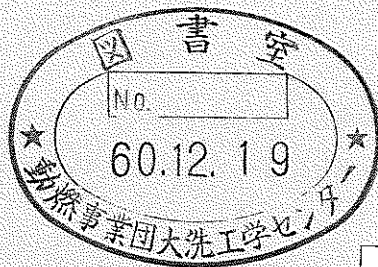
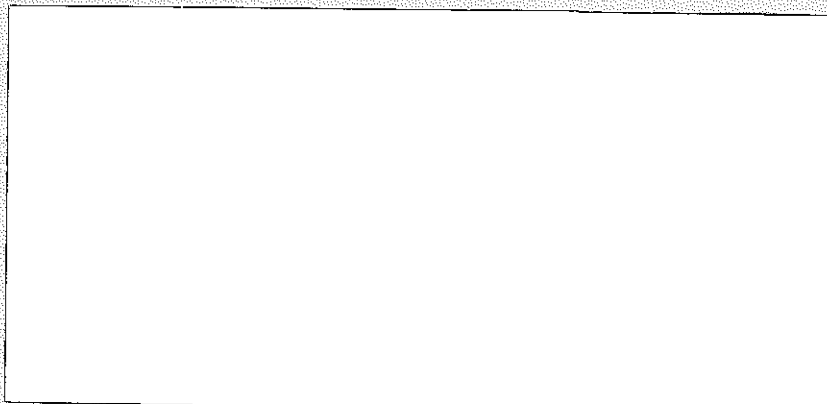


区分変更	
変更後資料番号	二
決定期月日	平成 13 年 7 月 31 日

Incorporation of Sodium-Concrete Reaction Module and Water Release Module into CONTAIN



November, 1985



OARAI ENGINEERING CENTER

POWER REACTOR AND NUCLEAR FUEL DEVELOPMENT CORPORATION

本資料の全部または一部を複写・複製・転載する場合は、下記にお問い合わせください。

〒319-1184 茨城県那珂郡東海村大字村松4番地49
核燃料サイクル開発機構
技術展開部 技術協力課

Inquiries about copyright and reproduction should be addressed to:
Technical Cooperation Section,
Technology Management Division,
Japan Nuclear Cycle Development Institute
4-49 Muramatsu, Tokai-mura, Naka-gun, Ibaraki, 319-1184
Japan

© 核燃料サイクル開発機構 (Japan Nuclear Cycle Development Institute)



Incorporation of Sodium-Concrete Reaction Module
and Water Release Module into CONTAIN

Takao Kudou**, Ikuo Miyaji**, Hiroshi Hiroi*,

Osamu Miyake*, Yoshiaki Himeno*, and Kimihide Miyaguchi*

Abstract

The containment response analysis code, CONTAIN, released from Sandia National Laboratories (SNL) U.S.A. is not yet provided with the function to deal with the sodium-concrete reaction, as well as the water release from a heated concrete, which is thought to be the important reaction occurring in the event of an LMFBR hypothetical accident. Thus, the current CONTAIN code can not consistently evaluate the influence of water vapor, hydrogen gas, and reaction heat during the course of the containment analysis.

This report presents the incorporation of the SLAM code (the sodium-concrete reaction code developed by SNL) into CONTAIN and its modification for the analysis of water migration within a concrete. The transition into the sodium-concrete reaction due to a liner failure is also included. Details of program structure and variable usage are presented.

* Plant Safety Section, Safety Engineering Division,
O-arai Engineering Center, PNC

** Nuclear Data Corporation.

Table of Contents

1. Introduction	1
2. Current Concrete Model in CONTAIN	2
3. Outline of the Original Stand-Alone Code SLAM	5
4. Incorporation of SLAM into CONTAIN	10
4-1. Design Philosophy	10
4-2. Input/Output	11
4-3. Storage Array Variable ICH, CH	11
4-4. Time Step Calculation Routine (STIME)	11
4-5. Concrete Module	12
5. Design of Subroutines	15
5-1. Modification of the Original SLAM Code for CONTAIN Module	15
5-2. Newly Established Subroutines	17
5-3. Description of Subroutines	17
5-3.1 SLINPT and SLCHEM	17
5-3.2 STIME	19
5-3.3 CONCRM	22
5-3.4 SLAM	28
5-3.5 CONEQS	30
5-3.6 NATCON	33
5-3.7 WTRRLS	35
5-3.8 SORGAS	36
5-3.9 CONCPT	36
5-3.10 LAYMOD	37
5-3.11 TRANA	37
5-3.12 TRANB	37
5-3.13 TRANC	38
6. Table of Common Variables	41
7. Program Configuration	50
8. Correction Set	65
9. Test Calculation	109
9-1. Sodium-Concrete Interaction	109
9-1.1 System and Conditions of Test Calculation	109
9-1.2 Results of Calculation and Discussion	116

