NOTICE

**CERTAIN DATA
CONTAINED IN THIS
DOCUMENT MAY BE
DIFFICULT TO READ
IN MICROFICHE
PRODUCTS.**

Design and Simulation of Latent Heat Storage Units

Final Report

N. Shamsundar, E. Stein, E. Rooz,
E. Bascaran, and T.C. Lee
University of Houston
Houston, Texas

NREL Technical Monitor: R. Gerald Nix

National Renewable Energy Laboratory
(formerly the Solar Energy Research Institute)
1617 Cole Boulevard
Golden, Colorado 80401-3393
A Division of Midwest Research Institute
Operated for the U.S. Department of Energy
under Contract No. DE-AC02-83CH10093

Prepared under Subcontract No. XX-7-07028-1

April 1992

MASTER

This report describes subcontracted research. The report is unreviewed and expresses only the opinions of the author[s]. It has been prepared for reproduction from the best available copy.

NOTICE

This report was prepared as an account of work sponsored by an agency of the United States government. Neither the United States government nor any agency thereof, nor any of their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States government or any agency thereof. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States government or any agency thereof.

Printed in the United States of America
Available from:
National Technical Information Service
U.S. Department of Commerce
5285 Port Royal Road
Springfield, VA 22161

Price: Microfiche A01
Printed Copy A04

Codes are used for pricing all publications. The code is determined by the number of pages in the publication. Information pertaining to the pricing codes can be found in the current issue of the following publications which are generally available in most libraries: *Energy Research Abstracts (ERA)*; *Government Reports Announcements and Index (GRA and I)*; *Scientific and Technical Abstract Reports (STAR)*; and publication NTIS-PR-360 available from NTIS at the above address.

SUMMARY

This report presents the results of two years of research and development on passive latent heat storage systems. Analytical models have been developed and extended, and a computer code for simulating the performance of a latent heat storage has been developed. The code is intended to be merged into a larger solar energy system simulation code and used for making realistic system studies. Simulation studies using a code which has a flexible and accurate routine for handling the storage subsystem should lead to the development of better systems than those in which storage is added on after the rest of the system has already been selected and optimised.

The heat storage material is a eutectic mixture of inorganic salts which fills the space outside the tubes in a shell and tube heat exchanger. During the heat storage phase, excess heat from the energy collection system is dumped into the storage unit by a heat transfer fluid or a power cycle working fluid flowing in the tubes. Almost all of this heat is stored as heat of fusion in the storage material, which melts in the process. In the heat recovery phase, the heat transfer fluid picks up the stored heat of fusion, causing the storage material to solidify.

Predicting the heat transfer rates and the transient response of such storage units involves the solution of a transient heat transfer problem with moving boundaries. In the heat recovery phase, the dominant mechanism of heat transfer in the storage material is heat conduction in the solidified zone.

During heat storage, both conduction and convection zones may exist. In the initial stages of melting the storage substance, both these zones will exist simultaneously. Later, the conduction zone will give way to a full-convection zone.

Analytical models for treating both the storage and recovery phases of operation have been developed. The heat recovery models are considerably more advanced, and have been tested by making comparisons to results from other calculations. They can be used with confidence in their range of applicability, which is delineated in this report. The simulation code incorporates only these models, and is at present restricted to passive shell-and-tube heat exchangers in which the heat storage material is on the shell side and exchanges heat with a fluid flowing in the tubes.

A heat storage model has been developed. This model accounts for simultaneously existing conduction and convection zones. Although this model accounts for axial temperature changes and other controlling effects that are present in a storage unit, it is based on experimental data for melting on long tubes with little axial variation. The model needs verification by comparison to experimental results from a configuration which is closer to that encountered in a heat exchanger. Comparisons should also be made by to results generated by purely numerical solution methods(finite-element or finite-difference).

A model has been set up for calculating the performance improvement resulting from the use of finned tubes instead of plain tubes. Fins can overcome the low conductance of common

heat storage materials, and allow the designer to choose a material for its high storage capacity and other desirable properties, without its low conductivity becoming a determining factor. This model pertains to heat recovery only, and has been verified by comparison to experimental results.

All the models mentioned here apply to shell-and-tube heat exchangers with any given tube size, spacing and length. They account for time-dependent flow-rate and inlet temperature of the tube-side fluid. The heat recovery models even allow discontinuous jumps in the coolant flow. All the models allow the inlet temperature of the fluid to be discontinuous. These features give the computer code considerable flexibility in faithfully tracing the performance of a storage unit which serves its proper role in the power system, without being artificially constrained to some simple modes of operation.

The simulation code will give results for any specified, time-dependent flow-rate and inlet temperature of the fluid in the tubes. The main simulation code should call this code with any time interval that is suitable, and specify the variation of these two quantities. The storage code will return the outlet temperature of the fluid, the instantaneous heat transfer rate, the extent of phase change, etc.

This report is in three main parts. The first part describes the analytical models underlying the computer code. The second part describes the code, its features, and how it interfaces with a solar-thermal system simulator. This part can be used as a user manual for the code. The first two parts, taken together, can be used as a reference manual by those wis-

hing to extend or otherwise alter the code prior to merging it with their own codes.

The third part describes the heat storage model, the heat recovery model for finned tubes, and some of the results given by these models. Fins are shown to be capable of giving striking improvements in storage system performance. They can lead the user close to the ideal of constant temperature operation which is the main motivation for using latent heat storage. This part also describes a simple computer program for quickly obtaining the design parameters for a latent heat storage unit with a specified performance. The development of the finned-tube model and the design program was supported by funding from the Johnson Space Center of NASA.

Efforts are continuing on extending the code to include these models, but at a reduced pace because we are unfunded now. All the code-testing undertaken by us was conducted using a small driver routine which called the storage simulation routines with test data introduced directly from the keyboard. The code has not been tested to see how well it works with any solar thermal system simulator. The code is being currently extended to give results in the form of plots in addition to printouts. This feature is likely to be less portable than the rest of the code.

CONTENTS

PART I MODELS FOR HEAT RECOVERY

FREEZING ON A SINGLE TUBE WITH STEADY FLOW 1-1

HEAT RECOVERY WITH VARIABLE COOLANT FLOW 2-1

PART II SIMULATION PROGRAM

USER MANUAL FOR HEAT RECOVERY SIMULATION PROGRAM 3-1

PART III OTHER MODELS AND PROGRAMS

DESIGN METHOD FOR LATENT HEAT STORAGE UNITS 4-1

PART I

MODELS FOR HEAT RECOVERY

I.1 FREEZING ON A SINGLE TUBE WITH STEADY FLOW

I.1.1. INTRODUCTION

This chapter contains a presentation of the model for the freezing of a substance outside a single circular tube, with a heat-transfer fluid flowing inside it. This fluid, which we refer to as the "coolant" in this chapter, enters the tube at a temperature which is specified, perhaps in a time-dependent way, and picks up heat from the latent heat storage material. This heat flow causes some of the heat storage material to freeze on the outside of the tube, and causes the temperature of the coolant to rise. Thus, the driving temperature difference, which is the difference between the melting temperature of the storage material and the local instantaneous temperature of the coolant, decreases in the direction of the coolant flow, and varies with time. The model for predicting the heat transfer and freezing rates for this basic situation is the core around which the rest of this report is built.

I.1.2 ANALYSIS

An analytical solution to this problem is very desirable to have since such a solution would help in understanding the interplay of various parameters and in making efficient calculations without having to resort to long numerical simulations. To obtain such a solution, a few simplifying assumptions will be made. These assumptions will be checked after the model is developed.

Assumptions:

- (i) The heat storage substance is completely liquid when the heat recovery phase begins.
- (ii) The initial superheat in the liquid is negligible compared to the latent heat of fusion.
- (iii) Freezing takes place with a smooth interface between liquid and solid, with no cavities in the frozen zone or on the tube itself.
- (iv) The tube has a length much larger than its diameter, and axial conduction in the frozen layer and in the coolant can be ignored.
- (v) The freezing rate is slow enough that the temperature distribution in the frozen zone can be approximated as being quasi-steady.
- (vi) The residence time of the coolant in the tube is very small compared to the typical freezing time.
- (vii) The convective heat transfer in the tube is fully developed[1] and uniform over the tube.
- (viii) As the heading of the chapter indicates, the flow rate of the coolant is assumed to be steady.

The reader can consider these assumptions and think up arguments to justify them. We prefer to go ahead with obtaining the solution, letting the justification of the assumptions rest upon how well the results compare with alternative numerical results which were obtained without making these assumptions.

Energy Balance:

We refer to Figure I.1.1 during the following derivation. Consider a thin slice of the tube lying between axial positions x and $x + dx$. Because of assumption (iv), the heat conduction in the solid is purely radial, and the heat transfer rate per unit length is given by

$$Q = 2\pi k(T_{\text{sat}} - T_w) / \ln(r/R) \quad (1.1.1)$$

Fig. 1: Sketches of (a) problem and (b) finite-element mesh.

