308
Scientific Notebook # 229
Q200006060001
21
150
R

Ron Green / Frank Dodge / Goodluck Ofoegbu

The Boorum & Pease® Quality Guarantee

The materials and craftsmanship that went into this product are of the finest quality. The pages are thread sewn, meaning they're bound to stay bound. The inks are moisture resistant and will not smear. And the uniform quality of the paper assures consistent rulings, excellent writing surface and erasability. If, at any time during normal use, this product does not perform to your expectations, we will replace it free of charge. Simply write to us:

Boorum & Pease Company
71 Clinton Road, Garden City, NY 11530
Attn: Marketing Services

Any correspondence should include the code number printed at the bottom of this page as well as the book title stamped at the bottom of the spine.

CNWRA
CONTROLLED
COPY 229

One Good Book Deserves Many Others.

Look for the complete line of Boorum & Pease® Columnar, Journal, and Record books. Custom-designed books also available by special order. For more information about our Customized Book Program, contact your office products dealer. See back cover for other books in this series.
Made in U.S.A.

Contents | Page

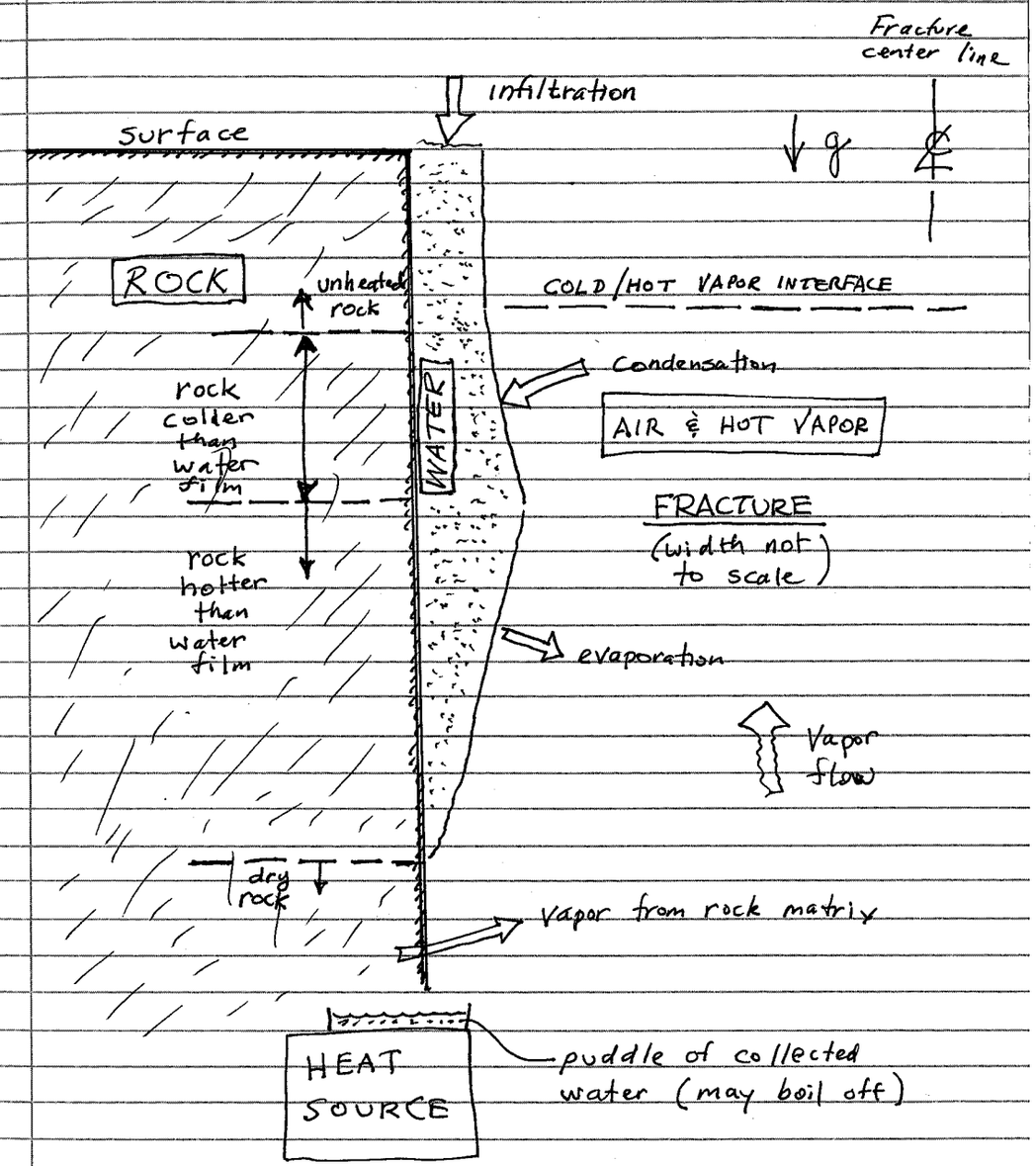
Contents | Page

Derivation of evaporation - condensation - movement models for the "Gravity Driven Refluxing Conceptual Model"

- liquid water falls slowly down the walls of a heated fracture (vertical)
- depending on whether the temperature of the walls is increasing or decreasing, water may evaporate or condense from the falling film
- water vapor may also be generated in the rock matrix by heating, and this vapor may be transported to the fracture where it can contribute to the net evaporation or condensation
- the liquid water that reaches the bottom of the fracture is assumed to pool on the heater, where it can evaporate (boil) if the heater temperature is high or remain liquid if the heater temperature is low
- a constant rate of infiltration (perhaps zero) is assumed at the top entrance of the fracture
- this model will interface with some finite element code of the rock matrix

F.D 7/29/97

Conceptual Physical Picture



F.D. 7/29/97

General Procedure

1. Use the rock thermal response code to compute the fracture wall temperature at time "t", using standard boundary conditions and the heat flux at the wall appropriate to the water evaporation/condensation rate

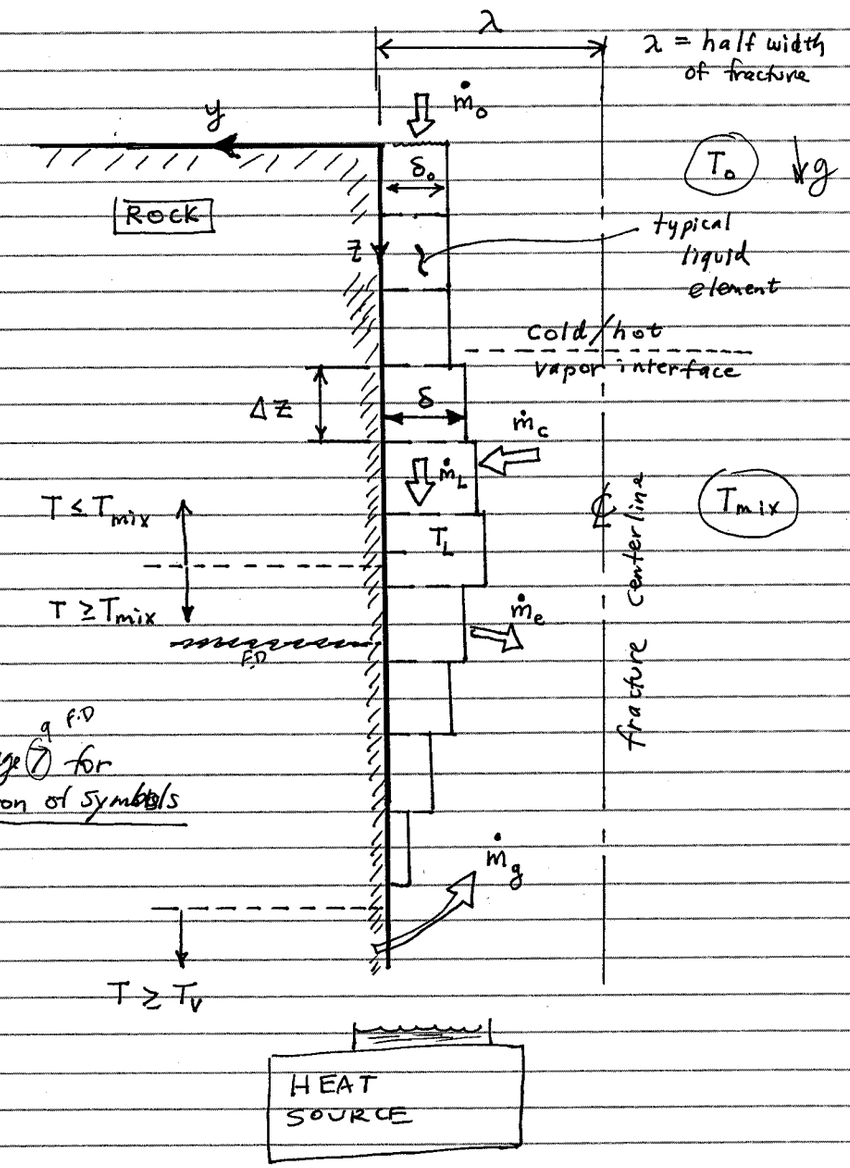
2. Assume gas mixture in the fracture is "well mixed". The gas below the "hot-cold" vapor interface is assumed to be all at one uniform temperature, and the gas above it is all at another uniform temperature. The "hot-cold" interface may be at the top of the fracture.
The temperature of the hot gas is computed at time "t" by summing all the heat transfers to the gas as well as the sensible heat of the water evaporated into it, less the sensible heat of the water condensed out of it.

3 "March" down the liquid film from the top of the fracture to compute the evaporation and condensation from the liquid film, and thus to compute the liquid ~~the~~ film thickness at time "t".

4. Iterate steps 1 – 3.

FD 1/29/97

FINITE DIFFERENCE GRID @ TIME = t



See page 9 F.D. for definition of symbols

F.D. 7/29/97

Definition of symbols – 1

$T(z,t)$ = temperature of rock wall

$T_{mix}(t)$ = temperature of hot air and vapor mixture below the "cold/hot vapor interface"

T_0 = initial temperature of rock, liquid, gas, etc, at time $t=0$. also the temperature of the cold air and vapor mixture above the "cold/hot vapor interface"

$T_L(z,t)$ = temperature of liquid film

T_V = boiling temperature of water at the gas mixture pressure existing in the fracture

$\dot{m}_0$ = constant infiltration rate at top of fracture (per unit depth of film)

F.D

$\cancel{\dot{m}_L}$ =

$\dot{m}_L(z,t)$ = downward water flow rate in film, per unit depth of film

$\dot{m}_c(z,t)$ = condensation rate, per unit depth of film, per unit length in z-direction

$\dot{m}_e(z,t)$ = evaporation rate, per unit depth of film, per unit length in z-direction

F.D 7/29/97

Definition of symbols – 2

$\dot{m}_g(t)$ = rate of generation of vapor injected into fracture by vaporization in rock matrix or from ~~boiling~~ boiling of accumulated puddle, per unit depth of liquid film

δ_0 = initial width of liquid film at time t_0. also, the width at the top of the fracture

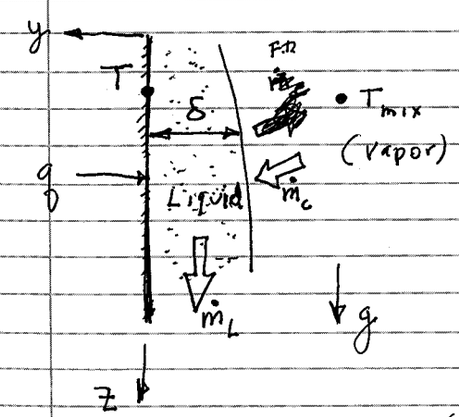
$\delta(z,t)$ = width of liquid film at location z, at time t.

g = acceleration of gravity.

λ = half width of fracture

FLOW RATE, FILM THICKNESS, & HEAT TRANSFER FOR A CONDENSING / EVAPORATING FILM

from <u>Heat Transfer</u>, by J. P. Holman, McGraw Hill, 1972



$$\dot{m}_L = \frac{\rho_L^2 g \delta^3}{3\mu_L}$$

$$q = k_L\left(\frac{T_{mix} - T}{\delta}\right)$$

$$q = -k_w \frac{dT}{dy}$$

(cont'd)

F.D. 7/30/97

$$\frac{d\delta}{dz} = \frac{k_L \mu_L (T_{mix} - T)}{\rho_L^2 g \delta^3 h_{fg}}$$

ρ_L = liquid density

μ_L = liquid viscosity

k_L = liquid thermal conductivity

k_W = rock thermal conductivity

q = heat transfer rate per unit depth of film

h_{fg} = liquid latent heat of condensation

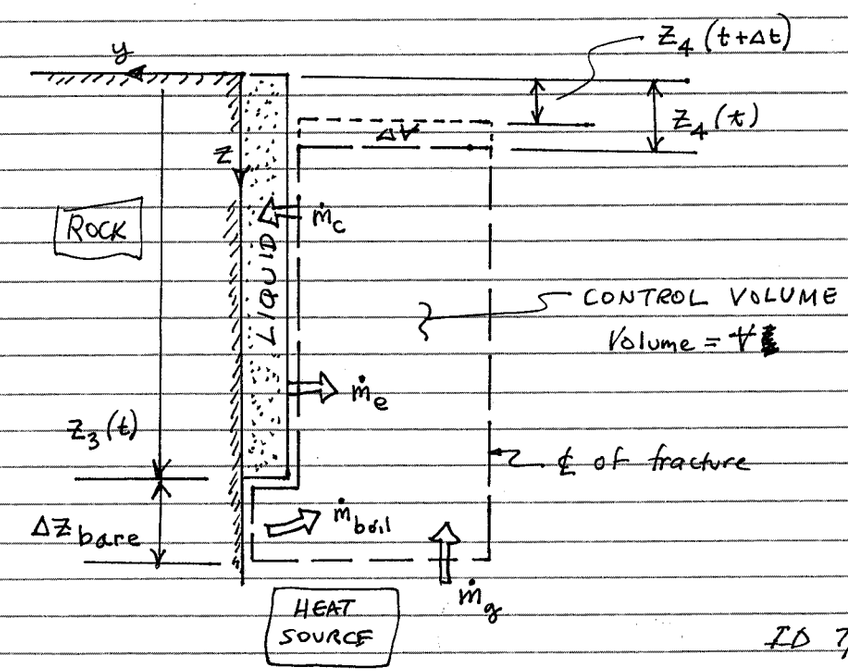
- The same relations can be used for evaporation

F.D. 7/30/97

MODEL TO DETERMINE HOT AIR & VAPOR MIXTURE TEMPERATURE "T_{mix}" AND THE VERTICAL LOCATION OF THE COLD/HOT GAS MIXTURE INTERFACE "z_4"

- Assumptions
  - air and vapor are well mixed $\Rightarrow T_{mix} \neq f(z)$
  - mixture has 100% relative humidity
  - any gas in contact with the dry part of the fracture wall has the same temperature as the rock but is rapidly mixed to temperature T_{mix}.

- CONTROL VOLUME FOR GAS MIXTURE



ID 7/30/97

- more assumptions
  - condensation fluid leaves the control volume with a temperature = T_{mix}
  - evaporation fluid enters the control volume with a temperature = T_L
  - hot vapor from rock matrix enters control volume with a temperature = T_g
  - boiling vapor from liquid film enters control volume with a temperature = T_v
  - gas pressure = constant = atmospheric pressure P_o
    (This assumption can be relaxed if necessary)

GEOMETRIC RELATION FOR $z_4 \geq 0$

$$\Delta \forall (t+\Delta t) = \left[z_4(t) - z_4(t+\Delta t) \right] \lambda \qquad (1)$$

- consider later cases when $z_4 < 0$

MASS BALANCE TO COMPUTE $\Delta \forall$

$$\Delta(\text{mass}) = (\text{Mass In}) - (\text{Mass Out})$$

so

FD. 7/30/97

$$\Delta t\left\{\frac{d}{dt}\left(\rho_{mix}\,\forall\right)\right\} = \Delta(\text{mass}) \qquad (2)$$

$$\Delta t\left\{\dot{m}_g(t) + \underbrace{\sum\left[\dot{m}_e(z,t)\right]\Delta z}_{\text{sum over elements that are evaporating}}\right\} + \underbrace{\sum\left[\rho_L\,\delta(z,t)\right]\Delta z}_{\text{sum over elements that boiled away from "t" to "t+}\Delta t\text{"}} = \text{mass in} \qquad (3)$$

$$\Delta t\left\{\sum\left[\dot{m}_c(z,t)\right]\Delta z\right\} = \text{mass out} \qquad (4)$$

$$\text{Eq. (2)} \rightarrow \quad \Delta t\left\{\forall(t)\frac{d\rho_{mix}}{dt} + \rho_{mix}\frac{d\forall}{dt}\right\} \qquad (5)$$

$$= \forall(t)\frac{d\rho_{mix}}{dt}\Delta t + \rho_{mix}\,\Delta\forall(t+\Delta t)$$

Combine Eqs. (3), (4), and (5) to give

$$\rho_{mix}\,\Delta\forall(t+\Delta t) = \left\{\dot{m}_g(t) + \sum\left[\dot{m}_e(z,t)\right]\Delta z\right.$$

$$\left. - \sum\left[\dot{m}_c(z,t)\right]\Delta z\right\}\Delta t + \sum\left[\rho_L\,\delta(z,t)\right]\Delta z \qquad (6)$$

$$- \forall(t)\frac{d\rho_{mix}}{dt}\Delta t$$

— need an expression for $d\rho_{mix}/dt$

F.D 7/30/97

Ideal gas law

$$\rho_{mix}(t) = \frac{P_{mix}(t)}{[R_{mix}(t)][T_{mix}(t)]} \qquad (7)$$

R_{mix} = gas constant
P_{mix} = constant

So

$$\frac{d\rho_{mix}}{dt} = -\frac{P_{mix}}{R_{mix}T_{mix}^2}\frac{dT_{mix}}{dt} - \frac{P_{mix}}{R_{mix}^2 T_{mix}}\frac{dR_{mix}}{dt}$$

$$= -\frac{P_{mix}}{R_{mix}T_{mix}}\left[\frac{1}{T_{mix}}\frac{dT_{mix}}{dt} + \frac{1}{R_{mix}}\frac{dR_{mix}}{dt}\right]$$

$$= -\frac{P_{mix}}{(R_{mix}T_{mix})^2}\left[R_{mix}\frac{dT_{mix}}{dt} + T_{mix}\frac{dR_{mix}}{dt}\right]$$

$$= -\frac{\rho_{mix}}{R_{mix}T_{mix}}\left[R_{mix}\frac{dT_{mix}}{dt} + T_{mix}\frac{dR_{mix}}{dt}\right]$$

This will get complicated because R_{mix} depends on the relative proportions of air and vapor in the mixture.