9-2. Water Release Calculation	133
9-2.1 System and Conditions of Test Calculation	133
9-2.2 Results of Calculation and Discussion	136
10. Future Tasks	150

List of Figures and Tables

Figure 2.1 Calculation Flow Diagram for CONTAIN Concrete
 Reaction Module 3

Figure 3.1 Configuration of the Original SLAM code 5

Figure 3.2 Calculation Flow Diagram of Original SLAM 9

Figure 4.1 Flow Diagram of Lower Cell Control Routine 13

Figure 4.2 Concrete Module Calculation Flow 14

Figure 5.1 Flow Diagram of STIME 21

Figure 5.2 Flow Diagram of CONCRM 27

Figure 5.3 Flow Diagram of SLAM 29

Figure 5.4 Flow Diagram of CONEQS 32

Figure 5.5 Flow Diagram of NATCON 34

Figure 9.1 System for Sodium-Concrete Interaction
 Test Calculation 110

Figure 9.2 Comparison of Original SLAM and CONTAIN
 Calculations of Pool Temperature during
 Sodium-Concrete Reaction117

Figure 9.3 Comparison of Original SLAM and CONTAIN
 Calculations of Concrete DRY Zone Temperature
 during Sodium-Concrete Reaction118

Figure 9.4 Comparison of Original SLAM and CONTAIN
 Calculations of Wet-Dry Interface Temperature
 during Sodium-Concrete Reaction119

Figure 9.5 Comparison of Original SLAM and CONTAIN
 Calculations of Wet-Dry Interface H2O Pressure
 during Sodium-Concrete Reaction120

Figure 9.6 Comparison of Original SLAM and CONTAIN
 Calculations of Ablation Velocity during
 Sodium-Concrete Reaction121

Figure 9.7 Comparison of Original SLAM and CONTAIN
 Calculations of Penetration Depth during
 Sodium-Concrete Reaction122

Figure 9.8 Comparison of Original SLAM and CONTAIN
 Calculations of Dry Zone Growth Rate during
 Sodium-Concrete Reaction123

Figure 9.9 Comparison of Original SLAM and CONTAIN
Calculations of Dry Zone Depth during
Sodium-Concrete Reaction 124

Figure 9.10 Comparison of Original SLAM and CONTAIN
Calculations of Interface H₂O Evaporated Flux
during Sodium-Concrete Reaction 125

Figure 9.11 Comparison of Original SLAM and CONTAIN
Calculations of Dry Zone Bound H₂O Flux during
Sodium-Concrete Reaction 126

Figure 9.12 Comparison of Original SLAM and CONTAIN
Calculation of Dry Zone Bound CO₂ Flux during
Sodium-Concrete Reaction 127

Figure 9.13 Comparison of Original SLAM and CONTAIN
Calculations of Concrete Reaction Heat during
Sodium-Concrete Reaction 128

Figure 9.14 Comparison of Original SLAM and CONTAIN
Calculation of Concrete Surface Heat Flux during
Sodium-Concrete Reaction 129

Figure 9.15 Comparison of Original SLAM and CONTAIN
Calculations of Wet-Dry Interface Heat Flux during
Sodium-Concrete Reaction 130

Figure 9.16 Comparison of Original SLAM and CONTAIN
Calculations of Dry Zone Heat Sources during
Sodium-Concrete Reaction 131

Figure 9.17 Comparison of Original SLAM and CONTAIN
Calculations of Heat Flux into Wet Zone during
Sodium-Concrete Reaction 132

Figure 9.18 System for Water Release
Test Calculation 134

Figure 9.19 Results of Low-Cell Temperature of
Water Release Calculation 137

Figure 9.20 Result of Wet-Dry Interface Temperature of
Water Release Calculation 138

Figure 9.21 Result of Wet-Dry Interface H₂O Pressure of
Water Release Calculation 139

Figure 9.22 Result of Dry Zone Growth Rate of
Water Release Calculation 140

Figure 9.23 Result of Dry Zone Depth of
Water Release Calculation 141

Figure 9.24 Result of Interface H₂O Evaporated Flux of
Water Release Calculation 142.