This heat must flow into the coolant by convection, so that

$$Q = 2\pi R h (T_w - T_f) \quad (1.1.2)$$

In turn, this heat causes the temperature of the coolant to rise, as given by the equation

$$Q = \dot{m} c \partial T_f / \partial x \quad (1.1.3)$$

The final term entering the energy balance is obtained by noting that this heat flow causes an appropriate amount of storage material to freeze. Therefore,

$$Q = \partial / \partial t (\pi (r^2 - R^2) \rho \lambda) \quad (1.1.4)$$

Solution:

To obtain an analytical solution of these equations needs some skill in manipulating the equations. Rather than leading the reader through these manipulations, we shall merely list the solution, asking the interested reader to verify that it is indeed the solution by substituting it into equations (1.1.1 - 1.1.4).

The solution will be expressed in terms of nondimensional variables and parameters. Of these, the most important is the frozen fraction, F , which is the area of the frozen region in the radial plane divided by the area of cross-section of the tube, that is,

$$F = (r^2 / R^2 - 1) \quad (1.1.5)$$

The value of F at the inlet, F_0 , plays the role of a time variable. The actual nondimensional time variable is τ ,

which is defined as

$$\tau = \int_0^t k(T_{\text{sat}} - T_0) dt / f \lambda R^2 \quad (1.1.6)$$

The relation between F_0 and τ is the equation

$$\tau = F_0 / 2Bi \left\{ (1/4) \left((1+F_0) \ln(1+F_0) - F_0 \right) \right\} \quad (1.1.7)$$

The frozen fraction at axial position x is given by the implicit equation

$$NTU_x = 2\pi R h x / mc = \ln(F_0/F) + (Bi/2) (G_1(F_0) - G_1(F)) \quad (1.1.8)$$

The nondimensional fluid temperature and the heat exchanger effectiveness, ϵ , are related by the amazing result

$$(T_f - T_0) / (T_{\text{sat}} - T_0) = \epsilon = 1 - F/F_0 \quad (1.1.9)$$

The total amount of solid formed from the beginning of the freezing over the entire tube is given in terms of the overall frozen fraction, $\bar{F}$, by

$$\begin{aligned} NTU_x F &= \int_0^{NTU_x} F^x dNTU_x \\ &= (F_0 - F) + (Bi/2) (G_0(F_0) - G_0(F)) \end{aligned} \quad (1.1.10)$$

The auxiliary functions $G_0(F)$ and $G_1(F)$ in the preceding equations are defined by

$$G_0(F) = \int_0^F \ln(1+F) dF, \quad G_1(F) = \int_0^F \ln(1+F) dF/F \quad (1.1.11)$$

The function G_0 is easily calculated since

$$G_0(F) = (1 + F) \ln(1 + F) - F \quad (1.1.12)$$

To calculate $G_1(F)$, we have developed an approximation to it -- no exact analytical representation exists. For the range of $F < 10$, which is usually the case, use

$$G_1(F) = p(1 + 17p^2/450)/(1 + p^2/100) + p^2/4 \quad (1.1.13)$$

where

$$p = \ln(1+F) \quad (1.1.14)$$

For $F > 10$, more work is necessary. First, find $F' = 1/F$ and calculate $p' = \ln(1 + F')$. Then, find G'_1 by using p' instead of p in equation (1.1.13). Finally, calculate

$$G_1 = \pi^2/6 + (p - p')^2/2 - G'_1 \quad (1.1.15)$$

When these approximations are employed in the appropriate ranges of F as indicated, the error in G_1 is less than 1 in 10^5 .

A first look at the form of these results would suggest that they are hard to use because the quantities F and ϵ appear in more than one place, and we need to solve implicit equations to obtain their values. It turns out that the Newton-Raphson method can be used to solve for them without any difficulty, using any reasonable initial guess. Later, we shall illustrate how such initial guesses can be obtained.

1.1.3 VERIFICATION

Two independent means of verification were used. The first was a complete analysis by Sparrow and Hsu[2], which did not use assumptions (iv) to (vi), and generated results by employ-

ying a finite-difference method. The second was the a set of finite-element results produced specifically to test the model. These results, described in E.Stein's thesis[3], conclusively showed that the analytical model and the solutions given above are highly accurate.

1.1.4 DISCUSSION

The chief deficiencies in the analytical model as regards application to storage systems are (i) restriction to a single tube, and (ii) restriction to steady flow. Removing the second restriction is the topic of the next chapter. Therefore, let us consider only the first restriction here, and ask what happens when we have an array of tubes rather than a single tube.

Until the solid layers on adjacent tubes contact one another, the freezing process on each tube is independent of the freezing on other tubes, and our results remain valid. After contact is established, the heat transfer falls rapidly, because the area of the liquid-solid interface diminishes because of the contact. The time interval in which this happens is, for most tube layouts, a small fraction of the time needed to initiate this problem. Numerical results for planar solidification around a tube array, presented by Shamsundar and Srinivasan[4], support this.

For this reason, and because of the low heat transfer rates, it is reasonable to regard the storage unit as fully discharged when the solid layers on adjacent tubes contact one another. Thus, we may disregard this small time interval altogether, at the expense of slightly overdesigning the storage unit

on slightly underestimating the heat recovered from a given storage unit. In keeping with these arguments, the simulation and design codes were written based on the single-tube formulae rather than on data for tube arrays.

REFERENCES:

1. W.M. Kays and M.C. Crawford, Convective Heat and Mass Transfer, McGraw-Hill,
2. E.M. Sparrow and C.F. Hsu, "Analysis of Two-Dimensional Freezing on a Coolant-Carrying Tube," International Journal of Heat and Mass Transfer, Vol. 24, 1981, pp. 1345-57.
3. E. Stein, M.S. Thesis, Department of Mechanical Engineering, University of Houston.
4. N. Shamsundar and R. Srinivasan, "Analysis of Energy Storage by Phase-Change in an Array of Cylindrical Tubes",

HEAT RECOVERY WITH VARIABLE COOLANT FLOW/INTRODUCTION

Our objective in this chapter is to remove the main limitation of the model of Chapter 1.1: the assumption that the flow rate of the coolant was independent of time. Removing this limitation is mandatory since any solar thermal system will definitely operate under variable insolation and load conditions. After all, one of the functions of storage is to act as a buffer between the solar collection system and the thermal-electric conversion system, and to cushion mismatches between the characteristics of these systems. Therefore, a thermal storage unit will rarely operate with constant coolant flow.

ANALYSIS

No exact analysis is possible for the complicated situation which we intend to resolve. Our approach is based on obtaining reference results using a finite-element method, and trying to adapt the equations of the previous chapter to the variable flow case. Many false starts were made before the right combination of variables was found in terms of which a successful prediction of the variable flow case could be made. Therefore, although the selection of the form of the correlation may seem illogical and unnatural, we feel that they are justified by success in predicting results.

FINITE ELEMENT CODE

To obtain results with sufficient precision to enable analytical approximations to be judged against them, we decided to use a novel form of the finite-element method, in which finite-element discretization is applied in time as well as in

space. Although this has seldom been done -- most people use finite-elements in time and finite-differences in time --, we feel that this is a logical decision to make, based on the following reasoning.

In a freezing problem, we have a moving boundary which moves at a nonuniform rate in time. Even in a plane freezing problem, we have a curved boundary in the position - time plane, and curved boundaries are what the finite-element method is for. Of course, we have the additional problem that the boundary in the position - time plane is unknown in addition to being curved, but that can be dealt with.

The finite-element method has the additional advantage of allowing us to include or drop any of the terms, in the governing differential equations, whose separate influence we wish to assess. We simply multiply each term in the equation by an influence coefficient before formulating the finite-element equations and writing the finite-element code. At run time, we set these coefficients equal to unity or zero, depending on whether or not we wish to keep the corresponding terms.

We shall include a few specific details of the finite-element formulation for reference, and to point out the key features. The frozen layer outside a single layer is discretized in space as shown in Figure 1.2.1, using a trapezoidal mesh. The figure also shows how a trapezoid deforms in time because of the moving boundary; note that the distortion is not known a priori. To obtain the finite-element equations, we apply the Galerkin method to the space-time domain formed by the longitudinal sections of the solid at times t and $t + dt$. A similar formulation

is made for the coolant in the tube, but this domain is just rectangular because it remains equal to the length of the tube at all times. The coupling between the fluid inside the tube and the frozen solid is automatically carried over into the finite-element formulation.

To represent the integrals within each oblique-box-shaped or rectangular element, we apply the usual isoparametric transformation to a cube or a square. Because of the moving solid-liquid boundary, the Jacobian of the transformation is not constant. Therefore, when the integrals are reduced by Gaussian integration, and the element equations are assembled, the coefficients in the equations contain terms which are dependent on the unknown location of the interface. This is treated by an implicit iteration scheme, which was found to converge in two or three iterations.

