

# DEWATERING FINE TAILS BY EVAPORATION: A MATHEMATICAL MODELLING APPROACH

**Dewatering fine tails by evaporation:  
A mathematical modelling approach**

Report

to

Syncrude Canada Ltd.  
Edmonton Research Centre  
Edmonton, Alberta

by

Xiaomei Li  
Alberta Environmental Centre

and

Yongsheng Feng  
Department of Renewable Resources  
University of Alberta

May, 1995

To purchase copies of this report for a nominal fee of \$10, contact the Alberta Environmental Centre, Communications and Office Support, Bag 4000, Vegreville, Alberta, T9C 1T4; phone (403) 632-8211; fax (403) 632-8379.

For copies of the fully operational computer program, please contact Xiaomei Li at the Alberta Environmental Centre, Bag 4000, Vegreville, Alberta, T9C 1T4; phone (403) 632-8213; fax (403) 632-8379.

This publication may be cited as:

Xiaomei Li, Yongsheng Feng 1995. Dewatering fine tails by evaporation: A mathematical modelling approach. Alberta Environmental Centre, Vegreville, AB. AECV95-R5. 42 pp.

ISBN 0-7732-1696-0

## TABLE OF CONTENTS

	PAGE
LIST OF TABLES .....	iv
LIST OF FIGURES .....	v
EXECUTIVE SUMMARY .....	vii
ACKNOWLEDGMENTS .....	xi
1. INTRODUCTION .....	1
2. OBJECTIVES .....	3
3. FORMULATION OF THE MODEL.....	3
3.1 The Modelling Approach.....	3
3.2 The Lagrangian Vs. Eulerian Coordinate Systems .....	4
3.3 Equations of Water Transport.....	5
3.4 Special Case: Saturated Condition.....	7
3.5 Equations of Heat Transport.....	8
3.6 Equations of Vapor Transport.....	9
3.7 Equations of Solid Phase Consolidation.....	10
3.8 Boundary Conditions .....	10
3.9 Numerical Solutions .....	11
4. EXPERIMENTAL VALIDATION .....	12
4.1 Materials and Methods.....	12
4.2 Results.....	14
4.3 Comparison of Experimental Data and the Predictions of the Model .....	14
5. EXAMPLES .....	16
6. CONCLUSIONS.....	17
7. RECOMMENDATIONS.....	18
REFERENCES .....	19
LIST OF SYMBOLS .....	21
APPENDIX I EVAP: The Numerical Analysis of Evaporation From Fine Tails User's Manual	
APPENDIX II The Computer Program	

LIST OF TABLES

PAGE

Table 5.0.1 The atmospheric conditions for a typical day in June (Fort McMurray) . . . . . 22

Table 5.0.2 The atmospheric conditions for a typical day in July (Fort McMurray) . . . . . 23

## LIST OF FIGURES

	PAGE
Figure 3.2.1. Eulerian (spatial) coordinate system .....	24
Figure 3.2.2. Lagrangain (material) coordinate system. ....	25
Figure 4.1.1. Evaporation apparatus: A, evaporation apparatus, 1.83 m in diameter and 0.38 m height, used to measure solid profile during evaporation process; B, evaporation pan, 0.6 m in diameter and 0.3 m height, used to measure maximum evaporation demand. ....	26
Figure 4.1.2. Evaporation demand during the experimental period. ....	27
Figure 4.2.1. Solids content of the top 1 cm. The bars represent the range of measurements..	28
Figure 4.2.2. Crust depth. The bars represent the range of measurements. ....	29
Figure 4.2.3. Total depth of fine tails. The bars represent the range of measurements. ....	30
Figure 4.2.4. Solids content profile of experiment I. The bars represent the range of measurements.....	31
Figure 4.2.5. Solids content profile of experiment II. The bars represent the range of measurements.....	32
Figure 4.2.6. Solids content profile of experiment III. The bars represent the range of measurements. ....	33
Figure 4.3.1. Simulated (—) and measured (●) depth of fine tails as a function of time. Experiment I. ....	34
Figure 4.3.2. Simulated (—) and measured (●) depth of fine tails as a function of time. Experiment II. ....	35
Figure 4.3.3. Simulated (—) and measured solids content profile at t=11(■), t=18 (◆), t=35 (●). The dashed line is predicted value at day 38. Experiment I.....	36
Figure 4.3.4. Simulated (—) and measured solids content profile at t=7(●), t=21 (■), different times during evaporation. Experiment II. ....	37
Figure 4.3.5. Simulated (—) and measured (●) solid content at the upper surface (1 cm) of the crust as a function of time. Experiment I. ....	38
Figure 4.3.6. Simulated (—) and measured (●) solid content at the upper surface (1 cm) of the crust as a function of time. Experiment II. ....	39

Figure 5.0.1.	Final solids content profiles after drying 37 days in June and 27 days in July.....	40
Figure 5.0.2.	The time required to complete the drying process as a function of initial ponding depth in June and July.....	41
Figure 5.0.3.	Solids content at the surface as a function of time for the initial ponding depth of 30 cm in June and July.....	42

## EXECUTIVE SUMMARY

Oil sand processing in northern Alberta generates a large amount of water saturated fine-grained material (or fine tails) as a byproduct. Fine tails are a mixture of fine mineral particles (<44µm in diameter), small amounts of bitumen, and a large amount of water. Some consolidation of the solids in the fine tails occurs quickly. The initial solids content of fine tails stream entering the tailing pond is approximately 5 % on a gravimetric basis. After one or two years, the solids consolidate to a solids content of 20% to 30%. However, under natural conditions, further consolidation is extremely slow. More than 500 million cubic meters of fine tails at 30% solids content have been produced over last 20 years by two operating extraction plants owned by Syncrude Canada Ltd. and Suncor Inc., in Fort McMurray, Alberta. Furthermore, fine tails are accumulating at a rate of approximately 20 million cubic meters per year.

The development of a system to reduce the enormous volume of fine tails, or to dry to a stable surface which permits further remediation treatments, presents a unique challenge for the sustainable management of the environment. Natural evaporation is one of the available options that can be used to reduce the volume of fine tails because it is both inexpensive and technically feasible on a large scale.

Evaporation of water from fine tails involves the complex interactions between the potential atmospheric evaporation demand and the physiochemical characteristics of the fine tails. When the surface of fine tails is wet (saturated with water) evaporation is controlled by the energy balance and the rate of vapour transport at the boundary between the atmosphere and the surface of the tails. The calculations needed to quantify dewatering at this stage are uncomplicated and well known. Eventually, evaporation at the surface exceeds the rate of transport of water within the layer of fine tails, and a surface crust develops. At this point further evaporation is controlled by the physical properties of the fine tails which govern the movement of water to the surface in either a liquid or vapour phase. The formation of a crust and the movement of water under unsaturated conditions are complex and not well understood.

A thorough understanding of the complex interactions among the solids, water and external atmospheric demand is required to develop a system to dewater fine tails by evaporation. However, the effects of ponding layers and initial solids content as well as the effect of

atmospheric conditions on the desiccation rate of fine tails are not fully understood. Furthermore, experiments can not be conducted for all possible conditions regarding different solids content, ponding depth, and evaporation demands. Thus, a mathematical model is an essential tool to identify potentially important parameters needed to understand these complex processes and identify the conditions at which optimum evaporation may be achieved.

A one dimensional approach was used to model the evaporation process from fine tails. To picture the modelling concept, divide the fine tails into a number of layers as shown Figure I.

The evaporation rate at the surface is described by

$$q_{\text{surface}} = f(S_n, T, RH, v, \theta_{\text{surface}}) \quad (1)$$

where  $S_n$  is net radiation at the surface,  $T$  is temperature,  $RH$  is relative humidity,  $v$  is wind speed, and  $\theta_{\text{surface}}$  is surface water content.

The volume of fine tails decreases as water is lost from the surface. However, the repulsion between the solid particles resists the volume change. Thus, water is drawn from the layer underneath. The rate that the thickness of each layer changes can be expressed as

$$\frac{\partial d}{\partial t} = f(h, q, K, \nabla\Psi, \nabla P, \nabla C) \quad (2)$$

where  $d$  is thickness,  $h$  is the initial ponding depth,  $q$  is water flux,  $K$  is the hydraulic conductivity,  $\nabla\Psi$  is the water potential gradient,  $\nabla P$  is the overburden pressure gradient, and  $\nabla C$  is the solid content gradient.

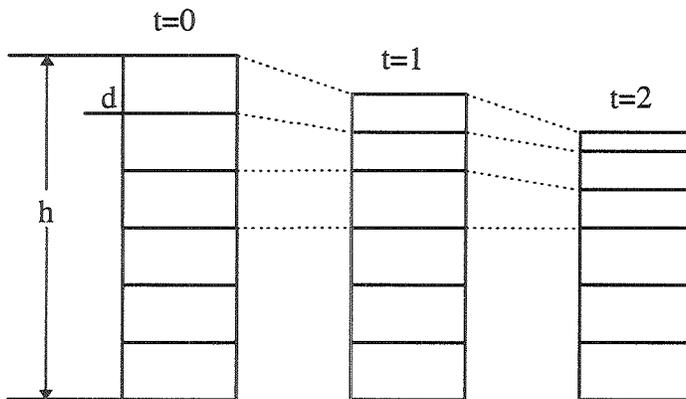


Figure I. A conceptual profile of fine tails during evaporation process.

As long as a given layer is saturated, the change in thickness is directly related to the loss of water from that layer so that

$$\frac{\partial d}{\partial t} = \frac{\partial q}{\partial x} \quad (3)$$

where  $\partial q/\partial x$  is water loss through the layer. When the layer has lost enough water so that it is unsaturated, its thickness is described by the shrinkage characteristics of the fine tails. A system of mathematical equations describing the formation of the solid phase and the interactions between the solid and the liquid phases were derived by quantifying the forces acting on the solid phase during the drying process. A corresponding system of mathematical equations describing the movement of water through the system was derived by including the resistance to water movement by the solid phase and the conservation of mass of the liquid phase. The hydrodynamic resistance of the solid phase to water movement at low solids content was described as the sum of resistances from individual particles; at high solids content, a permeability function was used. The energy exchange processes at the surface of fine tails were modeled to include the absorption of short wave radiation, the emission of long wave radiation, and the loss of latent heat due to evaporation.

As evaporation involves primarily the movement of water, either in liquid or vapor phase, to the surface, the first requirement of the mathematical model is an adequate description of the water transport process in the fine tails. Because heat energy is required for the vaporization of liquid water temperature affects the movement of water through its effects on viscosity and the potential energy of water, the model must also include adequate description of heat transport process. A third component of the model describes the change in the inter-particle distances as water is lost from the fine tails and its interactions with water movement. Thus the model consists of four components describing the movement of water, vapor, heat, and the volume changes of the solids.

The model uses a modular approach. The water, heat, and solids are the basis of individual modules. Each module is designed to function independently of the others. Interactions among the modules allow each module to access, but not modify, the outputs of the others. This approach lends the current model maximum flexibility for future expansion to include other related processes, such as freeze/thaw, formation and development of surface cracks, and possible effects of water uptake and transpiration by wetland vegetation.

The model was written in the DOS environment using FORTRAN 77 format. It can be run either in the DOS environment or Excel under windows.

Experimental validation was conducted using a tank, 1.83 m in diameter and 0.38 m in height, as an evaporation pan under greenhouse conditions. The total depth of fine tails, the thickness of the surface crust, and the solids content profile were measured.

The model successfully predicted the evaporation process from thin layers of fine tails under laboratory conditions. There was good agreement between the predicted and measured total depth of the fine tails, as well as the development of surface crust and the solids content profile during the evaporation process.

The model was used to simulate evaporation processes using two typical climate conditions in Fort McMurray. Under the medium evaporative demand, the increase in solids content at the surface was more gradual: the maximum solids content was reached near the end of the drying process. Under the higher evaporative demand, the solids content at the surface increased rapidly, with maximum solids content reached more than 10 days before drying of the whole profile was completed.

Only laboratory testing of the model was performed. Field studies would provide further validation of the model. In addition, a weak point in the model was its inability to describe the formation and effect of surface cracks on evaporation processes. Laboratory experiments showed that surface cracks may enhance the total rate of evaporation. The extent to which surface cracks affect evaporation from fine tails under field conditions was not clear.

With our modular system, the current model can be improved by incorporating the following components:

1. the formation of the cracks and their effects;
2. the freeze-thaw events;
3. the transpiration process;
4. the effects of amendments on the evaporation process.

Thus, a new revision of the model can provide a powerful tool for management of dewatering fine tails in Fort McMurray.

## ACKNOWLEDGMENTS

This report is the result of a joint research effort by Alberta Environmental Centre, the University of Alberta, and Syncrude Canada Limited. Financial support came from the Alberta Environmental Centre and Syncrude Canada, Limited. Facilities, scientific and technical support were provided by Alberta Environmental Centre. Department of Renewable Resources, University of Alberta, provided expertise in the formulation and development of the computer model.

Many individuals contributed to the research. Fruitful discussions with Dr. Graham Cuddy and Ted Lord of Syncrude Canada are gratefully acknowledged. The leadership of Richard L. Johnson, Alberta Environmental Centre, was important in ensuring the timely, consistent progress of the project.

Technical support for the project was provided by Paul Jablonski, Jim Storey, Kelly Gurski of Alberta Environmental Centre.

Assistance by Kurt Klingbeil in the development of graphic output of the model and by Hai Van Nguyen in computer programming is gratefully acknowledged.

Editorial work by Frank Geddes contributed significantly to improving the clarity of the final report. Thanks also go to Elaine Cannan, Michelle Myroniuk and Lorraine Yakemchuk who were responsible for the production of the final report.

## 1. INTRODUCTION

The extraction of bitumen from oil sands generates a large amount of fine tails as a waste byproduct. Fine tails comprise silt and clay, small amounts of bitumen, and water. They are normally deposited in large ponds for further consolidation and treatment. In Fort McMurray, Alberta, the total accumulated volume of fine tails in various ponds is over 500 million cubic meters. Furthermore, fine tails are accumulating at a rate of approximately 20 million cubic meters per year. The effective treatment to reduce the enormous volume of fine tails presents a unique challenge to the sustainable management of the environment.

Some consolidation of the solids of fine tails occurs quickly. For example, the solids content of the initial fine tails stream entering the pond is approximately 5% on a gravimetric basis (Cuddy and Lahaie, 1993), but after one or two years, the materials consolidate to a solids content of 20% to 30%. However, under natural conditions, further consolidation is extremely slow (Scott et al., 1985).

Natural evaporation has been used to dewater fine-grained slurries for decades (Sparrow, 1978; Ihle et al, 1983; Volker, 1982). It is one of the available options that can be used to reduce the volume of fine tails because it is both inexpensive and technically feasible on a large scale (Cuddy and Lahaie, 1993; Johnson et al. 1993). However, the effect of ponding depth and initial solids content on the desiccation rate of fine tails are not fully understood.

Evaporation from fine tails involves the complex interactions between the potential atmospheric evaporation demand and the physiochemical characteristics of the tails. Removal of water from fine tails by evaporation causes the relative movement of water and the solid phase in the fine tails. As long as a fluid surface is maintained, the rate of evaporation is controlled by the energy balance at the surface of tails and the vapor transport process at the atmospheric boundary layer. Once the atmospheric evaporation demand exceeds the rate of water transmission in the tails toward the surface, the surface dries and further evaporation is limited, largely controlled by the hydraulic properties of the surface crust.

As the surface crust becomes unsaturated, the vaporization of water occurs at a progressively greater depth from the surface. The transport of heat and water vapor in the unsaturated surface crust then becomes important limitations to the further dewatering of the fine tails.

The formation and properties of the surface crust depend to a large degree on the consolidation properties of the fine tails. As water evaporates from the surface of the tails, the total volume of the tails is reduced and the inter-particle distances near the surface increase. Sedimentation and consolidation of the solids are then controlled by gravity and the inter-particle stress gradients that have developed as water loss continues. The solids content of the surface layer continues to increase as evaporation progresses, eventually leading to the formation of a surface crust that limits the rate of further evaporation.

Physical characteristics of the surface crust, such as thickness, density and permeability depend not only on the nature of particle-to-particle and particle to water interactions in the tails during evaporation, but also on external factors such as the potential atmospheric evaporation demand and its diurnal fluctuation. Under high evaporative demands, thin surface crusts form with a higher bulk density that greatly reduces the subsequent rate of evaporation. On the other hand, moderate and low evaporative demands result in thick and low density surface crusts that are more permeable to water both in the liquid and vapor phases; these thick, low density crusts sustain evaporation for longer time.

Other external factors, such as temperature and the transport of heat energy are important in evaporation processes. The temperature within fine tails also affects the movement of water and solids by its effect on the viscosity of water. The transport of heat energy within fine tails has a large effect on the vaporization and movement of water vapor in the unsaturated surface crust.

A thorough understanding of the complex interactions among the solids, water, and external atmospheric demand is required to develop effective treatment procedures to dewater fine tails by evaporation. However, experiments cannot be conducted for all possible combinations of solids content, ponding depth, and evaporation demands. Thus, a mathematical model is an essential tool to identify potentially important parameters needed to understand these complex processes.

Mathematical models of the evaporation and consolidation processes in fine tails have been developed (Gibson et al. 1967; Gibson, et al. 1981; Pollock, 1988; Swabrick, 1992), but many of these considered only the limited case of saturated consolidation, where the solids content of the soil-water system increases monotonically. Swabrick (1992) adopted a unified

sedimentation/consolidation approach to model the behavior of tails, but did not consider the effect of heat and vapor transport on the movement of water and solids.

This report describes the development of a mathematical model that calculates the movement of water and heat in the consolidation of solids in fine tails during evaporation. After experimental validation, the model was used to simulate natural evaporation processes from fine tails under various atmospheric conditions. The results may be used to design optimum management practices for using natural evaporation to dewater fine tails.

## 2. OBJECTIVES

The two main objectives in this study are:

1. to develop a mathematical model that describes evaporation processes. The model should:
  - a. identify important parameters that control the evaporation process, and
  - b. identify conditions at which optimum evaporation may be achieved.
2. to validate the model by comparing its predictions with experimental data.

## 3. FORMULATION OF THE MODEL

### 3.1 The Modelling Approach

As evaporation involves primarily the movement of water, either in liquid or vapor phase, to the surface, the first requirement of the model is adequate description of the water transport process in the fine tails. Because heat energy is required for the vaporization of liquid water, and in addition, temperature affects movement of water through its effects on viscosity of water as well as the potential energy of water, the model must also include adequate description of heat transport process. A third component of the model describes the change in the inter-particle distances as water is lost from the fine tails and its interactions with water movement.

Thus the model consists of four components describing the movements of water, vapor, heat, and the volume changes of the solids. The model follows a modular approach, in which the main factors – water, heat, and solids – are the basis of individual modules. Each module is designed to function independently of the others. Interactions among the modules allow each module to access, but not modify, the outputs of the others. With this approach, the model can

be expanded later to include the effects of other factors, such as surface cracking and freezing/thawing events.

### 3.2 The Lagrangian Vs. Eulerian Coordinate Systems

Water loss from the fine tails during evaporation results in a reduction of total volume. In a one dimensional system, this is reflected in the reduction of the thickness. Before proceeding to the development of the model, we must first define a reference system in which the movements of water, heat, as well as the changes in the material volume can be described.

In modelling the movement of water and heat in non-deformable, rigid porous materials, a fixed (Eulerian) coordinate system is often used. In this case, the coordinate system is fixed in space to a reference point, usually chosen as the surface or the bottom of the material. This coordinate system is rigid so that the real distances between any pair of points in the system remain the same. For example, if  $x_1$  is 5 cm above the bottom of the layer of fine tails to be modeled, it is always 5 cm above the bottom. This system is not ideally suited in situations where the volume of the system itself undergoes change, such as during the dewatering of fine tails

For example, the top of the fine tail in this system is a variable, as its distance from the bottom changes with time (Figure 3.2.1). Thus, a point fixed in space may represent a different part of the solid material at different time, and the original surface of the materials may fall out of the range as the total depth of the tails decreases during evaporation

An alternative reference system is the Lagrangian coordinate system. It is also often referred to as the material coordinate system. In contrast to the Eulerian coordinate system, this is a flexible coordinate system in that there reference system deforms along with the deforming material. The coordinates representing various points in the system are fixed to the solid phase of the material so that they move with the fine tails as the total volume undergoes change. Each discrete point in this coordinate system is attached to a fixed part of fine tails frame (Figure 3.2.2). For example, in this reference system, the surface of the fine tails is a constant regardless of its actual position. This makes the mathematical formulation and programming considerably easier. In the end, the actual spatial position of the surface of the fine tails, at any time can be calculated from the density distributions.

The Lagrangian coordinate ( $X$ ), is linked to the Eulerian coordinate( $\chi$ ), by a one to one relationship

$$\chi = f(X, t) \quad (1)$$

where  $t$  is time. Because the Lagrangian coordinate system is attached to the solids phase, the total amount of solids between any two points e.g.,  $X_1$  and  $X_2$ , remains the same throughout time. Thus, we have

$$\int_{X_1}^{X_2} S_o dX = \int_{\chi_1}^{\chi_2} S d\chi \quad (2)$$

where  $\chi_1$  and  $\chi_2$  are points in the Eulerian coordinate system corresponding to  $X_1$  and  $X_2$  in the Lagrangian system.  $S_o$  and  $S$  are initial and final volume fractions of the solids phase. The differential form of equation 2 is,

$$\frac{d\chi}{dX} = \frac{S_o}{S} = J(X, t) \quad (3)$$

where  $J$  is called the Jacobian.

### 3.3 Equations of Water Transport

In a one-dimensional formulation, the surface of the fine tails is used as the reference and the downward direction as our positive direction for  $\chi$  and  $X$ . Relative movement of water and solids is driven by the gradients in water potential, gravity, and to lesser degree, temperature and water vapour density. For the spatial coordinate system, this can be written as,

$$\mathbf{q} = -\frac{K(\theta, S)}{\eta} \left( \frac{\partial \psi}{\partial \chi} - \rho_w g \right) - \frac{D_v(\theta, S, T)}{\rho_w} \frac{\partial \rho_v}{\partial \chi} - \frac{K_{wT}(\theta, S, T)}{\eta} \frac{\partial T}{\partial \chi} \quad (4)$$

where

- $\mathbf{q}$  is water flux ( $\text{m s}^{-1}$ )
- $K(\theta, S)$  the permeability of fine tails ( $\text{m}^2$ )
- $\eta$  is the viscosity of water ( $\text{kg m}^{-1} \text{s}^{-1}$ )
- $\psi$  is the water potential of fine tails (Pa)
- $\rho_w$  is density of water ( $\text{kg m}^{-3}$ )

- $g$  is the gravitational constant ( $9.8 \text{ m s}^{-2}$ )
- $D_v(\theta, S, T)$  is water vapour diffusivity of fine tails ( $\text{m s}^{-1}$ )
- $\rho_v$  is density of water vapour in the air filled pores of fine tails ( $\text{kg m}^{-3}$ )
- $K_{wT}(\theta, S, T)$  is the liquid water permeability induced by temperature gradient, ( $\text{N }^\circ\text{C}^{-1}$ )
- $\theta$  is percent saturation
- $S$  is volume fraction of solids
- $T$  is temperature ( $^\circ\text{C}$ ).

The first term on the right side of equation 4 describes water movement under the influence of potential energy differences of the liquid phase; the second term describes the movement of water by vapour diffusion; and the third term describes the movement of water induced by a temperature gradient.