Figure 9.25 Result of Dry Zone Bound H₂O Flux of
Water Release Calculation 143

Figure 9.26 Result of Integral of Bound + Evaporated H₂O of
Water Release Calculation 144

Figure 9.27 Result of Dry Zone Bound CO₂ Flux of
Water Release Calculation 145

Figure 9.28 Result of Integral of Bound CO₂ of
Water Release Calculation 146

Figure 9.29 Result of Concrete Surface Heat Flux of
Water Release Calculation 147

Figure 9.30 Result of Wet-Dry Interface Heat Flux of
Water Release Calculation 148

Figure 9.31 Result of Heat Flux into Wet Zone of
Water Release Calculation 149

Table 2.1 Comparison of DEBCON and SLAM Modules in
Solution Methods 4

Table 5.1 Modifications of SLAM Subroutines 15

Table 9.1 Limestone Concrete Compositions 109

Table 9.2 Calculation Cases for Original SLAM Code 112

Table 9.3 Calculation Cases for New CONTAIN Code 112

Table 9.4 Comparison of Results of SLAM and CONTAIN..... 116

List 9.1 SLAM (Case SLAM-1) Test Calculation
Input Data (Unit-9) 113

List 9.2 SLAM (Case SLAM-1) Test Calculation
Input Data (Unit-10) 114

List 9.3 CONTAIN (Case CONTAIN-1) Input Data
for Sodium-Concrete Interaction 115

List 9.4 CONTAIN Input Data for Water Release
Calculation 135

1. Introduction

The containment response analysis code, CONTAIN, released from Sandia National Laboratories U.S.A. is not yet provided with the function to deal with the sodium-concrete reaction, as well as the water release from a heated concrete, which is thought to be the important reaction occurring in the event of a thermal failure of the liner due to its contact with high temperature sodium or core debris following an LMFBR hypothetical accident. Thus, the current CONTAIN code can not consistently evaluate the influence of water vapor, hydrogen gas, and reaction heat during the course of the containment analysis.

Actually, the sodium-concrete reaction code SLAM is incorporated in the current CONTAIN code. But, since it is perfectly isolated from the other modules, its interaction with the other calculation modules can not be evaluated. A full integration of SLAM with other modules is needed to utilize CONTAIN as a consistent analytical tool. In addition, since concrete is heated by sodium or debris before the liner fails, water migration should be considered.

This report describes the incorporation of the SLAM code into CONTAIN, and its modification for analysis of water migration. Details of program structure and variable usage are presented.

2. Current Concrete Model in CONTAIN

Figure 2.1 shows a calculation flow diagram of the current concrete module, CONCRE, in CONTAIN. CONCRE consists of the DEBCON module and the SLAM module that handle the debris-concrete reaction and the sodium-concrete reaction, respectively. But the SLAM module is incorporated in the form of an isolated unit. Therefore, the problem associated with CONCRE is that one can not consistently calculate the sodium-concrete reaction in combination with other phenomena.

Another problem associated is that DEBCON and SLAM employ different solution methods as shown in Table 2.1. The consistency of these two modules is not considered at all.