The usual tests to find the proper number of elements and time steps were made. A mesh of ten X ten elements was found to be adequate. The time steps had to be made small enough to make the total freezing time equivalent to a few hundred steps. These steps were established by dropping terms in the governing equations to make the finite element model conform to the analytical model of the previous chapter, and by making the steps small enough to reproduce the analytical results.

Full details of the formulation and a listing of the finite-element program are given in E. Stein's M.S. Thesis[3].

FINITE-ELEMENT RESULTS FOR CONSTANT FLOW

Several runs were made to find out the influence of

terms dropped from the equations in developing the analytical model. The results were shown to justify the assumptions of the previous chapter for the range of values of parameters of interest in thermal energy storage applications. A sample of these results is shown in Table 1.2.1.

MODEL FOR VARIABLE COOLANT FLOW

There are two principal parameters in the constant flow analysis which are affected by the coolant flow becoming variable in time. The flow rate appears directly in the NTU, and changes in the flow rate will change the convective coefficient h if the flow is turbulent. Changes in h will in turn affect both NTU and Bi . To help understand the effects of these changes, let us consider two cases for study.

In the first, we consider a fixed NTU, and investigate the effect of time-dependent Bi by considering two limiting values of Bi , namely, zero and infinite Bi . If we express the instantaneous heat flow rate for the whole heat exchanger, we get

$$\dot{Q} = (T_{sat} - T_o) / (R_{conduction} + R_{convection}) \quad (1.2.1)$$

To find the convection resistance, note that if the solid were to be infinitely conductive, we would have the same convection resistance still, so that, from the definition of effectiveness, $R_{convection} \equiv Q / \Delta T = 1 / (\epsilon_0 \dot{m} c)$, where ϵ_0 is the effectiveness with infinite conductivity for the solid, and is equal to $1 - \exp(-NTU)$, as can be seen from equation (1.1.8). Let the conduction resistance be expressed as $1 / (k \times S)$, where S is the conduction shape factor of the frozen layer. Of course, S varies with time for a given NTU and Bi .

Putting these relations together, we get

$$(\epsilon_0/\epsilon - 1)/Bi = (2\pi\epsilon_0/NTU_x)/S \quad (1.2.2)$$

Now we ask the crucial question: is the quantity S truly a shape factor? That is, is it independent of the conductivity of the solid and, therefore, of Bi ? More specifically, can we assert that S is only a function of the overall frozen fraction $\bar{F}$, and that the shape of the solid is independent of Bi ?

Figures 1.2.2 and 1.2.3 present evidence to support these assertions. The first of these shows the influence of different values of Bi for fixed values of NTU . The second shows the shape of the frozen region for various values of effectiveness. Note that NTU does not appear explicitly in the second figure, but is different for each point on the figure. We immediately see that the assertions are true only in an approximate sense, and only for small values of effectiveness and NTU . However, for the range of parameter values we expect to encounter, the deviation are small.

Based on these results, which were obtained from the constant flow rate formulae of Chapter 1.1, we make a claim which extends the formulae to the variable flow case. If some quantity, like the shape factor S , is almost the same for $Bi=\text{constant}=0.1$ and $Bi=\text{constant}=10$, it should retain the same values when Bi varies within these two extremes. Thus, we simply use the instantaneous value of Bi in the constant flow formulae. Similarly, when NTU varies with time, we use the instantaneous value of NTU .

The reader may have accepted the preceding arguments, but may still ask: "How accurately can we predict quantities of

direct interest, rather than shape factors and frozen layer shapes?" This is a very valid question, and we proceed to answer it by comparing the predictions of the formulae with those obtained from the finite element code.

CALCULATION PROCEDURE FOR VARIABLE FLOW

It is useful to collect here the calculation procedure for variable flow. The following steps have to be performed.

- (i) For the specified variation of coolant inlet temperature, $T_0(t)$, calculate τ as a function of t from the definition of τ^0 in equation (I.1.5). From the specified variation of m with time, establish Bi as a function of τ .
- (ii) Integrate equation (I.1.5) to establish the relation between values of τ and values of F_0 . This integration is done analytically for laminar flow, since Bi is constant then. For turbulent flow, use the Runge-Kutta or any other method to integrate this first order ordinary differential equation.
- (iii) To find the effectiveness and amount of solid at any given time, use the corresponding F_0 from step (ii) in equation (I.1.6) and solve using the Newton-Raphson method.

SUDDEN JUMPS IN COOLANT FLOW

This case is slightly harder to treat than continuously variable coolant flow, but may be important to consider in some storage applications. The preceding formulae break down slightly at the instant of the jump because the effectiveness undergoes a jump without any accompanying changes in the amount frozen, that is, F_0 and $\bar{F}$. Let subscripts + and - represent conditions before and after a jump. Since $F_+ = F_-$, we shall not show any subscripts for F and $\bar{F}$. The heat transfer rate per unit length is given by

$$\begin{aligned} Q_+ &= -\dot{m}_+ c_p (T_{sat} - T_{b+}) / \partial x \\ &= (T_{sat} - T_{b+}) / \left(\frac{1}{2\pi R h_+} + \frac{1}{4\pi k} \ln(1 + F) \right) \end{aligned} \quad (I.2.3)$$

$$\begin{aligned}
 Q_- &= -\dot{m}_- c \partial(T_{\text{sat}} - T_{b-})/\partial x \\
 &= (T_{\text{sat}} - T_{b-}) / \left((1/(2\pi R h_-)) + (1/4\pi k) \ln(1 + F) \right) \quad (1.2.4)
 \end{aligned}$$

If we eliminate x from these equations and express $T_{\text{sat}} - T_b$ in terms of ϵ , we get

$$\frac{d \ln(1 - \epsilon_+)}{d \ln(1 - \epsilon_-)} = \frac{\dot{m}_- [2/Bi_+ + \ln(1 + F_0(1 - \epsilon_-))]}{\dot{m}_+ [2/Bi_+ + \ln(1 + F_0(1 - \epsilon_-))]} \quad (1.2.5)$$

The inlet conditions for this equation are $\epsilon_+ = \epsilon_- = 0$ at $x = 0$. For turbulent flow, Bi_+ is different from Bi_- and this equation has to be integrated numerically. For laminar flow, $Bi_+ = Bi_-$ so that

$$(1 - \epsilon_+) = (1 - \epsilon_-)^{\dot{m}_-/\dot{m}_+} \quad (1.2.6)$$

To avoid falsely predicting sudden jumps in $\bar{F}$, we should use the following equation instead of (1.1.8). It is obtained by performing an energy balance for the whole heat exchanger.

$$NTU \, d\bar{F}/d\tau = 2 \, Bi \, \epsilon \quad (1.2.7)$$

RESULTS AND DISCUSSION

There are two main modes of flow of the coolant that have to be considered -- laminar and turbulent. With laminar flow, which is likely to be encountered if the coolant is a liquid, the heat transfer coefficient and therefore the Biot number will be independent of the coolant flow rate for long tubes. Therefore, the only quantity affected by changing the coolant flow will be NTU, which is inversely proportional to the flow rate. With turbulent flow, which is likely to be experien-

ced in storage units with a gaseous coolant, on the other hand, the convective coefficient varies as the flow rate raised to the power 0.8. Therefore, Bi will be proportional to the flow rate raised to the power 0.8. The NTU depends on both the convective coefficient and the flow rate, and varies as the flow rate raised to the power -0.2.

Three types of variation of the coolant flow were selected for study. These are: (1) coolant flow proportional to time, (2) coolant flow varying quadratically with time, and (3) sudden jumps in coolant flow. The plot of coolant flow against time which corresponds to any flow control strategy may be approximated by a series of straight lines, parabolas and jumps.

For all the results to be presented next, the length to radius ratio of the tubes was fixed at 100, and the NTU was fixed at 0.6. Changing the values of these parameters slightly will have no effect on the qualitative nature of the results. To facilitate comparison with the finite-element results, the coolant inlet temperature was assumed to be independent of time.

Figures 1.2.4,, 1.2.5 and 1.2.6 show the overall frozen fraction $\bar{F}$ plotted against the nondimensional time variable τ . Each of these figures shows results for two different Biot numbers. The rates of increase of the flow rate of the coolant were arranged such that the flow doubled at roughly the same value of F_0 for either of the Biot numbers in each figure. For reference, curves for constant coolant flow are also shown. These correspond to a constant coolant flow rate equal to the initial flow rate of the other cases. Plots of effectiveness are shown for

the same cases in Figures 1.2.7, 1.2.8 and 1.2.9.

Inspection of the figures shows that the analytical model developed in this chapter yields results that are sufficiently accurate for simulation studies. The largest errors occur when the Biot number is large, and the error in the predicted freezing rate is larger than that in the predicted effectiveness. Even the largest errors are quite moderate and tolerable.

The results we have just seen are for turbulent flow. Similar results were obtained for the laminar case, as well. The agreement between the finite-element results and the predictions of the model were so close that no differences could be shown on the figures. Therefore, we do not show these figures in the report.