The total water flux,  $\mathbf{q}$ , can be divided into liquid water flux,  $\mathbf{q}_l$ , and water vapour flux,  $\mathbf{q}_v$ , so that

$$\mathbf{q} = \mathbf{q}_l + \mathbf{q}_v \quad (5)$$

where

$$\mathbf{q}_l = -\frac{K(\theta, S)}{\eta} \left( \frac{\partial \psi}{\partial \chi} - \rho_w g \right) - \frac{K_{wT}(\theta, S, T)}{\eta} \frac{\partial T}{\partial \chi} \quad (6)$$

and

$$\mathbf{q}_v = -\frac{D_v(\theta, S, T)}{\rho_w} \frac{\partial \rho_v}{\partial \chi} \quad (7)$$

Expressed in the Lagrangian coordinate system, where  $d\chi = JdX$  (equation 3), equations 5 to 7 can be expressed as

$$\mathbf{Q} = \mathbf{Q}_l + \mathbf{Q}_v \quad (8)$$

where

$$\mathbf{Q}_l = -\frac{K(\theta, S)}{J\eta} \left( \frac{\partial \psi}{\partial X} - J\rho_w g \right) - \frac{K_{wt}(\theta, S, T)}{J\eta} \frac{\partial T}{\partial X} \quad (9)$$

and

$$\mathbf{Q}_v = -\frac{D_v(\theta, S, T)}{J\rho_w} \frac{\partial \rho_v}{\partial X} \quad (10)$$

Consideration of mass balance leads to

$$\frac{\partial}{\partial t} (J(1-S)\theta) = -\frac{\partial \mathbf{Q}}{\partial X} \quad (11)$$

Expansion of the left hand side of equation 11 using equation 3 leads to

$$(1-S) \frac{\partial \theta}{\partial t} - \frac{\theta}{S} \frac{\partial S}{\partial t} = \frac{1}{J} \frac{\partial}{\partial X} \left[ \frac{K(\theta, S)}{J\eta} \left( \frac{\partial \psi}{\partial X} - J\rho_w g \right) + \frac{K_{wt}(\theta, S, T)}{\partial \eta} \frac{\partial T}{\partial X} \right] - \frac{1}{J} \frac{\partial \mathbf{Q}_v}{\partial X} \quad (12)$$

where  $\mathbf{Q}_v$  is given by equation 10.

### 3.4 Special Case: Saturated Condition

A special case of equation 12 occurs when the medium is saturated, such as in the initial stages of evaporation from fine tails. Under these conditions, the term describing water vapour transport in fine tails (in equation 4) becomes zero and the temperature gradient-induced liquid water flow become negligible, so that

$$\mathbf{Q} = -\frac{K(\theta, S)}{J\eta} \left( \frac{\partial P}{\partial X} - J\rho_w g \right) \quad (13)$$

Substituting this equation into equation 12,  $\theta = 1$  (saturated condition) and  $J = \frac{S_o}{S}$  (equation 3)

$$\frac{1}{S} \frac{\partial S}{\partial t} = -\frac{S}{S_o} \frac{\partial}{\partial X} \left( \frac{SK(\theta, S)}{S_o\eta} \left( \frac{\partial P}{\partial X} - \frac{S_o}{S} \rho_w g \right) \right) \quad (14)$$

From the principle of effective stress, in the absence of external load,

$$\int_0^{\chi} g(S\rho_s + (1-S)\rho_w)d\chi = \sigma + P \quad (15)$$

where the integral on the left side indicates the total load at depth  $\chi$ ,  $\sigma$  is effective stress in the solid phase,  $\rho_s$  is the density of fine tails, and  $P$  is pressure of liquid water. Differentiating equation 15 in the Eulerian coordinate system, with respect to  $\chi$ , yields

$$-\frac{d\sigma}{d\chi} + g(S\rho_s + (1-S)\rho_w) = \frac{dP}{d\chi} \quad (16)$$

In the Lagrangian coordinate system,

$$-\frac{d\sigma}{dX} + Jg(S\rho_s + (1-S)\rho_w) = \frac{dP}{dX} \quad (17)$$

Substituting equation 17 into equation 14 yields ( after simplification),

$$\frac{\partial S}{\partial t} = \left(\frac{S}{S_o}\right)^2 \frac{\partial}{\partial X} \left[ \frac{SK(\theta, S)}{\eta} \left( \frac{\partial \sigma}{\partial X} - S_o g(\rho_s - \rho_w) \right) \right] \quad (18)$$

Assuming effective stress ( $\sigma$ ) to be a function of solid content,

$$\frac{\partial \sigma}{\partial X} = \frac{\partial \sigma}{\partial S} \frac{\partial S}{\partial X} \quad (19)$$

so that

$$\frac{\partial S}{\partial t} = \left(\frac{S}{S_o}\right)^2 \frac{\partial}{\partial X} \left[ \frac{SK(\theta, S)}{\eta} \left( \frac{d\sigma}{dS} \frac{\partial S}{\partial X} - S_o g(\rho_s - \rho_w) \right) \right] \quad (20)$$

Equation 20 is similar to the well-accepted Gibson formulation (Pollock 1988, Townsend et al. 1990, Swabrick 1992).

To predict the consolidation of fine tails, equation 20 is used initially. The temperature of the fine tails has only a minor effect on the consolidation process through its effect on the viscosity of water,  $\eta$ . When an unsaturated layer develops at the surface ( $0 < 1$ ), the more general equation 12 was used.

### 3.5 Equations of Heat Transport

Transport of heat is by conduction, convection, and latent heat transfer with water vapour.

The total heat flux, in the spatial coordinate system, is

$$H = C_w Q_l T + L_v(T) Q_v - \frac{K_T(\theta, S, T)}{J} \frac{\partial T}{\partial X} \quad (21)$$

where

- $C_w$  is the volumetric heat capacity of water, ( $4.18 \times 10^6 \text{ J m}^{-3} \text{ }^\circ\text{C}^{-1}$ )
- $L_v(T)$  is the latent heat of vaporization ( $\text{J kg}^{-1}$ ). It is a function of temperature.
- $K_T(\theta, S, T)$  is thermal conductivity ( $\text{J m}^{-1} \text{ s}^{-1} \text{ }^\circ\text{C}^{-1}$ )

Consideration of the energy balance within a material volume element leads to

$$\frac{\partial}{\partial t}(JC(\theta, S)T) = - \frac{\partial H}{\partial X} \quad (22)$$

Substituting equation (21) into equation 22,

$$\frac{\partial}{\partial t}(JC(\theta, S)T) = - \frac{\partial}{\partial X} \left[ C_w \mathbf{Q}_t T + L_v(T) \mathbf{Q}_v - \frac{K_T(\theta, S, T)}{J} \frac{\partial T}{\partial X} \right] \quad (23)$$

The volumetric heat capacity of the fine tails,  $C(\theta, S)$ , is

$$C(\theta, S) = C_s S + C_w (1 - S) \theta \quad (24)$$

where  $C_s$  is the volumetric heat capacity of the solids ( $\text{J m}^{-3} \text{ }^\circ\text{C}^{-1}$ ). By combining equations 23, 24, 12, and 3,

$$\frac{\partial}{\partial t}(JC(\theta, S)T) = -C_w T \frac{\partial \mathbf{Q}}{\partial X} + JC(\theta, S) \frac{\partial T}{\partial t} \quad (25)$$

Substituting the expression in equation 25 into equation 23 leads to a final equation for heat transport

$$C(\theta, S) \frac{\partial T}{\partial t} = \frac{1}{J} \frac{\partial}{\partial X} \left( \frac{K_T(\theta, S, T)}{J} \frac{\partial T}{\partial X} \right) - \frac{C_w \mathbf{Q}_t}{J} \frac{\partial T}{\partial X} - \frac{L_v(T) - C_w T}{J} \frac{\partial \mathbf{Q}_v}{\partial X} \quad (26)$$

### 3.6 Equations of Vapor Transport

Movement of water vapour is by diffusion. The water vapour flux is given by equation 10 as

$$\mathbf{Q}_v = - \frac{D_v(\theta, S, T)}{J \rho_w} \frac{\partial \rho_v}{\partial X}$$

where  $D_v(\theta, S, T)$  is the diffusivity of water vapour and  $\rho_v$  is the water vapour density in the air phase in fine tails, itself a function of temperature and potential of water.

$$\rho_v(T, P) = \rho_v^s(T) e^{P/RT} \quad (27)$$

where  $\rho_v^s(T)$  is the saturation vapour density as a function of temperature,  $R$  is the gas constant, and  $T$  is temperature.

### 3.7 Equations of Solid Phase Consolidation

An equilibrium relation was used to describe the stress-strain relationship of the solid phase. Conservation of momentum leads to

$$f(P, s, \theta)P + \sigma = \int_0^X J(\rho_s S + \rho_w \theta(1 - S))g dX \quad (28)$$

where  $\sigma$  is the one-dimensional effective stress of the solid phase. A simple relation between  $\sigma$  and  $S$  is assumed, where

$$\sigma \leq \sigma_{\max} = a \left( \frac{1-S}{S} \right)^b \quad (29)$$

$\sigma_{\max}$  is the maximum stress at which further consolidation would occur,  $a$  and  $b$  are constants. At  $\sigma < \sigma_{\max}$ , the solid volume fraction is assumed to be independent of the stress. That is, any decrease in  $\sigma$  from the maximum value determined from the solid volume fraction will not cause a corresponding expansion of the solid phase. When the stress increases, the solid volume fraction will remain constant until the stress has exceeded the previously reached maximum.

### 3.8 Boundary Conditions

The fine tails are bounded at the top by the surface and at the bottom by the substratum. As a first approximation, drainage of liquid water from the lower boundary is assumed to be zero. However, water vapour may diffuse freely through the lower surface. Thus, for the movement of water and water vapour, the lower boundary condition is

$$Q_i = 0 \quad (30)$$

$$Q_v = - \frac{D_v(\theta, S, T)}{J\rho_w} \frac{\partial \rho_v}{\partial X} \quad (31)$$

A constant temperature is assumed for the lower boundary,

$$T = T_0 \quad (32)$$

At the upper surface, the boundary conditions are described from a consideration of energy and mass balance. The energy balance for the upper boundary yields

$$-K_T(\theta, S, T) \frac{\partial T}{\partial X} = S_n - L_v(T)ET - H_s \quad (33)$$

where  $S_n$  is net radiation, ( $\text{J m}^{-2} \text{s}^{-1}$ ),  $ET$  is the rate of evaporation, and  $H_s$  is sensible heat exchange between the fine tails surface and the atmosphere. The boundary condition for water at the upper surface is

$$Q_i + Q_v = Q_p - ET \quad (34)$$

where  $Q_p$  is rate of precipitation.

The rate of evaporation,  $ET$ , and the sensible heat flux,  $H_s$ , are given by

$$ET = h(\rho_v - \rho_{va}) \quad (35)$$

and

$$H_s = \rho_a c_a h(T - T_a) \quad (36)$$

where  $\rho_v$  ( $\text{kg m}^{-3}$ ) is the water vapour density at the surface of fine tails,  $\rho_{va}$  ( $\text{kg m}^{-3}$ ) is the water vapour density of the atmosphere,  $T$  is the soil surface temperature,  $T_a$  is the air temperature,  $\rho_a$  ( $\text{kg m}^{-3}$ ) is the density of air,  $c_a$  ( $\text{J kg}^{-1} \text{ }^\circ\text{C}^{-1}$ ) is the specific heat of air, and  $h$  ( $\text{m s}^{-1}$ ) is the boundary layer conductance, which is a function of wind velocity and surface characteristics, such as roughness. Substituting equations 35 and 36 into equations 33 and 34,

$$-K_T(f, s, T) \frac{\partial T}{\partial X} + \rho_a c_a hT = S_n - L_v(T)h(\rho_{vs} - \rho_{va}) + \rho_a c_a hT_a \quad (37)$$

and

$$Q_i + Q_v = Q_p - h(\rho_v - \rho_{va}) \quad (38)$$

During precipitation events, it is assumed that water does not accumulate on the surface. Once the surface is saturated, any precipitation in excess of infiltration is assumed to be surface run-off.

### 3.9 Numerical Solutions

Numerical solutions for the model have been formulated from central differences in the spatial dimension and from the combined implicit and Crank-Nicholson methods for integration over time. The model consists of three interacting modules: water, heat, and solids. There are three input files: two files contain information on conditions at the top and the bottom boundaries of the fine tails, while a third file contains specific parameters, such as hydraulic conductivity information for the water module and stress-strain relation for solids module. An overall

program control input file specifies program control parameters common to all modules, such as initial and final times, the minimum and maximum time steps allowed, and the approximate time intervals at which output of simulation results are desired.

Consolidation (stress/strain relation) behavior of the fine tails as well as saturated hydraulic conductivities for the simulations were obtained from Pollock (1988), Scott et al. (1985).

#### 4. EXPERIMENTAL VALIDATION

##### 4.1 Materials and Methods

Fine tails having an initial solids content of approximately 32% (gravimetric basis) were supplied by Syncrude Canada Limited. They were placed in a tank, 1.83 m in diameter and 0.38 m in height, which was used as an evaporation pan under greenhouse conditions. Two pairs of gamma-probe access tubes were constructed and installed in the evaporation pan to measure solids content. The tubes were located 0.5 m from the wall of the tank (Figure 4.1.1A). Each pair provided a set of experimental data.

Three independent greenhouse experiments were conducted under the following conditions. (The environmental conditions and determination of maximum evaporation demand are described below.)

Experiment I. Fine tails with a solids content of 32% were poured into the evaporation tank to a depth of 30 cm. They were monitored for 35 days during which the maximum evaporation demand was approximately  $2.8 \text{ mm day}^{-1}$ .

Experiment II. Initial conditions were identical to experiment I. However, the experiment lasted only 22 days and the evaporation demand was approximately  $1 \text{ mm day}^{-1}$ .

Experiment III. The surface crust from experiment II were removed from the evaporation pan and the remaining fine tails were stirred. The initial solids content (after stirring) was approximately 34%, and the ponding depth was 24 cm. The experiment lasted 30 days, during which time the evaporation demand was approximately  $1.2 \text{ mm day}^{-1}$ .

The duration of each experiment was chosen so that sufficient information was obtained for the validation of model.

The solids content in the profile of fine tails was measured every week using both  $\gamma$ -probe and pipette withdrawal methods. For the  $\gamma$ -probes, the attenuation of  $\gamma$ -radiation which is a function of solids and water content (Jury et al., 1991) was used as a measure of solids content. In the pipette method, a 25-ml sample was collected at the desired depth and completely transferred to an aluminum drying can; the samples were then oven dried (105 °C) for 48 hours to determine the solids content. At least 7 samples were taken to best characterize the solids content in the profile. More samples were taken at upper layer where the solids content changed fast.

The particle density of the fine tails was measured using the pycnometer method (Blake and Hartge, 1986). Results were used to convert solids content from a volumetric to a gravimetric basis.

The total depth of fine tails and the thickness of the surface crust were measured every 2 days using a ruler. For the surface crust (top 1 cm), where the both  $\gamma$ -probe and pipette methods were inappropriate, a gravimetric method was used to determine the solids content. Ten to fifteen grams of fine tails were taken for each sample at the surface.

Daily maximum evaporation demand was measured using an evaporation pan, 59 cm in diameter and 30 cm in height, to which fine tails were added to depth of 15 cm (Figure 4.1.1B). This provided the same relative height of tails as that in the tank, so they each received approximately the same amount of net radiation. The pan was set on a digital platform scale with a resolution of 50 g. The weight of the pan and contents were recorded daily, the fine tails were stirred to prevent the formation of a crust, and water was added to make up for any loss. The weight of water lost per day was considered to be the daily maximum evaporation demand (Figure 4.1.2). Due to a problem with the recorder, the data during the first three days of experiment 1 were not included.

## 4.2 Results

The solids content at fine tails of the top 1 cm increased with time ( Figure 4.2.1), and as a result , a crust gradually formed (Figure 4.2.2). During the first 15 days, both the solids content of the surface and solids depth increased at a similar rate in all three experiments. After 15 days, the increase was more rapid at higher evaporation rates (experiment I), because large cracks formed. Similarly, the total depth of fine tails decreased with time (Figure 4.2.3). In experiment I, where the evaporation demand was higher, the total depth of fine tails decreased at an accelerated rate after the appearance of large cracks (approximately 12 days). The cracks under experiment I conditions developed faster than under other conditions. The cracks would have enhanced evaporation by exposing the wet subsurface materials to ambient air, thus reducing the insulating effect of the crust. These cracks might also have enhanced the evaporation by increasing surface area and the interception of solar radiation. The detailed information about the formation and development of cracks was not recorded because the model was not formulated to describe these processes.

Solids content in the profile for all three experiments is shown in Figures 8 to 10. A transition zone, between the crust and liquid fine tails, was very clear in all cases (Figures 4.2.4 to 4.2.6). The solids content of fine tails under the crust remained relatively unchanged. Thus, the majority of water loss was from the top layer or crust. Consolidation sedimentation processes played a minor role in the dewatering process under all experimental conditions, as a slight increase of only 3 to 4% in solids content on the bottom was observed.

## 4.3 Comparison of Experimental Data and the Predictions of the Model

The mathematical model was used to predict the evaporation processes observed in the greenhouse evaporation experiments. The measured daily evaporation demands were used as inputs for the top boundary layer.

The saturated hydraulic conductivity was estimated from particle size distribution (Cuddy and Lahaie 1993). As solids content increases, interactions among individual particles normally result in an increase in hydrodynamic resistance, or a decrease in hydraulic conductivity. The saturated hydraulic conductivity was assumed to decrease as a function of the square of the average inter-particle distance: at a solids content of 32%, hydraulic conductivity was estimated

to be  $2 \text{ cm day}^{-1}$ , and decrease by a factor of 100 (to  $0.02 \text{ cm day}^{-1}$ ) as solids content increased to 75%. The compressibility data for the fine tails were obtained from Pollock (1988).

A comparison of the simulated and measured decrease in the total depth of fine tails in Experiments II and IV during evaporation are shown in Figure 4.3.1 and Figure 4.3.2, respectively. Although the simulated depth closely followed the measured values, it is clear that the actual decrease of the depth of fine tails exceeded that of the model prediction.

The predicted decrease in depth of fine tails in these experiments corresponded to the maximum evaporative demand, but actual decrease in depth indicated that the fine tails were losing water at a rate faster than the maximum evaporative demand. The extra water loss was probably due to the formation of surface cracks, which increased the effective area where radiative energy is adsorbed and water vapor is lost. The difference between the predicted and measured total depth of fine tails during evaporation suggests that the actual rate of evaporation from fine tails, with a wet surface crust and many cracks, may be higher than the maximum evaporative demand measured on a smooth surface.

Figure 4.3.3 and Figure 4.3.4 show the comparison between predicted and measured solids content profile at various times for Experiments I and II, respectively. It is obvious that a surface crust formed. For example, Figure 4.3.3 illustrates the formation of a 7 cm thick surface crust (in which the solids content was  $>50\%$ ) at the end of 35 days of evaporation. Little change in solids content occurred below the surface crust. At the bottom of fine tails, consolidation and sedimentation under the force of gravity added to a slightly higher solids content.

For Experiment I, the measured solids content at the surface was considerably lower than that predicted by the model (Figure 4.3.3). The predicted solids content profile at 38 days (dashed line) matched exactly with the measured solids profile for 35 days. This suggests that the actual rate of evaporation from the fine tails was approximately 10% higher than the maximum evaporative demand measured using current device.

Figure 4.3.5 and Figure 4.3.6 show a comparison of predicted and measured solids content at the crust surface as a function of time. The short term variations in predicted surface solids content were due to the fluctuations in evaporative demands. As shown by all of these figures the agreement between the model predictions and the experimental measurements was excellent, considering the uncertainties in the estimated material properties, such as hydraulic conductivity and compressibility.

A sudden decrease in evaporative demand results in a decrease in stress near the surface, as the movement of water from lower depths toward the surface exceeds the rate of evaporation. The unique relationship between stress and solids content, under these conditions, would lead to a decrease in solids content at the surface, or an expansion in volume. But an unconsolidated material, when stress is released, follows a different stress-strain curve, in which little volume expansion occurs. When the stress is reapplied, significant compression would not occur until the previous maximum stress is exceeded. Although experimentally these facts are simple and straight forward, they present a unique challenge to the modelling exercise. In the model reported upon here, a release of stress is assumed to yield a slight expansion of volume and some secondary compression. Because of a lack of original data, the increase in average inter-particle distance upon complete release of stress was assumed to average 5%.

## 5. EXAMPLES

The model was used to predict evaporation from a thin-layer of fine tails under typical summer conditions in Fort McMurray, Alberta. Simulations were carried out under medium evaporative demand (June) and high evaporative demand (July). For each month, detailed weather information for a 24-hr period were used for the first day; weather conditions were then assumed to repeat in the following days. Hourly weather information included solar radiation, air temperature, and relative humidity (Tables 5.0.1 and 5.0.2). The calculated maximum rate of evaporation from fine tails for the medium (June) and high (July) evaporative demand conditions were 6.0 and 7.8 mm day<sup>-1</sup>, respectively.

For each atmospheric condition, four simulations were carried out, at an initial solids content of 32% and initial depths of 5, 10, 20 and 30 cm. In each case, simulation was stopped after the solids content reached 50% at the bottom.

Figure 5.0.1 shows the profile of final solids content for a 30 cm deep trial. Under the medium evaporative demand (June), the drying process took 37 days, while under the high evaporative demand (July), it took 27 days. After the fine tails were dried, the two curves merged.

The time required to complete the drying process increased rapidly with increasing the depth of fine tails (Figure 5.0.2). Under atmospheric conditions for June, the drying process took 4 days when the initial depth was 5 cm. As the initial depth increased by a factor of 6 (to 30 cm),

the drying time increased by a factor of 9.3 (to 37 days). Similarly, under July conditions, the drying time increased from 3.3 to 27 days as the initial depth increased from 5 to 30 cm.

Under both the June and July conditions, the surface solids content showed a rapid, initial increase for the first few days (Figure 5.0.3). The rate at which the surface solids content increased with time then gradually decreased until reaching the shrinking limit, estimated at 78%. At this point, the surface was unsaturated. Under medium evaporative demand (June), the increase in solids content at the surface was more gradual; the maximum solids content was reached near the end of the drying period. Under higher evaporative demand (July), the solids content at the surface increased rapidly and the maximum solids content was reached more than 10 days before drying was complete.

Daily variation in solids content at the surface was due to diurnal variation in evaporative demand. During the day, under high evaporative demand, the solids content at the surface increased quickly. Conversely, at night, when there was minimal evaporation, the solids content at the surface decreased as water from lower layers moved up to replenish the water lost by evaporation during the day.

The model underestimated the time required to complete the drying process the formation and effect of surface cracks on evaporation was not included.

## 6. CONCLUSIONS

The model successfully predicted the evaporation process from thin layers of fine tails under laboratory conditions. There was good agreement between predicted and measured total depth of fine tails, as well as the development of surface crust and the increase in solids content during the evaporation process.

The model can be used to predict the effect of evaporation for given environmental conditions and to identify conditions under which optimal evaporation can be achieved.

A weak point in the model was its inability to describe the formation and effect of surface cracks on evaporation processes. Laboratory experiments showed that surface cracks may enhance the total rate of evaporation. The extent to which surface cracks affect evaporation from fine tails under field conditions was not clear.

## 7. RECOMMENDATIONS

1. Only limited laboratory testing of the model has been performed. Field studies would provide further validation of the model.

2. The model developed in this study lacks the ability to predict surface crack development. The formation of the cracks and their effects should be incorporated into future revisions of the model.

3. Freeze/thaw events have been identified as an inexpensive and effective technique of dewatering fine tails. However, the freezing layer, freezing time, and pumping rate have not yet been optimized for fine tails (Johnson et al., 1993). The mathematical model can be developed to describe freeze-thaw, and the consolidation/evaporation processes that follow by incorporating modules for freeze-thaw into future revisions.

4. Amending sand with fine tails can enhance the dewatering process. A mathematical model should be developed to help explain some of the uncertainties regarding the mixing process.

5. Dewatering through transpiration has been identified as an important process (Johnson et al., 1993). The model can be expanded to predict the quantity of water lost by transpiration process.