From the thermodynamics of an ideal gas mixture:

$$\frac{P_{mix}}{R_{mix}} = \frac{P_{air}}{R_{air}} + \frac{P_v}{R_v}$$

$$= \frac{(P_{mix} - P_v)}{R_{air}} + \frac{P_v}{R_v}$$

P_v = vapor partial pressure
R_v = vapor gas constant
P_{air} = air partial pressure
R_{air} = air gas constant

or

$$R_{mix} = \frac{P_{mix}}{P_v\left(\frac{1}{R_v} - \frac{1}{R_{air}}\right) + \frac{P_{mix}}{R_{air}}}$$

F.D 7/20/97

So

$$\frac{dR_{mix}}{dt} = -\frac{P_{mix}}{\left[P_V\left(\frac{1}{R_V}-\frac{1}{R_{air}}\right)+\frac{P_{mix}}{R_{air}}\right]^2}\left[\left(\frac{1}{R_V}-\frac{1}{R_{air}}\right)\frac{dP_V}{dt}\right]$$

$$= -\frac{R_{mix}^2}{P_{mix}}\left(\frac{1}{R_V}-\frac{1}{R_{air}}\right)\frac{dP_V}{dt}$$

$$= -\frac{R_{mix}^2}{P_{mix}}\left(\frac{1}{R_V}-\frac{1}{R_{air}}\right)\frac{dP_V}{dT_{mix}}\cdot\frac{dT_{mix}}{dt}$$

Since the mixture has 100% relative humidity, we can use the Clausius-Clapeyron equation to evaluate dP_V/dT_{mix}

$$\frac{dP_V}{dT_{mix}} = \frac{\rho_V h_{fg}}{T_{mix}} = \frac{h_{fg}\,P_{mix}}{R_V\,T_{mix}^2}$$

Estimate $P_V(t+\Delta t)$ as (8)

$$P_V(t+\Delta t) = P_V(t) + \left\{\frac{h_{fg}\,P_{mix}}{R_V\left[T_{mix}(t)\right]^2}\right\}\left\{T_{mix}(t+\Delta t)-T_{mix}(t)\right\}$$

If Eq. (8) predicts $P_V(t+\Delta t) \geq P_{mix}$ = atmospheric pressure

- then use $P_V(t+\Delta t) = P_{mix}$
- otherwise use Eq. (8) to compute ~~P_{mix}~~ FD $P_V(t+\Delta t)$

7/20/97

Combining all these relations, gives

$$\frac{d\rho_{mix}}{dt} = \left[\left\{ \frac{h_{fg}\, P_{mix}}{R_V \left[T_{mix}(t) \right]^3} \right\} \left\{ \frac{1}{R_V} - \frac{1}{R_{air}} \right\} \right. \tag{9a}$$

$$\left. - \frac{P_{mix}}{R_{mix} \left[T_{mix}(t) \right]^2} \right] \left\{ \frac{T_{mix}(t+\Delta t) - T_{mix}(t)}{\Delta t} \right\}$$

or

$$\frac{d\rho_{mix}}{dt} = - \frac{P_{mix}}{R_{mix} \left[T_{mix}(t) \right]^2} \left\{ \frac{T_{mix}(t+\Delta t) - T_{mix}(t)}{\Delta t} \right\} \tag{9b}$$

Eq. (9a) applies when $P_V(t+\Delta t) \leq P_{mix}$ from Eq. (8)

Eq. (9b) applies when $P_V(t+\Delta t) > P_{mix}$ from Eq. (8)

→ The right hand side of Eqs (9) ~~at~~ F.D. is known from conditions at time "t" (as soon as $T_{mix}(t+\Delta t)$ is computed)

F.D 7/30/97

— ENERGY BALANCE TO COMPUTE $T_{mix}(t+\Delta t)$

$$\Delta(\text{energy}) = \text{energy in} - \text{energy out}$$

$$\Delta(\text{energy}) = \Delta t \left\{ \frac{d}{dt}\left[\rho_{mix} \forall C_{mix} T_{mix} \right] \right\}$$

$$\text{Energy in} - \text{Energy Out} = \Delta t \left\{ \dot{m}_g C_v T_g + \underbrace{\sum \dot{m}_e C_v T_L \Delta z}_{\text{sum over all liquid elements evaporating}} \right.$$

$$\left. + \underbrace{\sum \rho_L \delta C_v T_v \Delta z}_{\text{sum over all elements that boiled away from "t" to "t+}\Delta t\text{"}} \right.$$

$$\left. + \underbrace{\sum \rho_{mix} C_{mix} T \lambda \Delta z}_{\text{sum over elements that came in contact with dry wall from "t" to "t+}\Delta t\text{"}} \right.$$

$$\left. - \Delta t \left\{ \underbrace{\sum \dot{m}_c C_v T_{mix} \Delta z}_{\text{sum over all elements condensing}} \right\} \right.$$

C_v = specific heat of vapor
C_L = specific heat of liquid
C_{mix} = specific heat of air — vapor mixture

FD 7/30/97

Note:

$$C_{mix} = C_{air}\left(\frac{R_{mix}}{R_{air}}\right)\left(1 - \frac{P_v}{P_{mix}}\right) + C_v\left(\frac{R_{mix}}{R_v}\right)\left(\frac{P_v}{P_{mix}}\right) \qquad (10)$$

– P_v evaluated as before

Combining preceding relations

$$T_{mix}(t+\Delta t) = T_{mix}(t) + \underbrace{\left\{\frac{T_{mix}(t)}{\rho_{mix}(t)\,C_{mix}(t)\,V(t)}\right\}\frac{d}{dt}\left\{\rho_{mix}\,C_{mix}\,V\right\}}_{\text{very small term}}$$

$$+\Big[\Delta t\Big\{\dot{m}_g(t)\,C_v\,T_g(t) + \sum \dot{m}_e(z,t)\,C_{v_L}\,T_L(z,t)\,\Delta z$$

MA.

$$- \sum \dot{m}_c(z,t)\,C_v\,T_{mix}(t)\,\Delta z\Big\}$$

$$+ \sum \rho_L\,\delta(z,t)\,C_v\,T_v\,\Delta z + \sum \rho_{mix}(t)\,C_{mix}(t)\,T(z,t)\,\lambda\,\Delta z\Big]$$

$$\times\left[\rho_{mix}(t)\,C_{mix}(t)\,V(t)\right]^{-1}$$

Neglecting the small term gives:

7/30/97

$$T_{mix}(t+\Delta t) = T_{mix}(t)$$

$$+ \frac{\Delta t}{\left[\rho_{mix} C_{mix} \forall\right]_t} \left\{ \dot{m}_g C_V T_g + \sum \dot{m}_e C_V T_L \Delta z - \sum \dot{m}_e C_V T_{mix} \Delta z \right\}_t \qquad (11)$$

$$+ \sum_{\substack{\text{Bare wall} \\ \text{elements}}} \left[\left(\frac{\lambda \Delta z}{\forall} \right) T \right]_t + \sum_{\substack{\text{elements} \\ \text{boiled away} \\ \text{from "t" to "t+}\Delta t\text{"}}} \left[\frac{\rho_L \delta C_V T_V \Delta z}{\rho_{mix} C_{mix} \forall} \right]_t$$

m.A.

where subscript "t" means parameters at evaluated at time "t"

- We have now enough relations to compute all the parameters of the hot gas mixture at time $t+\Delta t$, starting with known conditions at time t.

Procedure is given on next page

7/30/97

Corrected Eq. (11) after process gets started $t > 0$

$$T_{mix}(t+\Delta t) = T_{mix}(t) + \frac{\Delta t}{[\rho_{mix} C_{mix} \forall]_t} \left\{ \dot{m}_g C_V T_g + \sum_{\substack{\text{evap.}\\ \text{elements}}} \dot{m}_e C_L T_L \Delta z - \sum_{\substack{\text{condensing}\\ \text{elements}}} \dot{m}_c C_V T_{mix} \Delta z \right\} \quad \boxed{11}$$

$$+ \left\{ \sum_{\substack{\text{dry wall}\\ \text{elements}}} k_w \left(\frac{dT}{dy}\right)_t \Delta z \Delta t + \sum_{\substack{\text{elements that boiled}\\ \text{off from } t \text{ to } t+\Delta t}} \rho_L \delta C_V T_V \Delta z \right\} \frac{1}{(\rho_{mix} C_{mix} \forall)_t}$$

Note changed term

See page 83 for equations to get started $t=0$

F.D 11/7/97

STEPS IN ORDER

1. Compute new mixture temperature:

$$T_{mix}(t+\Delta t) = T_{mix}(t) + \frac{\Delta t}{[\rho_{mix} C_{mix} \forall]_t} \left\{ \dot{m}_g C_V T_g + \sum_{\text{evap. elements}} \dot{m}_e C_{V_L} T_L \Delta z - \sum_{\text{cond. elements}} \dot{m}_c C_V T_{mix} \Delta z \right\}_t \quad \boxed{11}$$

F.D

See rewrite on page 32 F.D.

$$+ \sum_{\substack{\text{new bare wall}\\ \text{elements "t" to "t+}\Delta t\text{"}}} \left[\frac{\lambda T \Delta z}{\forall} \right]_t + \sum_{\substack{\text{new boiled off}\\ \text{elements "t" to "t+}\Delta t\text{"}}} \left[\frac{\rho_L \delta C_V T_V \Delta z}{\rho_{mix} C_{mix} \forall} \right]_t$$

F.D. F.D.

2. Compute trial partial pressure of vapor in mixture: $\boxed{8}$

$$P_V(t+\Delta t) = P_V(t) + \left[\frac{h_{fg} P_{mix}}{R_V T_{mix}^2} \right]_t \left[T_{mix}(t+\Delta t) - T_{mix}(t) \right]$$

- If $P_V(t+\Delta t) \geq P_{mix}$ then $P_V(t+\Delta t) = P_{mix}$
- Otherwise $P_V(t+\Delta t) =$ Eq. (8)

FD 7/20/97

3. Compute $d\rho_{mix}/dt$:

$$\frac{d\rho_{mix}}{dt} = \left\{ \left(\frac{h_{fg} P_{mix}}{R_V T_{mix}^3} \right) \left(\frac{1}{R_V} - \frac{1}{R_{air}} \right) - \frac{P_{mix}}{R_{mix} T_{mix}^2} \right\}_t \left[\frac{T_{mix}(t+\Delta t) - T_{mix}(t)}{\Delta t} \right] \qquad \boxed{9a}$$

if $P_V(t+\Delta t) < P_{mix}$

or

$$\frac{d\rho_{mix}}{dt} = -\left(\frac{P_{mix}}{R_{mix} T_{mix}^2} \right)_t \left[\frac{T_{mix}(t+\Delta t) - T_{mix}(t)}{\Delta t} \right] \qquad \boxed{9b}$$

if $P_V(t+\Delta t) = P_{mix}$

4. Compute $\forall(t+\Delta t)$

$$\forall(t+\Delta t) = \forall(t) + \Delta\forall(t) \qquad z_4(t) = \frac{\forall(t)}{\lambda}$$

$$\Delta\forall(t) = \frac{\Delta t}{\rho_{mix}(t)} \left\{ \dot{m}_g + \sum_{\text{evap. elements}} \dot{m}_e \Delta z - \sum_{\text{cond. elements}} \dot{m}_c \Delta z \right\}_t \qquad \boxed{6}$$

$$+ \sum \left(\frac{\rho_L \delta \Delta z}{\rho_{mix}} \right)_t - \left(\frac{\forall}{\rho_{mix}} \right)_t \frac{\partial \rho_{mix}}{\partial t} \Delta t$$

F.D. new boil off elements "b" to "t+Δt"

$$z_4(t+\Delta t) = z_4(t) - \frac{\Delta\forall(t)}{\lambda}$$

F.D. 7/20/97

with the relations that

$$\rho_{mix} = \frac{P_{mix}}{R_{mix} T_{mix}}$$

$$R_{mix} = \frac{P_{mix}}{P_V\left(\frac{1}{R_V} - \frac{1}{R_{air}}\right) + \frac{P_{mix}}{R_{air}}}$$

$$C_{mix} = C_{air}\left(\frac{R_{mix}}{R_{air}}\right)\left(1 - \frac{P_V}{P_{mix}}\right) + C_V\left(\frac{R_{mix}}{R_V}\right)\left(\frac{P_V}{P_{mix}}\right)$$

NOTES

- If $z_4(t+\Delta t) < 0$ then $z_4(t+\Delta t) = 0$

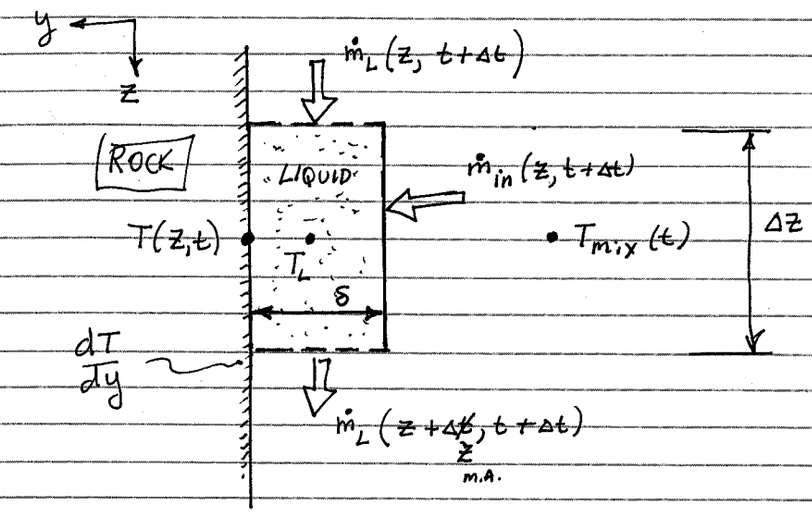
  and $\forall(t+\Delta t) = \forall(t)$

- "Boiled off" elements are computed as described later

NOW — Compute liquid film responses!

F.D 7/30/97

GENERAL LIQUID ELEMENT - MARCH FROM TOP TO BOTTOM



$$T_L(z,t) = \frac{1}{2}\left[T(z,t) + T_{mix}(t)\right] \qquad (12)$$

- If $T_L(z,t) \geq T_V$
  - liquid element boils off and its mass is added to the gas mixture at time $t+\Delta t$ (Eqs. 11 and 6). Its mass is $\rho_L \cdot \delta(z,t) \cdot \Delta z$

- If $T_L(z,t) < T_V$ and $T_L(z,t) \geq T_{mix}(t) > T_\infty$
  - liquid layer thickness decreases during Δt by evaporation
  - evaporation rate per unit depth $= \dot{M}_e$

F.D. 7/30/97

- If $T_L(z,t) < T_V$ and $T_L(z,t) < T_{mix}(t) > T_0$
  - liquid layer thickness increases because of condensation
  - condensation rate per unit depth $= \dot{m}_c$

- If $T_{mix}(t) = T_0$ and $T(z,t) = T_0$
  - liquid neither evaporates or condenses

<u>Assumption</u>

- Δz is chosen small enough so that any one liquid element either evaporates or condenses (or does neither) but does not evaporate over part of the interface and condense over the other part.

<u>MASS BALANCE</u>

$$\Delta(\text{mass}) = \text{mass in} - \text{mass out}$$

$$\Delta(\text{mass}) = \Delta t \cdot \frac{d}{dt}\left(\rho_L \delta \Delta z\right) = \rho_L \Delta z \frac{d\delta}{dt} \Delta t$$

$$\text{mass in} = \left\{\dot{m}_L(z, t+\Delta t) + \left[\dot{m}_{in}(z, t+\Delta t)\right] \times \Delta z\right\} \Delta t$$

$$\text{mass out} = \left\{\dot{m}_L(z+\Delta z, t+\Delta t)\right\} \Delta t$$

ED 7/30/97

So the result is

$$\dot{m}_L(z+\Delta z, t+\Delta t) = \dot{m}_L(z, t+\Delta t) + \left[\dot{m}_{in}(z, t+\Delta t)\right]\Delta z - \rho_L \frac{d\delta}{dt}\Delta z \qquad (13)$$

— need δ and $d\delta/dt$

— from page 11

$$\delta\left(z+\frac{\Delta z}{2}, t+\Delta t\right) = \left[\frac{3\mu_L}{\rho_L^2 g}\right]^{1/3}\left[\dot{m}_L\left(z+\frac{\Delta z}{2}, t+\Delta t\right)\right]^{1/3}$$

— so

$$\frac{d\delta}{dt} = \frac{1}{3}\left[\frac{3\mu_L}{\rho_L^2 g}\right]^{1/3}\left[\dot{m}_L\left(z+\frac{\Delta z}{2}, t+\Delta t\right)\right]^{-2/3}\frac{d\dot{m}_L\left(z+\frac{\Delta z}{2}, t+\Delta t\right)}{dt}$$

furthermore: (mass balance)

$$\frac{d\dot{m}_L\left(z+\frac{\Delta z}{2}, t+\Delta t\right)}{dt} \approx \frac{\dot{m}_L(z, t+\Delta t) - \dot{m}_L(z, t)}{\Delta t} + \frac{d\dot{m}_{in}\Delta z}{dt}\Delta t$$

since $\dot{m}_{in} = \dfrac{k_L(T_{mix}-T)}{\delta h_{fg}}$

$$\text{so} \quad \frac{d\dot{m}_{in}}{dt} = -\frac{k_L}{\delta h_{fg}}\frac{dT}{dt} - \frac{k_L(T_{mix}-T)}{\delta^2 h_{fg}}\frac{d\delta}{dt} + \frac{k_L}{\delta h_{fg}}\frac{dT_{mix}}{dt}$$

E.D. 7/30/97

or

$$\frac{d\dot{m}_{in}}{dt} = -\frac{\dot{m}_{in}}{T_{mix}-T}\frac{dT}{dt} - \frac{\dot{m}_{in}}{\delta}\frac{d\delta}{dt} + \frac{\dot{m}_{in}}{T_{mix}-T}\frac{dT_{mix}}{dt}.$$

Altogether:

$$\frac{d\delta}{dt} = \frac{1}{3}\left[\frac{\delta}{\dot{m}_L^{1/3}}\right]\left[\dot{m}_L\right]^{-2/3} *$$

$$\left\{\frac{\dot{m}_L(z,t+\Delta t) - \dot{m}_L(z,t)}{\Delta t} - \frac{\dot{m}_{in}\Delta z}{T_{mix}-T}\left[\frac{T(t+\Delta t)-T(t)}{\Delta t}\right] - \frac{\dot{m}_{in}\Delta z}{\delta}\frac{d\delta}{dt} + \frac{\dot{m}_{in}\Delta z}{T_{mix}-T}\left[\frac{T_{mix}(t+\Delta t)-T_{mix}(t)}{\Delta t}\right]\right\}$$

or collecting terms

$$\frac{d\delta}{dt}\left\{1 + \frac{[\dot{m}_{in}(z,t+\Delta t)]\Delta z}{3[\dot{m}_L(z,t+\Delta t)]}\right\} =$$

$$\frac{\delta(z,t)}{3\,\dot{m}_L(z,t+\Delta t)}\left[\frac{\dot{m}_L(z,t+\Delta t) - \dot{m}_L(z,t)}{\Delta t} + \left(\frac{\dot{m}_{in}\Delta z}{T_{mix}-T}\right)_{t+\Delta t}\left[\frac{T_{mix}(t+\Delta t) - T(z,t+\Delta t)}{\Delta t} - \frac{T_{mix}(t) - T(z,t)}{\Delta t}\right]\right]$$

very small ~ negligible term

F.D. 7/30/77

So, rewriting for clarity

$$\cancel{\delta(z,t) = \left[\frac{3\mu_L}{\rho_L^2 g}\right]^{1/3}\left[\dot{m}_L(z,\right.} \quad \text{F.D.}$$

$$\delta(z,t+\Delta t) = \left[\frac{3\mu_L}{\rho_L^2 g}\right]^{1/3}\left[\dot{m}_L(z,t+\Delta t)\right]^{1/3}$$

= thickness of element at time $t+\Delta t$

(Note, here we can use $\dot{m}_L$ at z rather than $z+\frac{\Delta z}{2}$)

and (neglecting small terms)

$$\frac{d\delta}{dt}(z,t+\Delta t) = \frac{\left[\delta(z,t+\Delta t)\right]}{3\left[\dot{m}_L(z,t+\Delta t)\right]}\left[\frac{\dot{m}_L(z,t+\Delta t) - \dot{m}_L(z,t)}{\Delta t}\right] \div$$

$$\left\{1 + \frac{\left[\dot{m}_{in}(z,t+\Delta t)\right]\Delta z}{3\left[\dot{m}_L(z,t+\Delta t)\right]}\right\}$$

or

$$\frac{d\delta}{dt} = \left\{\frac{\left[\dfrac{\dot{m}_L(z,t+\Delta t) - \dot{m}_L(z,t)}{\Delta t \cdot \dot{m}_L(z,t+\Delta t)}\right]}{\cancel{\left[1 + \dfrac{\dot{m}_{in}\Delta z}{3\dot{m}_L}\right]_{t+\Delta t}}}\right\}\frac{\delta(z,t+\Delta t)}{3} \qquad (14)$$

GW

F.D. 7/30/97

Mass addition term:

$$\dot{m}_{in} = \frac{k_L}{h_{fg}} \left[\frac{T_{mix}(t+\Delta t) - T(z, t+\Delta t)}{\delta(z, t+\Delta t)} \right] \qquad (15)$$

Note: Eq. (15) automatically compensates for $\pm$ sign for condensation or evaporation

ENERGY BALANCE

– not needed – the liquid temperature is determined by Eq. (12) on pg 39

ROCK Temperature Gradient @ WALL $y=0$

$$\frac{dT(z,t+\Delta t)}{dy} = - \frac{[\dot{m}_{in}(z,t+\Delta t)]\, h_{fg}}{k_W} \qquad (16a)$$

$$= - \frac{k_L}{k_W} \left[\frac{T_{mix}(t^{+\Delta t}) - T(z, t^{+\Delta t})}{\delta(z, t+\Delta t)} \right]^{F.D} \qquad (16b)$$

– Can use either Eq. (16a) or (16b).