From the results we see that raising the coolant flow rate does improve the heat recovery rate, as expected and desired. If the initial value of the Biot number is small, the increase in performance is substantial. With large Biot numbers, on the other hand, the thermal resistance of the solid dominates and the improvement in the performance is marginal. We thus infer a rule of thumb for the designer: make the initial Biot number small, say, less than unity, and increase the flow rate moderately as time progresses, so as to achieve improved performance without having to supply excessive pumping power for the maintaining coolant flow.

Although the rate of heat recovery can be increased by raising the coolant flow rate, there is a price to pay for this, since the effectiveness falls with increasing coolant flow. This implies that the irreversibility associated with the heat extrac-

tion increases as the coolant flow is raised.

The effects of sudden jumps in the coolant flow are illustrated in Figures I.2.10 and 11. We immediately that the error in the prediction of the model is largest immediately after the jump. We stress that equations (I.2.3) and (I.2.4) were not used in calculating the results in these figures. The value of $\epsilon + v$ was calculated using these equations, and was found to agree exactly with finite-element results.

CONCLUSIONS

A model has been developed for treating various cases of variable coolant flow. The performance of the storage unit can be predicted using the constant flow formulae of the previous chapter provided that equation (I.2.5) is used to calculate E . The tools necessary for the development of the simulation program are ready.

Fig. 1: Sketches of (a) problem and (b) finite-element mesh.

FROZEN FRACTION, $\bar{F} = V_{\text{SOLID}} / \pi R^2 L$

Fig. 3: Plot showing smallness of influence of Biot number and amount frozen on shape of solidified region.

Fig. 4: Overall frozen fraction as function of nondimensional time, moderate linear increase in coolant flow.

Fig. 5: Overall frozen fraction as function of nondimensional time, large linear increase in coolant flow.

Fig. 6: Overall frozen fraction as function of nondimensional time, moderate quadratic increase in coolant flow.

Fig. 7: Effectiveness as function of nondimensional time, moderate linear increase in coolant flow.

Fig. 8: Effectiveness as function of nondimensional time, large linear increase in coolant flow.

Fig. 9: Effectiveness as function of nondimensional time, moderate quadratic increase in coolant flow.

$L/R = 100$
 $NTU_o = 0.6$
 $Bi_o = 4$
 $Bi \sim \dot{m}^{0.8}$

DIMENSIONLESS TIME, τ

PART II

SIMULATION PROGRAM

PURPOSE: Program " STORAGE " computes the heat transfer from a freezing material to an immersed tube containing a coolant having arbitrary variations of flowrate and inlet temperature.

USAGE : CALL STORE (t1, t2, time1, time2, nstep, tcase, mcase, b2, print, njump, q, q1, q2, ihead, chead, fhead, flag, jump, jtime)

ARGUMENTS :

(i) The following variables must be supplied by the calling program:

t1 - (input/real) initial inlet temperature of the cycle.

t2 - (input/real) final inlet temperature of the cycle.

time1 - (input/real) the time corresponding to the initial temperature(t1). (see note (1))

time2 - (input/real) the time corresponding to the final temperature(t2).

nstep - (input/integer) the number of calculation step between time1 and time2.(nstep < 100)

tcase - (input/integer) inlet temperature type number.(tcase < 4)

=1: linear change in inlet temperature between t1 and t2. $t = t_0(1+a \times \text{time})$

=2: quadratic change in inlet temperature between t1 and t2. $t = t_0(1+a \times \text{time} \times \text{time})$

=3: constant inlet temperature between t1 and t2. $t = t_0$

where

t_0 = inlet temperature at time1

a = constant value.

mcase - (input/integer) mass flow rate variation type number between time1 and time2. (see note(5))

=1: linear increase in mass flow rate between time1 and time2, $m = m_0(1+b \times \text{time})$

=2: quadratic increase in mass flow rate between time1 and time2. $m = m_0(1 + b \times \text{time} \times \text{time})$

=3: constant mass flow rate between time1 and time2. $m = m_0$

=4: mass flow rate variation type same as in previous calculation cycle. NOTE: mcase=4 cannot be specified for the first cycle.(see note(5))

Here,

m_0 = mass flowrate at time1

b,b2 =constant values.

b2 - (input/real) the constant b for linear or quadratic increase mass flow rate.

NOTE: supply dummy value for b2, if mcase = 4.(see note(5))

print - (input/integer) printout code number.

1 = print all the properties and the results of this calculation cycle. (see note (4))

0 = suppress all printout of data and results for this calculation cycle.

njump - (input/integer) total number of sudden jumps of mass flow rate between time1 and time2. (njump < (or =) 10 or njump > (or =) 0)

For example:

njump=0 : no sudden jump in mass flow rate between time1 and time2.

njump=3 : three sudden jumps in mass flow rate between time1 and time2.

The following two variable arrays must be specified if njump > 0

jump(i) - (input/real array) the ratio of mass flow rate before jump i to mass flow rate after jump. (see note(2))

jtime - (input/real array) the time corresponding to jump i. (see notes(2,3))

flag - (optional input/integer) (output/integer) properties code. If the properties of freezing and coolant materials are constant for all calculation cycles. ignore this input. If flag is set to 0 for the specific calculation cycle, the user will be asked to input new values for the properties of storage and coolant materials through subprograms input1 and input2.

(ii) The following variables are output from program STORAGE.

q - (output/real array) array containing all the properties and results from previous calculation cycle: pf, kf, cp, uf, lheat, p, k, c, Tsat, R, xmin, xmax, dx, v, mass, touo, rmax, b2, Fo, time2.

q1 - (output/integer array) array containing all the code numbers and type numbers from previous calculation cycle: sys, mcase, ptype, iimax, cycle, flag2.

q2 - (output/real array) array containing all fbar(iimax,j) values from previous cycle.

ihed - (output/character) problem title from previous cycle. (max. 80 character)

chead - (output/character) identity of coolant material
from previous cycle. (max. 80 character)
fhead - (output/character) identity of freezing
material from previous cycle. (max. 80 character)

DESCRIPTION OF THE PROGRAM:

Program "STORE" computes the heat transfer from a freezing material to an immersed tube containing a coolant having an arbitrarily specified variation of flowrate and inlet temperature in a specified interval of time. Before calling STORE the user must collect and have the following information:

- (1) freezing material properties
- (2) coolant material properties
- (3) tube geometry
- (4) allowable maximum radius of frozen solid ($1/2$ tube pitch)
- (5) variation of inlet temperature with time
- (6) variation of mass flowrate with time

Next, the user should divide the range of time into several small time intervals, such that the variation of the inlet temperature and mass flowrate within each time interval can be represented well by the available types of variation in the program. each time interval is defined as one calculation cycle in the program. After selecting the time interval, the user can now execute the program by inputting the times and the corresponding inlet temperatures and flowrates.

In the first time interval (first cycle), the user follows the instructions of subprograms INPUT1 and INPUT, entering the coolant material and freezing material properties. in the succeeding cycles, the program automatically saves all the property data from the previous cycle. if the user wishes to change material properties in a cycle, he can set (flag=0) at the beginning of the cycle. this will result in calls to the INPUT1 and INPUT2 subroutines.

Two systems of physical units are available in the program: si-unit system and english-unit system. the user must keep the property data units consistent with the input data from the calling program. the output data units are consistent with the input data units.

The heat transfer results and material properties data can be printed at the end of each calculation cycle by setting print = 1. the user can also suppress all the output for a specific cycle

by setting print = 0. For detailed information on output format, see note(4).

The variables that must be provided for each cycle by the calling program are t1;t2,time1,time2,mcase,and tcase. If mcase is not equal to 4, the user must also input b2. If njump > 0, the user must also input print, nnjump, jump and jtime. If flag = 0, the user must also input new material properties.

In general, the program " STORAGE " can perform two different types of analyses. The first type is that for constant inlet temperature and mass flowrate without sudden jumps. The other is for the variable inlet temperature and mass flowrate case with or without sudden jumps. It is possible to consider the first type as a special case of the second, at the price of adding unnecessary calculations. Many different combinations of inlet temperature and mass flowrate types are possible. The user must select the most likely types of inlet temperature and mass flowrate for an economical analysis.

For more detailed information, see these references:

- (1) N. Shamsundar, "Formulae for freezing outside a circular tube with axial variation of coolant temperature" International J. of Heat and Mass Transfer, vol. 24, 1981, pp. 1345-57.
- (2) E. Stein, N. Shamsundar and E. Roosz, "Heat transfer from a freezing substance to an immersed tube containing a coolant having arbitrary variations of flowrate and inlet temperature", ASME paper 83-HT-99.