6. Besides modelling, a high speed cultivator could be used in the field to mix appropriate portions of tailing sand and fine tails and to form stable aggregates leading to a soil-like profile that would enhance the growth of plants. Through this study, a practical technology might be developed that could be used to reclaim the surface of fine tails to support a sustainable plant and animal community.

## REFERENCES

- Blake, G.R. and K.H. Hartge, 1982. Particle density, p377-382, *In* Methods of soil analysis Part 1, physical and mineralogical methods, 2nd edition. eds by A. Klute. 1188p.
- Cuddy, G. and R. Lahaie, 1993. Dewatering of mature fine tails by natural evaporation. Syncrude Canada Ltd. Research and Development Research Report 93-01.
- Gibson, R.E., G.L. England and M.J.L. Hussey, 1967. The theory of one-dimensional consolidation of saturated clays. I. Finite non-linear consolidation of thin homogeneous layers. *Geotech.* 17:261-273.
- Gibson, R.E., R.L. Schiffman, K.W. Cargill, 1981. The theory of one-dimensional consolidation of saturated clays. II. Finite nonlinear consolidation of thick homogeneous layers. *Canadian Geotech. J.* 18:280-293.
- Ihle, S.W., B.B. Johnson, T.A. Reven, and G.J. Sparrow. 1983. Field tests of an equation that describes dewatering of clay suspensions on a sand bed. *International Journal of mineral Processing*, 11:239-253.
- Johnson, R.L., P. Bork, E.A.D. Allen, W.H. James, and L. Koverny, 1993. Oil sand sludge dewatering by freeze-thaw and evapotranspiration. Alberta Conservation and Reclamation Council Report No. RRTAC 93-8. ISBN 0-7732-6042-0 247p.
- Jury, W.A., W.R. Gardner, and W.H. Gardner, 1991. *Soil physics*. 5th edition. John Wiley & Sons, Inc. New York. 328p.
- Neiman, H. and Y. Feng. 1995. EVAP: the numerical analysis of evaporation from fine tailing pond. User's manual.
- Pollock, G.W. 1988. Large strain consolidation of oil sand tailings sludge. M.Sc. Thesis, University of Alberta, Canada.
- Scott, J.D., M.B. Dusseault, and W.D. Carrier III. 1985. Behavior of the clay/bitumen/water sludge system from oil sands extraction plants. *Journal of Applied Clay Science*, 1:207-218.
- Sparrow, G.J. 1978. Dewatering a clay slime: Equilibrium conditions for drainage and sedimentation, a summary of the theory. Commonwealth Scientific and Industrial Research Organization. Report MCC 117. 17p.

- Swabrick, G.E. 1992. Transient unsaturated consolidation in desiccating mine tailings. Ph.D. Thesis, The University of New South Wales, Australia.
- Townsend, F.C. and M.C. McVay, 1990. SOA: Large strain consolidation predictions. ASCE J. Geotech. Eng. 116:222-243.
- Volker, A. 1982. Polders: An ancient approach to land reclamation. Nature and Resources, 18:2-13.

## LIST OF SYMBOLS

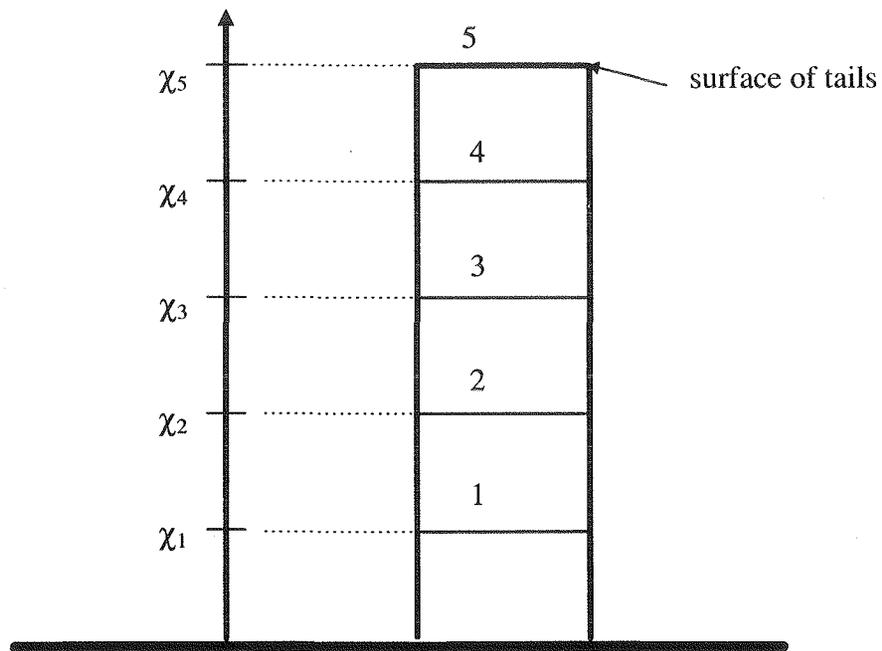
$\chi$	spatial coordinate
$\rho_a$ ,	density of air ( $\text{kg m}^{-3}$ )
$\rho_v$ ,	density of water vapor in the air filled pores ( $\text{kg m}^{-3}$ )
$\rho_{va}$ ,	water vapor density of the atmosphere ( $\text{kg m}^{-3}$ )
$\rho_w$ ,	density of water ( $\text{kg m}^{-3}$ )
$\rho_s$	density of fine tails ( $\text{kg m}^{-3}$ )
$\rho_v^s$	saturation vapor density ( $\text{kg m}^{-3}$ )
$C(\theta, S)$	volumetric heat capacity of fine tails ( $\text{J m}^{-3} \text{ }^\circ\text{C}^{-1}$ )
$c_a$ ,	specific heat of air ( $\text{J kg}^{-1} \text{ }^\circ\text{C}^{-1}$ )
$C_w$ ,	volumetric heat capacity of water ( $4.18 \times 10^6 \text{ J m}^{-2} \text{ }^\circ\text{C}^{-1}$ )
$C_s$	volumetric heat capacity of solids ( $\text{Jm}^{-3} \text{ }^\circ\text{C}^{-1}$ )
$D_v(\theta, S, T)$	water vapor diffusivity of fine tails ( $\text{m}^2 \text{ s}^{-1}$ )
$ET$	rate of evaporation
$g$	gravitational constant ( $9.8 \text{ m s}^{-2}$ )
$h$	boundary layer conductance ( $\text{m s}^{-1}$ )
$H_s$	sensible heat exchange between surface of fine tails and the atmosphere
$J$	Jacobian
$K(\theta, S)$	permeability of fine tails ( $\text{m}^2$ )
$K_T(\theta, S, T)$	thermal conductivity ( $\text{Jm}^{-1} \text{ s}^{-1} \text{ }^\circ\text{C}$ )
$K_{wT}(\theta, S, T)$	liquid water permeability induced by temperature gradient ( $\text{N}^\circ\text{C}^{-1}$ )
$L_v(T)$	latent heat of vaporization ( $\text{J kg}^{-1}$ )
$\psi$	water potential of fine tails (Pa)
$q$	total water flux ( $\text{m s}^{-1}$ )
$q_l$	liquid water flux ( $\text{m s}^{-1}$ )
$q_v$	water vapor flux ( $\text{m s}^{-1}$ )
$Q_p$	rate of precipitation
$S$	final volume fraction of the solid phase
$S_n$	net radiation ( $\text{Jm}^{-2} \text{ s}^{-1}$ )
$S_o$	initial volume fraction of the solid phase
$T_a$	air temperature ( $^\circ\text{C}$ )
$X$	material coordinate
$\theta$	percent saturation
$\sigma$	one-dimensional effective stress of the solid phase

Table 5.0.1. The atmospheric conditions for a typical day in June (Fort McMurray).

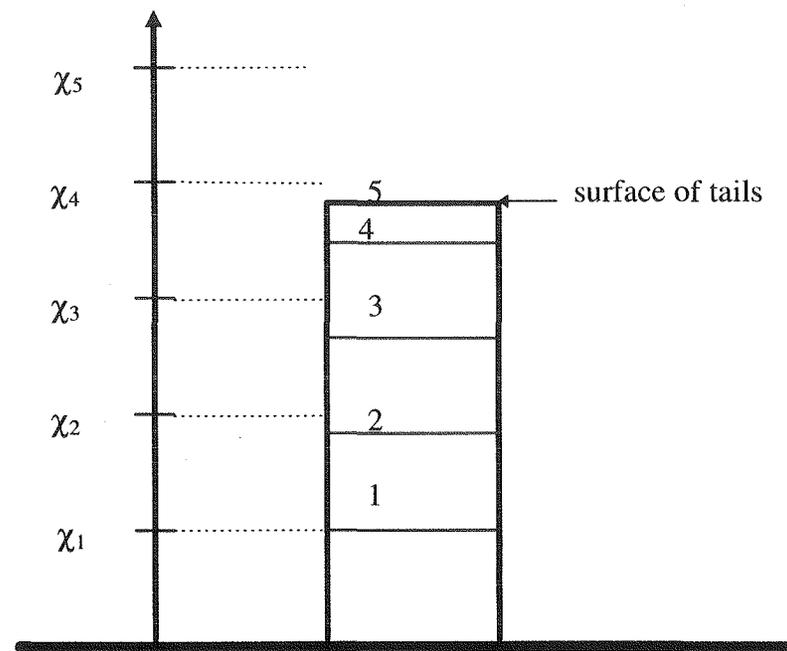
Time (Hr)	Ta (°C)	RH (%)	Wind (km/hr)	Solar R (w/m <sup>2</sup> )
0000	6.5	97	2	0
0100	5.8	98	4	0
0200	5.1	98	1	0
0300	4.6	99	1	0
0400	4.3	99	3	3
0500	4.2	99	2	29
0600	4.5	90	2	80
0700	5.6	98	4	212
0800	7.3	90	4	354
0900	8.8	74	5	527
1000	10.4	65	5	548
1100	11.5	57	4	540
1200	12.8	49	4	684
1300	14.1	37	4	739
1400	15.0	36	3	773
1500	16.2	34	4	674
1600	15.9	32	4	492
1700	16.2	33	7	349
1800	15.9	34	6	219
1900	15.4	39	4	143
2000	14.3	53	1	62
2100	12.7	73	0	17
2200	10.4	83	0	1
2300	8.8	91	0	0
2400	7.1	94	2	0

Table 5.0.2 The atmospheric conditions for a typical day in July (Fort McMurray).

Time (Hr)	Ta (°C)	RH (%)	Wind (km/hr)	Solar R (w/m <sup>2</sup> )
0000	12.1	78	0	0
0100	10.1	95	4	0
0200	8.9	97	1	0
0300	8.1	98	0	0
0400	7.6	98	0	0
0500	6.6	98	0	22
0600	7.1	100	0	107
0700	9.4	98	2	234
0800	12.4	92	2	377
0900	15.8	74	1	515
1000	18.6	59	4	639
1100	19.7	42	4	733
1200	20.9	36	5	796
1300	21.6	32	5	819
1400	22.3	31	5	792
1500	23.1	29	5	723
1600	23.7	28	4	626
1700	24.2	28	4	493
1800	24.1	27	3	338
1900	23.8	27	2	223
2000	23.2	37	0	100
2100	19.9	59	0	21
2200	15.4	69	0	1
2300	12.3	82	0	0
2400	11.2	91	0	0



(a) ( $t=0$ )



(b) ( $t > 0$ )

Figure 3.2.1. Eulerian (spatial) coordinate system

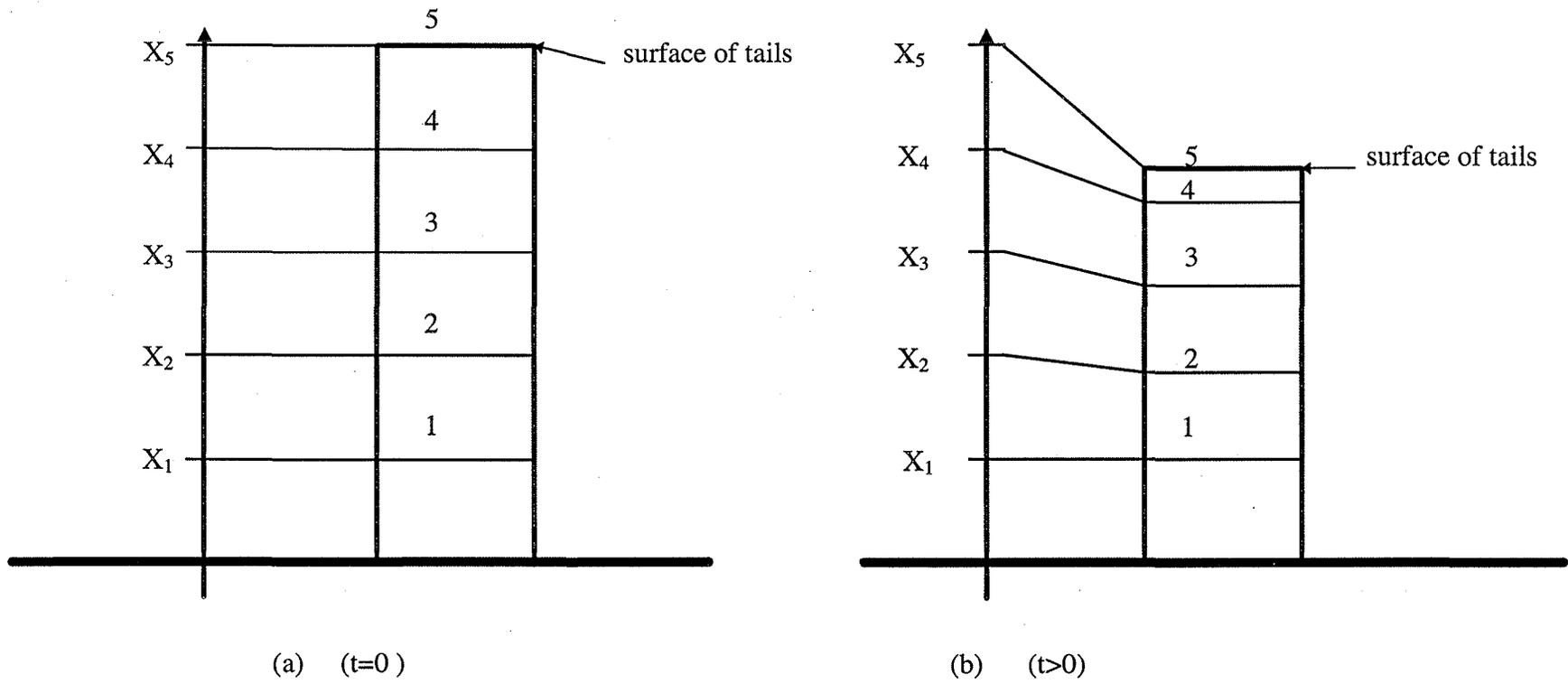


Figure 3.2.2. Lagrangian (material) coordinate system.

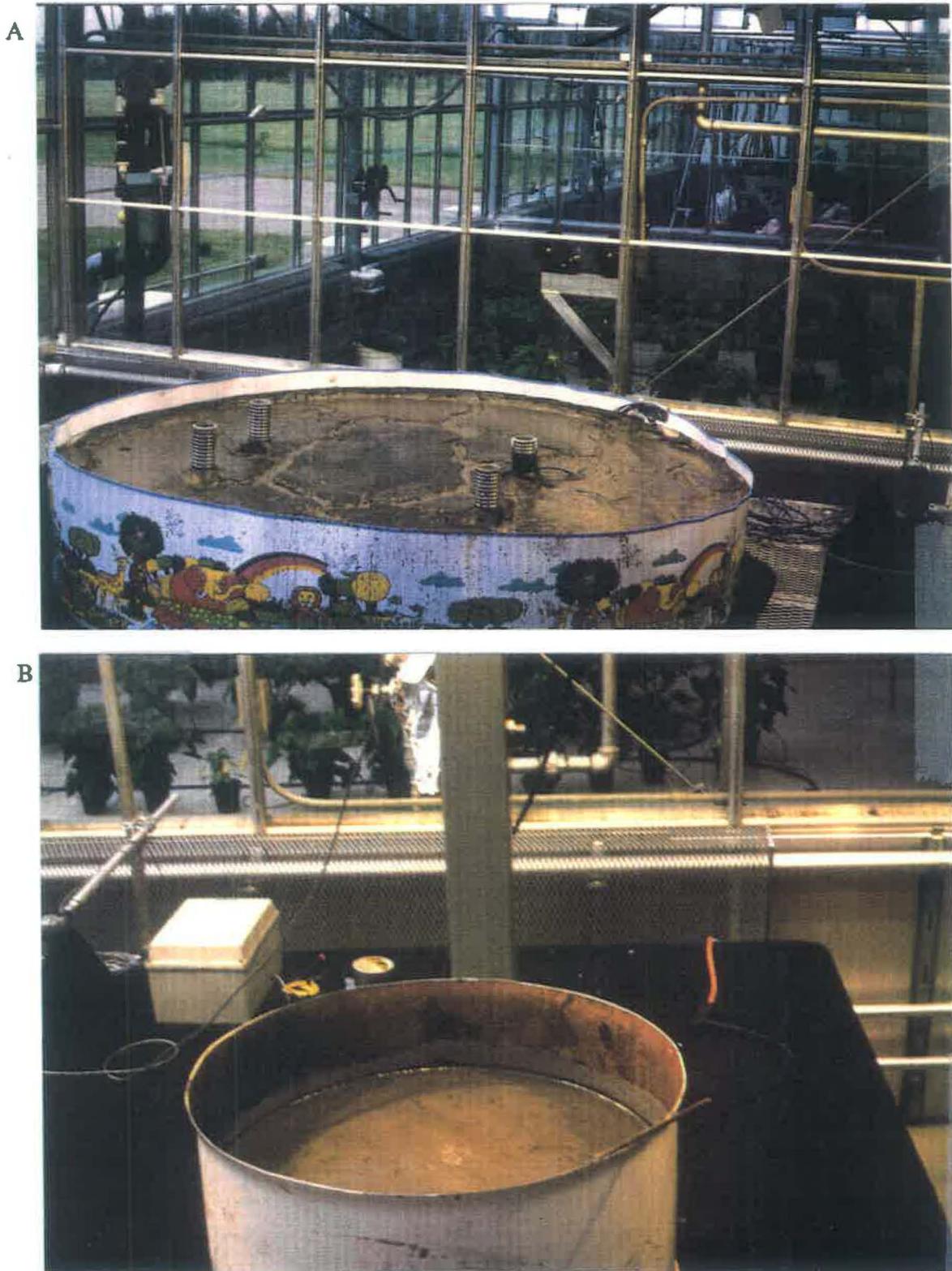


Figure 4.1.1 Evaporation apparatus: A, evaporation apparatus, 1.83 m in diameter and 0.38 m height, used to measure solid profile during evaporation process; B, evaporation pan, 0.6 m in diameter and 0.3 m height, used to measure maximum evaporation demand.

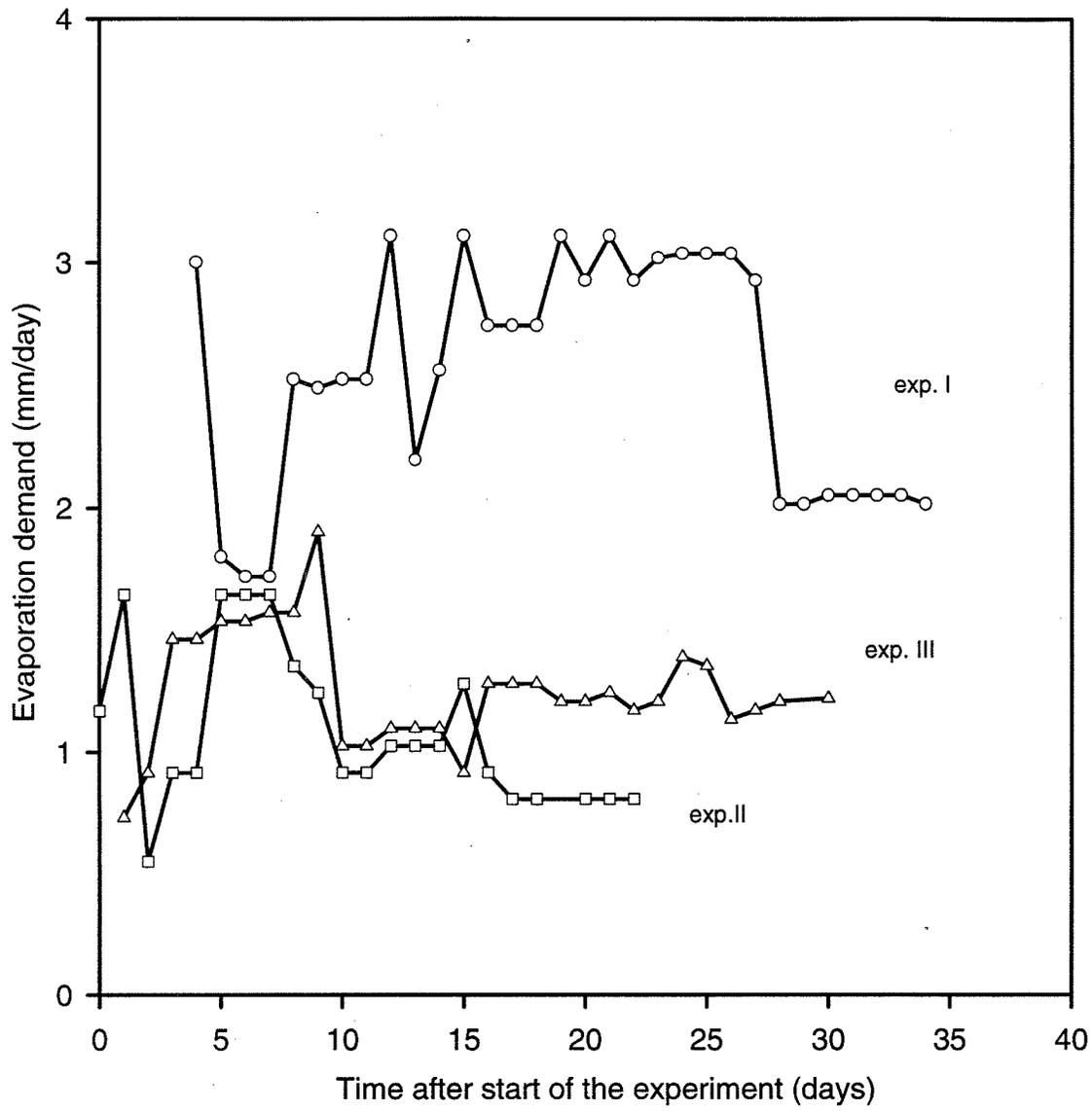


Figure 4.1.2. Evaporation demand during the experimental period.