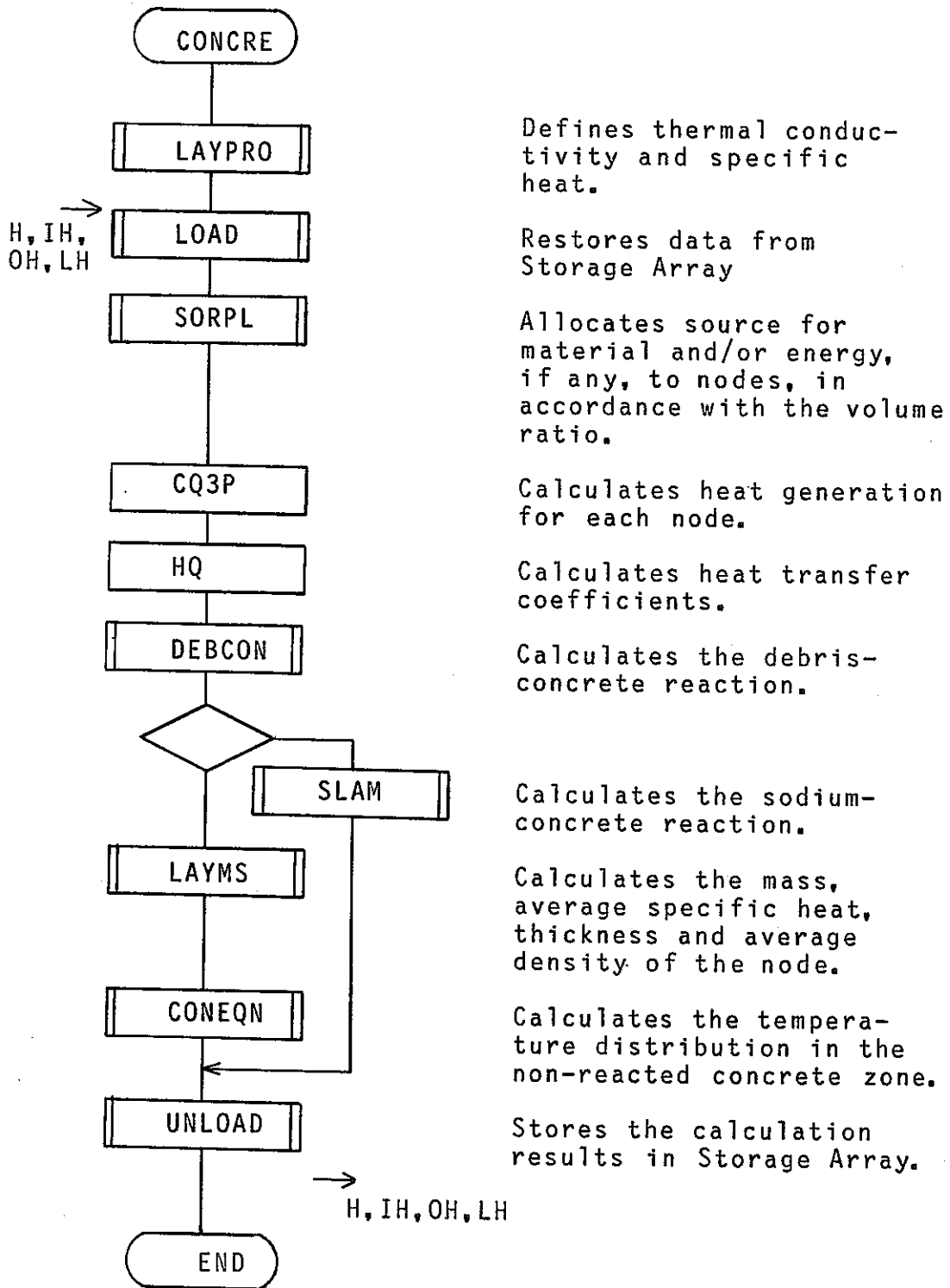


Figure 2.1 Calculation Flow Diagram for CONTAIN Concrete Reaction Module

Table 2.1 Comparison of DEBCON and SLAM Modules in Solution Methods

	CONTAIN	SLAM
Coordinate System	One-dimensional fixed coordinate system	One-dimensional moving Eulerian coordinate system
Number of Concrete Nodes	Several	Several tens
Concrete Node Height	uniform intervals	Dry zone ; uniform intervals Wet zone ; non-uniform intervals
Movement	Handles only water contained in the ablated concrete.	Handles both evaporable and bound water. Handles also movement thru the dry / wet boundary, using the Darcy's law.
Solving of Energy Conservation Equation	ASWR method	Galerkin method
Concrete Properties		
Specific Heat	$C_p = \sum W_i C_{pi}$	Constant
Thermal Conductivity	$T_k = \sum W_i T_{ki}$ (W_i =weight ratio)	$T_k = T_{ka} * T + T_{kb}$ (T_{ka}, T_{kb} ; constant)

3. Outline of the Original Stand-Alone Code SLAM

The original SLAM code consists of seventeen subroutines. Figure 3.1 presents the configuration of the program.