PROGRAMMING NOTES:

- (1) - time1 must be input as zero at the beginning of the calculations (i.e. first cycle) time1 must be same as time2 of previous calculation cycle.(i.e. no discontinuities in time are allowed)
- (2) - Sudden jumps in mass flow rate are treated as an independent feature, and they can be added to all types of continuous flowrate variations. the jump ratio ($\text{jump} = m^+ / m^-$) and its corresponding time, jtime(sec) must follow in the proper order, starting from the first jump time and its jump ratio to the last jump time and its jump ratio. the 1st jump time is less than 2nd jump time, and so on.
- (3) - An example of the jump and jtime arrays are as follows:
jump(1),jump(2),.....jump(njump)
jtime(1),jtime(2),.....jtime(njump),
Here, $\text{time2} \geq \text{jtime(njump)}$,

```

jtime(njump) >.....>
jtime(2) > jtime(1)
jtime(1) always > time1

```

In addition, the specified first jump time is not allowed to be the same as time1. if $jtime(1) = time1$, then this specified jump time, $jtime(1)$, should be specified in the previous calculation cycle, not in the current cycle.

- (4) - The printouts have the following four types :
- ptype=1 : prints $x, tf, tw, r\%$ and effectiveness
 - ptype=2 : prints $x/r, \theta, (r\%-r)/r, f$, and $\bar{f}$
 - ptype=3 : prints $x, tf, tw, r\%$, effectiveness, x/r , $\theta, r\%-r/r, f$ and $\bar{f}$
 - ptype=4 : optional output
- Two additional inputs are required for ptype=4; they are:

(a) total number of output parameters ($0 < iimax < = 10$)

(b) output code numbers, each code number corresponding to one parameter

Here,

```

code(1) = x, axial coordinate
code(2) = tf, coolant temperature at any x
code(3) = tw, wall temperature of tube at any x
code(4) = r%, outer radius of frozen layer at any x
code(5) = eff, effectiveness
code(6) = x/r, dimensionless axial coordinate
code(7) = theta, dimensionless coolant temp. at any x
code(8) = r%/r, dimensionless outer radius of frozen layer
code(9) = f, (area of solid/area of tube) at any x
code(10)= fbar, volume of solid/volume of tube

```

- (5) - At the end of each calculation cycle, the program automatically saves the values of the variables m_o , b and $mass(iimax)$, where

m_o (or $b3, b1$) is the mass flowrate at time $time1$, b (or $b2$) is the constant value implies a gain of the flowrate, and $mass(iimax)$ is the mass flowrate at the end of the cycle, $time2$.

Before going to the next cycle, the program automatically sets the new m_o (or $b3, b1$) = $mass(iimax)$. That is, the mass

flowrate at the end of the current cycle now becomes the mass flowrate at the beginning of next cycle, time1.

For constant flowrate, if the user specifies mcase=4 in the next calculation cycle, this indicates that the mass flowrate variation is the same type as in the previous cycle. the program automatically sets the new mo = mass(iimax). **

WARNING: For the linear or quadratic flowrate variation cases, if the user specifies mcase=4 in the next calculation cycle, this indicates that the mass flowrate variation type is the same as in the previous cycle. The program automatically calculates the new b (or b2) value, and the user can ignore the input of a new b2. However, these two cases have not been checked for correct working. The user should check the results carefully. The user must specify a new b2 value for the next calculation cycle, if mcase = 1, 2 or 3.

The following are the descriptions of each subprogram and its corresponding variables in the program "STORE".

CHECK:

The subroutine "CHECK" checks the variables from the calling program only. If an error exists, an error message will be printed and the program terminated.

INPUT1 and INPUT2:

Subroutine "INPUT1" reads all the properties of the coolant and freezing materials for the constant inlet temperature and mass flow rate case.

Subroutine "INPUT2" reads all the properties of the coolant and freezing materials for the variable inlet temperature and mass flow rate case.

Here, pf : density of coolant.
kf : conductivity of coolant.
cp : specific heat of coolant.
v : mean velocity of coolant.
uf : dynamic viscosity of coolant.
lheat : latent heat of fusion.
p : density of freezing material.
k : conductivity of freezing material.
c : specific heat of freezing material.
Tsat : freezing temperature.
R : radius of tube.

rmax :maximum allowable radius for freezing material.
 xmin :minimum x.
 xmax :maximum x.
 dx :calculation increment in x.
 ptype :printout type number.(see subroutine "printout")
 KK :printout parameters.(see subroutine "printout")
 tcase :inlet temperature type number.
 mcase :mass flow rate type number.
 a1,a2 :two constants for variable inlet temperature case.
 b1(or b3),b2 :two constants for variable mass flow rate case.
 t1 :initial time for calculation step(sec)
 t2 :final time for calculation step(sec)

TOCONST and TOMASS

Subroutine " TOCONST " computes the dimensionless time and the Biot number for constant inlet temperature and mass flow rate case.

Subroutine " TOMASS " computes the dimensionless time and corresponding Biot number, Stefan number and Stanton number for the variable inlet temperature and mass flow rate case.

Here,

touo :dimensionless time variable
 Bi1,Bi :Biot number.
 St,St1 :Stanton number.
 Ste,Ste1 :Stefan number.
 touo2 :dimensionless time variable (previous cycle) .
 h :convective coefficient inside tube.
 Tol,To :constant inlet temperature.
 a1,a2 :two constants for variable inlet temperature case.
 b1(b3) ,b2 :two constants for variable mass flow rate case.
 mass :mass flow rate of coolant.
 massj :mass flow rate of coolant after sudden jump.
 v1,v :velocity of coolant for constant flow rate case.
 njump :total number of jumps in mass flow rate.
 Re1,Re :Reynolds number of tube flow.

INLET and INLET2

Subroutine " INLET " determines the ratio for area of solid to area of tube, the thickness of freezing materials, and the radius of freezing material at the inlet ($x=0$) for the constant inlet temperature and mass flow rate case.

Subroutine " INLET2 " determines the same parameters for variable inlet temperature and mass flow rate case.

Here,

x :axial coordinate.
Fo :area of solid/area of tube at inlet (x=0)
tko :thickness of freezing material at the inlet(x=0).
r10 :radius of freezing material at the inlet(x=0).
Fao :cross-sectional area of freezing solid at inlet(x=0).

SOLID and SOLID2

Subroutine " SOLID " determines the ratio for the area of solid to the area of tube, the thickness of freezing materials, and the radius of freezing material along the tube for constant inlet temperature and mass flow rate case.

Subroutine " SOLID2 " determines the same parameters for variable inlet temperature and mass flow rate case.

Here :

f :area of solid/area of tube at any x.
tk :thickness of freezing material at any x.
r1 :radius of freezing material at any x.
fa :cross-sectional area of freezing solid at any x.

TEMP and TEMP2

Subroutine " TEMP " computes the effectiveness, coolant temperature, and tube wall temperature for constant inlet temperature and mass flow rate.

Subroutine " TEMP2 " computes the same parameters for variable inlet temperature and mass flow rates.

Here,

eff :effectiveness.
Tf :coolant temperature.
Tw :tube wall temperature.

SOLIDV and SOLIDV2

Subroutine " SOLIDV " computes the ratio for the volume of solid (from 0 to x) to the volume of the tube (from 0 to x), and total volume of solid from 0 to x for the constant inlet temperature and mass flow rate case.

Subroutine " SOLID2 " computes the same parameters for the variable inlet temperature and mass flow rate case.

Here,

Fbar :volume of solid from 0 to x/volume of tube between 0 to x.

Fvx :volume of solid from 0 to x.

MJUMP

Subroutine " MJUMP " determines

(a) the coolant temperature, tube wall temperature,
(b) the effectiveness after a sudden jump in flowrate, and
(c) the frozen fraction, fbar, before and after sudden jump in flowrate for variable (or constant) inlet temperature and mass flowrate conditions.

Here,

Tf :coolant temperature

Tfj :tf after sudden jump in flowrate.

mass :mass flow rate.

mass :mass flow rate after sudden jump.

Fbar :frozen fraction before sudden jump.

Fbarj :fbar after sudden jump in flowrate.

F : (area of solid/area of tube) at any x

Fo :f at inlet, x=0

Fj :f after sudden jump in flowrate.

Foj :fo after sudden jump in flowrate.

jump :ratio of mass flowrate after jump to the mass flowrate before jump, $\text{jump} = m^+/m^-$

jtime :time corresponding to the jump.

DTOUT1 and DTOUT2

Subroutine " DTOUT1 " prints all the properties of coolant and freezing materials and range of calculation for constant inlet temperature and mass flowrate case.

Subroutine " DTOUT2 " prints the same parameters for the variable inlet temperature and mass flowrate case.

PROUT

Subroutine " PRINTOUT " prints the heat transfer results for freezing outside a circular tube containing a coolant having arbitrary variations of flowrate and inlet temperature.

The printouts are of the following four types :

ptype=1 : prints x, Tf, Tw, rX and eff

ptype=2 : prints x/r, theta, $(rX-R)/R$, F, and Fbar

ptype=3 : prints x, Tf, Tw, rX, eff, x/R, theta, $(rX-R)/R$, F, Fbar

ptype=4 : optional output (see note(4)).