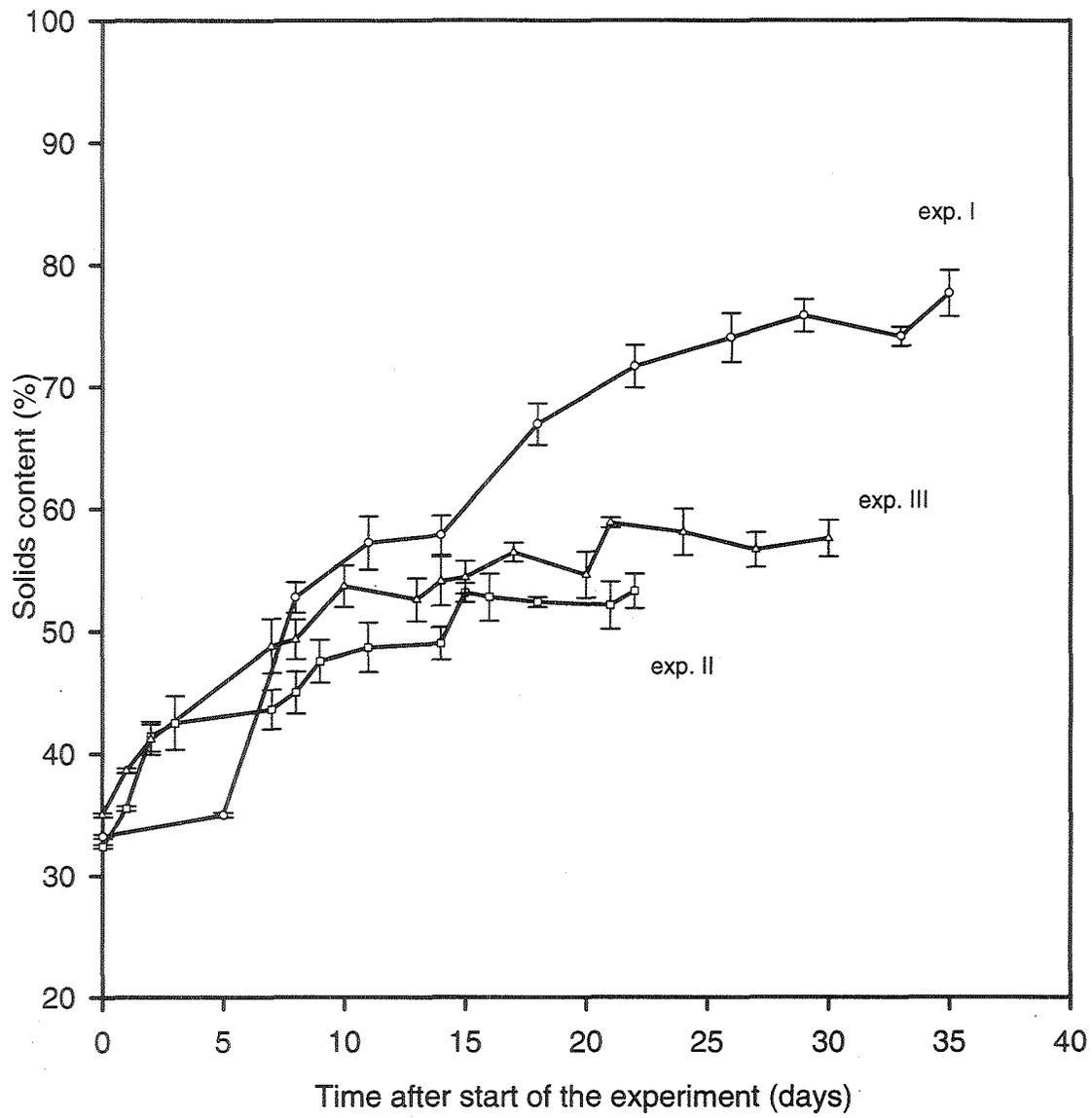


Figure 4.2.1 Solids content of the top 1 cm. The bars represent the range of measurements.

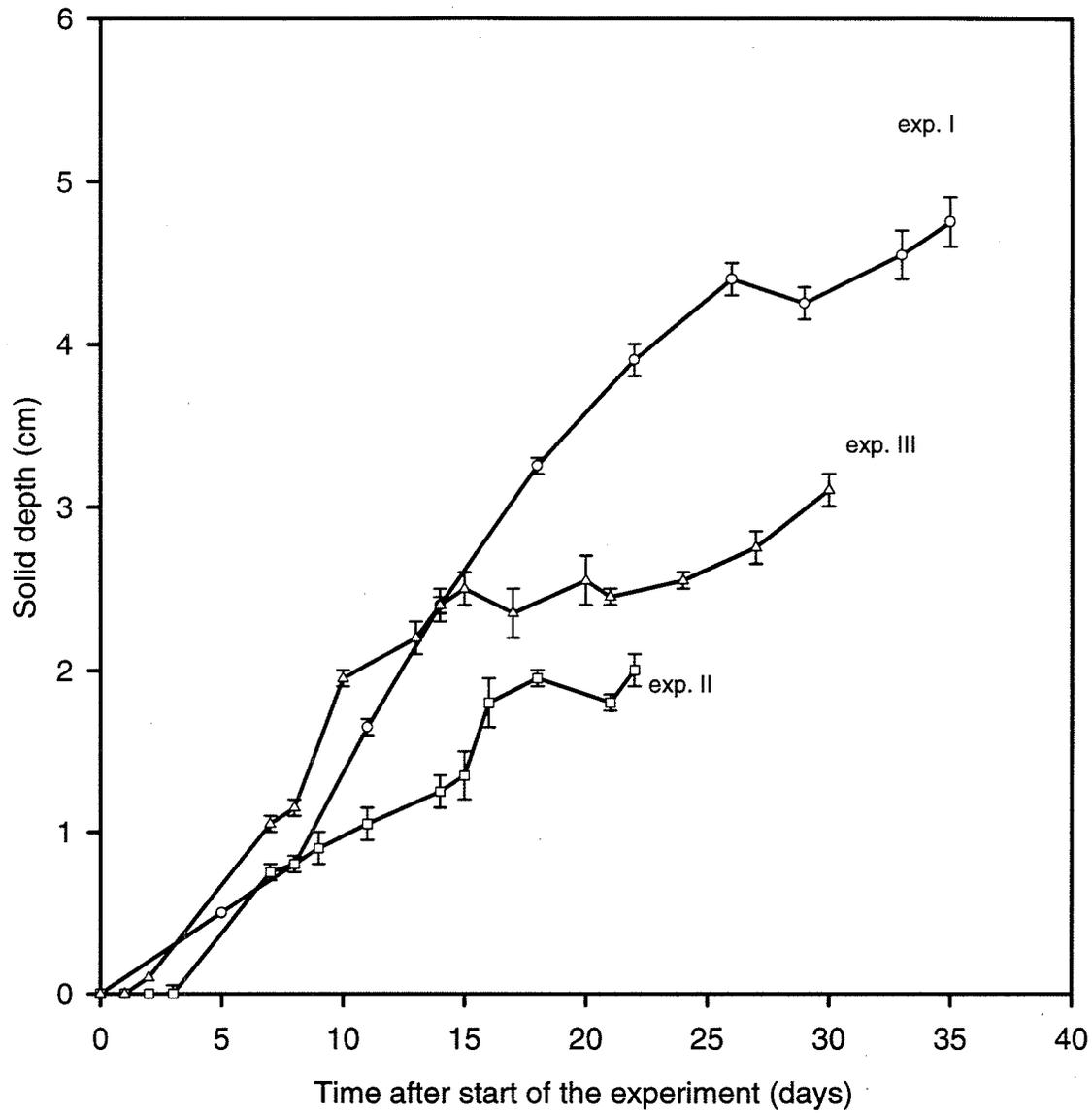


Figure 4.2.2. Crust depth. The bars represent the range of measurements.

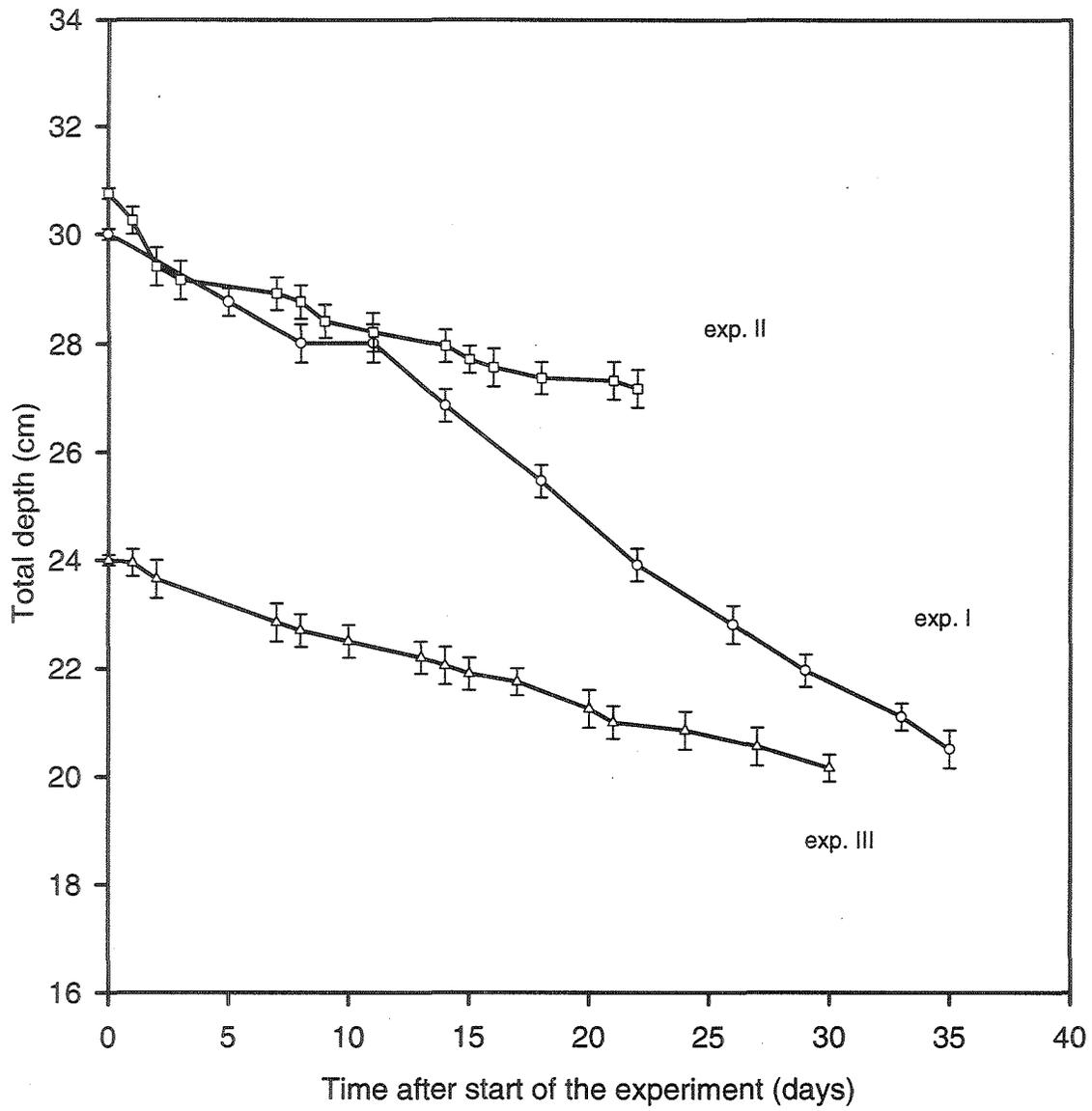


Figure 4.2.3. Total depth of fine tails. The bars represent the range of measurements.

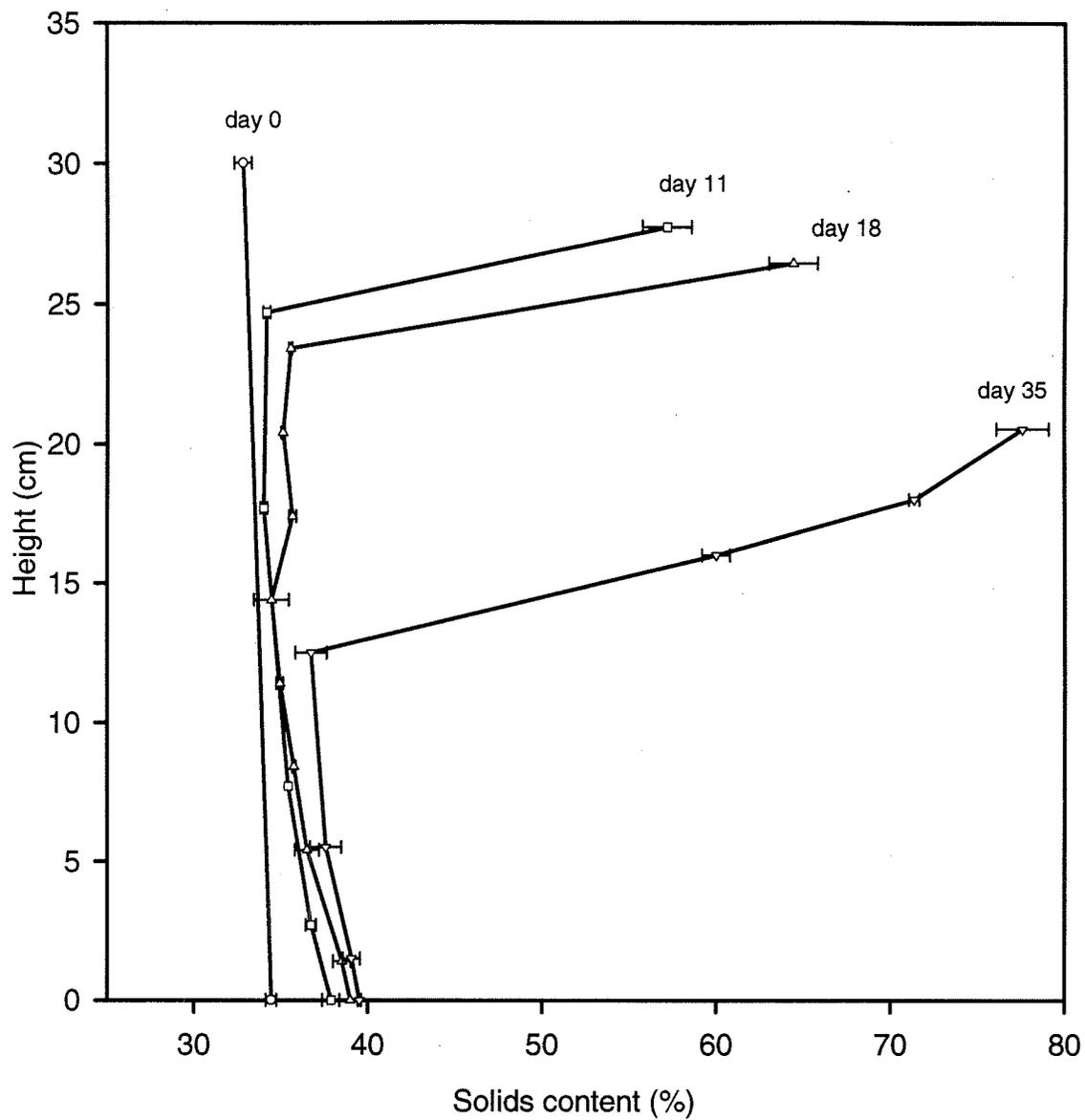


Figure 4.2.4. Solids content profile of experiment I. The bars represent the range of measurements.

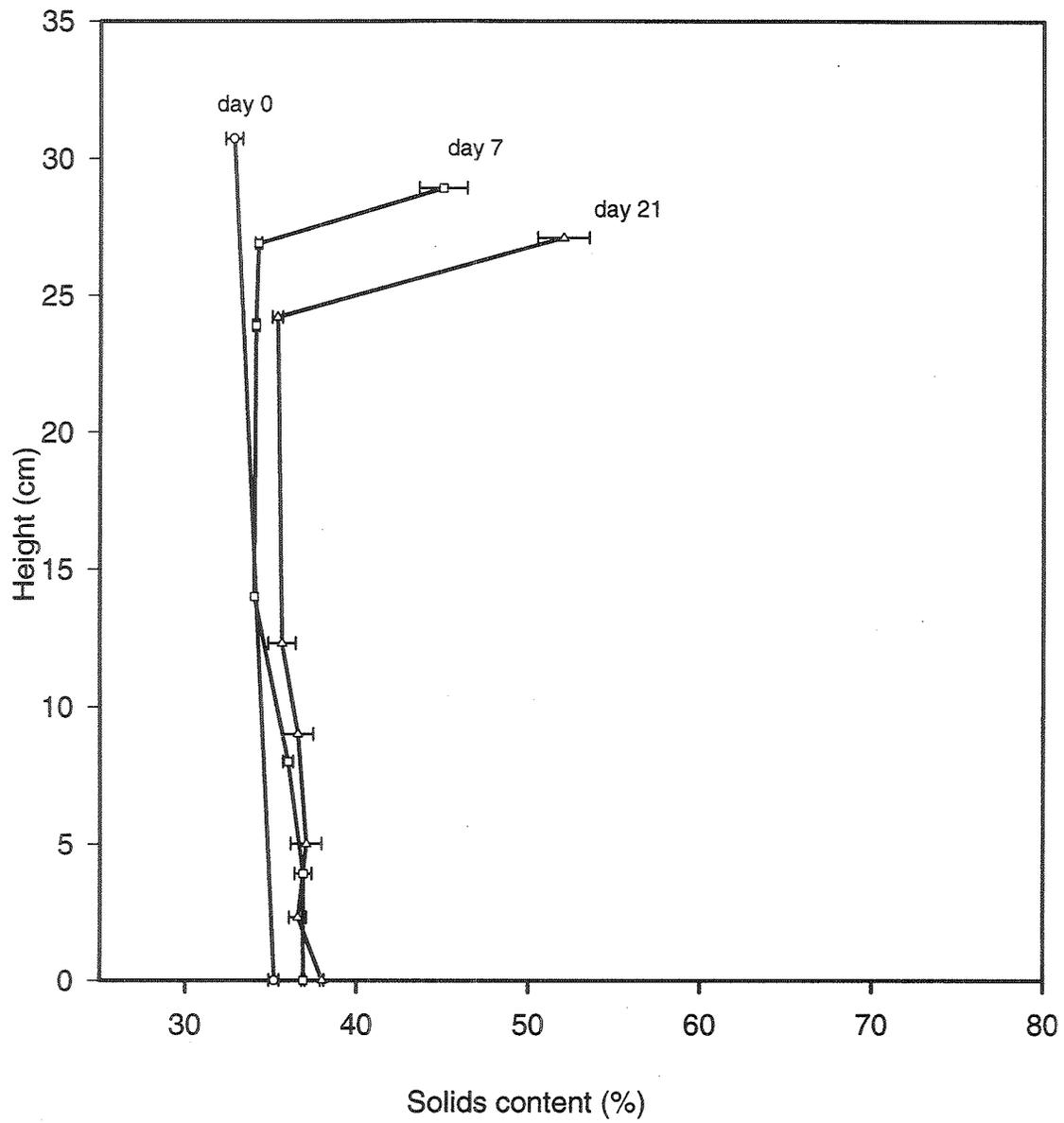


Figure 4.2.5. Solids content profile of experiment II. The bars represent the range of measurements.

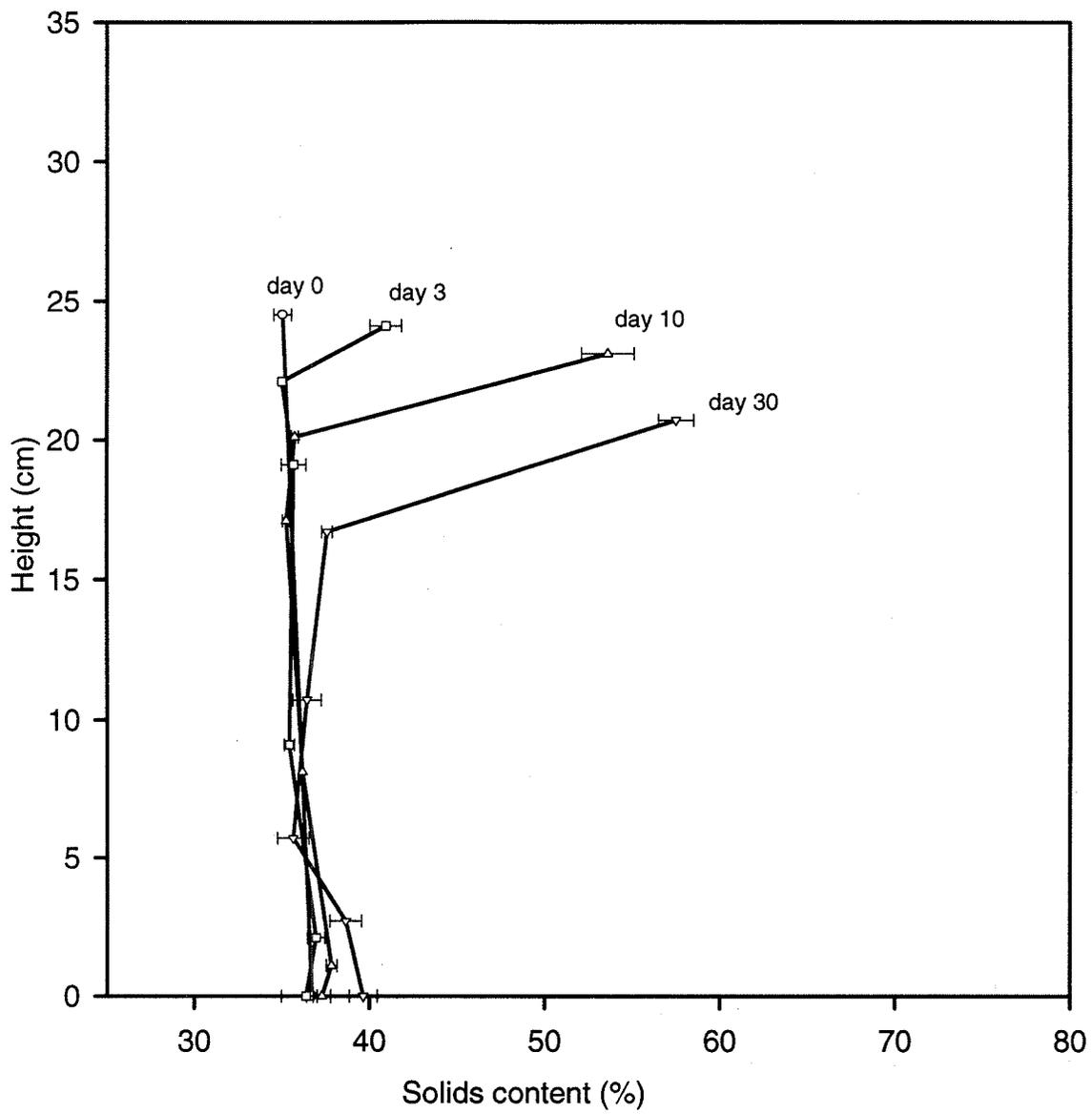


Figure 4.2.6. Solids content profile of experiment III. The bars represent the range of measurements.