– ~~Note that the gradient actually is computed using parameters at time "t"~~ F.D

F.D 7/30/97

– Note that $\frac{dT}{dy}(z, t+\Delta t)$ cannot be computed until, for example, $T(z, t+\Delta t)$ is computed. Thus, the boundary condition at the rock wall [i.e., dT/dy] will always lag behind by one Δt — this boundary condition is needed for the rock matrix computer code.

SOLUTION PROCEDURE FOR LIQUID FILM

1. Everything is known for time "t"
2. March downward to compute everything at time $t + \Delta t$
3. Start at top, $z=0$, where

$$\dot{m}_L(0, t+\Delta t) = \dot{m}_L(0, t) = \dot{m}_0$$

4. Compute $\dot{m}_L(z=\Delta z, t+\Delta t)$ based on $\dot{m}_L(z=0, t+\Delta t)$, etc., by preceding equations (summarized later).

F.B. 7/30/97

5. If $\dot{m}_L(z=\Delta z, t+\Delta t) > 0$ proceed to next element, to compute $\dot{m}_L(z=2\Delta z, t+\Delta t)$, and so on.

6. If for any element $\dot{m}_L(z+\Delta z, t+\Delta t) =$ flow out bottom of element is < 0, that element has completely evaporated away, and is removed. All lower elements should already have been removed.

7. If $\dot{m}_L(z, t+\Delta t) > 0$ for the lowermost element (i.e., at the heater), the flow collects on the heater in a puddle. The computer code for the heater and rock must be used to determine if the flow boils off to generate a $\dot{m}_g$. Presumably the boiling will be "instantaneous" if the heater temperature $> T_v$.

E.D 7/30/97

8. Check to ensure that the rock temperature $T(z,t)$ is still above T_v for all the "dry" elements where the liquid has evaporated or boiled away.

9. When heater power starts to decay, the higher elevations of the rock will begin to decrease in temperature (from the computer code). When the temperature of the "dry" elements decreases below T_v, vapor will start to condense on the dry wall. This also means that $\dot{m}_L(z, t+\Delta t)$ for the liquid element just above the dry wall element is now > 0. [If not, the calculations will have to proceed further along in time ~~until~~ to until $\dot{m}_L > 0$ for the above element.]

Add an element to the dry wall with a thickness of

from element above.

$$\delta = \frac{1}{3}\left[\frac{\mu_L}{\rho_L^2 g}\right]^{1/3}\left[\dot{m}_L\right]^{1/3}$$

F.2 7/30/57

10. Since ~~If~~ FD the flow $\dot{m}_L(z+\Delta z, t+\Delta t)$ out the bottom of this new element is > 0, then:

(a) If the rock wall temperature below this element is still $> T_v$, vaporize the $\dot{m}_L(z+\Delta z, t+\Delta t)$ flow and add to the hot gas mixture as a $\dot{m}_g$ contribution

(b) If the rock wall temperature below this element is $< T_v$, add another new element, and proceed as above.

FD 7/30/97

SUMMARY OF LIQUID "MARCHING" EQUATIONS

- Liquid flow @ $z=0$:

$$\dot{m}_L(z=0, t+\Delta t) = \dot{m}_L(z=0, t) = \dot{m}_0 \qquad \boxed{17}$$

- Liquid temperature for any element:

$$T_L(z, t+\Delta t) = \frac{1}{2}\left[T(z, t+\Delta t) + T_{mix}(t+\Delta t)\right] \qquad \boxed{12}$$

- Liquid film thickness for any element

$$\delta(z, t+\Delta t) = \left[\frac{3\mu_L}{\rho_L^2 g}\right]^{1/3}\left[\dot{m}_L(z, t+\Delta t)\right]^{1/3} \qquad \boxed{18}$$

— Eq. (18) is valid at $z=0$

- Liquid flow <u>out</u> of any liquid element and <u>into</u> the element below:

$$\dot{m}_L(z+\Delta z, t+\Delta t) = \dot{m}_L(z, t+\Delta t) + \left[\dot{m}_{in}(z, t+\Delta t)\right]\cancel{\Delta t}\,\Delta z \quad \text{GW}$$
$$- \rho_L \frac{d\delta}{dt}\cancel{\Delta t}\,\Delta z \quad \text{GW} \qquad \boxed{13}$$

- Evaporation or Condensation flow into a liquid element:

$$\dot{m}_{in}(z, t+\Delta t) = \frac{k_L}{h_{fg}}\left[\frac{T_{mix}(t+\Delta t) - T(z, t+\Delta t)}{\delta(z, t+\Delta t)}\right] \qquad \boxed{15}$$

F.D 7/30/97

- rate of change of liquid film thickness

$$\frac{d\delta}{dt} = \frac{\dot{m}_L(z,t+\Delta t) - \dot{m}_L(z,t)}{3\,\Delta t \cdot \dot{m}_L(z,t+\Delta t)} \cdot \left\{ \frac{\delta(z,t+\Delta t)}{1 + \dfrac{\dot{m}_{in}(z,t+\Delta t)\cdot \Delta z}{3\,\dot{m}_L(z,t+\Delta t)}} \right\} \qquad \boxed{14}$$

- Boundary condition for rock wall ($y=0$) at time $t+\Delta t$

$$\frac{dT}{dy}(z,t+\Delta t) = -\frac{k_L}{k_W}\left[\frac{T_{mix}(t) - T(z,t)}{\delta(z,t)}\right] \qquad \boxed{16b}$$

INITIAL CONDITIONS @ $t=0$
(suggested)

- heater power = 0
- $\dot{m}_L = \dot{m}_0$ for all elements
- all temperatures $= T_0$
  (i.e., no evaporation or condensation)
- turn heater on at $t > 0$

F.D 7/30/97

Notebook handed over to Gt. Ofoegbu (from M. Ahola) on Sept 09 1997.

Investigators

Frank Dodge

Ron Green

Mikko Ahola (up to Sept 19 1997)

Goodluck Ofoegbu (GIO) — from Sept 09 1997.

October 9 1997 GIO

Gravity-Driven Refluxing Model
Based on Film Flow With No Imbibition

Equations by *Frank Dodge*
Material-parameter tables (parameter values versus temperature) by *Mikko Ahola*
Numerical Implementation by *Goodluck I Ofoegbu*

Outline of Numerical Algorithm

The model will be implemented through four code modules as follows.

Air-Vapor-Mix Module: This module will be implemented as a functional data structure *AirVaporMix* that is equiped with storage space and a group of internal functions to keep track of mix temperature (T_{mix}), mix volume (V), incremental mix volume (Δv), z-coordinate of mix base (Z_4), and the evaporation, condensation, boil-off, bare-rock, and rock-vapor terms in the energy and mass-balance equations.

Liquid-Element Module: This module will be implemented as a functional data structure *LiquidElem* that is equiped with storage space and a group of internal functions to keep track of properties and attributes of the liquid cells (i.e., film segments). This module provides the wall-normal temperature gradient to, and receives wall temperature from, the rock-temperature module. It receives mixture temperature, T_{mix}, and base coordinate, Z_4, from the *AirVaporMix* element, and provides state-variable values required to update the evaporation, condensation, boil-off, bare-rock, and rock-vapor terms of the energy and mass-balance equations, which are stored and evaluated in the *AirVaporMix* element.

Rock-Temperature Module: This module computes rock temperature using an explicit integrated finite difference algorithm. It supplies wall-rock temperature to *LiquidElem* elements, which, in turn, supply this module with values of wall-normal temperature gradient.

Iteration Module: This module invokes member functions of the *AirVaporMix* element to update the properties and attributes of the air-vapor mix. Thereafter, the module steps through the *LiquidElem* elements from top to base, updating their status, properties, and attributes. Also, it invokes member functions of the *AirVaporMix* element to update the evaporation, condensation, boil-off, bare-rock, and rock-vapor terms of the energy and mass-balance equations, depending on the status of the various *LiquidElem* elements.

GIO

The following implementation is developed for C++ language.

October 17 1997

GW

1.0 Liquid Element Data Members

itsHeight	Δz	[m]
itsBase	$z+\Delta z$	[m]
preWidth	$\delta(t)$	[m]
curWidth	$\delta(t+\Delta t)$	[m]
preFluxDown	$m_L(z+\Delta z,t)$	[kg/s/m]
curFluxDown	$m_L(z+\Delta z,t+\Delta t)$	[kg/s/m]
wallTemp	$T_w(z+\Delta z/2,t+\Delta t)$	[K]
fluxFromVapor	m_{in}	[kg/s/m/m]
dDdt	$d\delta/dt$	[m/s]
dTdy	dT/dy	[K/m]
boiledOff	TRUE/FALSE	[TRUE=1, FALSE=0]

Notes

Data member wallTemp is computed through separate analysis of wall rock, which is coupled to liquid-vapor-air analyses through wall-normal temperature gradient dTdy. Wall-rock analysis should ensure that wallTemp increases downward, i.e., k.wallTemp >= j.wallTemp for each element j underlain by element k.

Value of downward flux into a liquid element is stored with the overlying element. Downward flux into top element equals the input infiltration rate, m_o.

Liquid element data variables are evaluated at time $t+\Delta t$, starting with the top element and marching down one element at a time.

GW

2.0 March-Down Procedure For Liquid Elements

Element index: k=1 at top; $k=k_{max}$ at base.
Element pointer: pL for current element; poL for overlying element.
AirVaporMix element reference: ~~rMixElem~~ theMix (supplied to all functions that need it) GW
Top-boundary temperature: T_{bt}
Top-boundary flux: m_o
Boiling temperature: T_v
T_{mix} = ~~rMixElem~~ theMix.GetTemperature() GW

For k=1

T_w = pL->GetWallTemp()
$T_L = (T_w + T_{mix})/2$

```
if (T_L >= T_v)
        BoilOffFlux(pL,m_o,T_bt)
        if (pL->HasBoiledOff() is FALSE) BoilOffElem(pL)
        endif
else    UpdateLiquidElem(pL, m_o, m_o)          [T_L < T_v]
endif
```


GW

For k>1

ctf = current value of downward flux in element k-1
ptf = previous value of downward flux in element k-1

```
if (poL->HasBoiledOff() and pL->HasBoiledOff())
        BoilOffFlux(pL,0.0,0.0)
else if (poL->HasBoiledOff() and (pL->HasBoiledOff() is FALSE))
        BoilOffElem(pL)
else                                    [poL->HasBoiledOff() is FALSE]
        ptf = poL->GetPreFluxDown()
        ctf = poL->GetCurFluxDown()
        topTemp = (poL->GetWallTemp() + T_mix)/2
        T_L = (pL->GetWallTemp() + T_mix)/2
        if (T_L < T_v) UpdateLiquidElem(pL,ptf,ctf)
        else                            [T_L >= T_v]
                BoilOffFlux(pL,ctf, topTemp)
                if (pL->HasBoiledOff() is FALSE)
                        BoilOffElem(pL)
                endif
        endif
endif
```

GW

3.0 Functions that Operate on Liquid Elements

pL = pointer to liquid element; ctf and ptf are current and previous fluxes in overlying element
Reference to AirVaporMix element ~~(rMixElem)~~ theMix is assumed accessible GW
T_v = boiling temperature

3.1. BoilOffElem(pL) GW [Boil off existing liquid element]

Δz = pL->GetHeight()
δ = pL->GetCurWidth()
$C_v = C_v(T_v)$; $\rho_L = \rho_L(T_v)$ [Evaluate C_v and ρ_L at temperature T_v] GW
~~rMixElem~~ theMix.AddToBoilOff($\rho_L\delta C_v T_v\Delta z$, $\rho_L\delta\Delta z$) GW
pL->SetBoiledOff(TRUE)
pL->SetWidth(0.0)
pL->SetFluxDown(0.0)
pL->SetdTdy()
return

3.2. BoilOffFlux(pL,mTop, topTemp) [Vaporize incoming flux]

mTop = incoming flux
topTemp = temperature of incoming flux

if (mTop > 0.0)
$C_v = C_v$(~~topTemp~~ T_v) GW [Evaluate C_v at temperature ~~topTemp~~ GW T_v]
~~rMixElem~~ theMix.AddToRokVapor((mTop*C_v*~~topTemp~~ T_v), mTop)
endif

GW

```
if (pL->HasBoiledOff())                [Element is already bare-wall element]
        Δz = pL->GetHeight()
        T_w = pL->GetWallTemp()
        rMixElem.AddToBareWall(λ*T_w*Δz)
        pL->SetdTdy()
endif
return
```

3.3. UpdateLiquidElem(pL, ptf, ctf) [Update values of liquid-element data]

```
T_L = (pL->GetWallTemp() + T_mix)/2
pL->SetBoiledOff(FALSE)
pL->SetWidth(ctf)
pL->SetFluxFromVapor()
m_in = pL->GetFluxFromVapor()
if (m_in > 0.0)            [m_in is a condensation flux]
        C_v = C_v(T_mix)
        rMixElem.AddToCondensate((m_in*C_v*T_mix*Δz), (m_in*Δz))
endif
if (m_in < 0.0)            [m_in is an evaporation flux]
        C_L = C_L(T_L)
        rMixElem.AddToEvaporate((–m_in*C_L*T_L*Δz), (–m_in*Δz))
endif
pL->SetdDdt(ptf, ctf)
pL->SetFluxDown(ctf)
pL->SetdTdy()
return
```

GW

4.0 Liquid Element Member Functions

```
T_mix = rMixElem.GetTemperature()
T_L = (wallTemp + T_mix)/2
dTime = Time increment
```

4.1. SetWidth(topFlux) [Compute film width (δ) given flux at top of element]

```
preWidth = curWidth
curWidth = 0.0
if (topFlux > 0.0)
        g = gravitational acceleration    [m/s^2]
        ρ_L = ρ_L(T_L)                    [kg/m^3]
        μ_L = μ_L(T_L)                    [kg/m/s]
        curWidth = [Eq. (18), p. 59 of Notebook]
endif
return
```


GW

4.2. SetFluxFromVapor() [Compute condensation (or evaporation) flux, m_{in}]

```
z4 = rMixElem.GetZHotCold()
zTop = itsBase – itsHeight
fluxFromVapor = 0.0    [No condensation/evaporation if element is above cold/hot gas interface]

if (zTop >= z4)
        k_L = k_L(T_L)
        h_fg = h_fg(T_L)
        fluxFromVapor = [Eq. (15), p. 59 of Notebook]
endif
return
```

4.3. SetdDdt(preTopFlux, curTopFlux) [Compute dδ/dt]

Input Variables:

- curTopFlux = value of curFluxDown in overlying element [$m_L(z,t+\Delta t)$]
- preTopFlux = value of preFluxDown in overlying element [$m_L(z,t)$]
- curWidth
- fluxFromVapor
- dTime

```
dDdt = [Eq. (14), p. 61 of Notebook]
return
```

4.4. SetFluxDown(topFlux) [Compute downward flux out of element, m_L]

Input Variables:

- topFlux = value of curFluxDown in overlying element [$m_L(z,t+\Delta t)$]
- current value of fluxFromVapor [$m_{in}(t+\Delta t)$]
- current value of dDdt [$d\delta/dt(t+\Delta t)$]
- dTime = time increment

GW

```
preFluxDown = curFluxDown
curFluxDown = 0.0
if (curWidth > 0.0)        [m_L is nonzero only if δ>0]
        ρ_L = ρ_L(T_L)
        curFluxDown = [Eq. (13), p. 59 of Notebook]
endif
if (curFluxDown < 0.0) curFluxDown = 0.0    [Should m_in be adjusted after this correction?]
return
```

4.5. SetdTdy() [Compute wall-normal temperature gradient dT/dy]

GW

```
k_w = thermal conductivity of wall rock
z4 = rMixElem.GetZHotCold()
zTop = itsBase – itsHeight
```

GW

if (zTop < z4) [Element within cold-gas zone]
    dTdy = 0.0 GW
else if ~~(boiledOff)~~ (!(preWidth > 0)) [Use bare-rock approximation]
    dTdy = (T_{mix} – wallTemp)/λ
else
    $k_L = k_L(T_L)$ = liquid thermal conductivity at temperature T_L
    dTdy = [Eq. (16b), p. 61 of Notebook]
endif
return

4.6. Other Functions [Simple but important, mostly inline, functions]

4.6.1 SetBoiledOff(status)
    boiledOff = status [status=TRUE or status=FALSE]
    return

4.6.2 SetWallTemp(temp)
    wallTemp = temp
    return

4.6.3 GetCurWidth()
    return (curWidth)

GW

4.6.4 GetPreWidth()
    return (preWidth)

4.6.5 GetdDdt()
    return (dDdt)

4.6.6 GetFluxFromVapor
    return (fluxFromVapor)

4.6.7 GetdTdy()
    return (dTdy)

4.6.8 GetPreFluxDown()
    return (preFluxDown)

4.6.9 GetCurFluxDown()
    return (curFluxDown)

4.6.10 GetWallTemp()
    return (wallTemp)

4.6.11 HasBoiledOff()
    return (boiledOff)

GW

4.6.12 GetHeight()
    return (itsHeight)

4.6.13 GetBase()
    return (itsBase)

5.0 AirVaporMix Element Data Members

preTemp	T_{mix}~~(t)~~ (t−Δt)	[K]
curTemp	T_{mix}(~~t+Δt~~) (t)	[K]
preVol	V~~(t)~~ (t−Δt)	[m^3/m]
dVol	ΔV(~~t+Δt~~)(t) GW	[m^3/m]
~~prePv~~ GW	~~P_v(t) (t−Δt)~~	~~[Pa]~~
~~curPv~~ pVapor GW	P_v(~~t+Δt~~) (t)	[Pa]
preZHotCold	z_4~~(t)~~ (t−Δt)	[m]
curZHotCold	z_4(~~t+Δt~~)(t)	[m]
eRokVapor	$m_gC_vT_g$ accumulation term	[Eq. (11), p. 33 of Notebook]
eEvaporate	$m_eC_LT_L$ accumulation term	[Eq. (11), p. 33 of Notebook]
eCondensate	$m_cC_vT_{mix}$ accumulation term	[Eq. (11), p. 33 of Notebook]
eBareWall	$\lambda T_w\Delta z$ accumulation term	[Eq. (11), p. 33 of Notebook]
eBoilOff	$\rho_L\delta C_vT_v\Delta z$ accumulation term	[Eq. (11), p. 33 of Notebook]
vRokVapor	m_g accumulation term	[Eq. (6), p. 35 of Notebook]
vEvaporate	$m_e\Delta z$ accumulation term	[Eq. (6), p. 35 of Notebook]
vCondensate	$m_c\Delta z$ accumulation term	[Eq. (6), p. 35 of Notebook]
vBoilOff	$\rho_L\delta\Delta z$ accumulation term	[Eq. (6), p. 35 of Notebook]

GW

Notes

One AirVaporMix element (named mixElem) will be declared inside of main() and ~~the pointer~~ GW reference to this element, rMixElem, will be made available to functions that need it. AirVaporMix data members are evaluated at time t+Δt using liquid element properties at time t.