PART III

OTHER MODELS AND PROGRAMS

The following chapters are in preparation and are not included here:

Chapter III.1 Analytical Model for Heat Storage

Chapter III.2 Analytical Model for Heat Recovery
from Finned Tubes

III.3 DESIGN METHOD FOR LATENT HEAT STORAGE UNITS

A method for designing latent heat storage units and a computer program to do a quick sizing of a latent heat storage unit for a specified storage capacity was developed in the summer of 1984 for the Johnson Space Center of NASA. Because of the relevance of this program to this report, we describe a modified version of that program and the underlying design method here. The storage unit in the NASA project was required to provide sufficient hot gas to drive a propulsion unit for the space station. We have modified the computer program by taking out the calculations of propulsive thrust.

The sizing calculations that we are about to describe are rather simplified, and we only seek to obtain a reasonable shell and tube heat exchanger configuration with conservative size and performance. A real design would need optimization, and this will be done at a later date.

Design Method:

There is no standard design procedure available. After a few false starts, we developed the strategy described here as a simplified approach to the design. A computer program has been written to carry out the design calculations conveniently. Both the design method and the computer program consider only the case of constant coolant flow rate, and have no provision for finned tubes. These limitations can be removed in future revisions, but the program is still useful in obtaining preliminary design parameters. The resulting design can be run through the simulation program to ascertain if it will have satisfactory performance as part of a solar thermal system with variable coolant flow

and inlet temperatures.

The underlying analytical model for the design method was described in Chapter I.1. The following description draws upon the results given there, but can be used as a stand-alone recipe. Reference should be made to Chapter I.1 for the definitions of the nondimensional parameters used here.

The principal design parameters for latent heat storage units, just as for conventional heat exchangers, are the effectiveness and the Number of Transfer Units (NTU). We want the heat exchanger to have high effectiveness, and to show as little variation of effectiveness as possible as the phase change material continues to solidify.

The definitions of effectiveness and NTU that we shall adopt for latent heat storage units are slightly different than the ones commonly used for fluid-fluid heat exchangers. In defining NTU, we use only the fluid-side heat transfer coefficient rather than the usual overall heat transfer coefficient -- the latter is not only unknown, but varies with time in a complicated way. The definition is

$$NTU = 2\pi h R_n L / \dot{m} c, \quad (III.3.1)$$

where

R_i = inside radius of tubes of heat exchanger,
 h_i = heat transfer coefficient of fluid flowing in tube,
 L = length of tube,
 n = number of tubes in heat exchanger,
 $\dot{m}$ = mass flow rate of fluid in tube,
 c = specific heat of fluid at constant pressure.

Similarly, we have to modify the definition of effectiveness, which is defined for conventional fluid-fluid heat exchangers.

ngers as the ratio of the actual heat transfer to the heat transfer when the surface area of the heat exchanger approaches infinity. For a latent heat storage unit, the latter quantity is time-dependent and is not easy to calculate. Therefore, it is improper for use as a reference heat transfer rate, and we replace it with the heat transfer rate corresponding to not only infinite area but also to infinite conductance in the phase change material. This heat transfer rate is clearly that which results in the coolant fluid being heated to the melting temperature of the phase change material. Thus, we have the definition

$$\epsilon = (T_{fo} - T_0) / (T_{sat} - T_0) \quad (III.3.2)$$

for the effectiveness, where

T_f = outlet temperature of coolant from heat storage unit,
 T_0 = inlet temperature of coolant entering storage unit,
 T_{sat} = melting temperature of phase change material.

The first step in the design is to specify the temperature of the heat source used to charge up the latent heat storage unit. Assuming that a temperature difference of about 20 degrees F to charge the storage unit, we find that the melting point of the phase change material (PCM) should be about 20 degrees F below the charging temperature. This enables the selection of a set of heat storage materials with the right melting point.

The second step is to estimate the heat exchanger effectiveness at the start of the discharge period, when the storage material is all liquid. At this stage, the outlet temperature of the coolant will probably be higher than at any other time since, at later instants of time, there will be a build up

of solid which impedes heat transfer. Thus, we specify a value for T_{max} , the maximum desired outlet temperature of the coolant. With this as T_f , we calculate the maximum effectiveness, ϵ_{max} , using the fact that the storage material has no thermal resistance at this instant of time. The effectiveness is given by

$$\epsilon_{max} = 1 - \exp(-NTU) \quad (III.3.3)$$

From this equation we find the value of NTU, and check to see if it lies within the range of reasonable values, that is, $0.5 < NTU < 5$. If not, some of the data will have to be revised and the above steps repeated.

We similarly specify a minimum outlet fluid temperature, T_{min} , and find the corresponding effectiveness, ϵ_{min} .

The next important parameter is the Biot number, Bi , which is

$$Bi = hR_o/k_{pcm} \quad (III.3.4)$$

To estimate Bi , we have to first find the Nusselt number, $Nu = hR_i/k_{coolant}$, for convection between the flowing coolant and the tube. Since we do not know the tube size and the number of tubes at this point, we have to make a good guess. For gaseous coolants, we are likely to encounter turbulent flow. For liquid coolants, the higher viscosity will probably lead to laminar flow. If we expect that the coolant will experience laminar flow, we have $Nu \approx 2$ for fully developed heat transfer. Furthermore, it is characteristic of laminar flow that this value is independent of the flowrate. For turbulent flow, on the other hand, we must first estimate the Reynolds number of the flow in a

single tube, and use a correlation for fully-developed heat transfer such as the Dittus-Boelter correlation to estimate the Nusselt number. After obtaining Nu in this way, we estimate

$$Bi = Nu(k_{\text{fluid}}/k_{\text{pcm}})(R_i/R_o) \quad (\text{III.3.5})$$

This estimate of Bi may be revised slightly at a later stage, when we have found values for R_i and R_o . For the case of turbulent flow, it may be useful to treat the Reynolds number as an additional adjustable parameter.

Next, we find F_0 , the fraction of phase change material that has solidified at the tube inlet at the end of the discharge period. This fraction is defined as the ratio of the area of the frozen region to the area of the outside surface of the tube. To calculate this frozen fraction, we have to solve the transcendental equation (I.1.8)

$$NTU = -\ln(1 - \epsilon_{\min}) + (Bi/2)(G_1(F_0) - G_1[F_0(1 - \epsilon_{\min})]) \quad (\text{III.3.6})$$

This equation is easily solved by the Newton-Raphson method (see Ref.[1]), but it is necessary to start off with a good approximation to the solution. A reasonable approximation is obtained by observing that $G_1(F)$ is approximately equal to p for small F (see equation (I.1.13)). This leads to the approximate result

$$F_0 \approx (\gamma - 1)/[1 - \gamma(1 - \epsilon_{\min})], \quad (\text{III.3.7})$$

where

$$y = \exp[NTU + \ln(1 - \epsilon_{\min})](2/Bi) \quad (III.3.8)$$

If the estimated F_0 is not in the range of 0 to 10, we reject this estimate and generate a new value based on an approximation of G_1 for large F , that is, $G_1 = p^2/2$ (see equation (I.1.12)). This leads to the estimate

$$F_0 \approx [y/(1 - \epsilon_{\min})]^{1/2}, \quad (III.3.9)$$

where

$$y = \exp[(4/Bi)(-NTU/\ln(1 - \epsilon_{\min}) - 1)] \quad (III.3.10)$$

From the estimated value of F_0 , we calculate the nondimensional time variable

$$\tau_0 = F_0/(2Bi) + (1/4)G_0(F_0) \quad (III.3.11)$$

Then, from the definition of the variable τ , we calculate R_0 as

$$R_0 = [k_{pcm}(T_{\text{sat}} - T_0)t_{\text{discharge}}/(\tau\rho\lambda)]^{1/2} \quad (III.3.12)$$

From the outside radius, we may compute the inside radius such that the hoop stress in the wall of the tube is an acceptable value, or, since the wall thickness obtained in this way usually falls below the usual commercial tube thicknesses, simply pick a reasonable value. The design program uses the latter approach.

We then find the tube pitch from the formula

$$\text{pitch} = 2R_0(1 + F_0)^{1/2} \quad (III.3.13)$$

We now have the information to find the length L and the number n of the tubes. We calculate the product nL from

$$nL = (NTU mc) / (2\pi k_{pcm} Bi) \quad (III.3.14)$$

To ensure that the flow in the tubes is laminar, we need

$$Re = 2mL / (\pi R_i nL) < 2300. \quad (III.3.15)$$

This condition gives us the maximum length of the tubes, if we have laminar flow. Then, we can choose various lengths below this value and calculate all the remaining parameters of the design. For turbulent flow, we simply choose the length so as to obtain the value of Re assumed earlier.

For each L , we find the corresponding $n = (nL)/L$. Then, we calculate the tank diameter D from

$$D = \text{pitch} (2\sqrt{3}n/\pi)^{1/2} \quad (III.3.16)$$

Finally, we can compute items such as the volume of the storage unit, the mass of phase change material in it, the amount of metal used in the tubes and in the outer shell, etc. This completes the design of the latent heat storage unit.