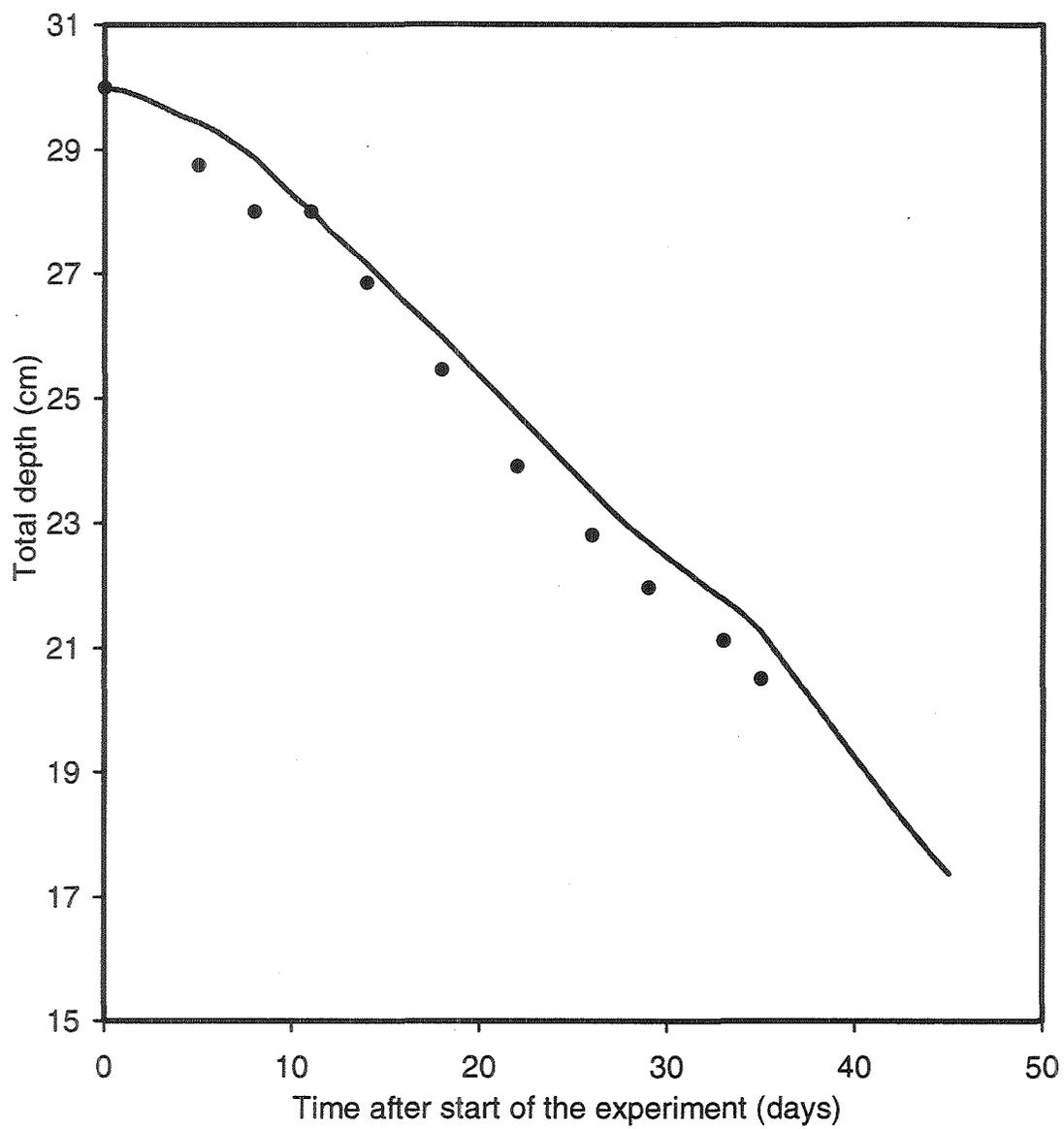


Figure 4.3.1 Simulated (—) and measured (●) depth of fine tails as a function of time. Experiment I.

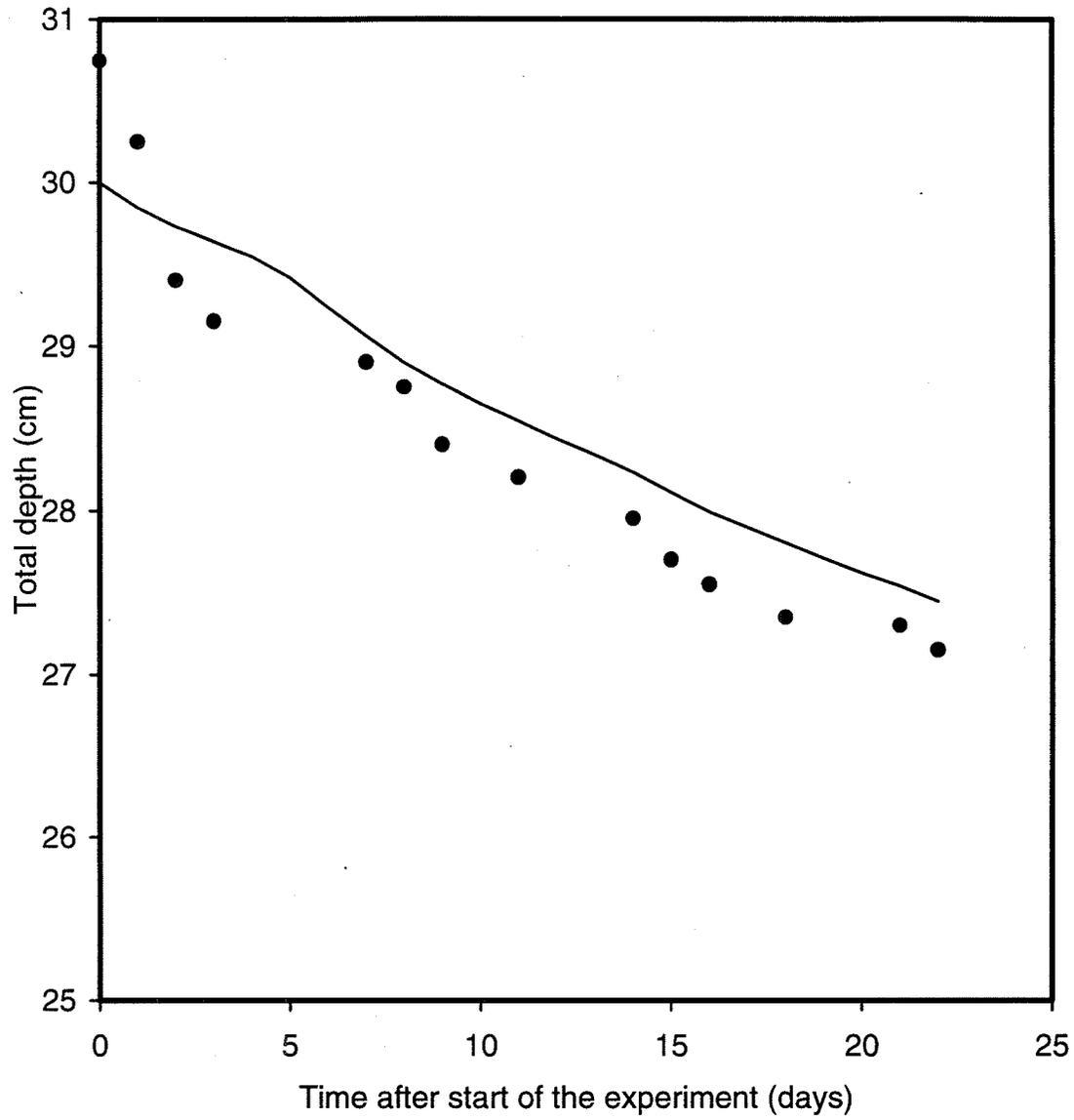


Figure 4.3.2 Simulated (—) and measured (•) depth of fine tails as a function of time. Experiment II.

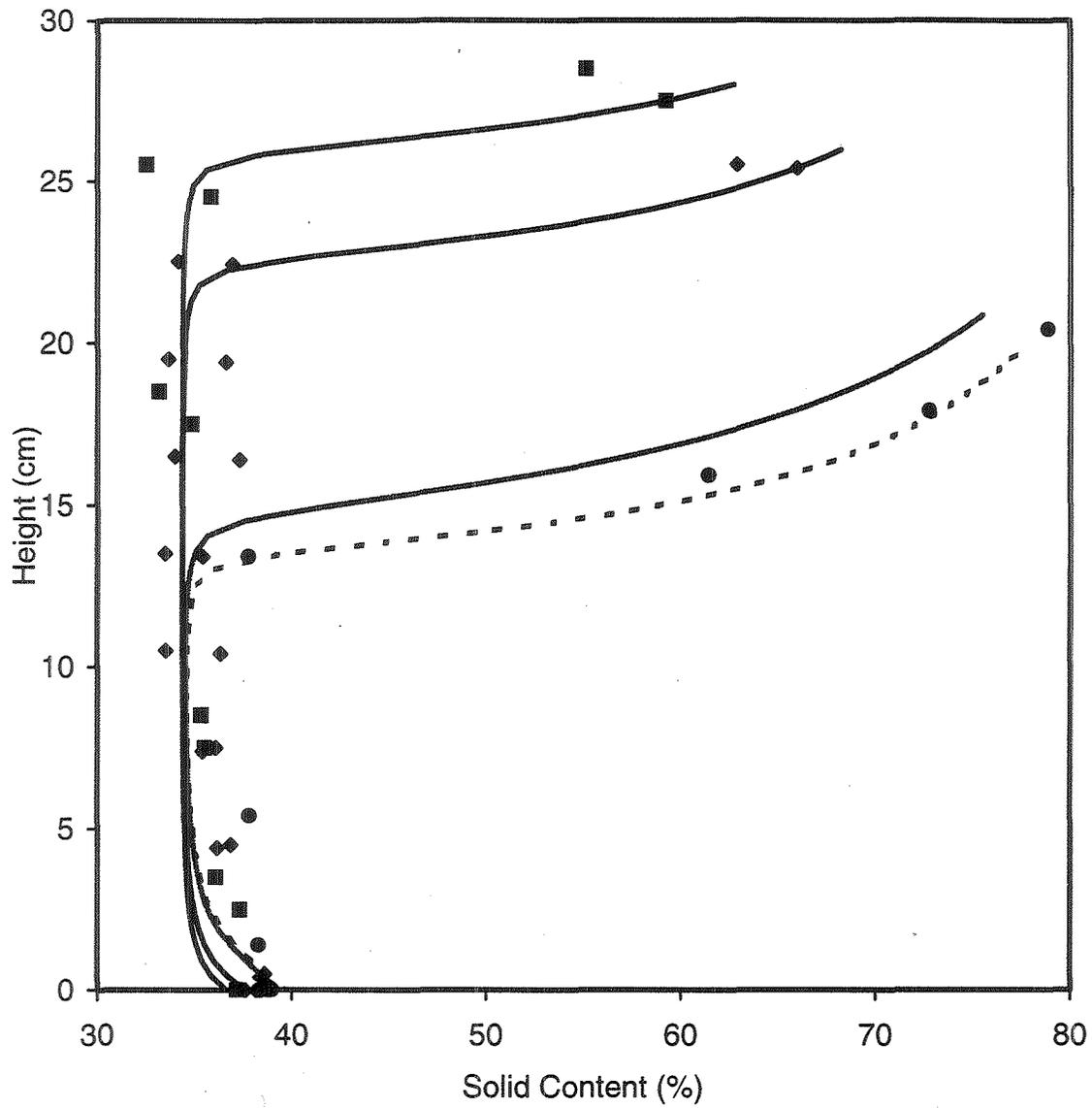


Figure 4.3.3. Simulated (—) and measured solids content profile at  $t=11$ (■),  $t=18$ (◆),  $t=35$ (●). The dashed line is predicted value at day 38. Experiment I.

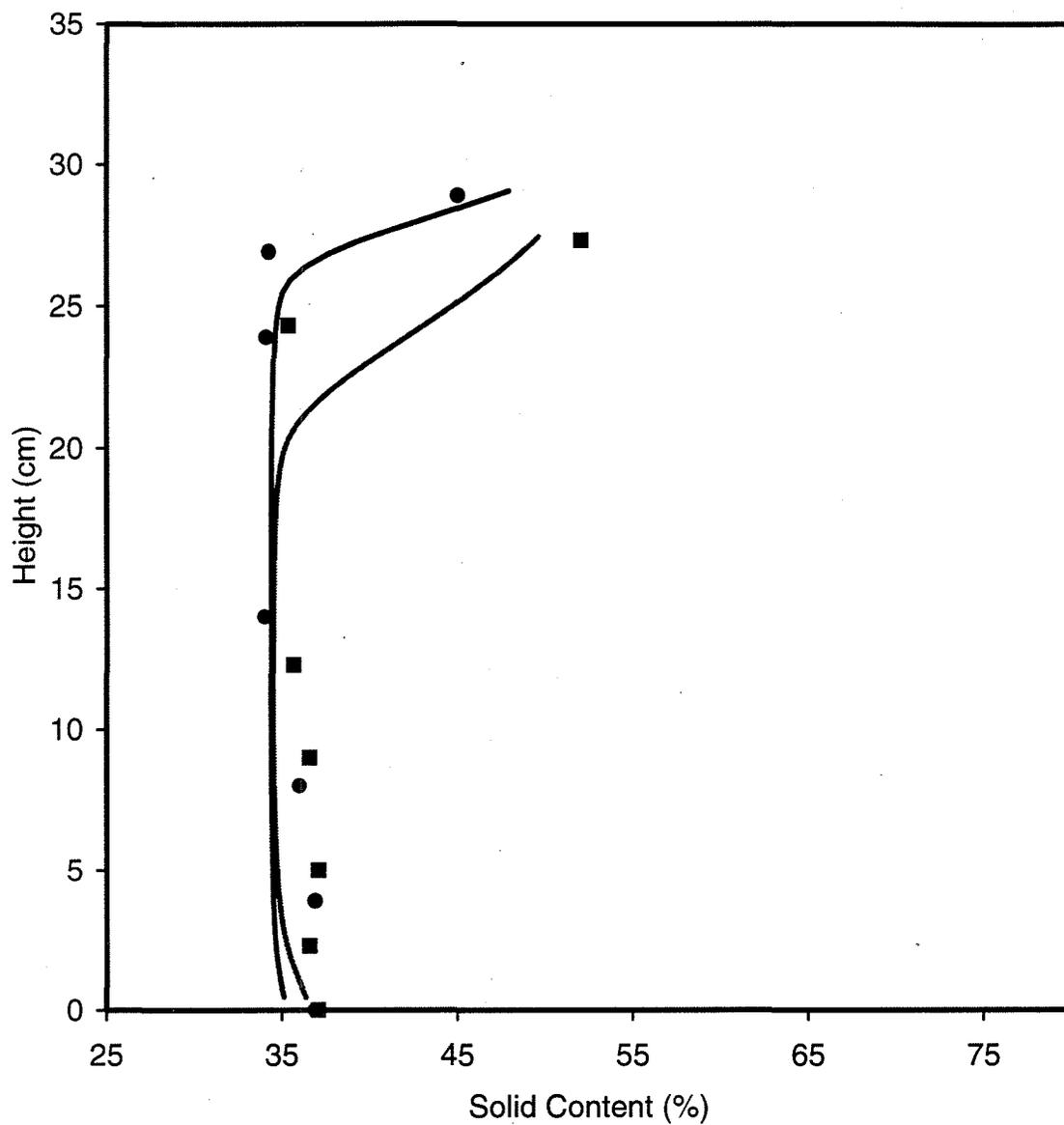


Figure 4.3.4. Simulated (—) and measured solids content profile at  $t=7$ (●),  $t=21$  (■), different times during evaporation. Experiment II.

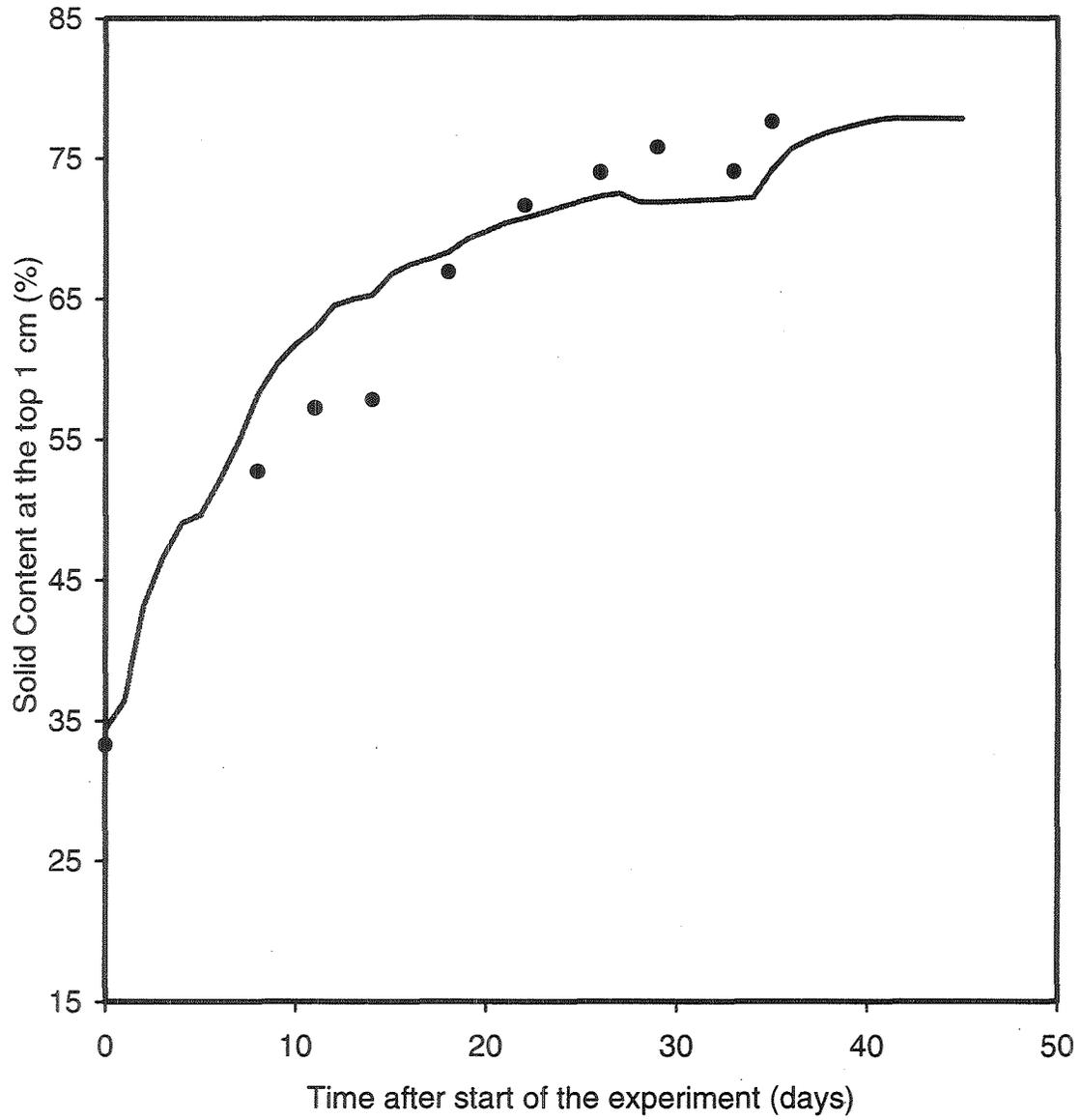


Figure 4.3.5. Simulated (—) and measured (•) solid content at the upper surface (1 cm) of the crust as a function of time. Experiment I.

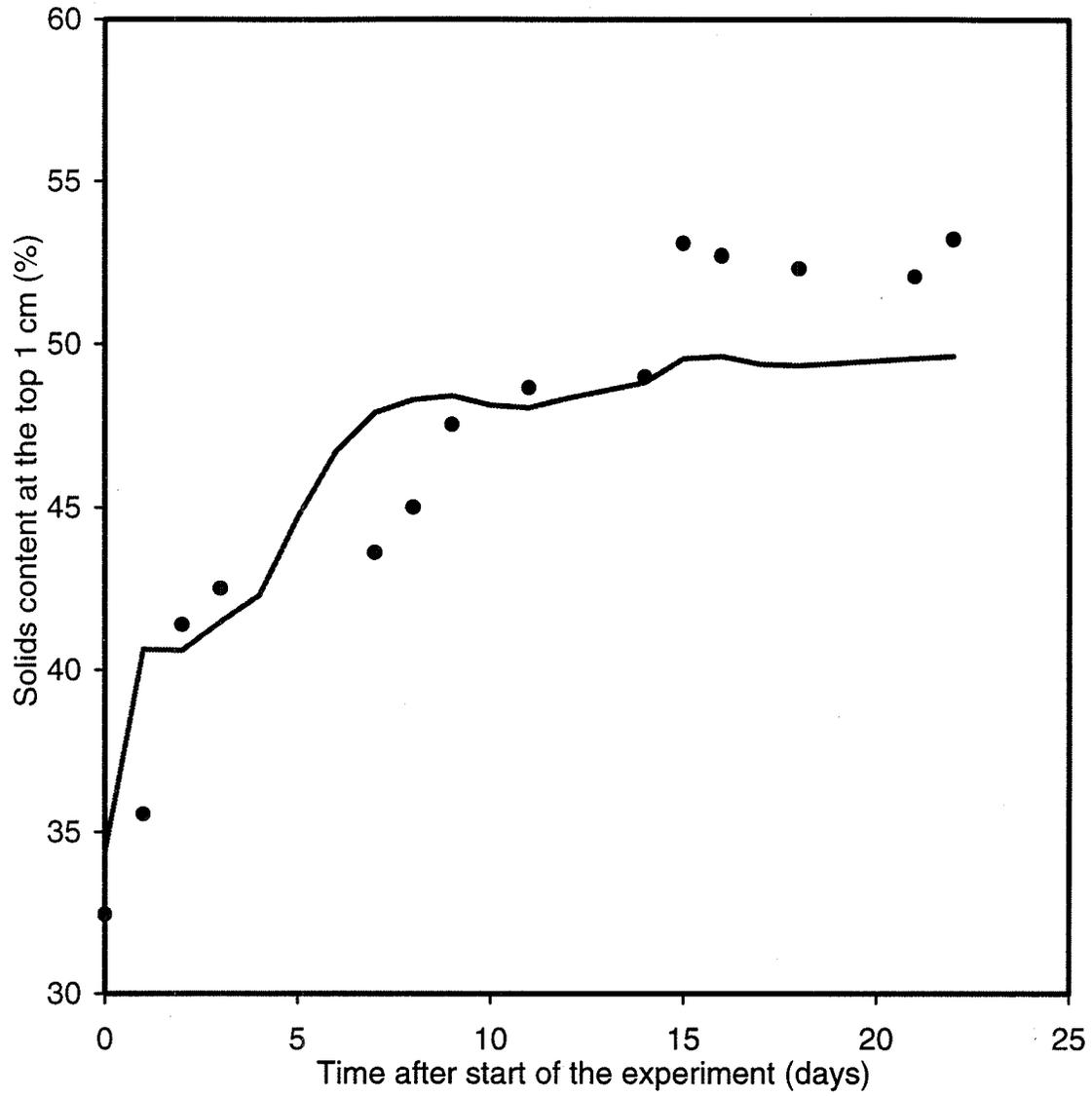


Figure 4.3.6. Simulated (—) and measured (•) solid content at the upper surface (1 cm) of the crust as a function of time. Experiment II.

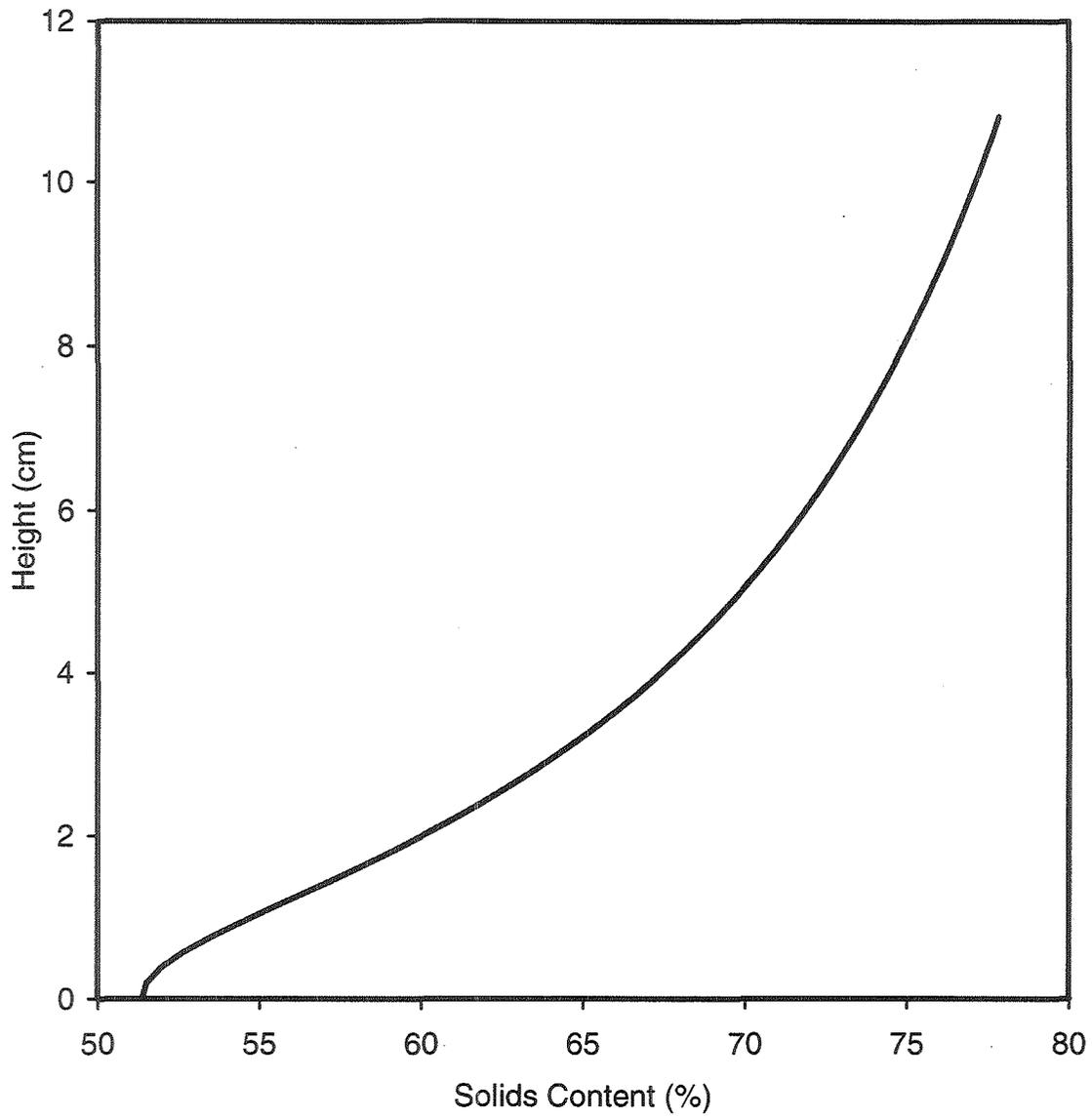


Figure 5.0.1. Final solids content profiles after drying 37 days in June and 27 days in July.