6.0 AirVaporMix Element Member Functions

6.1 ResetAccumulators()

eRokVapor = eEvaporate = eCondensate = eBareWall = eBoilOff = 0.0
vRokVapor = vEvaporate = vCondensate = vBoilOff = 0.0
return

6.2 AddToRokVapor(de,dv)

eRokVapor += de
vRokVapor += dv
return

GW

6.3 AddToEvaporate(de,dv)

eEvaporate += de
vEvaporate += dv
return

6.4 AddToCondensate(de,dv)

eCondensate += de
vCondensate += dv
return

6.5 AddToBoilOff(de,dv)

eBoilOff += de
vBoilOff += dv
return

6.6 AddToBareWall(de)

eBareWall += de
return

GW

6.7 SetTemperature() [Compute T_{mix}]

$P_{mix} = 1.01325 * 10^5$ [Pa, atmospheric pressure]
$R = 8314$ [J/kmol/K, universal gas constant]
$M_{air} = 28.97$ [kg/kmol, molecular mass of air]
$M_{water} = 18.02$ [kg/kmol, molecular mass of water]
$R_{air} = R/M_{air} = 287.0$ [J/kg/K, gas constant for air]
$R_v = R/M_{water} = 461.4$ [J/kg/K, gas constant for water vapor]
R_{mix} = [Eq. on p. 37 of Notebook]
$\rho_{mix} = P_{mix}/(R_{mix} * \cancel{preTemp})$ curTemp GW
$C_{air} = C_{air}(\cancel{preTemp})$ curTemp GW [Specific heat of air at temperature ~~preTemp~~ curTemp] GW
$Cv = Cv(\cancel{preTemp})$ curTemp [Specific heat of vapor at temperature ~~preTemp~~ curTemp]
C_{mix} = [Eq. on p. 37 of Notebook] [Specific heat of air-vapor mixture]
curVol = preVol + dVol [curVol now stores volume at time t]
eBareWall /= curVol [divide accumulated eBareWall by curVol]
eBoilOff /= ($\rho_{mix} * C_{mix}$ * curVol) [divide accumulated eBoilOff by product indicated]
ratio = dTime/($\rho_{mix} * C_{mix}$ * curVol)
preTemp = curTemp
curTemp += ratio * (eRokVapor + eEvaporate – eCondensate) + eBareWall + eBoilOff GW
return

6.8 SetPv() [Compute partial vapor pressure P_v]

$h_{fg} = h_{fg}$(preTemp)
prePv = curPv

GW

curPv += [$h_{fg} * P_{mix}/(R_v$*preTemp*preTemp)]*(curTemp – preTemp)
if (curPv > P_{mix}) curPv=P_{mix}
return

6.9 SetVolume() [Compute air-vapor mixture volume]

if (curPv < P_{mix})
    $h_{fg} = h_{fg}$(preTemp)
    $d\rho_{mix}/dt$ = [Eq. (9a), p. 35 of Notebook]
else
    $d\rho_{mix}/dt$ = [Eq. (9b), p. 35 of Notebook]
endif

preVol += dVol [preVol now stores volume at time t]
$\rho_{mix} = P_{mix}/(R_{mix} * preTemp)$
ratio = dTime/ ρ_{mix}
vBoilOff /= ρ_{mix} [Divide accumulated vBoilOff by ρ_{mix}]
dVol = ratio*(vRokVapor + vEvaporate – vCondensate) + vBoilOff – ratio*preVol*$d\rho_{mix}/dt$
return

GW

6.10 SetZHotCold() [compute z_4]

preZHotCold = curZHotCold
curZHotCold = preZHotCold – dVol/λ
return

6.11 GetPreVolume()
    return (preVol)

6.12 GetDVolume()
    return (dVol)

6.13 GetZHotCold()
    return (curZHotCold)

6.14 GetTemperature()
    return (curTemp)

October 31 1997

Governing Equation for Rock-Temperature Module

$$\rho C \frac{\partial T}{\partial t} + q_{i,i} = 0 \qquad (1)$$

ρ = rock density (kg/m^3)

C = Specific heat (J/kg-K)

T = temperature (K or C)

t = time (s)

$q_{i,i}$ = heat-flux gradient (J/s/m^2/m)

By integrating eq.(1) over a small volume V and using Gauss's theorem to convert volume integral of flux to surface integral (i.e., $\int_V q_{i,i}\, dv = \int_S q_n\, ds$) the following is obtained

$$\rho C \int_V \frac{\partial T}{\partial t} dv + \int_S q_n\, dS = 0 \qquad (2)$$

where q_n is flux normal to surface S that encloses volume V.

Numerical solution of eq. (2) for given boundary and initial conditions will be implemented using an Integrated Finite Difference scheme, as follows.

	m=3	
m=4	n	m=2
	m=1	

The medium is discretized into cells on a rectangular grid, such that, for a typical cell n, equation (2) is solved using information for cell ~~an~~ (Fig) n and 4 neighboring cells, m=1, 2, 3, and 4. The equation solved for cell n is as follows

$$[\rho C V]_n \left[\frac{\partial T}{\partial t}\right]_n = \sum_{m=1}^{m=4} \left[q_{mn} A_{mn}\right] \qquad (3)$$

q_{mn} = flux from cell m to cell n normal to boundary (of area A_{mn}) between cells m and n.

The expression $\partial T/\partial t$ is evaluted at time

$t + \Delta t$ using the explicit scheme:

$$\frac{\partial T}{\partial t} = \frac{{}^{t+\Delta t}T - {}^{t}T}{\Delta t} \qquad (4)$$

Δt = time increment.

${}^{t+\Delta t}T$ = temperature at $t + \Delta t$

${}^{t}T$ = temperature at t

The evaluation of q_{mn} depends on whether cell n is an interior or ~~exterior~~ boundary (Fig) cell. For an interior cell, all neighboring m cells exist, such that

$$q_{mn} = \frac{K}{L_{mn}}\left(T_m - T_n\right) \qquad (5a)$$

K = thermal conductivity (J/s-m-K)

L_{mn} = Distance of m to n normal to mn boundary.

T_n = temperature at n, at time t

T_m = temperature at m, at time $t+\Delta t$ or time t (e.g., for an upward-leftward sweep direction, T_m is known at time $t+\Delta t$ for m=1 and m=2 and at time t for m=3 and m=4, for the figure on p.74).

When n is a boundary cell, q_{mn} is evaluated

using one of two equations:

$$q_{mn} = K \frac{dT}{d(mn)} \qquad (5b)$$

for cell n next to the fracture wall, or

$$q_{mn} = q_{f(in)} \qquad (5c)$$

for cell n next to the heat source, where

$\frac{dT}{d(mn)} = \frac{dT}{dy}$ (−ve, so inward flow is +ve) = wall-normal temperature gradient (p. 61)

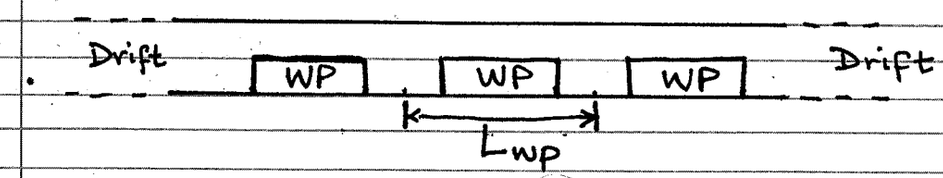
$q_{f(in)}$ = inward flux (into cell n) from the heat source.

The equation solved for each cell n is obtained by substituting eq. (4) into eq. (3), which gives

$$^{t+\Delta t}T_n = {}^{t}T_n + \frac{\Delta t}{[\rho C V]_n} \sum_{m=1}^{m=4} \left(A_{mn}\, q_{mn} \right) \qquad (6)$$

~~with~~ with q_{mn} evaluated using either eq (5a), (5b) or (5c).

Thermal-Load Calculation



- WP = Waste package.

L_{wp} = Waste-package spacing (along drift)

$$q_{f(in)} = \frac{Q_{wp}}{D_p L_{wp}}$$

Q_{wp} = power output (J/s) of ~~or~~ (Gro) 1 waste package (evaluated using computer code developed by R.D. Manteufel)

D_p = Drift cross-sectional-perimeter length.

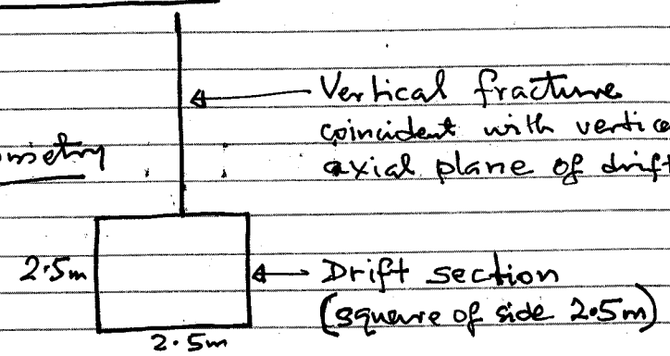
For example, $L_{wp} = 19$ m for 83 MTU/acre case with drift spacing of 22 m.

$D_p = 5\pi$ for 5-m diameter cross-section

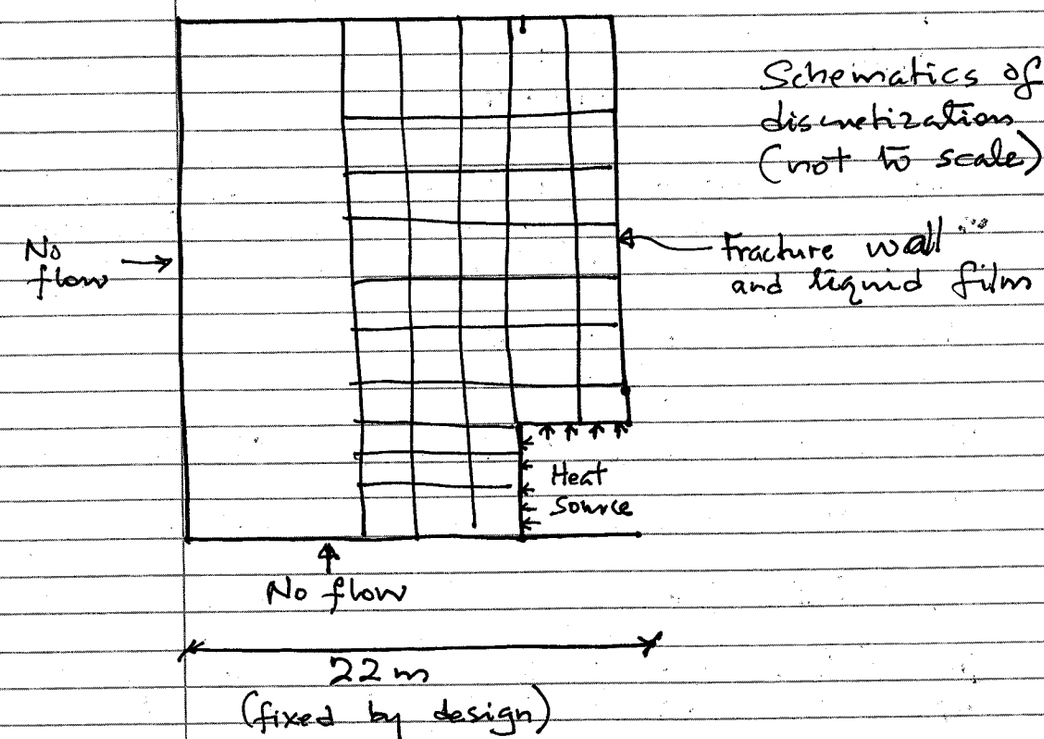
$D_p = 4L$ for drift of square cross-section of side length L.

Discretization

Problem Geometry



Vertical section normal to drift axis



Height of discretized section will vary.

Liquid-element index, say k, increases from 0 at the top boundary to ~~k_{max}~~ $k_{max}-1$ at the

drift roof. At the fracture wall, the vertical index of rock elements, say j, increases from n_d at the drift roof to ~~j_{max}~~ $j_{max}-1$ at the top boundary, where n_d is the number of ~~cells in the vert.~~ drift-section cells in the vertical direction.

$$k_{max} = j_{max} - n_d$$

Liquid elements supply wall-normal temperature gradient dT/dy to adjacent rock cells on the fracture wall. In turn, the rock cells supply wall temperature to the adjacent liquid elements

LiqElem[k] supplies $\frac{dT}{dy}$ to rockElem[~~$k_{max}-1-k+n_d$~~] $[k_{max}-1-k+n_d]$

rockElem[j] ($j >= n_d$) supplies ~~$\frac{dT}{dy}$~~ wall temperature to liqElem[$j_{max}-1-j$]

Verification of Rock-Temperature Code Module

A The rock-temperature problem without liquid film was solved to examine temperature profiles calculated using the code. Parameters used for the example are:

$\rho = 2{,}300\ kg/m^3$

$C = 932\ J/kg\text{-}K$

$K = 2.1\ J/m\text{-}s\text{-}K$

Model width = drift spacing = 22 m.

Model height = 50 m

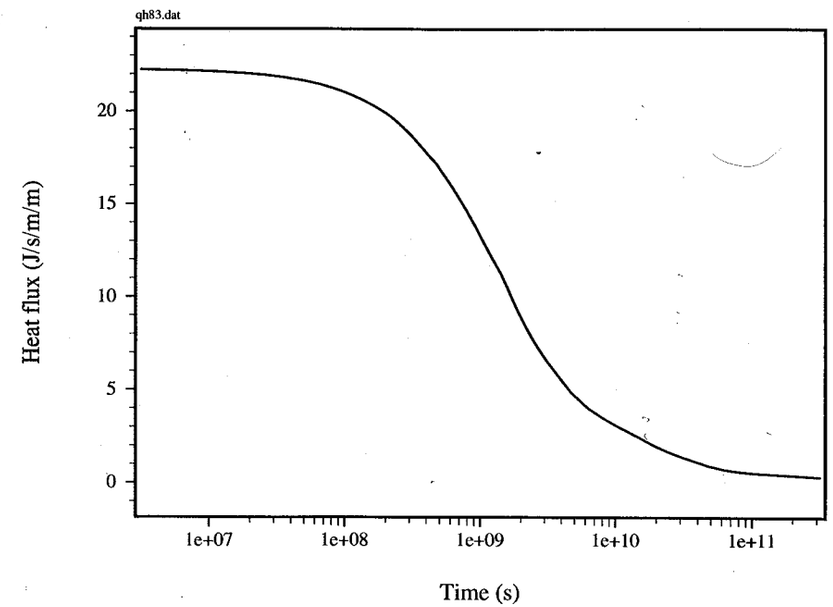
Cell size = 0.5 m square

Drift section = square of side ~~2.5 m~~ 5m (5 cells along each side of drift in model; only half of drift discretized).

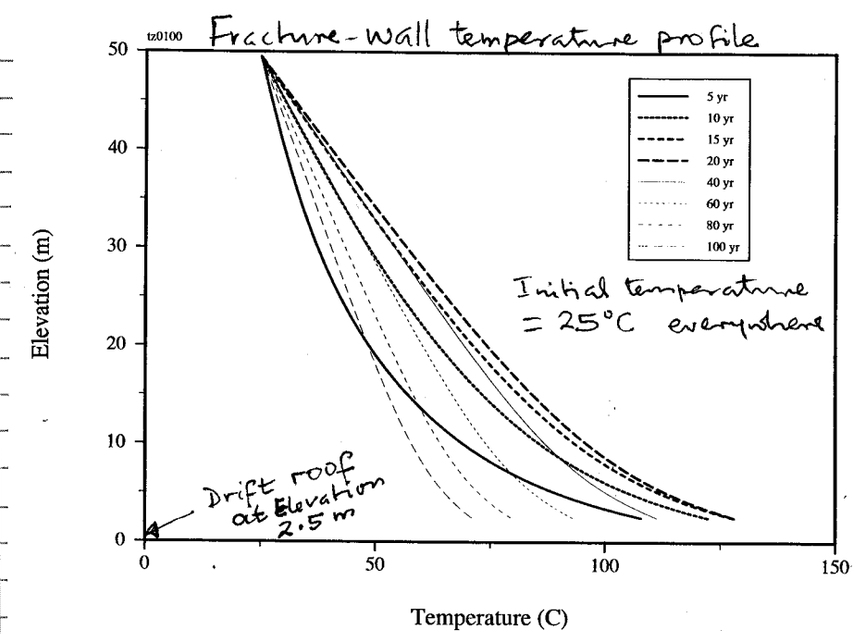
Boundary conditions:

No flow normal to fracture
No flow normal to base
No flow normal to left vertical boundary.
Temperature fixed at top of model (50 m above drift axis)
Heat flux applied to drift wall.

