11,file='sspktp.dat',status='new')
c      open (unit=12,file='ssvtpds.dat',status='new')
c      open (unit=13,file='ssz/tim.dat',status='new')
c      open (unit=14,file='sshtls.dat',status='new')
c      open (unit=15,file='specal.dat',status='new')
c
c      *****
c      *      INPUTS:      *
c      *      tpour saltstone pour temperature (K)      *
c      *      nhpour number of hours of pouring each day      *
c      *      dt time step size (sec)      *
c      *      wkmax number of weeks that the code runs      *
c      *      dayt start day (Jan 15 + "dayt" days)      *
c      *      hrday hour of the day that pouring starts -1      *
c      *      *****
c
c      tpour=318.15
c      nhpour=8
c      nhpp=nhpour+1
c      dt=3600.0
c      wkmax=14.0
c      dayt=335.0
c      hrday=8.0
c
c      *****
c      *      OUTPUT FILES:      *
c      *      sspktp.dat peak temp vs time      *
c      *      ssvtpds.dat weekly vertical tmp distribution      *
c      *      ssz/tim.dat saltstone depth vs time      *
c      *      sshtls.dat surface & bottom heat transfer losses      *
c      *      specal.dat special output not of general interest      *
c      *      *****
c
c      write (11,400)tpour
c      srtday=dayt+15.0
c      write (11,401)srtday
c      write (11,402)nhpour
c      daymax=wkmax*7.0
c      write (11,206)

```

```

write (13,1300)
dday=0.0
depth=0.0
write (13,207)dday,depth
c
c   These interger variables define the computational mesh.
c   The first 40 nodes model the ground under the vault with a
c   variable mesh. The balance of the nodes, (41 to 224), model the
c   poured saltstone, and this region has a uniform mesh.
c
      ibot=38
      ibotp=ibot+1
      imax=222
c
c   This loop reads input file 'gint.dat'. This file contains the mesh
c   widths and temperatures for the 40 nodes in the ground below the vault.
c
      do 10 i=1,ibot
        read (10,2000)z(i),t(i)
10    continue
c
c   This loop inputs the mesh widths for the pouring saltstone. The mesh
c   width is a function of the pour rate and the timestep size.
c   delta"z"=.0415 m corresponds to dt=1 hour and pour rate of 1362 ft^3/hr.
c
      do 11 i=ibotp,imax
        z(i)=.0415
11    continue
c
c   These are the physical properties:
c
c      rhocn  density of concrete (kg/m^3)
c      cpcn   specific heat of concrete (J/kg K)
c      zkcnc  thermal conductivity of concrete (W/m K)
c
c      rhogr  density of gravel (kg/m^3)
c      cpgr   specific heat of gravel (J/kg K)
c      zkgr   thermal conductivity of gravel (W/m K)
c
c      rhocl  density of clay (kg/m^3)
c      cpcl   specific heat of clay (J/kg K)
c      zkcl   thermal conductivity of clay (W/m K)
c
c      rhodt  density of dirt (kg/m^3)
c      cpdt   specific heat of dirt (J/kg K)
c      zkdt   thermal conductivity of dirt (W/m K)
c
c      rhoss  density of saltstone (kg/m^3)
c      cpss   specific heat of saltstone (J/kg K)
c      zkss   thermal conductivity of saltstone (W/m K)
c
      rhocn=2307.0
      cpcn=753.0
      zkcnc=1.4
      rhogr=2050.0
      cpgr=1840.0
      zkgr=.52
      rhocl=1457.7
      cpcl=880.0
      zkcl=1.3
      rhodt=1500.0
      cpdt=1842.0
      zkdt=.35
      rhoss=1797.63

```