One final calculation which can be made at our option is to find the performance of the storage unit we have just designed. To do this, we select various instants of time during the discharge period, and calculate the corresponding values of the time variable by using its definition. Then, we solve equation(3) by the Newton-Raphson method for F_0 . Using this F_0 , we then solve equation(2) for the instantaneous value of the effectiveness. Then, we calculate the instantaneous fluid outlet temperature from the definition of the effectiveness, equation(1).

A more thorough test of the performance of the design can be made by using the storage system simulation package instead of the calculation just described.

Computer Program for Design:

The method just described is implemented in Pascal in a program consisting of about 200 lines of code. Because the amount of input data is modest, the code is completely interactive. Several designs can be calculated in a single run, and a summary of the design parameters for each case can be printed out if desired.

A listing of the program and a sample run are given in Appendix A.

REFERENCE:

1. O.J. Dahlquist and A. Bjork, Numerical Methods, Translated by N. Anderson, Prentice-Hall, Inc., 1974.

STORET.FAS

```

-----
See Chapter III.3 of the Final Report from the University of Houston
Mechanical Engineering Department for an explanation of the design
method. The following program implements that method.
    It runs on any machine that runs TurboPascal or Microsoft Pascal.
Slight modifications may be necessary to use other Pascal versions.
-----

```

```

{ $STANDARD }

```

```

program store(input,output);

```

```

label exit;

```

```

const pi=3.14159265;

```

```

var tm,tch,ti,tmax,tmin,epsmax,epsmin,ntu,bir,x,y,f0,f1,p0,p,p20,p2,
    g1,g10,f0n,er,g0,g,tau,rc,ri,kp,kg,mdot,cp,nl,l,ls,vis,lmax,
    pitch,tankdia,tankvol,gasvol,tubevol,pcmvol,heatcap,tdisch,
    eps,epsn,tstep,tau1,tout,thrust,re,pr: real;
    i,n,iter: integer;
    ch,pflag: char;
    print.laminar: boolean;
    resfil: text;

```

```

function pow(x,y:real):real;

```

```

{-----x to the power y-----}

```

```

begin

```

```

    pow:=exp(y*ln(x))

```

```

end;

```

```

function fn_g1(f:real):real;

```

```

{-----integral of ln(1 + F)/F from F=0 to F-----}

```

```

{-----see equations (III.3. to III.3. -----)}

```

```

var p,p2:real;

```

```

begin

```

```

    if f<10.0 then

```

```

        begin p:=ln(1+f); p2:=p*p; fn_g1:=(1+17*p2/450)/(1+p2/100)*p+p2/4 end

```

```

    else begin

```

```

        p:=ln(f); fn_g1:=pi*pi/6+p*p/2-fn_g1(1.0/f)

```

```

    end

```

```

end;

```

```

begin{ Main Program}

```

```

repeat begin

```

```

    writeln;

```

```

    writeln('Design Program for Latent Heat Storage Unit');

```

```

    writeln('-----');

```

```

    writeln; writeln('Any numbers input may or may not have a decimal point');

```

```

    writeln('No number can begin or end with a decimal point--pad with a zero');

```

```

    writeln('Separate numbers with at least one space');

```

```

    writeln('To questions that need "yes" or "no" as answers, enter "y" or "n"');

```

```

    writeln('To abort an incomplete run, enter ^C in CP/M or in MS Pascal;');

```

```

    writeln('This does not work in the MS/DOS version. In this case, enter any');

```

```

writeln('invalid character when a number is asked for');
writeln:
  repeat
    begin
      {-----input parameters and properties-----}

      writeln('Input (i) Melting Point of PCM, (ii) Charging Temperature,');
      writeln(' (iii) Cold Gas Inlet Temperature, (iv) Max. disch. temp.,');
      write(' (v) Min. disch. temp. all in deg. F : ');
      readln(tm,tch,ti,tmax,tmin);
      if (tch-tm<10.0) then writeln('Charging Temp. probably not high enough');
      if (tmax>=tm) then writeln('Discharge temp. cannot be this high.')
    end
  until (tmax<tm);
  repeat
    begin
      {-----maximum effectiveness and NTU-----}
      {-----see equations III.3. and III.3. ----}
      epsmax:=(tmax-ti)/(tm-ti); ntu:=-ln(1.0-epsmax);
      writeln('NTU = ',ntu:6:2);
      if ((ntu<0.5) or (ntu>5.0)) then
        begin write('NTU not in proper range; change max. disch. temp : ');
              readln(tmax)
        end
      end
    until ((ntu>0.5) and (ntu<5.0));

    {-----input remaining parameters-----}

    writeln('Input thermal conductivity of ');
    write('(i) gas and (ii) solid PCM in Btu/hr.ft.deg.F : ');
    readln(kg,kp);
    write('Input Reynolds number estimate - 0 for laminar flow : ');
    readln(re);
    if re<=2300.0 then begin laminar:=true; bir:=2.00*kg/kp end
    else
      begin
        laminar:=false;
        write('Prandtl number of coolant : ');

        {-----Calculate Biot Number from Dittus-Boelter Correlation-----}

        readln(pr); bir:=0.0115*pow(re,0.9)*pow(pr,0.4)*kg/kp
      end;

      {-----Calculate minimum effectiveness allowed.-----}
      {-----Calculate estimate of FO-----}

      epsmin:=(tmin-ti)/(tm-ti); p:=-ln(1-epsmin);
      x:=(ntu-p)*2.0/bir; y:=exp(x); f0:=(y-1.0)/(1-y*(1-epsmin));

```

STORET.PAS

```

if (f0>10.0) or (f0<0.0) then
  begin y:=exp(2*x/p); f0:=sqrt(y/(1-epsmin)) end;
iter:=0;

{-----Solve equation III.3. for F0-----}

writeln('Solving for F0 --'); writeln('Iteration    F0');
repeat
  begin
    f1:=(1.0-epsmin)*f0; p0:=ln(1.0+f0); p20:=p0*p0;
    g10:=fn_g1(f0);
    p:=ln(1.0+f1); p2:=p*p;
    g1:=fn_g1(f1);
    x:=-ln(1.0-epsmin)+bir/2.0*(g10-g1)-ntu;
    y:=bir/2.0*(p0/f0-(1.0-epsmin)*p/(1.0+f1));
    f0n:=f0-x/y; er:=abs(f0/f0n-1.0); f0:=f0n; iter:=iter+1;
    (writeln(iter:3,f0:13:3));
    if f0>1000 then
      begin writeln('F0 > 1000 !',chr(7));
        writeln('Check input data; Min. outlet temp. may be too low');
        goto exit
      end
    end
  until (er< 1.0e-5) or (iter>80);
if iter>80 then begin writeln('Iterations diverged in finding F0'); goto exit end;

{-----Calculate tau_0 from equation III.3.11,-----
----- and outer radius of tubes from III.3.12-----}

g0:=(1.0+f0)*p0-f0; g:=(1.0+f1)*p-f1;
tau:=f0/(2.0*bir)+g0/4.0;
write('Bi-Ro = ',bir:6:2,' F0 = ',f0:7:3,' F1 = ',f1:7:3);
writeln(' Tau = ',tau:7:2);
write('Min. effectiveness = ',epsmin:6:3);
writeln(' Max. effectiveness = ',epsmax:6:3);
writeln('Input (i) discharge period in minutes, ');
write('      (ii) Latent heat of PCM in Btu/ft3 : ');
readln(tdisch,heatcap);
ro:=12.0*sqrt(tdisch/60.0*kp*(tm-ti)/(heatcap*tau));
write('Tube outside radius = ',ro:8:4,' inches');

{-----Calculate tube pitch from equation III.3.13-----}

pitch:=2.0*ro*sqrt(1.0+f0); writeln(' Pitch = ',pitch:8:2,' inches');

{-----Input remaining parameters not yet read-----}

write('Input inside radius : '); readln(ri);
write('Input (i) mass flow of gas in lb/hr, ');
write(' (ii) its sp.ht. in Btu/lb.deg. F : ');
readln(mdot,cp); nl:=(ntu/bir)*mdot*cp/(2.0*pi*kp);