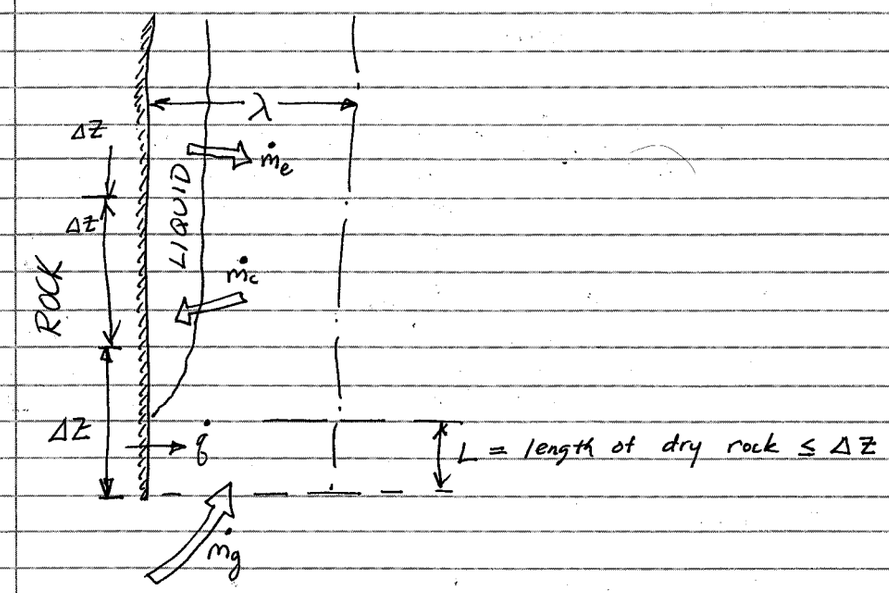
Drift Flux for 83 MTU/acre Case



Heat Conduction: No Water Film



Equation modifications to get started at time $t=0$



<u>USE THESE EQUATIONS UNTIL</u> ~~$L = \Delta Z$~~ F.D $\forall = \lambda \Delta Z$

- Instead of Eq. (6)

$$\frac{d}{dt}(\rho_{mix}\forall) = \dot{m}_g + \sum_{\substack{\text{evap.}\\ \text{elements}}} \dot{m}_e - \sum_{\substack{\text{condensing}\\ \text{elements}}} \dot{m}_c + \sum_{\substack{\text{boil-off between}\\ t \text{ and } t+\Delta t}} \rho_L \delta \, \Delta Z \quad \text{F.D}$$

~~Note that there are no boiled off elements to consider until $L = \Delta Z$~~ F.D

(NEXT PAGE)

F.D 11/17/97

So

$$\forall_{t+\Delta t} = \forall_t \left\{ 1 - \frac{\Delta t}{(\rho_{mix})_t} \left(\frac{d\rho_{mix}}{dt} \right)_t \right\}$$

$$+ \frac{\Delta t}{(\rho_{mix})_t} \left\{ (\dot{m}_g)_t + \sum_{\substack{\text{evap.}\\ \text{elements}}} (\dot{m}_e \Delta z)_t - \sum_{\substack{\text{cond.}\\ \text{elements}}} (m_c \Delta z)_t \right\}$$

$$+ \sum_{\substack{\text{elements that}\\ \text{boiled off from}\\ t=t \text{ to } t=t+\Delta t}} (\rho_L \delta \Delta z)_t \Big/ (\rho_{mix})_t \quad \text{FD} \qquad (6')$$

• Instead of Eq. (11)

$$\frac{d}{dt}\left(\rho_{mix} C_{mix} \forall T_{mix}\right) = \dot{m}_g C_v T_g + \sum \dot{m}_e \overset{C_L}{\rho_L} T_L \Delta z$$

$$- \sum \dot{m}_c C_v T_{mix} \Delta z + \dot{q} + \sum_{\text{new boil off}} \rho_L \delta C_v T_v \Delta z$$

or

$$(T_{mix})_{t+\Delta t} = (T_{mix})_t \left\{ 1 - 2\overset{\text{F.D}}{\cancel{\Delta t}} \left[\frac{\forall_{t+\Delta t} - \forall_t}{\forall_{t+\Delta t} + \forall_t} \right] \right\} \qquad (11')$$

$$+ \frac{\overset{\text{F.D}}{\cancel{\Delta t}}\ 2\Delta t}{(\rho_{mix} C_{mix})_t (\forall_{t+\Delta t} + \forall_t)} \left\{ (\dot{m}_g C_v T_g)_t \right.$$

$$\left. + \sum_{\substack{\text{evap.}\\ \text{elements}}} (\overset{\text{F.D}}{\dot{m}_e} \overset{C_L}{\cancel{\rho_L}} T_L \Delta z)_t - \sum (\overset{\text{F.D}}{\dot{m}_c} \overset{\text{F.D}}{\rho_v} T_{mix} \Delta z)_t \right\} + \overset{\text{F.D}}{\lambda^*} \left\{ \left[K_w \left(\frac{\partial T}{\partial y} \right) \overset{\text{F.D}}{\Delta z'} \right]_t \right.$$

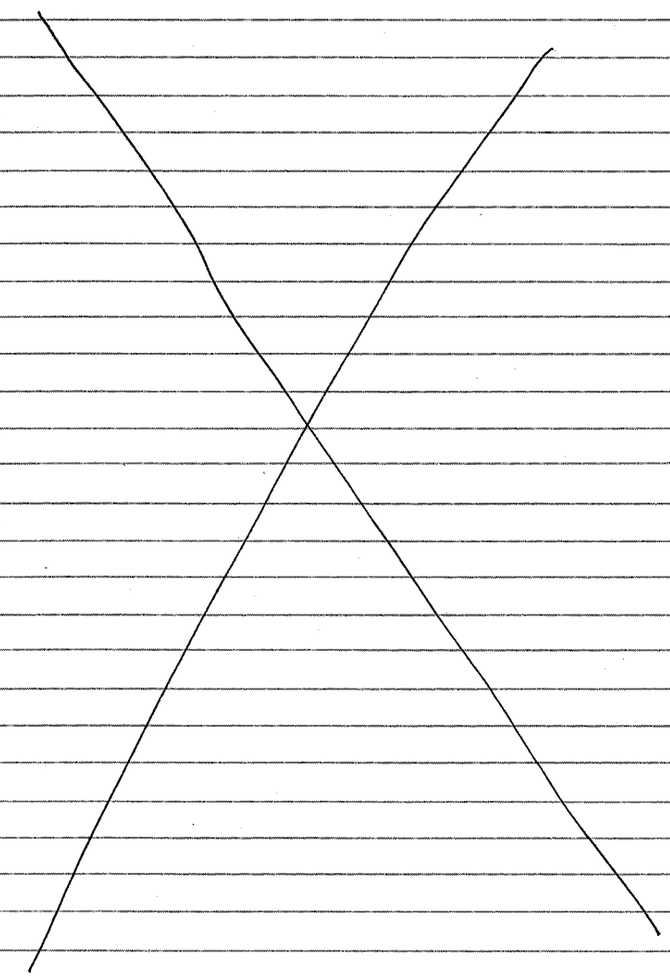
dry wall elements

$$\left. + \sum_{\text{elements that boil off "t" to "t+Δt"}} \rho_L \delta C_v T_v \Delta z \right\} \Big/ \left[\rho_{mix} C_{mix} (\forall_{t+\Delta t} + \forall_t) \right]$$

Steps need to be rearranged until $\forall \geq \lambda \Delta z$

(1) Compute $\forall_{t+\Delta t}$ from Eq. (6')

(2) Then compute $T_{t+\Delta t}$ from Eq. (11')



Revised set of equations for the gas — to improve convergence.

MASS BALANCE EQUATION

$$\underbrace{(\rho_{mix}\forall)_{t+\Delta t}}_{\text{new mass}} = \underbrace{(\rho_{mix}\forall)_t}_{\text{old mass}} \qquad \boxed{1}$$

These are only the liquid elements that lie below the top of $\forall_{mix}$.

$$+ \sum \rho_L \delta \Delta Z \qquad \text{boiloff from } t \text{ to } \Delta t$$

$$+ \Delta t (\dot{m}_g)_t \qquad \text{vapor generation at } t$$

$$+ \left(\sum \dot{m}_e \Delta Z\right)_t \Delta t \qquad \text{liquid evaporation at } t$$

$$- \left(\sum \dot{m}_c \Delta Z\right)_t \Delta t \qquad \text{liquid condensation at } t$$

ENERGY BALANCE EQUATION

$$\underbrace{(\rho_{mix} C_{mix} T_{mix} \forall)_{t+\Delta t}}_{\text{new energy}} = \underbrace{(\rho_{mix} C_{mix} T_{mix} \forall)_t}_{\text{old energy}} \qquad \boxed{2}$$

These are only the liquid elements that lie below the top of $\forall_{mix}$.

$$+ \Delta t (\dot{q})_t \qquad \text{heat transfer}$$

$$+ \sum \rho_L C_v T_v \delta \Delta Z \qquad \text{boiloff from } t \text{ to } t+\Delta t$$

$$+ \Delta t (\dot{m}_g C_v T_g)_t \qquad \text{vapor generation at } t$$

$$+ \Delta t \left(\sum \dot{m}_e C_L T_L \Delta Z\right)_t \qquad \text{liquid evaporation at } t$$

$$- \Delta t \left(\sum \dot{m}_c C_v T_{mix} \Delta Z\right)_t \qquad \text{liquid condensation at } t$$

PROPERTIES

$$(P_{mix})_t = \text{constant} = 1 \text{ atmosphere} \qquad \boxed{3}$$

$$(\rho_{mix})_t = \left(\frac{P_{mix}}{R_{mix} T_{mix}}\right)_t \qquad \boxed{4}$$

$$(R_{mix})_t = \left[\frac{R_V}{\left(\frac{P_V}{P_{mix}}\right)\left(1-\frac{R_V}{R_{air}}\right)+\frac{R_V}{R_{air}}}\right]_t \qquad \boxed{5}$$

$$(C_{mix})_t = \left\{C_{air}\left(\frac{R_{mix}}{R_{air}}\right)\left(1-\frac{P_V}{P_{mix}}\right)+C_V\left(\frac{R_{mix}}{R_V}\right)\left(\frac{P_V}{P_{mix}}\right)\right\}_t \qquad \boxed{6}$$

EQS. (3) → (6) ALSO ARE VALID FOR $t+\Delta t$
(when $t+\Delta t$ variables are used)

SOLUTION METHOD FOR $t>0$ AFTER
$\forall > 0$ (see the following for a starting method)

STEP

$\boxed{1}$ Use Eq. (1) to solve for the combined term

$$(\rho_{mix}\forall)_{t+\Delta t} = RHS$$

All terms on RHS are known from calculations at time t

STEP

$\boxed{2}$ From Step 1, the magnitude of $(\rho_{mix}\forall)_{t+\Delta t}$ is known. This is used to compute $(T_{mix})_{t+\Delta t}$ from Eq. (2). To do so, it must be assumed that $(C_{mix})_{t+\Delta t} \approx (C_{mix})_t$. This is a pretty accurate assumption. Thus, if $(C_{mix})_{t+\Delta t} = (C_{mix})_t$ is assumed <u>only for the purpose of determining $(T_{mix})_{t+\Delta t}$</u>, then Eq. (2) gives

FO

so

$$(T_{mix})_{t+\Delta t} = (T_{mix})_t\left[\frac{(\rho_{mix}\forall)_t}{(\rho_{mix}\forall)_{t+\Delta t}}\right]$$

$$+\frac{1}{(\rho_{mix}\forall)_{t+\Delta t}(C_{mix})_t}\left\{\sum \rho_L C_V T_V \delta\,\Delta z\right. \quad \text{boiloff } t \text{ to } t+\Delta t$$

$$+\Delta t\left[(\dot{q})_t\right. \quad \text{heat transfer}$$

$$+(\dot{m}_g C_V T_g)_t \quad \text{gas generation}$$

$$+\sum(\dot{m}_e C_L T_L \Delta z)_t \quad \text{evaporation}$$

$$\left.\left.-\sum(\dot{m}_c C_V T_{mix}\Delta z)_t\right]\right\} \quad \text{condensation}$$

STEP $\boxed{3}$

Use Eqs. (3)–(6) to compute

P_{mix}
ρ_{mix}
R_{mix}
C_{mix}

at time $t+\Delta t$

STEP $\boxed{4}$ Calculate $V_{mix} = \frac{(\rho V)_{mix}}{\rho_{mix}}$ for $t+\Delta t$

HOW TO GET STARTED AT TIME $t=0$

The same procedure as for $t>0$ will work because none of the denominators in Eqs (1) → (6) will be zero at time $t=0$

January 13 1998

G I Ofoegbu

Computational Algorithm

Step 1 Previous time t
New time $t+\Delta t$

Update air-vapor-mix properties for time $t+\Delta t$ using information from time t.

1.1 Update mix mass M_{mix} ($= \rho_{mix} \forall$ as shown on p. 87). $M_{mix} = 0$ at $t=0$.

1.2 If $M_{mix} = 0$ at $t+\Delta t$ (current M_{mix}), no further update of mix properties is necessary, i.e., mix retains its state at time t (previous time). On the other hand if $(M_{mix})_{t+\Delta t} > 0$, continue update of mix properties as follows.

1.2.1 Update T_{mix} using Eq. on p. 89.

1.2.2 Update P_v following Proc. 6.8 on p. 70.

1.2.3 Update density ρ_{mix} using $(T_{mix})_{t+\Delta t}$ and $(P_v)_{t+\Delta t}$ in Eq. (4), p. 88.

1.2.4 Update specific heat C_{mix} using

$(T_{mix})_{t+\Delta t}$ and $(P_v)_{t+\Delta t}$ in Eq. (6), p. 88.

1.2.5 Update control volume $\forall$ using

$$\forall_{t+\Delta t} = (M_{mix}/\rho_{mix})_{t+\Delta t}, \text{ p. 90.}$$

1.2.6 Update Z_4 using

$$(Z_4)_{t+\Delta t} = (Z_4)_t - (\forall_{t+\Delta t} - \forall_t)/\lambda$$

1.3 Reset mix accumulator variables to zero (whether or not change in mixture properties occurred) following Proc. 6.1, p. 69.

Step 2

Solve for rock temperature at $t+\Delta t$ using dT/dy values (from liquid elements) at time t and current $(t+\Delta t)$ input-flux value from waste package. Each $dT/dy = 0$ at $t=0$. Result gives wall temperature $(T_w)_{t+\Delta t}$ for each liquid element.

Step 3

Update infiltration rate to obtain current value of $\dot{m}_o$ [This feature is not being used now, as constant value is still specified for $\dot{m}_o$].

Step 4

March through liquid elements in top-down order following "March-Down Procedure for Liquid Elements" (p. 64). Mix accumulator variables and liquid-element properties are updated to time $t+\Delta t$ during this procedure.

End → Increment time and start Step 1 again if necessary.

The liquid-elements march-down procedure was modified to account for gas (vapor) generation from waste-package vaporization of water exiting from the bottom of the base liquid element. As a result, the following

procedure is executed at the end of the march-down process:

Set liquid-element pointer pL to the base liquid element.

T_w = pL → GetWallTemp()

If ((pL → HasBoiledOff()) == FALSE)

if (T_w >= T_v)

theMix.AddToRokVapor(~~pL~~ G10 ($m.C_v \cdot T_v$, m)

having set [

m = pL → GetCurFluxDown()

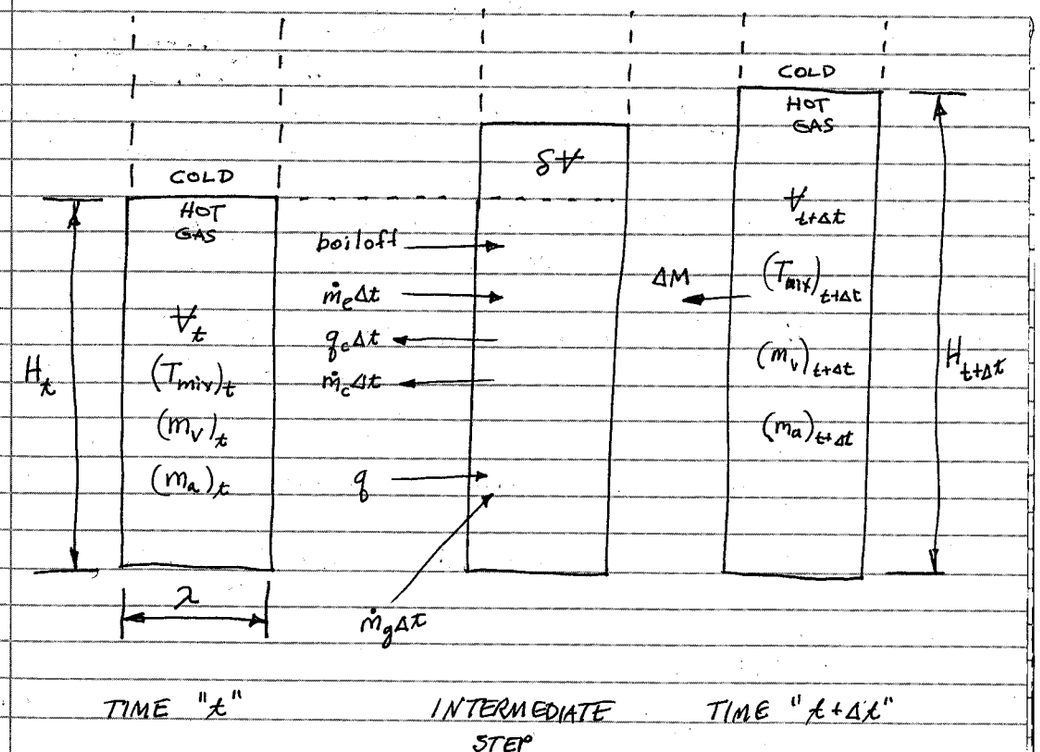
C_v = vapor specific heat at temperature T_v

T_v = boiling temperature.

NEW PROCEDURE TO COMPUTE HOT GAS MIXTURE PROPERTIES

- The previous procedures failed to take into account that in general some vapor must condense (or evaporate) to maintain the mixture at 100% relative humidity. This condensation or evaporation is in addition to the condensation or evaporation caused by heat transfer at the wall. This condensation or evaporation changes the temperature of the mixture to generate the appropriate amount of energy.

- The computation from time "t" to time "t+Δt" now involves an intermediate fictitious mixing step. The final properties are correct.



- TOTAL PRESSURE IS ALWAYS P_0 = atmospheric pressure

F.D. 2/3/98

Definitions

ΔM = additional condensation or evaporation (kg)

F.D $\cancel{M_g = \sum \dot{m}_e}$

F.D. $(M_g) = (\dot{m}_g \Delta t)_t$ = vapor added by evaporation from rock matrix (kg)

$M_e = \left(\sum_i \dot{m}_e \Delta t \, \Delta z\right)_t$ = vapor added by evaporation from all wet wall elements (kg)

$M_c = \left(\sum_i \dot{m}_c \Delta t \, \Delta z\right)_t$ = vapor removed by condensation to all wall elements (kg)

$q = (q \Delta t)_t$ = heat transfer from dry part of the wall, equal to $K \frac{dT_{wall}}{dy}$ (joules)

$q = K\left(\frac{dT_{wall}}{dy}\right)(\Delta Z)$ GW

$M_e C_v T_L = \left(\sum_i \dot{m}_e \Delta t \, \Delta z \, C_v T_L\right)_t$ = enthalpy of vapor added by ~~condensa~~ evaporation (joules)

$Q_c = (q_c \Delta t)_t$ = heat transfer to cooler parts of wall to generate condensation M_c

$M_b = \left(\sum_i \rho_L \delta \, \Delta z\right)_t$ = mass of all wet wall elements that boil away from t to $t+\Delta t$ (kg)

$M_b C_v T_o = \left(\sum \rho_L \delta \Delta z C_v T_o\right)_t$ = enthalpy of boil off (joules)

T_o = boiling temperature at p_o (100°C) (373°K)

- NOTE THAT

$$Q_c = M_c h_{fg} \qquad (A1)$$

where h_{fg} = latent heat, (joules/kg)

F.D 2/3/98

Also, to save writing, define

$T_i = (T_{mix})_t$ °K

$T_f = (T_{mix})_{t+\Delta t}$ °K

$P_i = (P_v)_t$ N/m²

$P_f = (P_v)_{t+\Delta t}$ N/m²

$\Delta T = T_f - T_i$ °K

$\Delta P = P_f - P_i$ N/m²

$P_o = 101.3 \times 10^3$ N/m²

Other definitions that apply at time "t"

T_g = temperature of rock matrix vapor (°K)

T_L = temperature of liquid of each wall element (°K)

C_a = specific heat of air at constant pressure (joule/kg-°K)

C_v = specific heat of vapor at constant pressure (joule/kg-°K)

$\Delta T_o = T_g - T_i$ °K

EQUATIONS

- At time, "t", the gas mixture properties are known.