```

      cpss=1302.06
      zkss=.76
c
      do 25 i=1,ibot
         rho(i)=rhodt
         cp(i)=cpdt
         zk(i)=zkdt
25      continue
         do 26 i=17,ibot
            rho(i)=rhocl
            cp(i)=cpcl
            zk(i)=zkcl
26      continue
         do 27 i=25,ibot
            rho(i)=rhogr
            cp(i)=cpgr
            zk(i)=zkgr
27      continue
         do 28 i=29,ibot
            rho(i)=rhocn
            cp(i)=cpcn
            zk(i)=zkcن
28      continue
         do 29 i=ibotp,imax
            rho(i)=rhoss
            cp(i)=cpss
            zk(i)=zkss
29      continue
         do 30 i=1,imax
            qstar(i)=0.0
            telap(i)=0.0
30      continue
         npx=ibot
         dz=z(ibotp)
         zdepth=0.0
         hour=0.0
         week=0.0
         day=1.0
         zkoz=2.0*zk(ibot)*zk(ibotp)/(z(ibot)*zk(ibotp)+z(ibotp)*zk(ibot))
         qtpsum=0.0
         qbtsum=0.0
c
c      *****
c      Start the computational loop
c      *****
c
      xxxx=.5
100     nday=1
         week=week+1.0
         if (week .gt. wkmax)go to 160
105     if (nday .gt. 6)go to 140
         if (npx .eq. imax)go to 140
c
c      This loop (do 120 ...) calculates the heat transfer during
c      the pouring phase of a day. nhpour is the number of hours
c      that the saltstone is poured a day.
c
110     do 120 nhr=1,nhpour
         tsurf=t(npx)
         hrday=hrday+1.0
         call surfht(dayt,hrday,tsurf,tr,heq,tair)
c      write (15,3000)xxxx,tsurf,tr,heq
         npx=npx+1
         call pour(npx,dt,tpour,tr,heq)

```

```

      zdepth=zdepth+dz
      telap(npix)--dt/2.0
      do 115 i=ibotp,npix
        telap(i)=telap(i)+dt
        elapt=telap(i)
        ccp=cp(i)
        call intht(elapt,dt,ccp,source)
c      call inthtx(elapt,dt,ccp,source)
        qstar(i)=source
115      continue
c      write (15,208)xxxx,qstar(39),qstar(50),qstar(58)
        xxxx=xxxx+1.0
        hour=hour+1.0
        depth=zdepth/.30548
        dday=hour/24.0
        qbott=zkoz*(t(ibotp)-t(ibot))
        qtop=heq*(t(npix)-tr)
        write (14,4000)dday,qtop,qbott
        qtpsum=qtpsum+qtop*3600.0
        qbtsum=qbtsum+qbott*3600.0
120      continue
        write (13,207)dday,depth
        deltaz=.0415
        axpos=-deltaz/2.0
        do 121 i=ibotp,npix
          axpos=axpos+deltaz
          tdegc=t(i)-273.15
c        write (12,207)axpos,tdegc
121      continue
c        write (12,5000)
5000      format('end pouring period')
c
c      This loop (do 130...) calculates the heat transfer during
c      the nonpouring phase of a day.
c
      do 130 nhr=nhpp,24
        tsurf=t(npix)
        if (hrday .lt. 24.0)then
          hrday=hrday+1.0
        else
          hrday=1.0
          dayt=dayt+1.0
        end if
        call surfht(dayt,hrday,tsurf,tr,heq,tair)
c      write (15,3000)xxxx,tsurf,tr,heq
        call nopour(npix,dt,tr,heq,heff)
        do 125 i=ibotp,npix
          telap(i)=telap(i)+dt
          elapt=telap(i)
          ccp=cp(i)
          call intht(elapt,dt,ccp,source)
c          call inthtx(elapt,dt,ccp,source)
          qstar(i)=source
125      continue
c      write (15,208)xxxx,qstar(39),qstar(50),qstar(58)
        xxxx=xxxx+1.0
        hour=hour+1.0
        depth=zdepth/.30548
        dday=hour/24.0
        qbott=zkoz*(t(ibotp)-t(ibot))
        qtop=heff*(t(npix)-tr)
        write (14,4000)dday,qtop,qbott
        qtpsum=qtpsum+qtop*3600.0
        qbtsum=qbtsum+qbott*3600.0

```