```

STORET.PAS

```

write('Input viscosity of gas in lbm/hr.ft : '); readln(vis);

{-----Calculate length of tubes-----}

lmax:=2300.0*pi*ri/12.0*vis*n1/(2.0*mdot);
if laminar then
begin
  write('Maximum heat exchanger length = ',lmax:6:2);
  writeln(' feet to retain laminar flow');
  write('Input desired length in feet : '); readln(l)
end else l:=lmax;

{-----make hardcopy of results if so requested-----}

  write('Do you want printed output? (y/n)');
  readln(pflag); print:=(pflag='y') or (pflag='Y');

if print then
begin
  assign(resfil,'prn:'); rewrite(resfil);
  write('TURN PRINTER ON.--ENTER "y" WHEN READY: ');
  repeat readln(ch) until (ch='y') or (ch='Y');
  writeln(resfil); writeln(resfil); writeln(resfil);
  writeln(resfil,'-----');
  writeln(resfil); writeln(resfil,'DESIGN OF A LATENT HEAT STORAGE UNIT');
  writeln(resfil); writeln(resfil);
  write(resfil,'T_m =',tm:4:0,' T_ch =',tch:4:0,' T_i =',ti:4:0);
  writeln(resfil,' T_out_max =',tmax:4:0,' T_out_min =',tmin:4:0,' deg R');
  writeln(resfil,'k_gas =',kg:7:3,' k_s =',kp:7:3,' btu/hr_ft_F');
  writeln(resfil,'Max. effectiveness =',epsmax:6:3,' Min. eff. =',epsmin:6:3);
  writeln(resfil,'NTU =',ntu:6:2,' Bi_Ro =',bir:6:2,
    ' FO =',f0:7:3,' F1 =',f1:7:3,' tau =',tau:7:3);
  writeln(resfil,'t_disch =',tdisch:5:0,' min. Latent ht. =',heatcap:6:0,
    ' btu/ft_3');
  writeln(resfil,'Gas flow =',mdot:5:0,' lb/hr cp =',cp:6:2,' btu/lb-F visc =',
    vis:6:2,' lb/hr-ft');
end;

{-----Find dimensions and volumes of storage unit-----}

write('Tank dia(ft) Length(ft) n-tubes Vol(ft3)');
writeln(' Gas vol PCM vol Tube-wall vol Re'); writeln;
repeat
  begin n:=round(n1/l); ls:=1;
    tankdia:=sqrt(2.0*sqrt(3.0)/pi*n)*pitch/12.0;
    tankvol:=pi*tankdia/2.0*tankdia/2.0*l;
    gasvol:=pi*n*ri*ri*l/144.0;
    tubevol:=pi*(ro+ri)*n*l*(ro-ri)/144.0;
    pcmvol:=tankvol-tubevol-gasvol;
    if laminar then re:=2.0*mdot/(pi*ri/12.0*vis*n);
    write(tankdia:6:2,l:14:2,n:11,tankvol:10:2);
  
```

STORET.PAS

```

writeln(gasvol:10:3,pcmvol:10:2,tubevol:8:2,round(re):10);
if laminar then
begin
  writeln('Try another length? If so, just enter length in ft. ');
  write('If not, enter a dummy negative length.: '); readln(l)
end;
end
until (l<0.0) or (not laminar); if laminar then l:=1s;

{-----Make hardcopy of dimensions and volumes-----}

if print then
begin
  writeln(resfil); writeln(resfil,'Heat exchanger particulars:'); writeln(resfil);
  writeln(resfil,'  Length =',l:5:1,' ft  Dia =',tankdia:5:2,' ft  Vol =',
    tankvol:3:0,' ft_3');
  write(resfil,'  Tube dimensions:');
  writeln(resfil,'  Ri =',ri:6:2,'      Ro =',ro:6:2,'  pitch =',pitch:6:2,
    ' in. ');
  writeln(resfil,n:5,' tubes,      vol. of gas in them =',gasvol:6:1,' ft_3');
  writeln(resfil,'  Vol. of PCM =',pcmvol:6:0,' ft_3,  tube-wall-vol =',
    tubevol:6:2,' ft_3');
  writeln(resfil);
  writeln(resfil,'  Reynolds number in tubes =',re:6:0);
end;

{-----Calculate variations of outlet temperature and -----
-----effectiveness of the design just obtained-----}

writeln; writeln('At what time-intervals(min.) do you want the outlet');
write('temperature and the corresponding ideal thrust to be printed? ');
readln(tstep); n:=trunc(tdisch/tstep);
eps:=epsmax;
writeln; writeln('Time(min.)  Disch. Temp(deg F) Effectiveness'); writeln;
if print then
begin
  writeln(resfil); writeln(resfil,'Time(min.)  Disch. Temp(deg F) Effectiveness');
  writeln(resfil)
end;
for i:=0 to n do
begin
  tau1:=tau*tstep*i/tdisch;
  f0:=2.0*bir*tau1;
  if i > 0 then
  begin
    repeat
    begin
      p0:=ln(1.0+f0); f0n:=(f0-p0+4.0*tau1)/(2.0/bir+p0);
      er:=abs(f0/f0n-1.0); f0:=f0n
    end
    until (er < 1.0e-5);
  end
end

```

STORET.PAS

```

p20:=p0*p0;
g10:=fn_g1(f0); iter:=0;
repeat
  begin
    f1:=(1.0-eps)*f0; p:=ln(1.0+f1); p2:=p*p;
    g1:=fn_g1(f1);
    x:=-ln(1.0-eps)+bir/2.0*(g10-g1)-ntu;
    y:=(1.0+bir/2.0*p)/(1.0-eps);
    epsn:=eps-x/y; er:=abs(eps/epsn-1.0);
    eps:=epsn; iter:=iter+1;
  end
until (er < 1.0e-5) or (iter > 50);
end;
tout:=ti+eps*(tm-ti);
writeln(i*tstep:7:1,tout:13:1,eps:19:3);
if print then writeln(resfil,i*tstep:7:1,tout:13:1,eps:19:3)
end;
if print then begin writeln(resfil); writeln(resfil); writeln(resfil) end;

{-----Repeat design for another case -----}

writeln: write('Do another case? '); readln(ch);
if (ch <> 'y') and (ch <> 'Y') then goto exit
end until (false);
exit:
end.

```

SAMPLE PRINTOUT FROM PROGRAM STORET.PAS

----- DESIGN OF A LATENT HEAT STORAGE UNIT

T_m = 500 T_{ch} = 540 T_i = 100 T_{out_max} = 400 T_{out_min} = 200 deg R
k_{gas} = 0.020 k_s = 0.040 btu/hr_ft_F
Max. effectiveness = 0.750 Min. eff. = 0.250
NTU = 1.39 Bi_{Ro} = 3.80 FO = 7.483 F1 = 5.612 tau = 3.649
t_{disch} = 120 min. Latent ht. = 5000 btu/ft₃
Gas flow = 200 lb/hr cp = 2.50 btu/lb-F visc = 0.02 lb/hr-ft

Heat exchanger particulars:

Length = 10.5 ft Dia = 2.13 ft Vol = 37 ft₃
Tube dimensions: Ri = 0.48 Ro = 0.50 pitch = 2.93 in.
69 tubes, vol. of gas in them = 3.6 ft₃
Vol. of PCM = 33 ft₃, tube-wall-vol = 0.35 ft₃

Reynolds number in tubes = 4000

Time(min.) Disch. Temp(deg F) Effectiveness

0.0	400.0	0.750
10.0	285.0	0.462
20.0	255.3	0.388
30.0	240.0	0.350
40.0	230.1	0.325
50.0	223.1	0.308
60.0	217.7	0.294
70.0	213.4	0.283
80.0	209.8	0.275
90.0	206.8	0.267
100.0	204.3	0.2

Document Control Page	1. SERI Report No. SERI/TP-253-4280	2. NTIS Accession No. DE91002145	3. Recipient's Accession No.
4. Title and Subtitle Design and Simulation of Latent Heat Storage Units, Final Report			5. Publication Date April 1992
			6.
7. Author(s) N. Shamsundar, E. Stein, E. Rooz, E. Bascaran, T. C. Lee			8. Performing Organization Rept. No.
9. Performing Organization Name and Address University of Houston 4800 Calhoun Houston, Texas 77004			10. Project/Task/Work Unit No.
			11. Contract (C) or Grant (G) No. (C) XX-7-07028-1 (G)
12. Sponsoring Organization Name and Address Solar Energy Research Institute 1617 Cole Boulevard Golden, Colorado 80401-3393			13. Type of Report & Period Covered Technical Report
			14.
15. Supplementary Notes SERI Technical Monitor: R. Gerald Nix, (303)231-1757			
16. Abstract (Limit: 200 words) This report presents the results of two years of research and development on passive latent heat storage systems. Analytical models and a computer code for simulating performance were developed. The heat storage material studied was a eutectic mixture of inorganic salts that fill the space outside the tubes in a shell-and-tube heat exchanger. Predicting the heat transfer rates and the transient response involved the solution of a transient heat transfer problem with moving boundaries. Heat recovery models were tested by comparing results from other calculations. The heat storage model accounted for simultaneously existing conduction and convection zones, however, it was based on experimental data for melting on long tubes with little axial variation. A model was set up for calculating performance improvement resulting from the use of finned tubes instead of plain tubes. Fins were shown to improve storage system performance. A simple computer program for quick calculation of design parameters for heat storage is also described.			
17. Document Analysis a. Descriptors Photodegradation; high flux solar radiation; solar simulated radiation; photodegradation; photoirradiation b. Identifiers/Open-Ended Terms c. UC Categories 201			
18. Availability Statement National Technical Information Service U.S. Department of Commerce 5285 Port Royal Road Springfield, VA 22161			19. No. of Pages 65
			20. Price AO4

END

**DATE
FILMED**

6 / 24 / 92