$(m_v)_t = m_{vi}$ = KNOWN (A2)

$(m_a)_t = m_{ai}$ = KNOWN (A3)

Also $\frac{m_{vi}}{m_{ai}} = 0.622 \frac{P_i}{P_o - P_i}$ (A4)

F.D. 2/3/98

Eq. (A4) ensures 100% relative humidity. Rearranging gives

$$m_{ai} = \frac{P_o - \cancel{P_i}}{\cancel{P}} \left(\frac{P_o - P_i}{0.622 P_i}\right) m_{vi} \qquad (A5)$$

FD

or

$$m_{ai} = A_i \, m_{vi} \qquad A_i = \frac{P_o - P_i}{0.622 P_i} \qquad (A6)$$

- Vapor pressure relation: previously we used

$$\Delta P = \frac{h_{fg} P_i}{R_v T_i^2} \Delta T$$

This is very inaccurate unless $\frac{\Delta P}{P_i} \ll 1$, which is not the case.

Instead use the correlation:

$$P_v = -3604 + 502.06(T_v - 273) - 13.16(T_v - 273)^2 + 0.1846(T_v - 273)^3 \qquad (A7)$$

which is pretty accurate for $T_v - 273 \gtrsim 10°K$

- Intermediate volume change δV

$$\delta V = \frac{R_v T_g M_g}{P_o} + \frac{R_v T_L M_e}{P_o} + \frac{R_v T_o M_b}{P_o} - \frac{R_v T_i M_c}{P_o} + \frac{R_i}{P_o C_i}(\varphi - \varphi_o) \qquad (A8)$$

May want to discard this term.

R_v = vapor gas constant $\left(\frac{\text{Joules}}{\text{kg-°K}}\right)$

FD

R_a = ~~vapor~~ air gas constant $\left(\frac{\text{Joules}}{\text{kg-°K}}\right)$

$$R_i = \frac{m_{ai} R_a + m_{vi} R_v}{m_{ai} + m_{vi}} = \frac{A_i R_a + R_v}{A_i + 1}$$

$$C_i = \frac{m_{ai} C_a + m_{vi} C_v}{m_{ai} + m_{vi}} = \frac{A_i C_a + C_v}{A_i + 1}$$

$A_i = \frac{m_{ai}}{m_{vi}}$ from eq. (A6)

- Additional air and vapor brought into control volume by δV:

$$\delta m_v = \left(\frac{P_i}{R_v T_i}\right) \delta V \qquad (A9)$$

$$\delta m_a = A_i \, \delta m_v \qquad (A10)$$

- Total air and vapor (that ~~can be~~ was at 100% RH) initially in the control volume when temperature $= T_i$

F.D.

$$M_v = m_{vi} + \delta m_v \qquad (A11)$$

$$M_a = A_i M_v \qquad (A12)$$

Note that $m_{vi} = m_{ai} = 0$ @ time $t = 0$

- When the temperature changes to T_f ($> T_i$), some vapor will have to condense or evaporate in addition to the fluxes driven by heat transfer

ΔM = additional vapor condensed (kg)

For 100% RH at the final conditions:

$$\frac{M_v + M_g + M_e - M_c + M_b - \Delta M}{M_a} = 0.622 \frac{P_f}{P_o - P_f} \qquad (A13)$$

After algebraic rearrangement:

$$\frac{\Delta M}{M_g + M_e + M_b - M_c} = 1 - \left[\frac{M_v}{M_g + M_e + M_b - M_c}\right]\left[\frac{\frac{P_o}{P_i} \cdot \frac{\Delta P}{P_i}}{\frac{P_o}{P_i} - 1 - \frac{\Delta P}{P_i}}\right] \qquad (A14)$$

Eq. (A14) is the first equation needed to solve for the unknown ΔM, ΔP, and ΔT.

- Vapor pressure correlation Eq. (A7) is rearranged to give

$$\frac{\Delta P}{P_i} = \left\{ \frac{F_1 + F_2(\Delta T/\Delta T_o) + F_3(\Delta T/\Delta T_o)^2}{P_i} \right\} \frac{\Delta T}{\Delta T_o} \qquad (A15)$$

where

$$F_1 = \left[502.06 - 26.32\,\Delta T_e + 0.5538\,\Delta T_e^2\right]\Delta T_o \qquad (A16a)$$

$$\Delta T_e = T_i - 273 \qquad \Delta T_o = T_g - T_i$$

F.D. 2/3/98

$$F_2 = (0.5538\,\Delta T_e - 13.16)\,\Delta T_o^2 \qquad (A16b)$$

$$F_3 = 0.1846\,\Delta T_o^3 \qquad (A17b)$$

Eq. (A15) is the second needed equation.

- Energy relation (everything occurs @ constant pressure)

$$\left[M_a C_a + M_v C_v + M_g C_v + M_e C_v + M_b C_v - M_c C_v - \Delta M\, C_v\right] T_f$$

$$= \left[M_a C_a + M_v C_v\right] T_i + M_g C_v T_g + M_e C_v T_L + M_b C_v T_o$$
$$- M_c\left(C_v T_i - h_{fg}\right) - \Delta M\left(C_v T_i - h_{fg}\right) \qquad (A18)$$
$$+ Q - Q_c$$

Note: $C_v T_i - h_{fg} = h_e = h_v - h_{fg}$ = liquid enthalpy

$$Q_c = M_c h_{fg}$$

Rearranging gives

$$\left[M_a C_a + (M_v + M_g + M_e + M_b - M_c) C_v - \Delta M\, C_v\right](T_f - T_i) \qquad (A19)$$
$$= M_g C_v (T_g - T_i) + M_e C_v (T_L - T_i) + M_b C_v (T_o - T_i)$$
$$+ \Delta M\, h_{fg} + Q$$

F.D. 2/3/98

or

$$\frac{\Delta T}{\Delta T_o} = \frac{M_g C_v + M_e C_v \left(\frac{T_L - T_i}{\Delta T_o}\right) + M_b C_v \left(\frac{T_o - T_i}{\Delta T_o}\right) + \frac{\Delta M h_{fg} + Q}{\Delta T_o}}{A_i M_v C_a + (M_v + M_g + M_e + M_b - M_c) C_v - \Delta M C_v} \qquad (A20)$$

Define

$$m_1 = M_g + M_e \left(\frac{T_L - T_i}{\Delta T_o}\right) + M_b \left(\frac{T_o - T_i}{\Delta T_o}\right)$$

$$m_2 C_2 = A_i M_v C_A + (M_v + M_g + M_e + M_b - M_c) C_v$$

Thus

$$\frac{\Delta T}{\Delta T_o} = \frac{1 + \frac{\Delta M}{m_1}\left(\frac{h_{fg}}{C_v \Delta T_o}\right) + \frac{Q}{m_1 C_v \Delta T_o}}{\frac{m_2 C_2}{m_1 C_v} - \frac{\Delta M}{m_1}} \qquad (A21)$$

Eq. (A21) is the third equation needed

SUMMARY

$$\frac{\Delta M}{M_g + M_e + M_b - M_c} = 1 - \left[\frac{M_v}{M_g + M_e + M_b - M_c}\right]\left[\frac{\left(\frac{P_o}{P_i}\right)}{\frac{P_o}{P_i} - 1 - \frac{\Delta P}{P_i}}\right]\frac{\Delta P}{P_i} \qquad (A14)$$

$$\frac{\Delta P}{P_i} = \left[\frac{F_1 + F_2 (\Delta T/\Delta T_o) + F_3 (\Delta T/\Delta T_o)^2}{P_i}\right]\frac{\Delta T}{\Delta T_o} \qquad (A15)$$

$$\frac{\Delta T}{\Delta T_o} = \frac{1 + \frac{\Delta M}{m_1}\left(\frac{h_{fg}}{C_v \Delta T_o}\right) + \frac{Q}{m_1 C_v \Delta T_o}}{\frac{m_2 C_2}{m_1 C_1} - \frac{\Delta M}{m_1}} \qquad (A21)$$

FD 2/3/98

One solution method

Guess $\frac{\Delta T}{\Delta T_o} = .01$

* Use Eq. (A15) → compute $\frac{\Delta P}{P_i}$
* Use Eq. (A14) → compute $\frac{\Delta M}{M_g + M_e + M_b - M_c}$ (using $\frac{\Delta P}{P_i}$)
* Use $\frac{\Delta P}{P_i}$ and $\frac{\Delta M}{M_g + M_e + M_b - M_c}$ in Eq. (A21) to compute

$$X = \frac{\Delta T}{\Delta T_o} - \frac{1 + \frac{\Delta M}{m_1}\left(\frac{h_{fg}}{C_v \Delta T_o}\right) + \frac{Q}{m_1 C_v \Delta T_o}}{\frac{m_2 C_2}{m_1 C_1} - \frac{\Delta M}{m_1}}$$

* Note sign of X
* Increase $\left(\frac{\Delta T}{\Delta T_o}\right)_{new} = \left(\frac{\Delta T}{\Delta T_o}\right)_{old} + 0.01$
* Repeat & note sign of X
* When X changes sign, the true value of $\frac{\Delta T}{\Delta T_o}$ lies between $\left(\frac{\Delta T}{\Delta T_o}\right)_{new}$ and $\left(\frac{\Delta T}{\Delta T_o}\right)_{old}$
* Start over with

$$\frac{\Delta T}{\Delta T_o} = \left(\frac{\Delta T}{\Delta T_o}\right)_{old} + 0.001$$

* GO TO
* When $X \approx 0$, you have found the root.

FD 2/13/98

- After finding solution @ $t + \Delta t$, then compute

$$(T_{mix})_{t+\Delta t} = (T_{mix})_t + \Delta T$$

$m_{vi} + \delta m_v$ [eq. (A11)] (G10)

$$(m_v)_{t+\Delta t} = (m_v)_t + (\dot{m}_g \Delta t)_t + \left(\sum \dot{m}_e \Delta t \Delta z\right)_t$$

$$- \left(\sum \dot{m}_c \Delta t \Delta z\right)_t + \left(\sum \rho_L \delta \Delta z\right)_t$$

$$- (\Delta M)_t$$

$$(P_v)_{t+\Delta t} = (P_v)_t + \Delta P$$

$$(m_a)_{t+\Delta t} = 0.622 \left[\frac{(P_v)_{t+\Delta t}}{P_o - (P_v)_{t+\Delta t}}\right] (m_v)_{t+\Delta t}$$

$$H_{t+\Delta t} = \frac{\left[(m_v)_{t+\Delta t} R_v + (m_a)_{t+\Delta t} R_a\right] (T_{mix})_{t+\Delta t}}{\lambda P_o}$$

But if $H_{t+\Delta t} > L_f$ = length of fracture, then

$$H_{t+\Delta t} = L_f$$

and re-compute $(m_v)_{t+\Delta t}$ & $(m_a)_{t+\Delta t}$ as

$$(m_v)_{t+\Delta t} = \frac{P_o \lambda L_f}{R_v (T_v)_{t+\Delta t}}$$

$$(m_a)_{t+\Delta t} = 0.622 \left[\frac{(P_v)_{t+\Delta t}}{P_o - (P_v)_{t+\Delta t}}\right] (m_v)_{t+\Delta t}$$

F.D. 2/3/98

- Then proceed on to $t + 2\Delta t$ and so on!

REVISED RELATIONS FOR EVAPORATION, CONDENSATION, AND BOILOFF

- Previously, condensation to the liquid film was computed by

$$\dot{m}_c = \frac{k_L}{\delta h_{fg}} (T_{mix} - T) \qquad T = \text{wall temperature}$$

This did not allow for any contact resistance. Also it takes a fair amount of temperature difference to unbalance the natural condensation – evaporation exchange.

- The new recommened way is

$$\dot{m}_c = \frac{k_L}{\delta h_{fg}} \left[(T_{mix} - T) - \Delta T_R\right] \quad \text{if } (T_{mix} - T) > \Delta T_R$$

$$= 0 \quad \text{if } (T_{mix} - T) \leq \Delta T_R$$

ΔT_R = temperature difference to compensate for contact resistance, etc

$\Delta T_R = 5°K$ to $10°K$

(need to evaluate the choices, start with 5°K)

F.D. 2/3/98

- Likewise

$$\dot{m}_e = \frac{k_L}{\delta h_{fg}}\left[(T - T_{mix}) - \Delta T_R\right] \quad \text{if } T - T_{mix} > \Delta T_R$$

$$= 0 \quad \text{if } T - T_{mix} \le \Delta T_R$$

- Liquid film will boil away instantaneously between time "t" and time "t+Δt" if the wall temperature exceeds $T_{boilg} + \Delta T_R$

$$T \ge T_{boil} + \Delta T_R \Rightarrow \text{liquid boils off instantaneously}$$

FB 2/4/98

May 14 1998 G I Ofoegbu.

Following are changes to the computational algorithm, necessitated by model changes described on pages 95 – 106.

AirVaporMix Class Data Members (All data members declared private)

```
class AirVaporMix
{
private:
   float curTemp;
   float tempCorrection;
   float pVapor;
   float pVCorrection;
   float mVapor;
   float mVCorrection;
   float delMVapor;
   float hotAirHeight;
   float vapSpecHeat;
   float airSpecHeat;
   float vapAirMassRatio;
   float eBareWall;
   float eCondensate;
   float mEvapTLiqProduct;
   float mRokVapor;
   float mEvaporate;
   float mCondensate;
   float mBoilOff;
```

- curTemp ← $(T_{mix})_t$ and $(T_{mix})_{t+\Delta t}$
- tempCorrection ← ΔT
- pVapor ← $(P_v)_t$ and $(P_v)_{t+\Delta t}$
- pVCorrection ← ΔP
- mVapor ← $(M_v)_t$ and $(M_v)_{t+\Delta t}$
- mVCorrection ← ΔM
- delMVapor ← δM_v
- hotAirHeight ← $(H)_t$ and $(H)_{t+\Delta t}$
- vapSpecHeat ← $(C_v)_t$ and $(C_v)_{t+\Delta t}$
- airSpecHeat ← $(C_a)_t$ and $(C_a)_{t+\Delta t}$
- vapAirMassRatio ← M_a/M_v at t and t+Δt
- eBareWall ← $(\Delta t)(\Sigma q)_t$
- eCondensate ← $(\Delta t)(\Sigma q_c)_t$
- mEvapTLiqProduct ← $(\Delta t)(\sum \dot{m}_e \Delta z T_L)_t$ (GW)
- mRokVapor ← $(\Delta t)(\sum \dot{m}_g)_t$
- mEvaporate ← $(\Delta t)(\sum \dot{m}_e \Delta z)_t$
- mCondensate ← $(\Delta t)(\sum \dot{m}_c \Delta z)_t$
- mBoilOff ← $(\sum \rho_L \delta \Delta z)_t$

The data members store data values at time t on entrance to the AirVaporMix controller (MixIterator) and values at time t+Δt on exit from the controller.

AirVaporMix Member Functions

1.0 AirVaporMix::AirVaporMix(float iniTemp)

Constructor.

1.1 Initialize curTemp to iniTemp.
1.2 Initialize pVapor using curTemp in eq. (A7) - p. 98
1.3 Initialize mVapor (p. 99)
1.4 Initialize hotAirHeight
1.5 Initialize vapAirMassRatio using eq. (A6) - p. 98
1.6 Initialize vapSpecHeat and airSpecHeat by interpolation of curTemp in the appropriate material property tables.
1.7 Initialize all accumulator data members to zero.

2. All the other AirVaporMix member functions are called by the AirVaporMix controller [function MixIterator()] at the beginning of time step $t+\Delta t$. The functions are called in the following order (MixIterator is called by main() to get $t+\Delta t$ step started).

2.0 AirVaporMix::SetDelMVapor()

Compute delMVapor (δM_v) at time $t+\Delta t$ using data values at end of time t in eq. (A8) and (A9) . . . p. 99

3.0 AirVaporMix::SetVaporCorrectors()

Compute ΔT, ΔP and ΔM using eq. (A14), (A15) and (A21) with an iterative algorithm currently being developed

4.0 AirVaporMix::SetTemperature()

curTemp += tempCorrection

5.0 AirVaporMix::SetPVapor()

pVapor += pVCorrection

6.0 AirVaporMix::SetMVapor()

mVapor += (. . . .) equation on p. 104.

7.0 AirVaporMix::SetVaporAirMassRatio()

Compute $A_{t+\Delta t} = (M_a/M_v)_{t+\Delta t}$ using equation on p. 104.

8.0 AirVaporMix::SetHotAirHeight()

Compute $H_{t+\Delta t}$ using eq. on p. 104

Correct M_v if necessary.

9.0 AirVaporMix::SetSpecHeat()

Compute C_v and C_a by interpolation of curTemp in material-property tables.

10.0 AirVaporMix::ResetAccumulators()

eBareWall = eCondensate = 0
mEvapTLiqProduct = 0
mRokVapor = mEvaporate = mCondensate = 0
mBoilOff = 0

Calculation of Vapor Flux Due to Condensation or Evaporation

See pp. 59 [eq. (15)] and 105-106.

Condensation flux: $\cancel{\dot{m}_{in}}$ GW

$$\dot{m}_c = \dot{m}_{in}$$

for $T_{mix} > T_{wall}$

Evaporation flux

$$\dot{m}_e = -\dot{m}_{in}$$

for $T_{mix} < T_{wall}$

Calculation ΔT_R = wall resistance temperature (see p. 105-106)

If $T_{mix} - T_w > \Delta T_R$

$$\dot{m}_{in} = \frac{K_L}{\delta h_{fg}}\left(T_{mix} - T_w - \Delta T_R\right)$$

$$\dot{m}_c = \dot{m}_{in} \quad \text{(condensation flux)}$$

If $T_w - T_{mix} > \Delta T_R$

$\dot{m}_{in}$ given by same equation, but

$$\dot{m}_e = -\dot{m}_{in} \quad \text{(evaporation flux)}$$

$\dot{m}_{in}$ is stored in ~~var~~ fluxFromVapor data member of liquid elements and is assigned to condensation or evaporation

depending on its sign (+ve for condensation and −ve for evaporation)

Calculation (and Accumulation) of AirVaporMix accumulator data members:

The AirVaporMix accumulator data members (

Q = eBareWall
Q_c = eCondensate
M_eT_L = mEvapTLiqProduct
M_g = mRokVapor
M_e = mEvaporate
M_c = mCondensate
M_b = mBoilOff

) are evaluated as follows by LiqElem-iterator functions.

Q = eBareWall

Calculated in function BoilOffFlux(..) if the liquid element passed to the function is found to have boiled off previously. A contribution to Q (denoted Q_i) is calculated given by

$$Q_i = (K_w)\left(\frac{dT}{dy}\right)(\Delta z)(\Delta t)$$ see p. 96

Q_c = eCondensate and M_c = mCondensate

Calculated in function UpdateLiquidElem(..), which calculates contributions Q_{ci} (to Q_c) and M_{ci} (to M_c) as follows, if the liquid element data member fluxFromVapor ($\dot{m}_{in}$) is positive.

$$Q_{ci} = (\dot{m}_{in})(\Delta z)(h_{fg})(\Delta t)$$

$$M_{ci} = (\dot{m}_{in})(\Delta z)(\Delta t)$$

$M_e T_L$ = mEvapTLiqProduct and M_e = mEvaporate

Calculated in function UpdateLiquidElem(...), which calculates contributions $(M_e T_L)_i$ and M_{ei} (to $M_e T_L$ and M_e, respectively) as follows, if the liquid element data member fluxFromVapor ($\dot{m}_{in}$) is negative.

$$(M_e T_L)_i = (-\dot{m}_{in})(\Delta z)(T_L)(\Delta t)$$

$$M_{ei} = (-\dot{m}_{in})(\Delta z)(\Delta t)$$

M_g = mRokVapor

Calculated in function BoilOffFlux()(two forms, one operating on a liquid element and the other on flux) when nonzero flux ~~is~~ is detected at the top of a liquid element that has boiled off or is about to boiloff, or when nonzero exit flux occurs at the base liquid element ~~with heater~~ GW and heater temperature is above boiling. A contribution M_{gi} (to M_g) is calculated as follows

$$M_{gi} = (\dot{m}_L)(\Delta t)$$

where $\dot{m}_L$ is top flux (for the first case) or exit flux (for the second case).