```

130 continue
   write (9,212) dday, daymax
   write (13,207) dday, depth
   deltaz=.0415
   axpos=-deltaz/2.0
   do 131 i=ibotp,npx
       axpos=axpos+deltaz
       tdegc=t(i)-273.15
c   write (12,207) axpos, tdegc
131 continue
c   write (12,5001)
5001 format ('end pour day')
c
c   This loop (133) calculates and reads out the peak temperature
c   once a day.
c
       tpeak=t(ibotp)
       do 133 i=ibotp,npx
           if (t(i) .lt. tpeak) go to 133
           tpeak=t(i)
133 continue
       tpeak=tpeak-273.15
       write (11,207) dday, tpeak
       nday=nday+1
       day=day+1.0
c
c   This loop (do 135...) calculates the heat transfer during
c   the nonpouring day.
c
       do 135 nhr=1,24
           tsurf=t(npx)
           if (hrday .lt. 24.0) then
               hrday=hrday+1.0
           else
               hrday=1.0
               dayt=dayt+1.0
           end if
           call surfht (dayt, hrday, tsurf, tr, heq, tair)
c       write (15,3000) xxxx, tsurf, tr, heq
           call nopour (npx, dt, tr, heq, heff)
           do 134 i=ibotp,npx
               telap(i)=telap(i)+dt
               elapt=telap(i)
               ccp=cp(i)
               call intht (elapt, dt, ccp, source)
c               call inthtx (elapt, dt, ccp, source)
               qstar(i)=source
134 continue
c       write (15,208) xxxx, qstar(39), qstar(50), qstar(58)
       xxxx=xxxx+1.0
       hour=hour+1.0
       depth=zdepth/.30548
       dday=hour/24.0
       qbott=zkoz*(t(ibotp)-t(ibot))
       qtop=heff*(t(npx)-tr)
       write (14,4000) dday, qtop, qbott
       qtpsum=qtpsum+qtop*3600.0
       qbtsum=qbtsum+qbott*3600.0
135 continue
   write (9,212) dday, daymax
   write (13,207) dday, depth
   deltaz=.0415
   axpos=-deltaz/2.0
   do 137 i=ibotp,npx

```

```

        axpos=axpos+deltaz
        tdegc=t(i)-273.15
c      write (12,207)axpos,tdegc
137 continue
c      write (12,5002)
5002 format ('end nopour day')
c
c      This loop (136) calculates and reads out the peak temperature
c      once a day.
c
        tpeak=t(ibotp)
        do 136 i=ibotp,npx
            if (t(i) .lt. tpeak)go to 136
            tpeak=t(i)
136 continue
        tpeak=tpeak-273.15
        write (11,207)dday,tpeak
        nday=nday+1
        day=day+1.0
        go to 105
c
c      This loop (do 150...) calculates the heat transfer on Sunday
c      for a M/W/F pour schedule.
c
140 do 150 nhr=1,24
        tsurf=t(npx)
        if (hrday .lt. 24.0)then
            hrday=hrday+1.0
        else
            hrday=1.0
            dayt=dayt+1.0
        end if
        call surfht(dayt,hrday,tsurf,tr,heq,tair)
c      write (15,3000)xxxx,tsurf,tr,heq
        call nopour(npx,dt,tr,heq,heff)
        do 145 i=ibotp,npx
            telap(i)=telap(i)+dt
            elapt=telap(i)
            ccp=cp(i)
            call intht(elapt,dt,ccp,source)
c          call inthtx(elapt,dt,ccp,source)
            qstar(i)=source
145 continue
c      write (15,208)xxxx,qstar(39),qstar(50),qstar(58)
        xxxx=xxxx+1.0
        hour=hour+1.0
        depth=zdepth/.30548
        dday=hour/24.0
        qbott=zkoz*(t(ibotp)-t(ibot))
        qtop=heff*(t(npx)-tr)
        write (14,4000)dday,qtop,qbott
        qtpsum=qtpsum+qtop*3600.0
        qbtsum=qbtsum+qbott*3600.0
150 continue
        write (9,212)dday,daymax
        write (13,207)dday,depth
        deltaz=.0415
        axpos=-deltaz/2.0
        do 151 i=ibotp,npx
            axpos=axpos+deltaz
            tdegc=t(i)-273.15
            write (12,207)axpos,tdegc
151 continue
        write (12,215)week

```