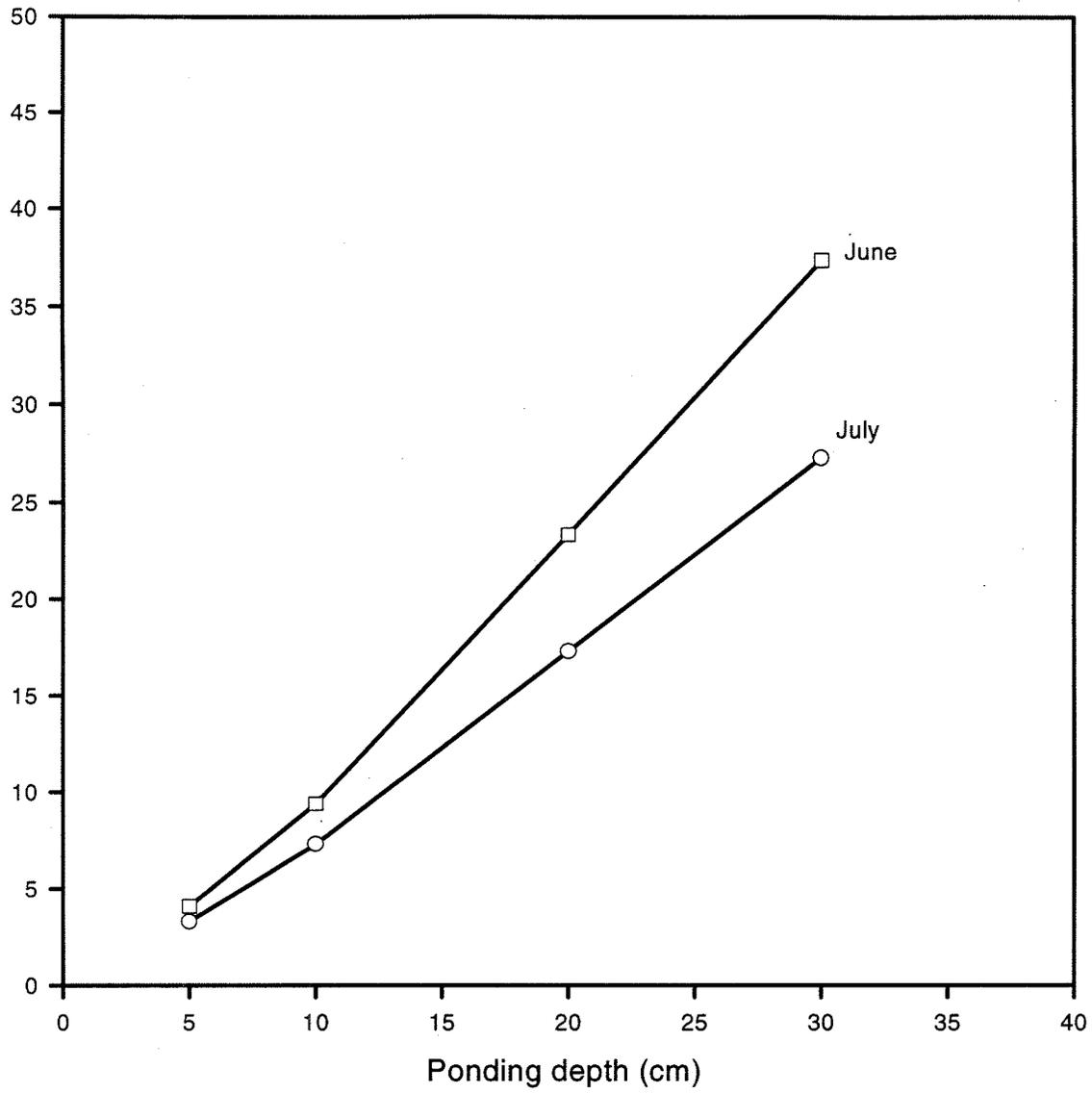


Figure 5.0.2. The time required to complete the drying process as a function of initial ponding depth in June and July.

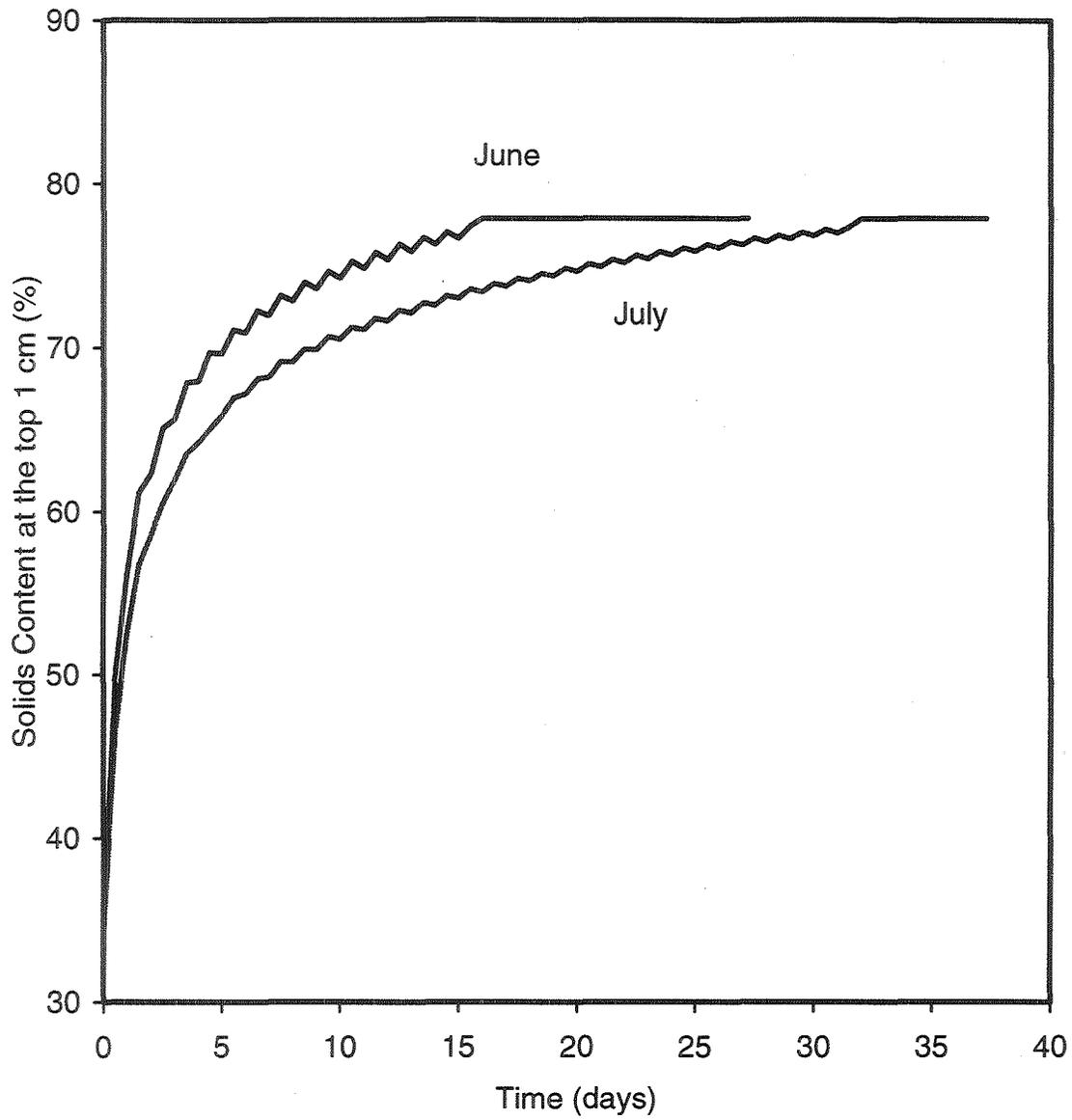


Figure 5.0.3. Solids content at the surface as a function of time for the initial ponding depth of 30 cm in June and July.

Appendix I:

EVAP: The Numerical Analysis of Evaporation From Fine Tails User's Manual

Heidi Neiman and Yongsheng Yeng



## TABLE OF CONTENTS

	PAGE
LIST OF FIGURES .....	iv
LIST OF TABLES .....	iv
1. INTRODUCTION .....	1
2. HOW TO RUN THIS PROGRAM .....	1
2.1 In the DOS .....	1
2.1.1 Prepare The Input Files .....	1
2.1.2 Run The Program .....	5
2.1.3 Examine The Results .....	5
2.2 In the Excel .....	5
3. PROGRAM ORGANIZATION AND COMPILATION .....	6
3.1 Organization .....	6
3.2 Compilation .....	8
4. PROGRAM VARIABLE DESCRIPTIONS .....	12
5. SUBROUTINE/FUNCTION LISTING AND DESCRIPTION .....	17
APPENDIX A: GETDAT MANUAL	
APPENDIX B: FORMATION OF EVAP-E	
APPENDIX C: NUMERICAL PROCEDURES - THE LAGRANGIAN INTERPOLATION POLYNOMIAL	

## LIST OF FIGURES

	PAGE
Figure 3.1 Simplified flow chart illustrating the major steps in the execution of program EVAP .....	9
Figure 3.2 Flow chart illustrating the major steps in subroutine ITER.....	10
Figure 3.3 Flow chart illustrating major steps in subprogram WATER.....	11

## LIST OF TABLES

	PAGE
Table 2.1-1 Sample Raw Data Input File.....	3
Table 2.1-2 Sample Boundary Condition Input File.....	4
Table 5.1-1 General Alphabetic Listing of Subprogram Names. ....	18

## APPENDIX I - 1

### 1. INTRODUCTION

This manual is in regards to the use and organizations of computer program. Some knowledge of FORTRAN would be helpful, but is not necessary. The program can be run either in the DOS or in the Microsoft EXCEL. In the DOS environment, the input parameters required are supplied to the program via three input files: a program supplied and two user supplied files specified at run time. Program output is written into a file named "raw.out". This file then in turn be used as input for program GETDAT, which allows the user to examine specific output of interest. In the Excel, the input parameters are supplied mainly through worksheet. The output data are also generated to worksheet. The detailed translations from the FORTRAN to C, which allows to run the program in Excel is in Appendix A. Program variable names have been selected to share a mnemonic relationship with the model parameters they represent.

### 2. HOW TO RUN THIS PROGRAM

#### 2.1 In the DOS

##### 2.1.1 Prepare The Input Files

Three separate input files are required for program operations. All these files must present in the same directory as the program. The first one, entitled INPUT is supplied with the program, and once set should not need to be altered. It contains parameters controlling the internal workings which specify changes in time step size and convergence criterion. The second and third files are user supplied. They dictate the data parameters for a particular situation to be examined. They are a raw data file and a boundary conditions data file. The two files' names must have the same prefix and end with '.raw' and '.bc'. For example, they could be called 'my.raw' and 'my.bc'. This is so that when running the program you need only to specify the prefix (in this case 'my') instead of typing in the names of both of your files.

All input data files will be given with the program, with the parameters set for a certain situation. If these files should be lost, they can be reconstructed by examining the code in subroutine INPUT, which calls the data files and reads from them one variable at a time.

## APPENDIX I - 2

Remember to put in a comment line everywhere the code does a read on variable TMPC. This comment line needs to be surrounded by apostrophies.

Program EVAP operates with time units of days. In regards to the other variable units, it doesn't matter as long as the input data is consistent throughout.

Program Supplied Input File: The file, entitled INPUT, consists of FIVE variables: MINI, MAXI, MAXP, SAME and WSMTH on separate lines in the mentioned order. The first three control the iterative process by either causing a change in the size of the time step taken for the next step according to present speed of convergence, or causing the present time step to be recalculated from its beginning using a smaller time differential (variable DT) if the difference of values between two iterations is too large and may cause parameter oscillations to occur. The variable SAME is the largest percentage difference between two iteration's parameters that will allow the process to act as if convergence has occurred.

MINI and MAXI are the minimum and maximum number of iterations taken that will affect the next time step. If the amount of iterations which actually occurs is less than MINI, the next time differential between steps will double. Conversely, if the number of iterations is greater than MAXI, the next time differential will be halved. If overflow problems occur while running the program, altering these variables may eliminate them. Try lowering the value of MINI and tightening the difference between MINI and MAXI. Conversely, if the program is running well, widening the difference between MINI and MAXI while increasing MINI may make the program run faster. MAXP is the maximum percentage difference allowed between two iterations before the present time step will be recalculated as described above.

The last variable in the file is WSMTH which is used the subroutine SMOOTH. The subroutine "smooths out" the array which is passed to it. Basically, it takes the choppiness out of the data incurred by the solving method to give smooth graphs. WSMTH is a factor of weight with a value between zero and one. Zero means that no smoothing occurs, and a one means total smoothing.

User Defined Raw Data Input Data File: The data file consists of unformatted input separated by comment lines dictating the data to follow, with each parameter occupying a separate line. Descriptions of the program variables are detailed in section 1.3 in alphabetical order. Table 2.1-1 is a partial example of a raw input file.

## APPENDIX I - 3

Table 2.1-1 Sample raw data input file

```
'WATER INPUT: RHOG'  
1  
0.5  
'TEMPERATURE INPUT: T10, T20, LV, CW'  
30  
30  
2466.2  
1  
'SOLID INPUT: RHOS, RHOW, SO'  
2.65  
1  
0.15  
'GENERAL INPUT: DEPTH, ENDT, FREQ, NNODES, DT, DTMAX, DTMIN'  
10  
1  
0.1  
11  
0.0005  
0.1  
0.00001  
'ROUTINE SUBDVT'  
0  
0  
0  
0  
0  
.  
..  
.  
.  
'ROUTINE SUBXS'  
0.2  
0.25  
0  
0  
0
```

The lines surrounded by apostrophes are comment lines, stating the list of variables to immediately follow. Those sections delimited by ROUTINE such as in 'ROUTINE DVT', for example, contain numbers to be entered into "subroutine arrays"; up to five numbers can be

## APPENDIX I - 4

entered into the arrays. The numbers following 'ROUTINE DVT' are needed to evaluate the formulas in function DVT and are stored in array SUBDVT. All subroutine array names start with the three letters SUB. Therefore, to change a formula's constant parameters, find the appropriate routine heading in the raw input file and change the array.

*User Defined Boundary Conditions File:* This file contains a table of values pertaining to weather used to calculate the boundary conditions for temperature and the volume fraction of liquids. Table 2.1-1 below is a sample boundary condition input file.

Table 2.1-2 Sample boundary condition input file

TIME	TA	RH	WIND	SN'
0	25	0.5	1	1300
0.5	30	0.502	13	1305
1.3	20	0.03	2	1209
15	25	0.5	1	1340

The first column is the time from the start of running the program. The second column is air temperature in degrees Celsius, the third relative humidity, the fourth wind, and the fifth is net radiation. The last row of data **must be** for a time greater than or equal to the end time of the program run time that is specified in the raw data file. The data entered should be ordered increasing in time from time zero. For calculations which occur at times not specified in the data file, a linear interpolation of the surrounding data takes place to approximate the boundary conditions.

These files must be in the format depicted by section 3. Of particular importance, the variable NNODES in the user supplied raw data file must be less than or equal to the parameter MNODES in file vars.for, which declares all variables used by the program. As well, the end time specified in the raw data file must be less than or equal to the time entered in the last row of the boundary conditions file. The boundary conditions file must also have an entry for time is equal to zero and the times must be in increasing order. Finally, variable FREQ in the raw data file specifies the frequency of printout of the data to the output file. If FREQ is small, the output file created may become quite large.

## APPENDIX I - 5

### 2.1.2 Run The Program

Program EVAP creates a row output file named Raw.out. If you have already run EVAP and a file raw.out exists, then before running EVAP the existing raw.out needs to be copied to another file name.

Type EVAP will start the program. You will then be prompted to type in the prefix of the boundary conditions input and raw data input files that you have previously created. When typing that in, it needs to be typed surrounded by apostrophes. For example, if your files are called my.raw and my.bc, you would type 'my'. The program will stop automatically when the end time specified in the raw data file is reached. The time it is working on as well as the value of the time steps that are being taken will be periodically written to the screen to indicate its progress. The program may take considerable time to run, depending upon the input. Please account for this.

### 2.1.3 Examine The Results

To examine the results, run program GETDATA which is described in Appendix A.

## 2.2 In the Excel

1. Run Excel, load the sample EXAMPLE.CSV file, {File|SaveAs} TEST.CSV. Make any modifications you want to the simulation parameters. Prepare the Boundary conditions file using the same format as Table 2.1-2 and place it in the same directory as EXAMPLE.CSV file. Then close TEST.CSV. EXAMPLE.CSV is a standard formation for input file.
2. From the DOS prompt type "C:\> evap-E TEST.CSV". Switch back to excel, hit {File} and TEST.CSV should be #1. PageDown a couple of times to see the data.
3. To graph the data, load the sample EVAPGRAF.XLS, select the Graph Data sheet, switch back to TEST.CSV, select All the data and copy it into the GraphData sheet of

EVAPGRAF.XLS. Selecting any of the pre-prepared Graphs should automatically display the updated data. To save, {File|SaveAS} WHATEVER.XLS

Note: work is in progress on an Excel macro to automate the above procedure.

### 3. PROGRAM ORGANIZATION AND COMPILATION

The code was written to follow ANSI FORTRAN 77 format. Since FORTRAN 77 character strings are delimited by apostrophes, when a file name is prompted from the keyboard it must be entered as 'filename'.

#### 3.1 Organization

Program EVAP is designed to integrate equations that describe evaporation from a free water surface, namely, equations of water transport, heat transport, vapor transport, and solid phase consolidation. The equations are managed via the Crank-Nicolson method, iterating to solve for required criterion at successive moments in time.

The complete set of files needed to compile and run this program include:

1. EVAP.EXE - the executable main program
2. VARS.FOR - used for variable declarations
3. INPUT - program supplied input file
4. a raw data user supplied input file, whose name is specified during run time
5. a boundary condition user supplied input file, also specified at run time
6. GETDAT.EXE - allows the examination of the raw data produced by EVAP.EXE

Program EVAP is structured so that each major operation is divided into its own subroutine or function. The main program's overall structure and procedure is:

1. retrieve input
2. initialize arrays and other variables
3. echo initial values
4. evaluate functions for partial derivatives of the volume fraction of liquids, and temperature (THETA, TEMP) using the data for time equals zero

## APPENDIX I - 7

5. start of main loop: do until process is complete
6. start iteration process to find parameters at the next moment in time
7. evaluate functions for partial derivatives of the volume fraction of liquids, and temperature using the recently converged data.
8. output converged data values at this particular time if near specified frequency of output
9. store the temperature at the bottom of the fine tail for this time step
10. increment time step and repeat process from 5.
11. end when analysis is complete for specified length of time

A simplified flow chart illustrating the major steps in the main program can be seen in Figure 3.1.

The main four controlling routines are ITER, WATER, HEAT, and SOLID. Subroutine ITER controls the iteration process, determining when the three central program parameters (THETA, TEMP, SIGMA) are converged, and calling the appropriate routines to continue their development when they are not. WATER, HEAT, and SOLID contribute the newest iteration results for the volume fraction of liquids, temperature, and the one-dimensional compressional stress of the solid phase respectively. For the parameters THETA and TEMP, this is achieved by calling function routines to calculate the partial derivatives with the most recent iteration information, and use these to update their values. This process is briefly explained in the subroutine synopses WATER and HEAT. The parameter SIGMA is updated using an empirical formula dependent upon itself as well as the changing values of water potential and volume fraction of liquids. Data at the boundaries for THETA and TEMP are determined using the boundary condition equations and the Lagrangian interpolation polynomial. (See subroutines BCW and BCH.) At the end of each iteration, all other needed parameters are evaluated using the newest values of THETA, TEMP and SIGMA.

A flow chart for subroutine ITER is given in Figure 3.2. Since subroutines WATER, and HEAT follow a similar structure, a flow chart for WATER will only be given. WATER's flow chart is shown in Figure 3.3. Subroutine SOLID differs from WATER and HEAT in that SIGMA can be calculated directly using an empirical formula.

3.2 Compilation

The program was written in a DOS personal computer environment. If recompilation takes place, the file named VARS.FOR must be present in the same directory as the source code. This file contains all the variable declarations needed by program EVAP. If changes are made to the parameter MNODES in vars.for, then the program GETDAT, which also uses VARS.FOR for its variable declarations, must be recompiled as well.