M_b = mBoilOff

Calculated in function BoilOffElem(), which calculates a contribution M_{bi} (to M_b) as follows

$$M_{bi} = (\rho_L)(\delta)(\Delta z)$$

May 29 1998

Evaluation of ΔT, ΔP, and ΔM

Through iterative solution of Eq. (A14), (A15) and (A21) using a binary search algorithm incorporating procedure described on p. 103.

Procedure is accomplished through the following AirVaporMix Member functions:

3.0 AirVaporMix::SetVaporCorrectors()
— Described briefly on p. 108 (item 3).

3.1 AirVaporMix::DoA14Equation()
— Solves eq. (A14) to calculate ΔM using an existing value of ΔP

3.2 AirVaporMix::DoA15Equation(r_{guess})
— Solves eq. (A15) to calculate ΔP using an assumed value of $\Delta T/\Delta T_0$ passed in as argument r_{guess}.

3.3 AirVaporMix::DoA21Equation()
— Solve eq. (A21) to calculate ΔT using an existing value of ΔM.

More Details About Procedure 3.0

AirVaporMix::SetVaporCorrectors()

The following information is known at entry into this procedure:

(1) Values of all AirVaporMix data members (p. 107) at end of the previous time step
(2) Current value of δm_v (p. 99), and
(3) a starting guess for $(\Delta T/\Delta T_0)$.

First, the code checks to see if a solution for equation (A21) is likely to be found within the range $0 \leq \Delta T/\Delta T_0 \leq 1$. If not, an error message is issued and the process terminates.

Next (where a solution in the $0 \leq \Delta T/\Delta T_0 \leq 1$ range is found to be likely), the code invokes procedures

3.2 DoA15Equation(r_{guess})
3.1 DoA14Equation() and
3.3 DoA21Equation()

in that order to compute ΔP, ΔM and ΔT and each time compares the computed value of $\Delta T/\Delta T_0$ (ΔT_0 is a known constant for each time step) against the

estimated value. The process terminates (with success) if the value of $\overline{X}$ (p. 103) is small enough, or continues with a modified estimate for $\Delta T/\Delta T_0$.

June 10 1998

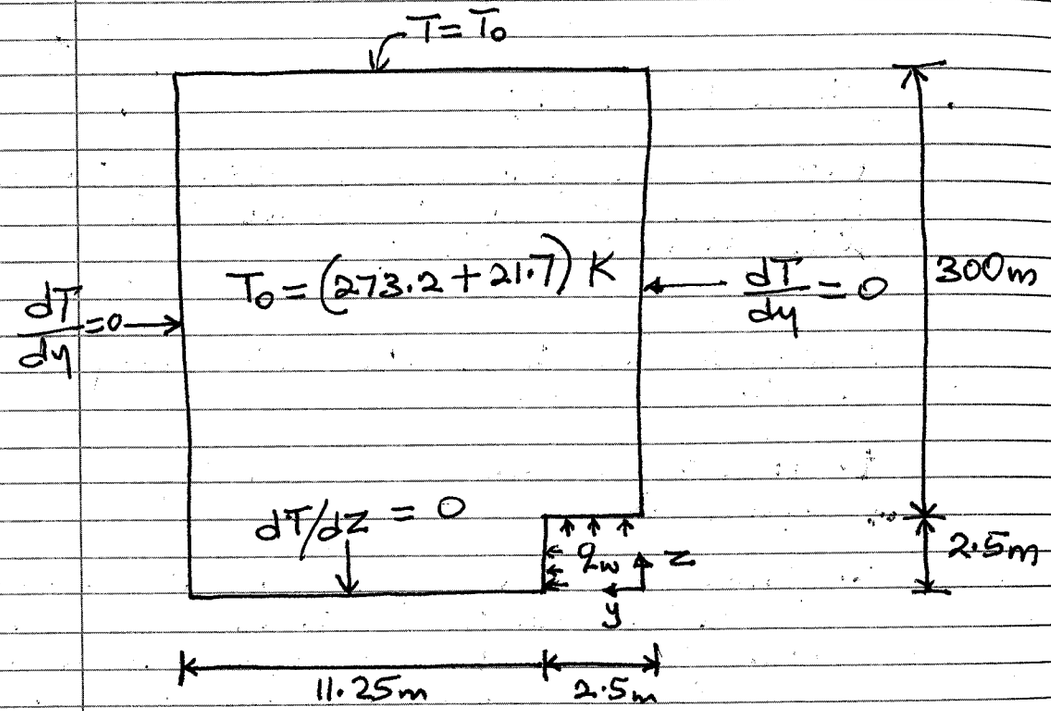
Procedure for Sequentially Coupled Analysis of Heat Conduction and Liquid-Film Flow

Analysis consists of two steps as follows:

1. Heat conduction analysis without film flow.
2. Film flow analysis coupled with heat conduction within a subset of the rock medium. Temperature boundary conditions in the rock-medium subset are taken from results of conduction-only analysis.

— — — Next page.

Heat Conduction Without Film Flow

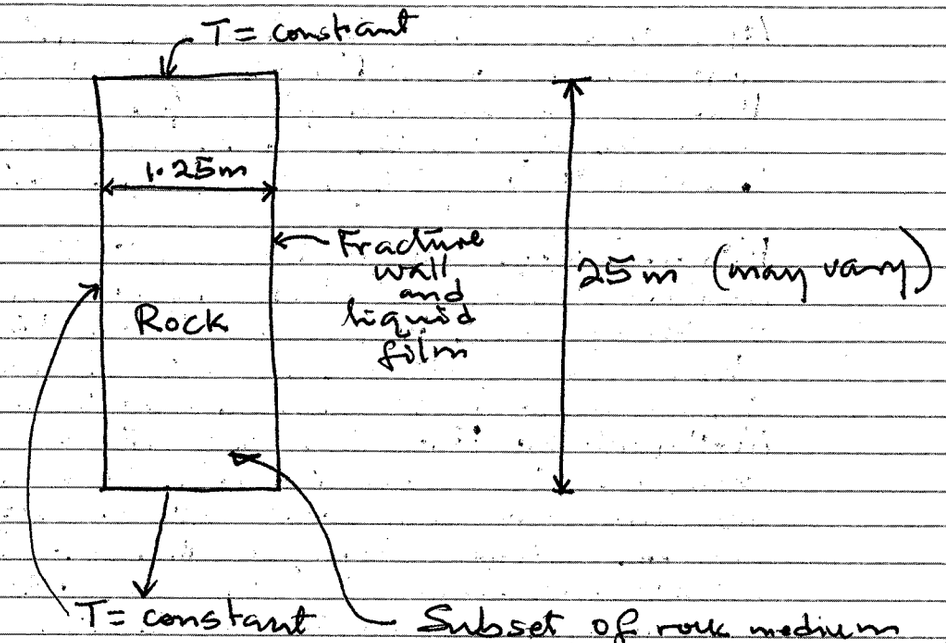


T = temperature (K)
T_0 = initial temperature
q_w = thermal flux on drift wall

Spatial discretization: $\Delta z = \Delta y = 1.25\,m$
Time stepping: $\Delta t = 0.1$ day $= 8.64 \times 10^3$ s

—— —— Next page.

Coupled Film Flow and Heat Conduction



Subset of rock medium modeled in conduction-only analysis.

Initial and boundary temperatures from conduction-only analysis.

Spatial discretization: $\Delta z = \Delta y = 0.05\,m$
Time stepping: $\Delta t = 1$ s
Solve for a very short time (on the order of s).

June 26 1998

Approach described on p. 115 - 117 was not successful. Values calculated for the condensation flux for liquid elements near the top of the model were excessive, leading to very high values for dT/dy.

Frank Dodge suggested that the sequentially coupled approach may not work because it implies large temperature difference at the start of the problem between the rock and the Air-Vapor Mix. For the algorithm of the model to work the rock body and the gas mix should start at about the same temperature, so the state of the system begins to change with the occurrence of boiling near the heater source.

Initializing the system following Frank Dodge's suggestion requires solving the rock problem and gas-mix problem concurrently, which is difficult (if at all possible) because of the large difference in both spatial and temporal scales between the two systems.

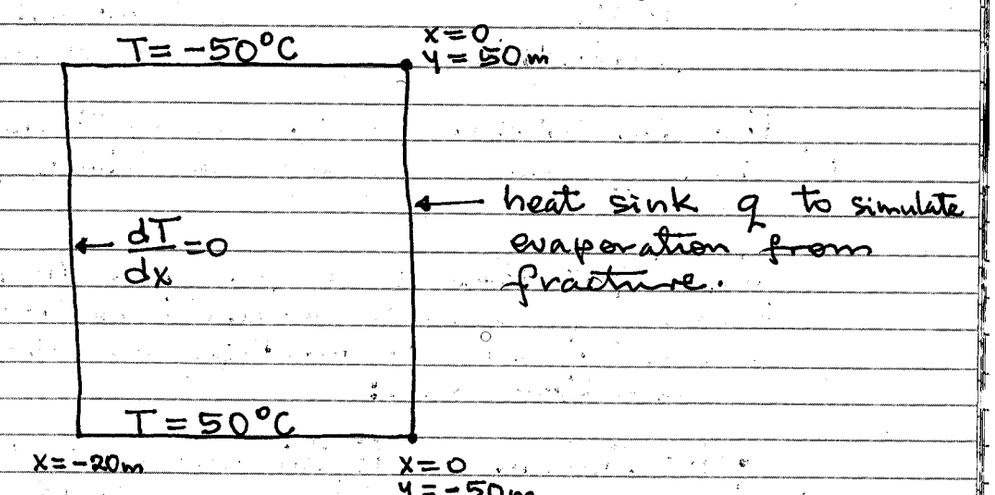
A possible approach to solving the problem may be by using a boundary element formulation, which may eliminate the need to discretize the rock domain.

July 28 1998

Stuart Stothoff suggested a boundary element formulation of the problem. The formulation is documented in Notebook #163. A boundary element code that he developed earlier (as part of his Ph.D. research) will be modified to implement the formulation in Notebook #163.

A test problem was developed to test the basic boundary element code (without the proposed modifications for this project). The test problem will be solved using both the boundary element code and the finite element code ABAQUS.

Problem Geometry (Steady state heat conduction in a rock slab)



Calculation of Sink Magnitude q

$$-q = \rho_w \left(\frac{V_w}{A_f}\right) h_{fg}$$

ρ_w = density of water $\approx 1000\ kg/m^3$

A_f = fracture surface area

$= 100\ m^2/m$

h_{fg} = latent heat of vaporization

$$V_w = \left(10\ \frac{mm}{yr}\right)\left(10^{-3}\ \frac{m}{mm}\right)\left(3.17\times10^{-8}\ \frac{yr}{s}\right)\left(1\ s\right)\left(100\ \frac{m^2}{m}\right)$$

Infiltration rate (can be varied).

Funnel factor or catchment area.

GW ~~For the initial set of calculations, the sink q was calculated as follows:~~

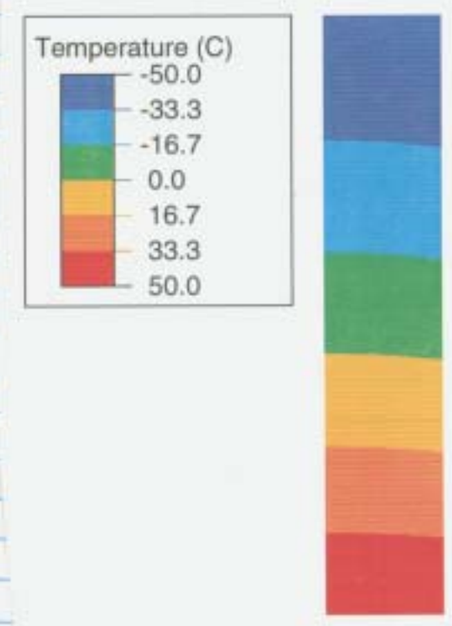
The input files used for the ABAQUS and boundary element calculations have been placed in an 8-mm tape archive. The file directories are:

- ./FracAsSink : Finite element input files for ABAQUS Version 5.6

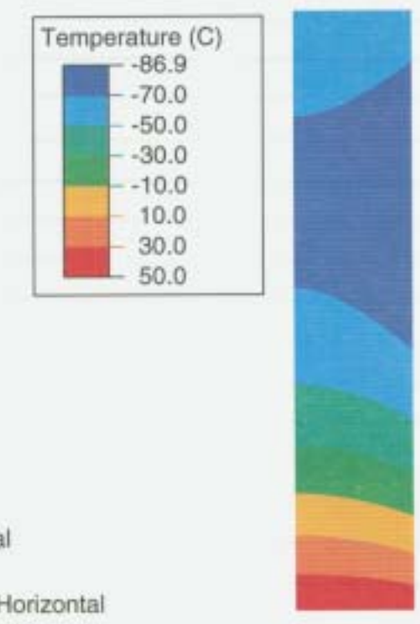
- ./Stu : Boundary element code, input files and output files.

GW

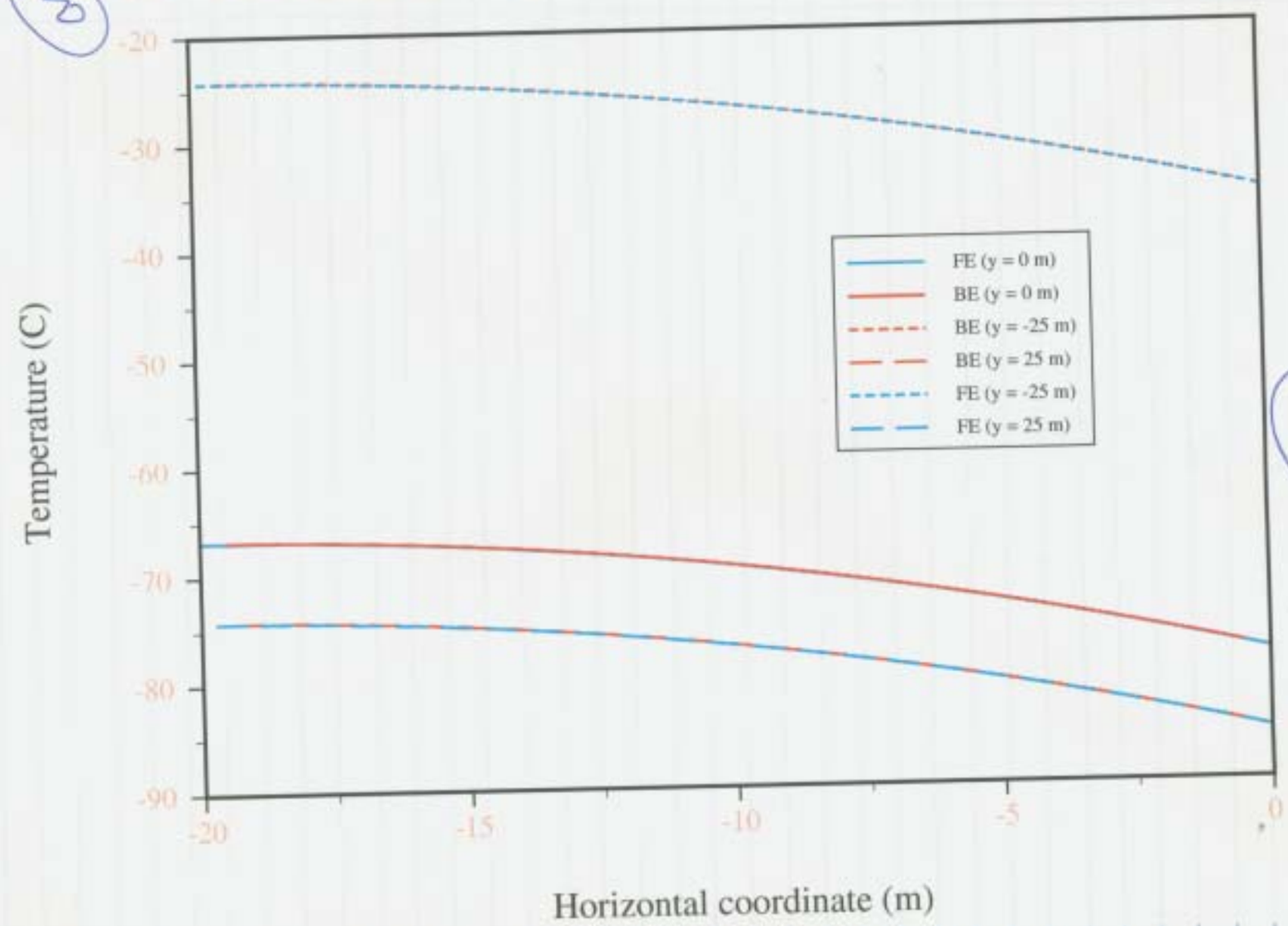
Infiltration = 10 mm/yr
Funnel Factor = 100 m
Evaporation Sink = -0.237 J/m^2