```

        write (12,5003)
5003 format ('end week')
c
c   This loop (153) calculates and reads out the peak temperature
c   once a day.
c
        tpeak=t(ibotp)
        do 153 i=ibotp,npx
            if (t(i) .lt. tpeak)go to 153
            tpeak=t(i)
153 continue
        tpeak=tpeak-273.15
        write (11,207)dday,tpeak
        nday=nday+1
        day=day+1.0
        if (nday .gt. 7)go to 100
        go to 105
160 write (9,210)
        write (14,215)wkmax
        write (14,220)qtpsum
        write (14,225)qbtsum
        qltot=qtpsum+qbtsum
        qtrat=qtpsum/qltot
        qbrat=qbtsum/qltot
        write (14,207)qtrat,qbrat
c
c   *****
c   end of the computational loop
c   *****
c
        week=week-1.0
        write (12,210)
        write (12,2500)week
        write (12,2501)
        deltaz=.0415
        axpos=-deltaz/2.0
        do 1100 i=ibotp,npx
            axpos=axpos+deltaz
            tdegc=t(i)-273.15
            write (12,207)axpos,tdegc
1100 continue
        depthg=z(ibot)/2.0
        grtmp=t(ibot)-273.15
c        write (15,207)depthg,grtmp
        do 1110 i=39,1,-1
            depthg=depthg+.5*(z(i+1)+z(i))
            grtmp=t(i)-273.15
c        write (15,207)depthg,grtmp
1110 continue
        pause
200 format (g15.6,i5,2g15.6,'pour')
205 format (g15.6,i5,2g15.6,'no pour')
206 format (1x,'      DAY      PEAK TEMP. ')
207 format (2g16.8)
208 format (4g16.8)
210 format ('end')
212 format (1x,'day=',g14.6,' daymax=',g14.6)
215 format (1x,'week=',g14.6)
220 format (1x,'top heat trans loss=',g20.8,' J/m^2')
225 format (1x,'bottom heat trans loss=',g20.8,' J/m^2')
400 format (1x,'pour temp =',g14.6,' "K"')
401 format (1x,'start date =',g14.6)
402 format (1x,'pour schedule = ',i4,' hrs/day, 3 days/wk')
1300 format (1x,'      day      depth (ft)')

```



```

2000 format (1x,2g18.6)
2010 format (1x,i5,g18.6)
2500 format (1x,'week = ',g18.6)
2501 format (1x,'    height (m)          temp (K) ')
3000 format (1x,4g16.8)
4000 format (1x,3g18.8)
      stop
      end

```

```

      subroutine airtmp(dayt,hrday,tair)
c
c *****
c This subroutine calculates the air temperature based on the
c day of the year and the hour of the day. The average daily
c temperature varies on an annual basis +/- 20 deg F about the mean
c of 65 deg F. The diurnal variation is +/- 12 deg F about the
c daily average temperature.
c *****
c
c dayt      day of the year starting with Jan 15.
c hrday     hour of the day.
c
c
      c1=.017214206
      c2=.261799388
      tbar=273.15+18.3333
      if (dayt .le. 365.0)go to 100
      dayt=dayt-365.0
100  tavg=tbar-11.11111*cos(c1*dayt)
      if (hrday .le. 24.0)go to 200
      hrday=hrday-24.0
200  tair=tavg-6.666666*cos(c2*hrday)
      return
      end

```

```

      subroutine intht(elapt,dt,ccp,source)
c
c *****
c This subroutine calculates the saltstone source term as a
c function of the elapsed time.
c *****
c
c elapt     elapsed time (sec)
c dt        time step size (sec)
c ccp       specific heat (j/kg K)
c source    source term (W/kg)
c
c
      pi=3.14159265
      bb=.065788
      ttt=(elapt-(7.0/16.0)*dt)/3600.0
      ddt=(dt/8.0)/3600.0
      qsum=0.0
      do 100 j=1,8
        If (ttt .lt. 2.5) then
          dtdt=0.0
        else if (ttt .lt. 7.5) then
          dtdt=8.2*sin((1.0-(ttt-2.5)/5.5)*pi)

```