APPENDIX I - 9

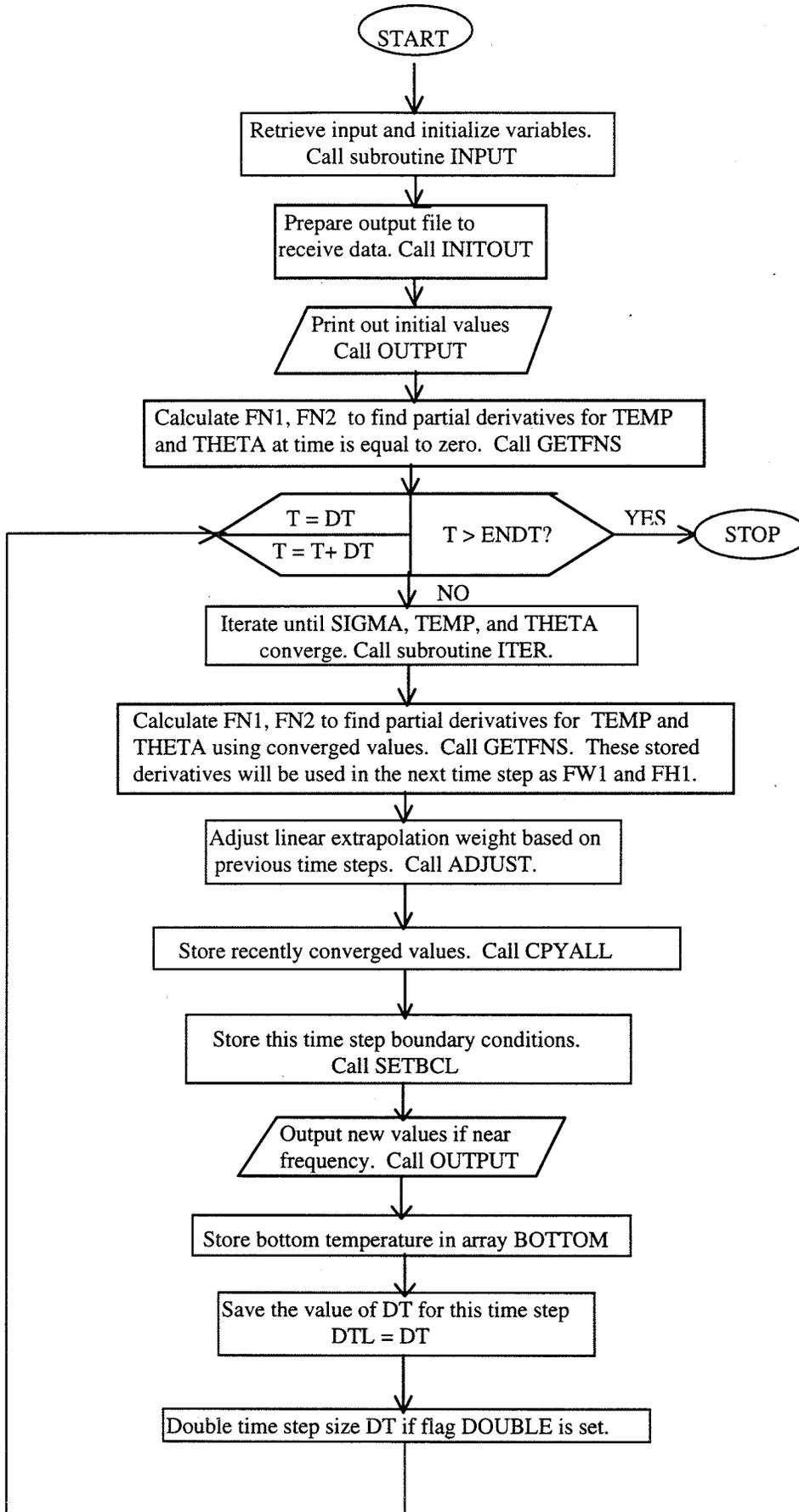


Figure 3.1 Simplified flow chart illustrating the major steps in the execution of program EVAP

APPENDIX I - 10

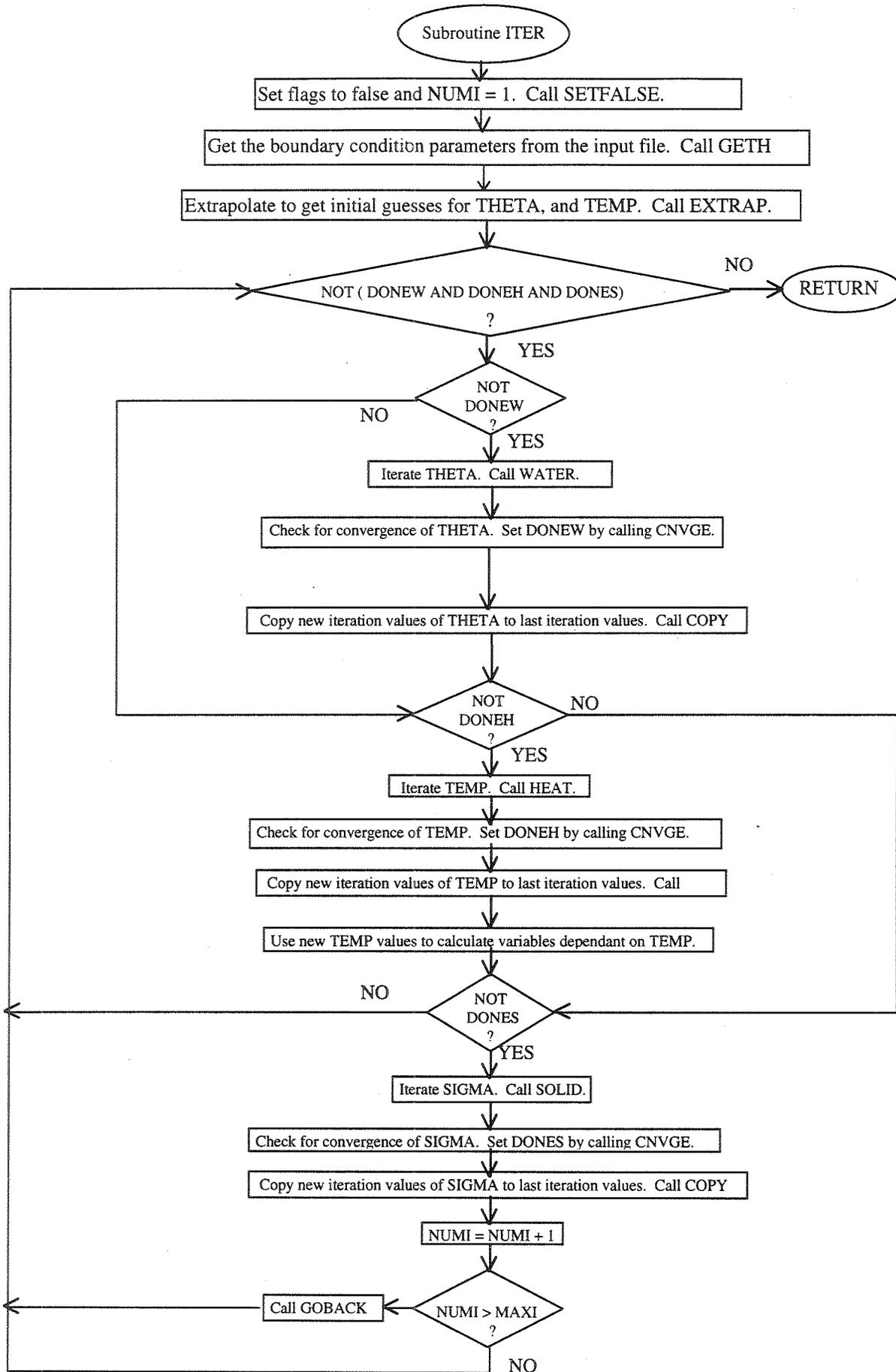


Figure 3.2 Flow chart illustrating the major steps in subroutine ITER

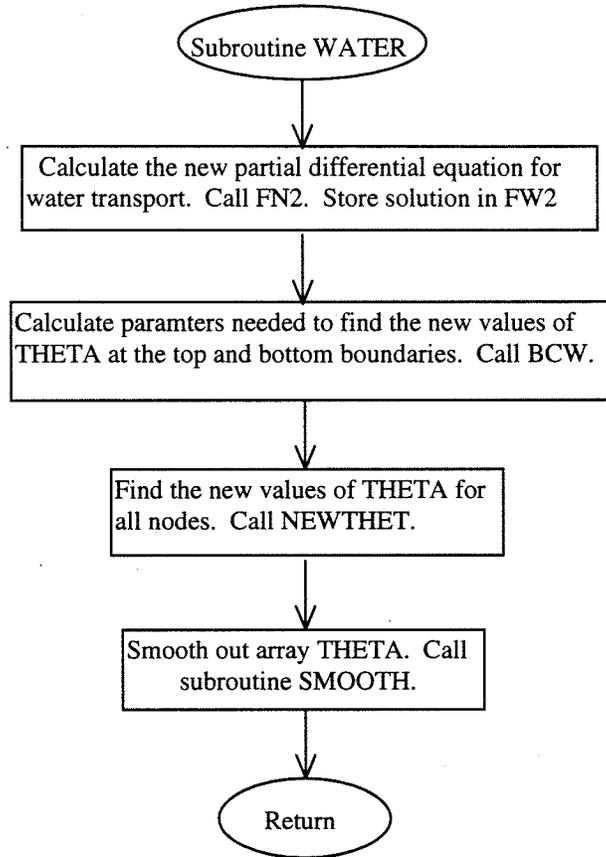


Figure 3.3 Flow chart illustrating major steps in subroutine WATER

## 4. PROGRAM VARIABLE DESCRIPTIONS

A list of the program variables is presented with a short explanation of their uses.

- BOTTOM** - Array containing temperatures at the bottom of the fine tailing pond at different times. This is used to find the bottom temperature boundary condition. The maximum amount of stored temperatures at one time is NUMT.
- DEPS** - Array containing the time derivative of EPS.
- DEPSL** - DEPS from the last completed time step encountered.
- DEPTH** - Total initial depth of the fine tailing pond.
- DFILM** - The change of rate of the depth of water sitting on the surface of the fine tailing pond.
- DOCOPY** - A flag used to indicate whether the copy routine needs to be called. It will not be called if convergence for an iteration is a problem and the program calls GOBACK to try again with a smaller time step.
- DONEH** - A flag used to indicate whether the temperature has reached convergence within an iterative step. It is determined by calling subroutine CNVGE.
- DONES** - A flag used to indicate whether SIGMA has reached convergence within an iterative step. It is determined by calling subroutine CNVGE.
- DONEW** - A flag used to indicate whether water potential has reached convergence within an iterative step. It is determined by calling subroutine CNVGE.
- DOSUMM** - A flag used to indicate whether the subroutine SUMM needs to be called. SUMM is used in the calculation of the bottom boundary condition for temperature. SUMM is only called if the temperature change on the bottom in previous time steps is large enough to warrant the calculation of the integral. It is initially set to false.
- DOUBLE** - A flag used to indicate whether DT needs to be doubled. Initially set to false, it turns true when the number of iterations needed to reach convergence is smaller than the variable MINI set in file INPUT.
- DT** - Incremental value for change in time between consecutive time steps.

## APPENDIX I - 13

- DTL** - DT used in the previous time step
- DTMAX** - The maximum allowable value for DT, the time step size.
- DTMIN** - The minimum allowable value for DT, the time step size.
- ENDT** - Specified total length of time to model the process.
- EXTH** - Determines the percentage weight given to the value of TEMP found at the last time step in calculating its initial guess using an extrapolated value. It ranges between 0 and 1 and is automatically adjusted in subroutine ADJUST depending upon the trend of previous time steps.
- EXTW** - Determines the percentage weight given to the value of THETA found at the last time step in calculating its initial guess for the present time step using an extrapolated value. It ranges between 0 and 1 and it is automatically adjusted in subroutine ADJUST, depending upon the trend of previous time steps.
- FILM** - The depth of water sitting on the surface of the fine tailing pond.
- FILML** - The depth of water sitting on the surface of the fine tailing pond at the previous time step.
- FIRSTH** - Array containing the linear extrapolation of TEMP
- FIRSTW** - Array containing the linear extrapolation of THETA
- FREQ** - Specified frequency the user wants the output written to the file. FREQ must be given in units of days..
- FW1,FH1** - Partial differentials for water and heat transport at distances of X using values from the last converged time step (constant throughout the next time step's iterations).
- HA** - Boundary condition for TEMP. Its usage is described in BCH.
- HAL** - HA saved from the previous time step.
- HB** - Boundary condition for TEMP. Its usage is described in BCH
- HBL** - HB saved from the previous time step.
- HC** - Boundary condition for TEMP. Its usage is described in BCH.
- HCL** - HC saved from the previous time step.
- HD** - Boundary condition for TEMP. Its usage is described in BCH.

APPENDIX I - 14

- HDL** - HD saved from the previous time step.
- HE** - Boundary condition for TEMP. Its usage is described in BCH.
- HEL** - HE saved from the previous time step.
- HG** - Boundary condition for TEMP. Its usage is described in BCH.
- HGL** - HG saved from the previous time step.
- HH** - Heat transfer coefficient of the surface boundary layer.
- HV** - Vapor transfer coefficient of the surface boundary layer.
- IN2** - Character string for the name of the user supplied raw data input file.
- IN3** - Character string for the name of the user supplied boundary conditions input file.
- INDEXT** - A pointer used to keep track of the present index in the circular arrays TIME and BOTTOM.
- J** - Array of the ratio of current/initial soil layer thickness.
- JL** - Array containing the values of J at the previous time step.
- KT2** - Thermal conductivity of underlying material.
- LV** - The latent heat of vaporization. Included in the raw data input file.
- MAXI** - If the number of iterations at convergence exceeds this variable then DT is halved in the next time step's calculations. Set it file INPUT
- MAXP** - Maximum percentage difference that can occur between values of a variable in two consecutive iterations before DT is halved. The present time step is reinitiated at the new lower time caused by the halved DT.
- MAXPO** - The maximum value of water potential seen at the top boundary.
- MINI** - If the number of iterations at convergence is less than this variable then DT is doubled for the next time step. Set in file INPUT.
- MINPO** - The minimum value of water potential seen at the top boundary.
- MNODES** - The maximum allowed number of nodes as set in file vars.for. Changing this parameter required recompilation of EVAP and GETDAT.
- N** - Viscosity of water
- NNODES** - Number of nodes used in the calculations for the fine tail pond.
- NUMI** - Count of the number of iterations occurring in a time step.

## APPENDIX I - 15

<b>NUMT -</b>	Dimension of arrays <b>BOTTOM</b> and <b>TIME</b> . Number of temperatures which can be stored. Set in file <b>vars.for</b> . If a new parameter is desired, the files must be recompiled.
<b>P -</b>	Potential of water.
<b>PL -</b>	Array containing the water potentials at the previous time step.
<b>POSTVE -</b>	A flag indicating whether certain terms should be used in the calculation of <b>HA</b> and <b>HC</b> in <b>BCH</b> . The flag is set within subroutine <b>BCW</b> .
<b>RH -</b>	Relative humidity. Used to calculate the water vapor density of air. Included in the boundary conditions input file.
<b>RHOV -</b>	Water vapor density.
<b>RHOVA -</b>	Water vapor density of air.
<b>RHOVL -</b>	<b>RHOV</b> at the previous time step.
<b>RHOS -</b>	Solid density.
<b>RHOG -</b>	Gravitational coefficient.
<b>RHOW -</b>	Density of water.
<b>S -</b>	Solid volume fraction.
<b>S0 -</b>	The initial value of the variable <b>S</b> . Included in the raw data input file.
<b>SAME -</b>	The largest percentage difference between two iterations that will allow the process to act as if convergence has occurred. Set in file <b>INPUT</b> .
<b>SIGMA -</b>	One dimensional compressional stress of the solid phase <ul style="list-style-type: none"><li>• 1st column contains information from the most recently completed iteration.</li><li>• 2nd column contains information from the present iteration calculations.</li></ul>
<b>SIGMAL -</b>	One-dimensional compressional stress of the solid phase which occurred at the last time step
<b>SIGMAM -</b>	Array holding the maximum values of <b>SIGMA</b> calculated at each node.
<b>SL -</b>	<b>S</b> at the previous time step.
<b>SN -</b>	Net radiation at the upper surface.
<b>STEP -</b>	Count of the number of time steps taken that does not exceed <b>NUMT</b> .

APPENDIX I - 16

<b>SUBC</b>	Array holding input parameters needed to compute the volumetric specific heat of the material.
<b>SUBDVT</b>	Array holding input parameters needed to compute the water vapor diffusion coefficient.
<b>SUBK</b>	Array holding input parameters needed to compute permeability.
<b>SUBKT</b>	Array holding input parameters needed to compute thermal conductivity.
<b>SUBKWT</b>	Array holding input parameters needed to compute movement of liquid water under temperature gradient.
<b>SUBL</b>	Array holding input parameters needed to compute the volumetric enthalpy of air.
<b>SUBPTH -</b>	Array holding input parameters needed to compute the potential of water from the volume fraction of liquids or visa-versa.
<b>SUBS -</b>	Array holding input parameters needed to relate the variables S and SIGMA.
<b>SUBRHOVS</b>	Array holding input parameters needed to compute vapor density.
<b>SUBXS -</b>	Array holding input parameters needed to compute SIGMA in subroutine SOLID.
<b>T -</b>	Current time.
<b>T10 -</b>	Initial temperature of the bottom of the fine tailing pond.
<b>T20 -</b>	Initial temperature of the underlying material.
<b>TA -</b>	Air temperature.
<b>TEMP -</b>	Temperature of fine tails <ul style="list-style-type: none"><li>• 1st column - information from last completed iteration</li><li>• 2nd column - present iteration information being calculated.</li></ul>
<b>TEMPL -</b>	Array containing temperature of fine tail pond at the previous time step
<b>TEMPL2 -</b>	Array containing the temperature as it was two time steps previous.
<b>THETA -</b>	Volume fraction of liquids <ul style="list-style-type: none"><li>• 1st column - information from the last completed iteration</li><li>• 2nd column - present iteration information being calculated.</li></ul>

## APPENDIX I - 17

<b>THETAL</b> -	Array containing the volume fraction of liquids at the last completed time step.
<b>THETL2</b> -	Array containing the volume fraction of liquids two time steps previous.
<b>TIME</b> -	An array to keep track of the times at which the time steps took place. Used in calculated the boundary conditions for TEMP.
<b>WA</b> -	Boundary condition for THETA. Its use is described in BCW.
<b>WAL</b> -	WA saved from the previous time step.
<b>WB</b> -	Boundary condition for THETA. Its use is described in BCW.
<b>WBL</b> -	WB saved from the previous time step.
<b>WC</b> -	Boundary condition for THETA. Its use is described in BCW.
<b>WCL</b> -	WC saved from the previous time step.
<b>WD</b> -	Boundary condition for THETA. Its used is described in BCW.
<b>WDL</b> -	WD saved from the previous time step.
<b>WE</b> -	Boundary condition for THETA. Its use is described in BCW.
<b>WEL</b> -	WE saved from the previous time step.
<b>WG</b> -	Boundary condition for THETA. Its use is described in BCW.
<b>WGL</b> -	WG saved from the previous time step.
<b>WSMTH</b> -	Holds a value between 0 and 1. A weight factor used in subroutine SMOOTH.
<b>X</b> -	An array containing downward displacements of nodes in the fine tailing pond. The first element in the array is the top of the pond.

### 5. SUBROUTINE/FUNCTION LISTING AND DESCRIPTION

A total of 62 subprograms and one main program are used in program EVAP. Table 5.1 exhibits a general listing of subprogram names in alphabetical order. A description of each subprogram follows. Important procedural assumptions are outlined.

Table 5.1-1 Alphabetic Listing of Subprogram Names

ADJUST	FN1NEW	LGRANG1	VISC
AVEBC	FN2	LGRANG1I	VDIFF
BCH	FN2NEW	LGRANG2I	VSHM
BCW	FSVD	NEWTEMP	WATER
CNVGE	GETGRAD	NEWTHT	WMOVE
COPY	GETH	OUTFREQ	
CPLAST	GETFNS	OUTPUT	
CPYALL	GETRHOV	PERM	
CPYBAK	GETRHOVS	PTHETA	
DTHDP	GETXS	SEESUMM	
DVT	GOBACK	SETBCL	
EXTRAP	HEAT	SETFALSE	
FDX1	INIT	SMOOTH	
FDX2	INITOUT	SOLID	
FDX3	INTGRL	SSIGMA	
F2DX1	ITER	SUMM	
F2DX2	K	SVD	
F2DX3	KT	THERM	
FN1	KWT	THETAP	

Subroutine ADJUST

Called by the main program. It adjusts the arrays EXTW and EXTH which govern the percentage weight that the values of THETA and TEMP (respectively) found at the last time step have in determining the initial guesses of their values at the present time step.

Subroutine AVEBC

This is called by subroutines BCH and BCW. It averages the newly computed boundary conditions with those from the previous time step.

Subroutine BCH

This is called by subroutine HEAT and is used to compute the boundary conditions for temperature. It evaluates six variables (listed below) that are used by subroutine NEWTEMP in finding the new temperatures at the boundaries.

When considering the surface, a combination of the mass balance equation and the energy balance equation is used. First, the mass balance equation is rearranged to solve for the term containing  $\partial P / \partial X$ . This in turn is substituted into the energy balance equation. What is left is an equation predominantly dependent on temperature. Also substituted into the equation is the approximation

$$\frac{\partial \rho_v}{\partial X} \cong \rho_v \left( \frac{1}{\rho_{vs}} \frac{\partial \rho_{vs}}{\partial T} \right) \frac{\partial T}{\partial X}$$

This approximation is made when the equation of vapor transport is differentiated, and the terms related to water potential are ignored due to their minimal addition.

The final resulting equation is arranged into the following form and HA, HB, and HC can be found.

$$(HA) \frac{\partial T}{\partial X} + (HB)(T_1) = HC \quad (5.1)$$

where: -  $T_1$  is the temperature at the surface

$$HA = \frac{K_T(f,s,T)}{J} + L_v \frac{D_v(f,s,T)}{J} \left( \rho_v \left( \frac{1}{\rho_{vs}} \frac{d\rho_{vs}}{dT} \right) \right)$$

$$HB = h_h$$

$$HC = S_n + h_h T_a + L_v h_v (\rho_v - \rho_{va})$$

When calculating HA and HC, not all the terms may necessarily be used. A global logical flag called POSTVE indicates the usage. If POSTVE is false, then the terms in 'HA' and 'HC' starting with  $L_v$  are ignored. POSTVE is set in subroutine BCW.

When considering the bottom of the fine tails

$$(HD) \frac{\partial TN}{\partial X} + (HE)(TN) = HG \quad (5.2)$$

where:

-TN is the temperature at the bottom boundary

$$HD = \frac{KT(f, s, T)}{J}$$

$$HE = \frac{-2\sqrt{\frac{K_{T_2} C_2}{\Pi}}}{\sqrt{\Delta t}}$$

$$HG = \sqrt{\frac{K_{T_2} C_2}{\Pi}} \left( \int_0^{t_k} \frac{T(\tau)}{\sqrt{t_{(k+1)} - \tau}} d\tau - \frac{2T_k}{\sqrt{\Delta t}} - \frac{\nabla T}{\sqrt{t_{(k+1)}}} \right)$$

$$\nabla T = (T20 - T10) * \left( \frac{\sqrt{K_{T_1} C_1}}{\sqrt{K_{T_1} C_1} + \sqrt{K_{T_2} C_2}} \right)$$

-T20 is temperature of the sand at the boundary

-T10 is temperature of the fine tails at the boundary

### Subroutine BCW

This is called by subroutine WATER. It is used to compute the boundary conditions for the volume fraction of liquids. It does this in an indirect fashion, solving for the boundary conditions for water potential. Six boundary condition variables are solved for ( listed below) which are used by subroutine NEWTHET to find the new values of THETA and P at the boundaries

When considering the surface, the mass balance equation is used. This can be rearranged into the following form and used to solve for WA, WB, and WC.

$$(WA) \frac{\partial P}{\partial X} + (WB)(P) = WC \quad (5.3)$$

where:  $-P_1$  is the potential at the surface

$$WA = \frac{K(s, \theta)}{J\eta}$$

$$WB = 0$$

$$WC = - \left[ -h_v(\rho_v - \rho_{va}) + \frac{D_v(f, s, T)}{J} \frac{\partial \rho_v}{\partial X} + \frac{K_{wT}(s, \theta, T)}{J\eta} \frac{\partial T}{\partial X} - WA \times \rho g \right]$$

The first two terms in finding WC are also used to set a global flag variable. If the value of these terms when added together is a positive one, the flag is set to “false” and visa-versa. The flag indicates whether terms in subroutine BCH should be included in calculations there. It reflects the flow of water potential.

When considering the bottom of the fine tails, water and vapor fluxes are zero, so  $P_N = P_{N-1}$  where N stands for the bottom node, and (N-1) stands for the point just above. Because of the following relation

$$(WD) \frac{\partial P_N}{\partial X} + (WE)(P_N) = WG \quad (5.4)$$

then

$$WD = 1$$

$$WE = 0$$

$$WG = 0$$

#### Function CNVGE

Determines the convergence of its argument by comparing the data of the last two iterations against a maximum percentage discrepancy ( variable SAME set in file INPUT). Returns true if converged, false if not. If it is noticed that convergence has taken place very quickly, ( i.e. the number of iterations taken before convergence is less than the variable MINI ) the flag DOUBLE is set that will double the time step size for the next step.