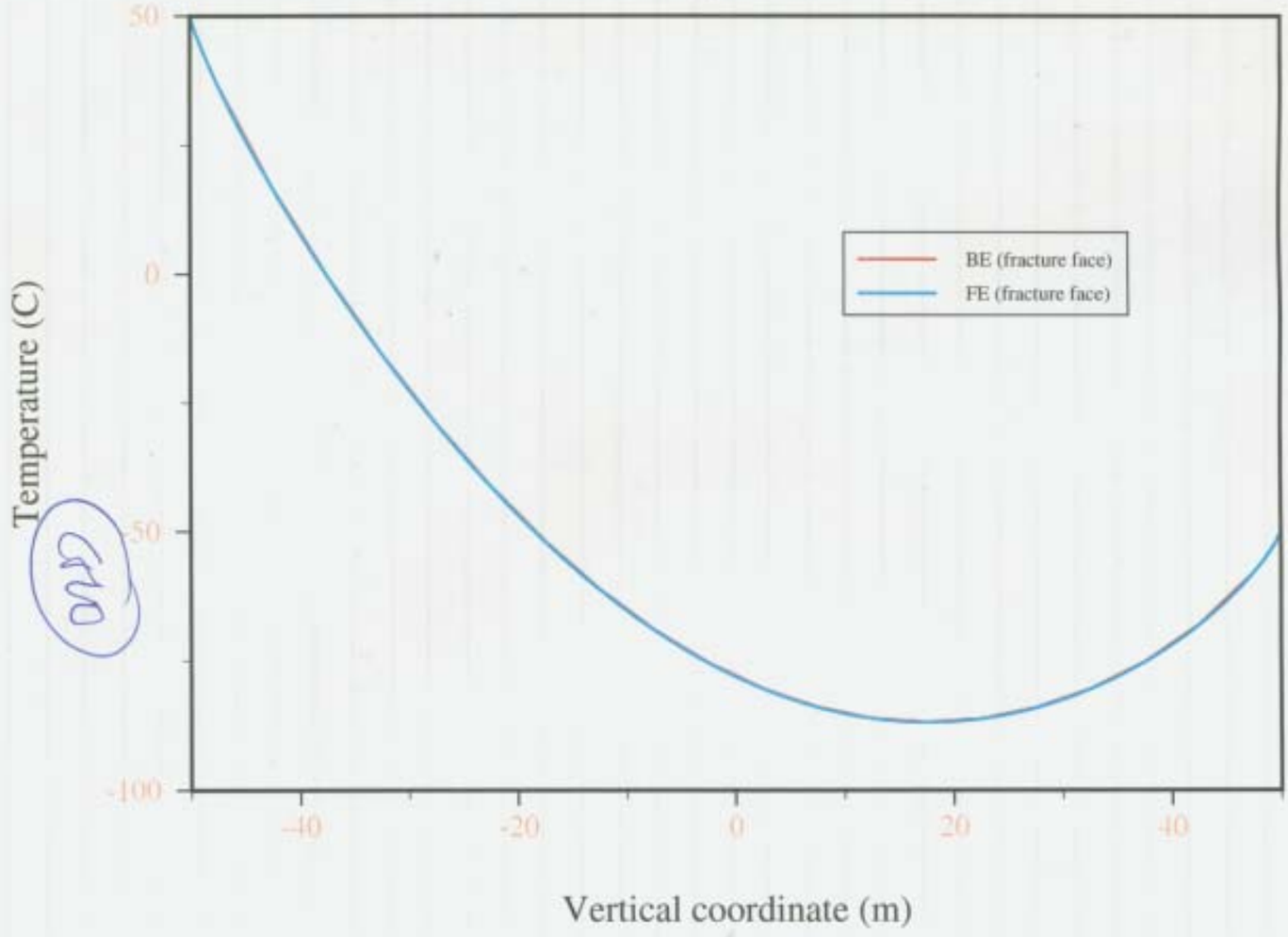


Infiltration = 100 mm/yr
Funnel Factor = 100 m
Evaporation Sink = -2.37 J/m^2

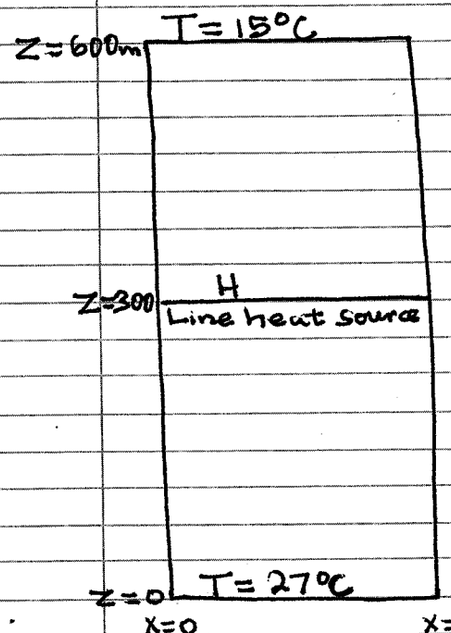


GW


Temperature (C)
Horizontal coordinate (m)
FE (y = 0 m)
BE (y = 0 m)
BE (y = -25 m)
BE (y = 25 m)
FE (y = -25 m)
FE (y = 25 m)
-20
-30
-40
-50
-60
-70
-80
-90
-20
-15
-10
-5
0


Temperature (C)
Vertical coordinate (m)
BE (fracture face)
FE (fracture face)
50
0
-50
-100
-40
-20
0
20
40

A second problem to be ~~was~~ solved that is based ~~similar~~ on analysis of heat conduction with the effects of fracture-wall evaporation and condensation represented as sinks and sources. This problem will be solved using ABAQUS and boundary element code (after the boundary element code is modified to allow for iterative relocation of the sinks and sources).



$\frac{dT}{dx} = 0$ along $x=0$

$x=50\,m$ is fracture wall

At the fracture wall, heat sink or source applied as follows:

Above heat source:

$q = q_{sink}$ if $95 \le T \le 105\,°C$

$q = 0$ otherwise

Below heat source:

$q = 0$ if $T > 95\,°C$

$q = q_{source}$ otherwise

Magnitudes of q_{sink} & q_{source} and their locations are varied iteratively with the temperature solution and height of fracture face of over which temperature satisfies the sink and source conditions.

GVO

$$-2q_{sink} = -(\text{heat sink on both faces of a fracture})$$

$$= \rho_w \left(\frac{V_w}{A_e}\right) h_{fg}$$

ρ_w = density of water $\approx 1000\,kg/m^3$

A_e = fracture-face area over which evaporation or condensation occurs (m^2/m)

h_{fg} = latent heat of evaporation $\approx 2.26 \times 10^6\,J/kg$

V_w = volume of water evaporating or condensing.

$$= \left(w_{in}\ mm/yr\right)\left(10^{-3}\frac{m}{mm}\right)\left(3.17\times10^{-8}\frac{yr}{s}\right)(1\,s)\left(L_{fc}\frac{m^2}{m}\right)$$

$$= (3.17\times10^{-11})(w_{in})L_{fc}\ m^3/m$$

$$-2q_{sink} = \left(10^3\frac{kg}{m^3}\right)\left(3.17\times10^{-11}\,w_{in}L_{fc}\frac{m^3}{m}\right)\left(\frac{1}{A_e}\frac{m}{m^2}\right)\left(2.26\times10^6\frac{J}{kg}\right)$$

$$= (7.16\times10^{-2})\left(\frac{w_{in}L_{fc}}{A_e}\right)\ J/m^2$$

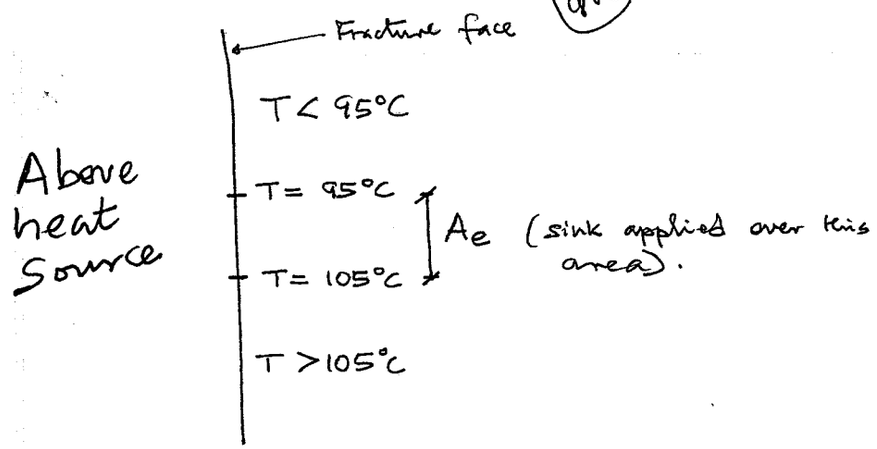
w_{in}	mm/yr
L_{fc}	m
A_e	m

Both w_{in} and L_{fc} will be set for each model. A_e is solution-dependent, and will be evaluated iteratively.

GVO

Evaluation of A_e for the Evaporation Zone

GW



For first set of 3 model cases:

$L_{fc} = 100$ m
$W_{in} = 1, 10,$ or 100 mm/yr.

$L_{fc} = 100$ m.

W_{in} (mm/yr)	$-2(A_e)(q_{sink})$ J	$A_e\, q_{sink}$ J.
1.0	7.16	−3.58
10.0	71.6	−35.8
100.0	716.	−358

GW

For q_{source}, replace q_{sink} with q_{source} in the table and set A_e equal to height of condensation zone.

Magnitude of applied heat source (line source along z = 300 m on p. 124) was varied to give a desired maximum temperature ~~for a case with no evaporation on~~ GW for a zero-infiltration case. A series of results showing the iteration steps are tabulated

GW

Analysis Case C01

Thermal Source Strength (J/m²) 0.725

Base-Case Maximum Temperature (C) 124.6

Infiltration Rate (mm/yr) 1.0

Fracture Catchment Area (m²/m) 100

Attempt	T_{max} (C)	Evaporation Zone (Z>300 m, and 95<T<105 C)			Condensation Zone (Z<300 m, and T<95 C)		
		Element Number Range (increment=100)	Elevation Range (m)	Source strength (J/m²)	Element Number Range (increment=100)	Elevation Range (m)	Source strength (J/m²)
Base	124.6	8625-8925	350.6-381.5	−0.1159	125-1025	0-208.2	0.0172
1	122	8525-8825	344.9-372.4	−0.1302	125-1025	0-208.2	0.0172
2	122	8525-8825	Same	Same	Same	Same	Same.

Final elevation of boiling isotherm ~~(T=195°C~~ (T = 100°C)
357 m (57 m above heat source)
227.6 m (72 m below ")

GW

GTD

Analysis Case C02

Thermal Source Strength (J/m²) 0.725

Base-Case Maximum Temperature (C) 125

Infiltration Rate (mm/yr) 10.0

Fracture Catchment Area (m²/m) 100

Attempt	T_{max} (C)	Evaporation Zone (Z>300 m, and 95<T<105 C)			Condensation Zone (Z<300 m, and T<95 C)		
		Element Number Range (increment=100)	Elevation Range (m)	Source strength (J/m²)	Element Number Range (increment=100)	Elevation Range (m)	Source strength (J/m²)
Base	125	8625-8925	350.6-381.5	−1.159	125-1025	0-208.2	0.172
1	102	5125-7425	300-313.2	−2.712	125-1425	0-243.0	0.147
2	97.9						

Fracture-wall temperature less than 95°C everywhere from Attempt #1. No boiling isotherm.

GTD

Analysis Case C03

Thermal Source Strength (J/m²) 1.08

Base-Case Maximum Temperature (C) 175

Infiltration Rate (mm/yr) 1.0

Fracture Catchment Area (m²/m) 100

Attempt	T_{max} (C)	Evaporation Zone (Z>300 m, and 95<T<105 C)			Condensation Zone (Z<300 m, and T<95 C)		
		Element Number Range (increment=100)	Elevation Range (m)	Source strength (J/m²)	Element Number Range (increment=100)	Elevation Range (m)	Source strength (J/m²)
Base	175	9425	431.1-447.6	−0.217	125-525	0-133.9	0.0267
1	174	Same as above					

Boiling isotherm (T = 100°C):
439.3 m (139.3 m above heat source)
143.2 m (157 m below heat source).

GTD

Analysis Case C04

Thermal Source Strength (J/m²) 1.08

Base-Case Maximum Temperature (C) 175

Infiltration Rate (mm/yr) 10

Fracture Catchment Area (m²/m) 100

Attempt	T_{max} (C)	Evaporation Zone (Z>300 m, and 95<T<105 C)			Condensation Zone (Z<300 m, and T<95 C)		
		Element Number Range (increment=100)	Elevation Range (m)	Source strength (J/m²)	Element Number Range (increment=100)	Elevation Range (m)	Source strength (J/m²)
Base	175	9425	431.1-447.6	−2.17	125-525	0-133.9	0.267
1	159	9025 (1.0)	~~381.5-397.6~~	−2.22	125-425 (1.0)	0-123.5	0.290
		9125 (0.5)		−1.11	525 (0.5)		0.145
2	150	8825 (1.0)		−2.82	125-525	0-133.9	0.267
		8925 (0.5)	364.2-376.9	−1.41			
3	148	8725	357-364.2	−4.97	125-525	0-133.9	0.267
4	146	8625	350.6-357	−5.598	125-525 (1.0)	0-143.2	0.250
					625 (0.5)		0.125

...contd.

GTD

Analysis Case C04 contd.

Thermal Source Strength (J/m²)

Base-Case Maximum Temperature (C)

Infiltration Rate (mm/yr)

Fracture Catchment Area (m²/m)

Attempt	T_{max} (C)	Evaporation Zone (Z>300 m, and 95<T<105 C)			Condensation Zone (Z<300 m, and T<95 C)		
		Element Number Range (increment=100)	Elevation Range (m)	Source strength (J/m²)	Element Number Range (increment=100)	Elevation Range (m)	Source strength (J/m²)
5	146	8525-8925	344.9-381.5	−0.978	125-525	0-133.9	0.267
6	146	8625-8725 (1.0)		−2.023			
		8825 (0.5)	350.6-368.3	−1.011	125-525	0-133.9	0.267
7	146	8625 (1.0)		−3.58			
		8725 (0.5)	350.6-360.6	−1.79	125-525	0-133.9	0.267
~~8~~	~~145~~	Distributions #6 and #7 are close enough. So #6 is accepted as a good estimate for the solution.					

Boiling isotherm (T = 100°C):
353.8 (54 m above heat source)
156.6 (143 m below heat source).

GTD

GW

Analysis Case C201 (Mesh gradation modified for all C2-- cases)

Thermal Source Strength (J/m²) 0.725

Base-Case Maximum Temperature (C) 125

Infiltration Rate (mm/yr) 1.0

Fracture Catchment Area (m²/m) 100

Attempt	T_{max} (C)	Evaporation Zone (Z>300 m, and 95<T<105 C)			Condensation Zone (Z<300 m, and T<95 C)		
		Element Number Range (increment=100)	Elevation Range (m)	Source strength (J/m²)	Element Number Range (increment=100)	Elevation Range (m)	Source strength (J/m²)
Base	125	8525-8825	354.3-383.6	-0.122	125-1225	0-206.2	0.0174
1	122	8325-8625	343.7-367.4	-0.151	125-1325	0-214.9	0.0167
2	122	8325-8725	343.7-375.1	-0.114	125-1325	0-214.9	0.0167
3	122	Same	as	case	2 (Converged).		

Boiling isotherm (T = 100°C)
y = 356 m (56 m above heat source)
y = 226.5 (74 m below heat source).

GW

Analysis Case C202

Thermal Source Strength (J/m²) 0.725

Base-Case Maximum Temperature (C) 125

Infiltration Rate (mm/yr) 10 ~~mm/yr~~

Fracture Catchment Area (m²/m) 100

GW

Attempt	T_{max} (C)	Evaporation Zone (Z>300 m, and 95<T<105 C)			Condensation Zone (Z<300 m, and T<95 C)		
		Element Number Range (increment=100)	Elevation Range (m)	Source strength (J/m²)	Element Number Range (increment=100)	Elevation Range (m)	Source strength (J/m²)
Base	125	8525-8825	354.3-383.6	-1.22	125-1225	0-206.2	0.174
1	103	5725-7225	301.4-314.2	-2.80	125-1725	0-242.7	0.448
2	97.9						

This case cannot converge without accounting for cooling effect of water in nonboiling and noncondensing region above the upper boiling isotherm.

GW

Analysis Case C203

Thermal Source Strength (J/m²) 0.725

Base-Case Maximum Temperature (C) 125

Infiltration Rate (mm/yr) 5.0

Fracture Catchment Area (m²/m) 100

Attempt	T_{max} (C)	Evaporation Zone (Z>300 m, and 95<T<105 C)			Condensation Zone (Z<300 m, and T<95 C)		
		Element Number Range (increment=100)	Elevation Range (m)	Source strength (J/m²)	Element Number Range (increment=100)	Elevation Range (m)	Source strength (J/m²)
Base	125	8525-8825	354.3-383.6	-0.610	125-1225	0-206.2	0.0870
1	114	7525-8125	317.9-339.2	-0.840	125-1425	0-222.9	0.0800
2	111	7025-7725	310-325.2	-1.18	125-1525	0-230.1	0.0778
3	111	6725-8025	306.8-335.1	-0.632	125-1525	0-230.1	0.0778
4	111	6725-7925	Converged.				

Boiling isotherm (based on T=100°C)
y = 316 m (16 m above heat source)
y = 250.6 (49 m below heat source)

GW

Analysis Case C204

Thermal Source Strength (J/m²) 1.08

Base-Case Maximum Temperature (C) 175

Infiltration Rate (mm/yr) 5.0

Fracture Catchment Area (m²/m) 100

GW

Attempt	T_{max} (C)	Evaporation Zone (Z>300 m, and 95<T<105 C)			Condensation Zone (Z<300 m, and T<95 C)		
		Element Number Range (increment=100)	Elevation Range (m)	Source strength (J/m²)	Element Number Range (increment=100)	Elevation Range (m)	Source strength (J/m²)
Base	~~175~~ ✓	~~9325~~ 9325-9425	~~426.3-442.7~~ 428.3-458.7	-0.589	125-625	0-131.8	0.136
1	168	9125-9225	403.6-428.3	-0.725	125-625	0-131.8	0.136
2	165	9125	403.6-415.2	-1.53	125-625	0-131.8	0.136
3	165	9125	Converged.		125-625		

T=100° isotherm: z_{upper} = 401 m $z_{upper} - z_h$ = 101 m
z_{lower} = 147 m $z_h - z_{lower}$ = 153 m.

GW

GTO

Analysis Case C205

Thermal Source Strength (J/m²) 1.08

Base-Case Maximum Temperature (C) 175

Infiltration Rate (mm/yr) 10 ~~mm/yr~~

Fracture Catchment Area (m²/m) 100

Attempt	T_{max} (C)	Evaporation Zone (Z>300 m, and 95<T<105 C)			Condensation Zone (Z<300 m, and T<95 C)		
		Element Number Range (increment=100)	Elevation Range (m)	Source strength (J/m²)	Element Number Range (increment=100)	Elevation Range (m)	Source strength (J/m²)
Base	175	9325-9425	428.3-458.7	-1.18	125-625	0-131.8	0.272
1	160	8925-9025	383.6-403.6	-1.79	125-525	0-114.7	0.312
2	150	8725	367.4-375.1	-4.65	125-725	0-147.3	0.243
3	149	8625	360.5-367.4	-5.19	125-625	0-131.8	0.272
4	146	8525	354.3-360.5	-5.77	125-625	0-131.8	0.272
5	145	8425-8625	348.7-367.4	-1.91	125-725	0-147.3	0.243
6	147	8425-8825	348.7-383.6	-1.03	125-625	0-131.8	0.272
7	147	8525-8625	354.3-367.4	-2.73	125-725	0-147.3	0.243
8	147	8525-8625			125-625		

Acceptable Convergence.

108°C isotherm: $Z_{upper} = 354$ $Z_{upper} - Z_h = 54$ m

$Z_{lower} = 147$ $Z_h - Z_{lower} = 153$ m

GTO

Analysis Case C206

Thermal Source Strength (J/m²) 1.08

Base-Case Maximum Temperature (C) 175

Infiltration Rate (mm/yr) ~~100~~ 20

Fracture Catchment Area (m²/m) 100

Attempt	T_{max} (C)	Evaporation Zone (Z>300 m, and 95<T<105 C)			Condensation Zone (Z<300 m, and T<95 C)		
		Element Number Range (increment=100)	Elevation Range (m)	Source strength (J/m²)	Element Number Range (increment=100)	Elevation Range (m)	Source strength (J/m²)
Base	175	9325-9425	428.3-458.7	~~-11.8~~ -2.38	125-625	0-131.8	~~2.7.2~~ 0.544
1	144	8425-8525	348.7-360.5	-6.07	125-425	0-95.91	0.747
2	108	6125-7225	302.9-314.2	-6.34	125-1425	0-222.9	0.321
3	119	4525-6025	298-302.9	-14.6	125-625	0-131.8	0.543
4	103						

Case requires accounting for cooling effect of ~~water~~ liquid water.

GTO

GTO

I have reviewed this scientific notebook and find it in compliance with QAP-501. There is sufficient information regarding procedures used for conducting tests, acquiring and analyzing data so that another qualified individual could repeat the activity.

Manager-MGFE

June 6, 2000