```

else if (ttt .lt. 15.5) then
  dtdt=2.31-.23875*(ttt-7.5)
else if (ttt .lt. 17.0) then
  dtdt=.4+1.7*sin((1.0-(ttt-15.5)/1.5)*pi)
else if (ttt .lt. 45.0) then
  allp=.036484
  dtdt=.3/(exp(allp*(ttt-17.0)))
else if (ttt .lt. 175.0) then
  dtdt=.10801-.0006*(ttt-45.0)
else if (ttt .lt. 275.0) then
  coef=.012
  dtdt=.03+coef*sin(((ttt-175.0)/100.0)*pi)
else if (ttt .lt. 450.0) then
  dtdt=0.02+.01/exp(bb*(ttt-275.0))
else if (ttt .lt. 800.0) then
  dtdt=.015+.005/exp(bb*(ttt-450.0))
else
  dtdt=0.0
end if
qsum=qsum+dtdt
ttt=ttt+dtdt
100 continue
qsum=qsum/8.0
source=(ccp*qsum)/3600.0
return
end

subroutine newton(a,b,c,tr)
c
c *****
c This subroutine solves the fourth order polynomial for the roof
c temperature:
c
c 
$$a*(tr^4)+b*tr+c=0$$

c
c Inputs are the three coefficients (a,b,& c), and the output is the
c roof temperature. The solution technique is Newton's iterations.
c *****
c
  tnew=300.0
100 told=tnew
  f=a*(told**4)+b*told+c
  dfdts=4.0*a*(told**3)+b
  tnew=told-f/dfdts
  diff=abs(tnew-told)
  if (diff .gt. .01) go to 100
  tr=tnew
  return
end

subroutine nopour(npix,dt,tr,heq,heff)
c
c *****
c This subroutine updates the temperature distribution in the
c saltstone during the periods when pouring is not occurring.
c *****
c
  dimension a(222),b(222),c(222),d(222),x(222)
  common t(222),z(222),rho(222),cp(222),zk(222),qstar(222)

```