Subroutine COPY

Copies the 2nd column of the input array to its first. Used to copy iteration information for variables THETA, TEMP and SIGMA to prepare for the next iteration.

Subroutine CPLAST

Saves information from the previous time step into an array designated for values from two steps ago for THETA, and TEMP.

Subroutine CPYALL

Copies all data from the next time step array values to the present time step ("last" arrays). The "last" arrays will then have final iteration values from the previous time step.

Subroutine CPYBAK

Opposite of subroutine CPYALL. This routine is called when a time step taken was too large. It reverts all arrays to the condition they were at the start of the time step.

Function DTHDP

This calculates the derivative of THETA with respect to P at a specified node. If the relations for THETA and P are altered, this subroutine needs to be altered as well.

Function DVT

This calculates the water vapor diffusion coefficient.

Subroutine EXTRAP

Performs a weighted linear extrapolation on THETA and TEMP, using saved data from the previous two time steps. For example, when considering THETA, first we perform a straight linear extrapolation.

$$\text{FIRSTW(I)} = \text{THETAL(I)} + \text{DT} * (\text{THETAL(I)} - \text{THETL2(I)}) / \text{DTL}$$

where:

- FIRSTW(I) is the extrapolation at node I
- THETAL(I) is THETA from the previous time step
- DT is the present incremental step size
- THETL2 is THETA from two steps previous
- DTL is the old incremental time step size

After the above has been completed, the result is “weighted” to result in the first guess for the next time step according to

$$\text{THETA}(I,1) = \text{THETAL}(I) * \text{EXTW}(I) + (1 - \text{EXTW}(I)) * \text{FIRSTW}(I)$$

where:

- EXTW(I) is a weight between zero and one that is determined in subroutine ADJUST by the trend of the last two time steps.
- THETA(I,1) is the initial guess for THETA at node I;
- $1 \leq I \leq \text{NNODES}$

#### Subroutine FDX1

Calculates the three terms of ratio of displacements in Lagrangian interpolation formulae used in evaluating the derivative of the first node. See the Appendix C for an explanation of the Lagrangian interpolation polynomial.

#### Subroutine FDX2

Calculates the three terms of ratio of displacements in Lagrangian interpolation formulae used in evaluating the derivative of an internal node. See the Appendix C for an explanation of the Lagrangian interpolation polynomial.

Subroutine FDX3

Calculates the three terms of ratio of displacements in Lagrangian interpolation formula used in evaluating the derivative of the final node. See the Appendix C for an explanation of the Lagrangian interpolation polynomial.

Subroutine F2DX1

Calculates the three terms of ratio of displacements in Lagrangian interpolation formulae used in evaluating the second derivative of the first node. See the Appendix C for an explanation of the Lagrangian interpolation polynomial.

Subroutine F2DX2

Calculates the three terms of ratio of displacements in Lagrangian interpolation formulae used in evaluating the second derivative of an internal node. See the Appendix C for an explanation of the Lagrangian interpolation polynomial.

Subroutine F2DX3

Calculates the three terms of ratio of displacements in Lagrangian interpolation formula used in evaluating the second derivative of the final node. See the Appendix C for an explanation of the Lagrangian interpolation polynomial.

Subroutine FN1

This subroutine evaluates the equation of water transport solving for  $\partial\theta/\partial t$ . using the iteration information from the last time step. The equation looks as follows:

$$\frac{\partial\theta}{\partial t} = \frac{1}{J} \left( \frac{\partial}{\partial X} \left[ \frac{D_{vT}(f,s,T)}{J} \frac{\partial p_v}{\partial X} + \frac{K_{wT}(s,\theta,T)}{J\eta} \frac{\partial T}{\partial X} + \frac{K(s,\theta)}{J\eta} \left( \frac{\partial P}{\partial X} - \rho g \right) \right] - J\theta \frac{\partial \epsilon}{\partial t} \right)$$

It solves the equation by breaking it up into parts. The three terms inside the large [] brackets are computed by functions VDIFF, WMOVE and PERM.

Subroutine FN1NEW

This subroutine evaluates the equation of water transport solving for  $\partial\theta/\partial t$ . using the most recent iteration information, or the initial guesses for a time step produced by subroutine EXTRAP. One term in the overall formula is ignored. (Compare with FN1.)

It is added into the equation in subroutine NEWTHET. The equation looks as follows:

$$\frac{\partial\theta}{\partial t} = \frac{1}{J} \left( \frac{\partial}{\partial X} \left[ \frac{D_{vT}(f,s,T)}{J} \frac{\partial\rho_v}{\partial X} + \frac{K_{wT}(s,\theta,T)}{J\eta} \frac{\partial T}{\partial X} - \frac{K(s,\theta)\rho g}{J\eta} \right] - J\theta \frac{\partial\varepsilon}{\partial t} \right)$$

It solves the equation by breaking it up into parts. The two terms inside the large [] brackets are computed by functions VDIFF, and WMOVE.

Subroutine FN2

This subroutine evaluates the equation of heat transport solving for  $\partial T/\partial t$  using the iteration information from the last time step. The equation looks as follows:

$$\frac{\partial T}{\partial t} = \left( \frac{\partial}{\partial X} \left[ \frac{K_T(f,s,T)}{J} \frac{\partial T}{\partial X} \right] - CWT \frac{\partial}{\partial X} \left[ \frac{D_{vT}(f,s,T)}{J} \frac{\partial\rho_v}{\partial X} \right] \right) \frac{1}{(C(s,\theta)*J)}$$

Subroutine FN2NEW

This subroutine evaluates the equation of heat transport solving for  $\partial T/\partial t$  using the most recent iteration, or the initial guesses found by EXTRAP. One term in the overall equation is ignored. ( Compare with FN2.) This term is added in during subroutine NEWTEMP. The equation looks as follows:

$$\frac{\partial T}{\partial t} = \left( CWT \frac{\partial}{\partial X} \left[ \frac{D_{vT}(f,s,T)}{J} \frac{\partial\rho_v}{\partial X} \right] \right) \frac{1}{(C(s,\theta)*J)}$$

Function FSVD

Calculates the approximated vapor transport differential used in subroutine BCH

Subroutine GETGRAD

This subroutine forms the gradient and divergence matrices GRAD and DIVR using the Lagrangian interpolation polynomial. See the Appendix C for an explanation of the polynomial.

Subroutine GETH

This subroutine reads the data from the boundary conditions input file, performing the necessary interpolations, and calculating HH and HV for a time step. These are then used in boundary condition calculations by BCH and BCW for the time step.

Subroutine GETFNS            Calls FN1 and FN2

Subroutine GETRHOV        Calculates RHOV for the specified temperature and node.

Subroutine GETRHOVS      Calculates RHOVS for the specified node.

Function GETXS

Called by subroutine SOLID, this is part of a calculation to find SIGMA.

Subroutine GOBACK

This is called by subroutine ITER and occurs when the time step taken was too large for the system of equations to handle and oscillations would occur. It sets DT to be half of its present value, and restores variables to pre-iteration status, enabling the time step to start over again with a smaller time difference. If the new DT is less than the minimum allowable value DTMIN, DT is set to DTMIN. If the new DT had already been set to DTMIN, the program it stopped. This means that either DTMIN was set to be too high, or there is a problem with convergence. DTMIN is set in the raw data file.

Subroutine HEAT

This subroutine calls routines to calculate new temperature values within the iterative process based on old data. The process incorporates the equation of heat transport. It works on the premise that

$$TEMP_{new} = TEMP_{old} + DT(FH1 + FH2) / 2$$

where:

- $TEMP_{new}$  is the newest iteration information
- $TEMP_{old}$  is the value from the last time step
- FH1 is the function for the partial derivative of TEMP with respect to time evaluated at the previous time step. It is constant throughout the present time step and is evaluated in the main program.

- FH2 is the same function without the term including the temperature derivative with respect to displacement evaluated using data from the latest iteration within the present time step. The missing term is added in subroutine NEWTEMP.

- DT is the time increment between steps

Boundary conditions are used to find parameters needed to calculate new values for the temperatures at the surface and the bottom of the fine tail. Subroutine NEWTEMP performs the calculations of the new temperature values at all of the nodes.

Subroutine INIT

This subroutine initializes variables according to the conditions specified by the raw data input file. It also sets the necessary flags and controlling parameters to start up status.

Subroutine INTOUT

This opens the output file raw.out and writes to it parameters needed by program GETDAT to process the data.

Function INTGRL

This evaluates an approximation of an integral needed to find the one-dimensional compressional stress in subroutine SOLID.

Subroutine ITER

This subroutine controls the iteration of the variables THETA, TEMP, and SIGMA (volume fraction of liquids, temperature, one-dimensional compressional stress of the solid phase). It calls routine CNVGE which checks for their convergence within a time step and then subsequently decides whether to continue with their iterations. When complete, THETA, TEMP, and SIGMA contain the final values for that time step, and all other variables dependent upon them will have been updated.

Function K      Calculates permeability.

Function KT      Calculates thermal conductivity.

Function KWT      Calculates the liquid water movement.

Subroutine LGRANGI

Calculates the derivatives according to vertical displacement for all nodes in a one-dimensional array using Lagrangian interpolation formulae. See Appendix C for a description of the process.

Subroutine LGRANGII

Calculates the derivative according to vertical displacement for a single node in a one-dimensional array using Lagrangian interpolation formula. See Appendix B for a full description of the process.

Subroutine LGRANG2I

Calculates the derivative according to vertical displacement for a single node in a two-dimensional array using Lagrangian interpolation formula. See Appendix C for a full description of the process.

Subroutine NEWTEMP

Called by subroutine HEAT. It calculates the new values of temperature for all nodes in an iterative step. See HEAT.

Subroutine NEWTHET

Called by subroutine WATER. It calculates the new values of THETA and P for all nodes in an iterative step. See WATER.

Subroutine OUTFREQ

Determines whether the present time is near a multiple of the specified desired output frequency. If so, it call subroutine OUTPUT to write the data to the raw data file.

Subroutine OUTPUT

Outputs the following variables to an unformatted raw data file called raw.out:

1. time
2. temperature
3. one-dimensional compressional stress of solid phase
4. water potential
5. water vapor density
6. strain
7. current/initial thickness of a layer

Program GETDAT can be used to extract data for examination.

Function PERM

$$\text{Calculates } \frac{K(s,\theta) * \left( \frac{\partial P}{\partial X} - \rho g \right)}{J\eta}$$

Function PTHETA

Finds the relative THETA for a given water potential value.

Subroutine SEESUMM

Determines whether the integral used in calculating the bottom boundary condition for TEMP will be evaluated. This is accomplished by comparing the temperatures on the bottom of the fine tailing pond from previous time steps to see if the differences are large enough to warrant performing the integration. SEESUMM sets the flag DOSUMM which governs the operation.

Subroutine SETBCL

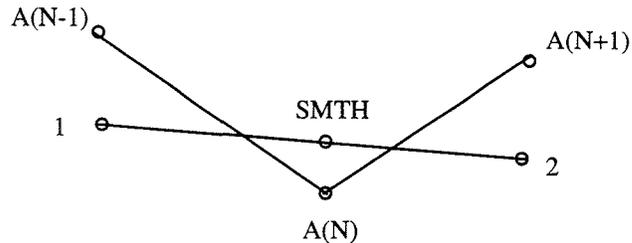
This subroutine is called by the main program. It saves the boundary conditions for THETA and TEMP before proceeding on to a new time step.

Subroutine SETFALSE

This subroutine set the flags DONEW, DONEH, and DONES to false, while setting the NUMI equal to one.

Subroutine SMOOTH

This subroutine smooths out an array to take out oscillatory values. It works according to the following diagram:



where:

- $A(N-1)$ ,  $A(N)$ ,  $A(N+1)$  are 3 consecutive nodes in an array.
- point 1 is the average of  $A(N-1)$  and  $A(N)$
- point 2 is the average of  $A(N)$  and  $A(N+1)$
- SMTH is the smoothed out value of  $A$  at node  $N$  found by taking the slope of the line between points 1 and 2.

The new value of  $A(N) = A(N) + WSMTH * (SMTH - A(N))$ .

WSMTH is in the program supplied input file entitled INPUT and is a value between zero and one. If WSMTH is zero then  $A(N)$  doesn't change. If WSMTH is one, then  $A(N)$  becomes SMTH.

This subroutine is called by WATER to smooth out THETA after each iteration.

Subroutine SOLID

This subroutine calculates new SIGMA or one-dimensional compressional stress values within the iterative process. These values are then used in turn to find EPS and J. Results are found using force balance considerations and consolidation relations of the solid phase.

Subroutine SSIGMA

Finds the corresponding SIGMA value for a particular S.

Subroutine SUMM

This evaluates a summation used to approximate an integral needed to find the boundary condition for temperature at the bottom of the fine tails. It is called by BCH. The summation is

$$\sum_{i=1}^{step-1} \frac{bottom(i) - bottom(i-1)}{\sqrt{t - (i\Delta t - \Delta t / 2)}}$$

where

-bottom(i) is the temperature at the 'ith' time step

-t is the current time

-\Delta t is the difference in time between time steps of the step in question

Function SVD      Calculates the volumetric enthalpy of air.

Function THERM

Calculates  $\frac{K_T(f, s, T) * \partial T}{J \quad \partial X}$

Function THETAP

Finds the relative water potential for a given THETA value.

Function VDIFF

Calculates  $\frac{D_{vT}(f, s, T) * \partial P_v}{J \quad \partial X}$

Function VISC      Calculates the viscosity of water.

Function VSHM

Calculates the volumetric specific heat of the material.

Subroutine WATER

This subroutine calls routines to calculate new THETA or volume fraction of liquid values within the iterative process based on old data. The process incorporates the equation of water transport and works on the premise that

$$\theta_{new} = \theta_{old} + DT(FW1 + FW2) / 2$$

where:

- $\theta_{new}$  is the newest iteration information
- $\theta_{old}$  is the value from the last time step
- FW1 is the function for the partial derivative of  $\theta$  with respect to time evaluated at the previous time step. It is constant throughout the present time step and is evaluated in the main program.
- FW2 is the same function without the term including the derivative of potential with respect to displacement evaluated using data from the last iteration within the present time step. The missing term is added in the subroutine NEWTHET.
- DT is the time increment between steps

Boundary conditions are used to find parameters needed to evaluate new values for the volume fraction of liquids and potential at the surface and the bottom of the fine tail. NEWTHET incorporates the above functions and parameters to find the new values of THETA and P at all of the nodes.

After the above process has been completed, subroutine SMOOTH is called to smooth out the THETA array.

Function WMOVE

Calculates  $\frac{K_{wr}(s, \theta, T)}{J\eta} \frac{\partial T}{\partial X}$

## Appendix A: GETDAT Manual

## APPENDIX 1A - 1

GETDAT was written for the examination of the raw data output created by EVAP. The raw data collected by program EVAP available for inspection includes:

1. temperature
2. volume fraction of liquids
3. one-dimensional compressional stress of the solid phase
4. water potential
5. water vapor density
6. consolidation
7. ratio of current and initial layer thickness

Before running GETDAT, first run program EVAP to create a raw data file. In order for GETDAT to work correctly, it is necessary for it to have been compiled with the same value of parameter MNODES in file vars.for as that with which the program EVAP was compiled. Therefore, if the parameter MNODES is altered in file vars.for for the purpose of program EVAP, GETDAT needs to be recompiled at the same time. GETDAT will then no longer work on the raw data files previously created with the old value of MNODES. To use the old raw data files, recompile GETDAT with the old value of MNODES. It would be much wiser to not alter the parameter MNODES as it could become quite confusing!

To start the program, type GETDAT. Upon commencement, GETDAT will ask you for the name of the raw data file created by EVAP. This file was created with the name raw.out. To continue, type the file name in single apostrophes. If the file name raw.out has not been changed, simply type 'raw.out' when prompted.

GETDAT prompts the user to make decisions. There are two main streams of choices: two examine data at a particular moment in time, or to examine how the data changes with time.

## APPENDIX 1A - 2

### **Choice one: examining data at a particular moment in time.**

Choice one occurs by typing a 1 at the main menu prompt. The first alternative within this option is to choose to examine either all or one particular variable's values at a particular moment in time. To choose to examine all of them, type a 1 followed by a return. Or, to only examine one variable in particular, type 2. You will then be asked what moment in time you would like to examine. The maximum length of time in the file will be provided for you. Type in a number between 0 and the maximum. The selected variable(s) are charted at the specified time alongside vertical displacement.

### **Choice two: examining how the data changes with time.**

Choice two involves viewing a single variable's changes over all time. To choose this option input a 2 at the main menu followed by a return. You can then look at either all depths of that variable, or you can choose to hone in on a specific depth of interest. Looking at all the depths will result in multiple charts of vertical displacement and the chosen variable headed by the time the data occurred. Conversely, choosing to look at only one depth of the variable will result in a single chart detailing that particular node's data for all time of the model process. To make the decision you want, simply enter the number shown on the screen associated with the task.

Appendix B:

Formation of Evap-E

## APPENDIX 1B - 1

Evap-E is a translation from FORTRAN to C of EVAP described in other sections. The standard unix f2c utility was used for the initial translation, and subsequent modifications to eliminate the need for the associated f2c library followed.

The only functional changes relate to the input and output methods and formats of simulation data. Rather than reinvent the data-entry and graphics output wheels, the decision was taken to use a "Comma Separated Variable" ASCII file format and then rely on regular spreadsheet software as the user interface.

In theory, *\_any\_* spreadsheet could read the .CSV files produced by evap3, but in practice, only EXCEL is unhampered by excessively restrictive line-length limitations. In addition, EXCEL's 3D graphing capabilities make visualizing the entire simulation run simple and easily manageable (subject to some restrictions).

Investigations into using other more powerful graphics software such as GNUPLOT continue, and requests for customized ASCII data output will be entertained.

In its current incarnation, evap3 can be run in two different modes - either as a filter or on a .CSV file in-place. The file EXAMPLE.CSV which accompanies this software is a example datafile showing the syntax for specifying the required simulation parameters.

```
C:\> copy EXAMPLE.CSV TEST.CSV
```

```
C:\> evap-E TEST.CSV in-place mode
```

```
C:\> evap-E <TEST.CSV >TESTOUT.CSV filter mode
```

Note: Do not make the output file the same as input file when running as a filter - the input file will be cleared first.

The first 49 lines of the input file remain unchanged or are simply copied to the output. The results of the simulation appear beginning on line 50. In this way, it is easy to keep the simulation parameters and results organized. To rerun a simulation with slightly different parameters, copy (or Save As) to a different file, modify the input parameters, and rerun. The old results will be overwritten/discarded/replaced with the new results.

## APPENDIX 1B - 2

Since Excel's 3D graphing is not true XYZ plotting, and since the simulation results are arranged node-wise, not depth-wise (i.e. the total depth decreases during the run while number of nodes is fixed), the default behavior is to skew the data to improve the graphical visualization. The depth of each node at the beginning of the run is used to locate the solids content values for subsequent time steps.

To view/plot/print the simulation results, open the file with Excel as a .CSV ASCII text file. The data will appear with two header rows suitable as axis labels with simulation iterations arranged row-wise below. Column-wise there are 3 groups of data each having the time-step value (in days) as its first column. The first group is total depth, the second group is the skewed solids content for each simulation node, and the third group is the unskewed solids content for each node.

Since this data-skewing is only useful when 3D-graphing the data with XYZ-impaired software, this default behavior can be changed with the -deskew flag. This flag causes the second columnwise group of data to be changed from "skewed solids content" to "depth at each node".

```
C:\> evap-E -deskew TEST.CSV          in-place mode
```

Initial forays using gnuplot prompted the implementation of an additional -gnuplot flag which radically alters the data output format. The input parameters are not copied to the output, and the data is presented as (Time-step, Depth, Solids-content) triples, three sets of ASCII values per line with a blank line separating groups of triples corresponding to each simulation Time-step.

```
C:\> evap-E -gnuplot <TEST.CSV >TEST.DAT  gnuplot compatible output
```

## Appendix C:

### Numerical Procedures - The Lagrangian Interpolation Polynomial

## APPENDIX 1C - 1

For all terms where a partial differential is needed with respect to spatial coordinates, the Lagrangian interpolation polynomial is used to achieve this goal. The ratios of displacement are calculated at the beginning of the program and stored in matrices GRAD and DIVR for first derivative and second derivative respectively.

For an arbitrary function,  $F(X)$ , the formula for three nodal points  $X_1, X_2, X_3$  is

$$F(X) = \frac{(X - X_2)(X - X_3)}{(X_1 - X_2)(X_1 - X_3)} f_1 + \frac{(X - X_1)(X - X_3)}{(X_2 - X_1)(X_2 - X_3)} f_2 + \frac{(X - X_1)(X - X_2)}{(X_3 - X_1)(X_3 - X_2)} f_3$$

where  $f_1, f_2,$  and  $f_3$  are values of  $F(X)$  at  $X_1, X_2, X_3$  respectively. The first derivative of the equation using the same notation is

$$F'(X) = \frac{(X - X_2) + (X - X_3)}{(X_1 - X_2)(X_1 - X_3)} f_1 + \frac{(X - X_1) + (X - X_3)}{(X_2 - X_1)(X_2 - X_3)} f_2 + \frac{(X - X_1) + (X - X_2)}{(X_3 - X_1)(X_3 - X_2)} f_3$$

When calculating the derivative at the first node in an array, it becomes

$$F'(X_1) = \frac{(X_1 - X_2) + (X_1 - X_3)}{(X_1 - X_2)(X_1 - X_3)} f_1 + \frac{(X_1 - X_3)}{(X_2 - X_1)(X_2 - X_3)} f_2 + \frac{(X_1 - X_2)}{(X_3 - X_1)(X_3 - X_2)} f_3$$

and subroutine FDX1 is called to find the three terms of the ratios of  $X$ . These can later be multiplied by the corresponding  $f_1, f_2$  and  $f_3$  values.

A similar procedure is followed in the evaluation of the partial derivatives of the other nodes in an array. For an internal node  $X(M)$ , FDX2 is called to find the three ratios of  $X$  in

$$F'(X(M)) = \frac{X(M) - X(M+1)}{(X(M-1) - X(M))(X(M-1) - X(M+1))} f^{(M-1)} + \frac{2X(M) - X(M-1) - X(M+1)}{(X(M) - X(M-1))(X(M) - X(M+1))} f^{(M)} \\ + \frac{X(M) - X(M-1)}{(X(M+1) - X(M-1))(X(M+1) - X(M))} f_3$$

For the bottom node,  $X(N)$ , FDX3 is called and the Lagrangian equation becomes

APPENDIX 1C - 2

$$F'(X(N)) = \frac{X(N) - X(N-1)}{(X(N-2) - X(N-1))(X(N-2) - X(N))} f^{(N-2)} + \frac{X(N) - X(N-2)}{(X(N-1) - X(N-2))(X(N-1) - X(N))} f^{(N-1)} + \frac{2X(N) - X(N-1) - X(N-2)}{(X(N) - X(N-2))(X(N) - X(N-1))} f^3$$

When finding the second derivative, the divergence is defined as the same for the all nodes. For example, in a three nodal system, the second derivative is defined by

$$F''(X1) = \frac{2}{(X1 - X2)(X1 - X3)} f1 + \frac{2}{(X2 - X1)(X2 - X3)} f2 + \frac{2}{(X3 - X1)(X3 - X2)} f3$$

The ratios of displacement are found by functions F2DX1, F2DX2, and F2DX3.

Appendix II:

The Computer Program

This material is provided under educational reproduction permissions included in Alberta Environment and Sustainable Resource Development's Copyright and Disclosure Statement, see terms at <http://www.environment.alberta.ca/copyright.html>. This Statement requires the following identification:

"The source of the materials is Alberta Environment and Sustainable Resource Development <http://www.environment.gov.ab.ca/>. The use of these materials by the end user is done without any affiliation with or endorsement by the Government of Alberta. Reliance upon the end user's use of these materials is at the risk of the end user.