```

c
c  t(i)      nodal temperatures
c  z(i)      nodal delta z
c  rho(i)    nodal densities
c  cp(i)     nodal specific heats
c  zk(i)     nodal thermal conductivities
c  qstar(i)  nodal internal heat generation
c
      npx=npx-1
      npmm=npx-2
      heff=2.0*heq*zk(npx)/(heq*z(npx)+2.0*zk(npx))
c
c  Elements of the tridiagonal matrix are calculated
c
      a(1)=-zk(1)*zk(2)*dt/
&          (rho(2)*cp(2)*z(2)*
&          (z(1)*zk(2)+z(2)*zk(1)))
      b(1)=t(1)+(zk(1)*zk(2)*dt/(rho(1)*cp(1)*z(1)*
&          (z(1)*zk(2)+z(2)*zk(1))))*(t(2)-t(1))
&          +qstar(1)*dt/cp(1)
      c(1)=-zk(1)*zk(2)*dt/
&          (rho(1)*cp(1)*z(1)*
&          (z(1)*zk(2)+z(2)*zk(1)))
      do 425 i=2,npmm
          a(i)=-zk(i)*zk(i+1)*dt/
&              (rho(i+1)*cp(i+1)*z(i+1)*
&              (z(i)*zk(i+1)+z(i+1)*zk(i)))
          b(i)=t(i)-(zk(i-1)*zk(i)*dt/(rho(i)*cp(i)*z(i)*
&              (z(i-1)*zk(i)+z(i)*zk(i-1))))*(t(i)-t(i-1))+
&              (zk(i)*zk(i+1)*dt/(rho(i)*cp(i)*z(i)*
&              (z(i)*zk(i+1)+z(i+1)*zk(i))))*(t(i+1)-t(i))
&              +qstar(i)*dt/cp(i)
          c(i)=-zk(i)*zk(i+1)*dt/
&              (rho(i)*cp(i)*z(i)*
&              (z(i)*zk(i+1)+z(i+1)*zk(i)))
425  continue
      a(npmm)=-zk(npmm)*zk(npmm)*dt/(rho(npmm)*cp(npmm)*z(npmm)*
&          (z(npmm)*zk(npmm)+z(npmm)*zk(npmm)))
      b(npmm)=t(npmm)+qstar(npmm)*dt/cp(npmm)
&          -(zk(npmm)*zk(npmm)*dt/(rho(npmm)*cp(npmm)*z(npmm)*
&          (z(npmm)*zk(npmm)+z(npmm)*zk(npmm))))*(t(npmm)-t(npmm))
&          +(zk(npmm)*zk(npmm)*dt/(rho(npmm)*cp(npmm)*z(npmm)*
&          (z(npmm)*zk(npmm)+z(npmm)*zk(npmm))))*(t(npmm)-t(npmm))
      c(npmm)=-zk(npmm)*zk(npmm)*dt/(rho(npmm)*cp(npmm)*z(npmm)*
&          (z(npmm)*zk(npmm)+z(npmm)*zk(npmm)))
      b(npmm)=t(npmm)*(1.0-zk(npmm)*zk(npmm)*dt/(rho(npmm)*cp(npmm)*z(npmm)*
&          (z(npmm)*zk(npmm)+z(npmm)*zk(npmm))))-
&          heff*dt/(2.0*rho(npmm)*cp(npmm)*z(npmm))+
&          zk(npmm)*zk(npmm)*dt*t(npmm)/(rho(npmm)*cp(npmm)*z(npmm)*
&          (z(npmm)*zk(npmm)+z(npmm)*zk(npmm)))+
&          heff*dt*t(r)/(rho(npmm)*cp(npmm)*z(npmm))
&          +qstar(npmm)*dt/cp(npmm)

```

```

d(1)=1.0+zk(1)*zk(2)*dt/(rho(1)*cp(1)*z(1)*
&      (z(1)*zk(2)+z(2)*zk(1)))

do 426 i=2,npmn

    d(i)=1.0+(zk(i)*dt/(rho(i)*cp(i)*z(i)))*
&      (zk(i+1)/(z(i)*zk(i+1)+z(i+1)*zk(i))+
&      zk(i-1)/(z(i-1)*zk(i)+z(i)*zk(i-1)))

426 continue

d(npm)=1.0+(zk(npm)*dt/(rho(npm)*cp(npm)*z(npm)))*
&      (zk(npx)/(z(npm)*zk(npx)+z(npx)*zk(npm))+
&      zk(npmm)/(z(npmm)*zk(npm)+z(npm)*zk(npmm)))

d(npx)=1.0+(dt/(rho(npx)*cp(npx)*z(npx)))*
&      (zk(npm)*zk(npx)/(z(npm)*zk(npx)+z(npx)*zk(npm))+heff/2.0)
c
c  subroutine tri solves a tridiagonal system of equations
c  with the Thompson algorithm.  n is the number of equations.
c
    n=npx
    call tri(n,a,d,c,b,x)
    do 450 i=1,npx
        t(i)=x(i)
450 continue
    return
    end

    subroutine pour(npx,dt,tpour,tr,heq)
c
c  *****
c  This subroutine updates the temperature distribution in the
c  saltstone during the pouring phase of the operation.
c  *****
c
    dimension a(222),b(222),c(222),d(222),x(222)
    common t(222),z(222),rho(222),cp(222),zk(222),qstar(222)
c
c  t(i)      nodal temperatures
c  z(i)      nodal delta z
c  rho(i)    nodal densities
c  cp(i)     nodal specific heats
c  zk(i)     nodal thermal conductivities
c  qstar(i)  nodal internal heat generation
c
    npm=npn-1
    npmm=npn-2
c
c  Elements of the tridiagonal matrix are calculated
c
    a(1)=-zk(1)*zk(2)*dt/
&      (rho(2)*cp(2)*z(2)*
&      (z(1)*zk(2)+z(2)*zk(1)))

    b(1)=t(1)+(zk(1)*zk(2)*dt/(rho(1)*cp(1)*z(1)*
&      (z(1)*zk(2)+z(2)*zk(1))))*(t(2)-t(1))
&      +qstar(1)*dt/cp(1)

    c(1)=-zk(1)*zk(2)*dt/
&      (rho(1)*cp(1)*z(1)*
&      (z(1)*zk(2)+z(2)*zk(1)))

```

```

do 425 i=2,npm
  a(i)=-zk(i)*zk(i+1)*dt/
&      (rho(i+1)*cp(i+1)*z(i+1)*
&      (z(i)*zk(i+1)+z(i+1)*zk(i)))

  b(i)=t(i)-(zk(i-1)*zk(i)*dt/(rho(i)*cp(i)*z(i)*
&      (z(i-1)*zk(i)+z(i)*zk(i-1))))*(t(i)-t(i-1))+
&      (zk(i)*zk(i+1)*dt/(rho(i)*cp(i)*z(i)*
&      (z(i)*zk(i+1)+z(i+1)*zk(i))))*(t(i+1)-t(i))
&      +qstar(i)*dt/cp(i)

  c(i)=-zk(i)*zk(i+1)*dt/
&      (rho(i)*cp(i)*z(i)*
&      (z(i)*zk(i+1)+z(i+1)*zk(i)))

425 continue
  a(npm)=-zk(npm)*dt/(rho(npx)*cp(npx)*z(npx)*z(npm))

  b(npm)=t(npm)+qstar(npm)*dt/cp(npm)
&      -(zk(npm)*zk(npm)*dt/(rho(npm)*cp(npm)*z(npm)*
&      (z(npm)*zk(npm)+z(npm)*zk(npm))))*(t(npm)-t(npm))
&      -(zk(npm)*dt/(rho(npm)*cp(npm)*z(npm)*z(npm)))*t(npm)

  c(npm)=-2.0*zk(npm)*dt/(rho(npm)*cp(npm)*z(npm)*z(npm))

  b(npx)=tpour+zk(npm)*dt*t(npm)/(rho(npx)*cp(npx)*z(npx)*z(npm))
&      +heq*dt*tr/(rho(npx)*cp(npx)*z(npx))

  d(1)=1.0+zk(1)*zk(2)*dt/(rho(1)*cp(1)*z(1)*
&      (z(1)*zk(2)+z(2)*zk(1)))

do 426 i=2,npm

  d(i)=1.0+(zk(i)*dt/(rho(i)*cp(i)*z(i)))*
&      (zk(i+1)/(z(i)*zk(i+1)+z(i+1)*zk(i))+
&      zk(i-1)/(z(i-1)*zk(i)+z(i)*zk(i-1)))

426 continue
  d(npm)=1.0+(zk(npm)*dt/(rho(npm)*cp(npm)*z(npm)))*
&      (zk(npm)/(z(npm)*zk(npm)+z(npm)*zk(npm))+1.0/z(npm))

  d(npx)=1.0+2.0*zk(npm)*dt/(rho(npx)*cp(npx)*z(npx)*z(npm))
&      +heq*dt/(rho(npx)*cp(npx)*z(npx))

c
c  subroutine tri solves a tridiagonal system of equations
c  with the Thompson algorithm.  n is the number of equations.
c
  n=npx
  call tri(n,a,d,c,b,x)
  do 450 i=1,npx
    t(i)=x(i)
450 continue
  return
end

  subroutine surfht(dayt,hrday,tsurf,tr,heq,tair)

c
c *****
c This subroutine calculates the vault roof temperature and the heat
c transfer coefficient for combined natural convection and thermal
c radiation heat transfer between the surface of the saltstone and the

```

```

c  roof of the vault.
c  *****
c
c  sigma      Stefan-Boltzmann constant
c  emmr       emissivity of vault roof
c  emmst      emissivity of saltstone
c  alpsol     absorptivity of roof to solar radiation
c  coefcv     forced convection heat transfer coefficient for roof top
c  coefnc     natural convection heat transfer coefficient (air space)
c  tbar       air space temperature
c  zkt        thermal conductivity of air space
c  znewt      kinematic viscosity of air space
c  zalpha     thermal diffusivity of the air space
c  beta       bulk coefficient of thermal expansion of the air space
c
c      pi=3.1415927
c      sigma=5.67e-8
c      emmr=.92
c      emmst=.94
c      alpsol=.3
c      coefcv=5.0
c      coefnc=0.0
c      g=9.80665
c      emstrf=(emmr*emmst)/(emmr+emmst-emmr*emmst)
c
c  This subroutine calculates the air temperature.
c  tsky is the effective sky temperature for infrared radiation.
c
c      call airtmp(dayt,hrday,tair)
c      tsky=.0552*(tair**1.5)
c
c  This "if" group calculates the solar flux qt (W/m^2)
c
c      if (hrday .lt. 7.0)then
c          qt=0.0
c      else if (hrday .le. 19.0)then
c          qt=800.0*sin(((hrday-7.0)/12.0)*pi)+100.0
c      else
c          qt=0.0
c      end if
c
c  The roof temperature (tr) is solved for with the assumption
c  that there is no natural convection with the saltstone.
c
c      a=(emmr+emstrf)*sigma
c      b=coefcv+coefnc
c      c=-(emmr*sigma*(tsky**4)+coefcv*tair+coefnc*tsurf+alpsol*qt
c      & +emstrf*sigma*(tsurf**4))
c      call newton(a,b,c,tr)
c
c  The natural convection heat transfer coefficient for the air
c  space (hcv) is calculated and the roof temperature is
c  recalculated.
c
c      tbar=(tr+tsurf)/2.0
c      zkt=.02227+7.76e-5*(tbar-250.0)
c      znewt=9.49e-6+1.13e-7*(tbar-250.0)
c      zalpha=.13161e-4+1.67e-7*(tbar-250.0)
c      beta=1.0/tbar
c      deltt=tsurf-tr
c      if (deltt .gt. 0.0)then
c          hcv=.0555*zkt*(((beta*g*deltt)/(znewt*zalpha))**.33333)
c      else
c          hcv=0.0

```

```

      end if
      coefnc=hcv
      a=(emmr+emstrf)*sigma
      b=coefcv+coefnc
      c=-(emmr*sigma*(tsky**4)+coefcv*tair+coefnc*tsurf+alpsol*qt
&      +emstrf*sigma*(tsurf**4))
      call newton(a,b,c,tr)
c
c   The equivalent heat transfer coefficient for both natural
c   convection and thermal radiation between the saltstone and
c   the vault roof is calculated.
c
      tbar=(tr+tsurf)/2.0
      zkt=.02227+7.76e-5*(tbar-250.0)
      znewt=9.49e-6+1.13e-7*(tbar-250.0)
      zalp=1.13161e-4+1.67e-7*(tbar-250.0)
      beta=1.0/tbar
      deltt=tsurf-tr
      if (deltt .gt. 0.0) then
        hcv=.0555*zkt*((beta*g*deltt)/(znewt*zalp))**.33333
      else
        hcv=0.0
      end if
      hrad=emstrf*sigma*(tr+tsurf)*((tr**2)+(tsurf**2))
      heq=hcv+hrad
      return
      end

      subroutine tri(n,a,d,c,b,x)
c
c   *****
c   This subroutine solves a tridiagonal matrix
c   *****
c
      dimension a(n),d(n),c(n),b(n),x(n)
      do 2 i=2,n
        xmult=a(i-1)/d(i-1)
        d(i)=d(i)-xmult*c(i-1)
        b(i)=b(i)-xmult*b(i-1)
2      continue
      x(n)=b(n)/d(n)
      do 3 i=n-1,1,-1
        x(i)=(b(i)-c(i)*x(i+1))/d(i)
3      continue
      return
      end

```

DATE

FILMED

3 / 22 / 94

END