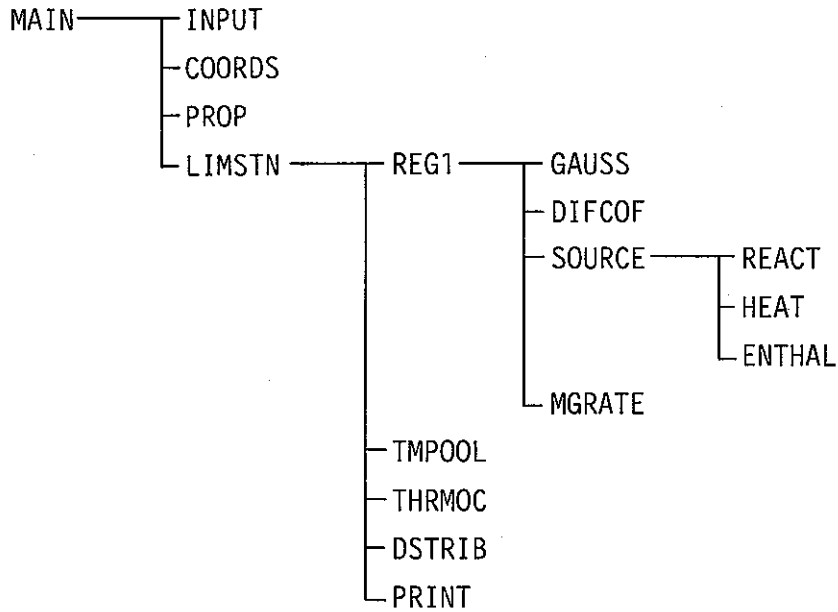


Figure 3.1 Configuration of the Original SLAM code

Individual subroutines are outlined below and the detailed calculation flow is presented in Figure 3.2.

(1) MAIN

Main routine calls an input data routine (INPUT), initialization routines (COORDS, PROP), and a transient calculation routine (LIMSTN).

(2) INPUT

Reads input data from Units 9 and 10.

(3) COORDS

Sets up a coordinate system for the dry and wet zones, weighting the coefficients of individual coordinate points, and coefficients of trial function for coordinate points.

(4) PROP

Sets up the properties of materials used in calculation (concrete, water, sodium, H_2 , NaOH, Na_2SiO_3 , Na_2CO_3 , Na_2O , CaO, $CaCO_3$, CO_2 , C, $MgCO_3$, MgO).

(5) LIMSTN

The driver routine for the transient calculation and output control. Calls the concrete zone routine (REG1), the pool zone routine (TMPPOOL), and output routines (THRMOC, DSTRIM, PRINT).

(6) REG1

Controls the iteration procedure of entire calculation of the concrete zone. The dry zone is calculated by REG1 itself using the slave routines (NEWTRM, GAUSS, DIFCOF, SOURCE). On the other hand, the wet zone is calculated by calling MGRATE.

(7) GAUSS

Solves the simultaneous equations by the Gauss method.

(8) DIFCOF

Calculates both diffusion (DIFF) and viscosity (XMU) coefficients with temperature and pressure as inputs.

(9) SOURCE

Calculates the densities and the gas source rate in the dry zone (for B/L, calls REACT to perform its calculation.) Next, calculates the penetration rate and the mass flux going out into the pool zone. Finally, calls HEAT to calculate the heat generation.

(10) REACT

Calculates the densities and the hydrogen gas source rate in the B/L region of the dry zone.

(11) HEAT

Calculates the heat generation in the dry zone.

(12) ENTHAL

Calculates the formation heat of reactants and products with temperature and phase as inputs.

(13) MGRATE

Calculates the moving velocity of the dry-wet boundary and temperature, density, and pressure in the wet zone. Provided that the density calculation is done at intervals of MIG (input data) step.

(14) TMPOOL

Calculates the densities and the temperature in the pool zone.

(15) THRMOL

The output routine for the results of calculation. (Outputs to Units 12, 13, 16, 18 and 17.)

(16) DSTRIB

The output routine for the results of calculation.
(Units 11, 14, 15)

(17) PRINT

The output routine for the results of calculation.
(Units 6, 20)

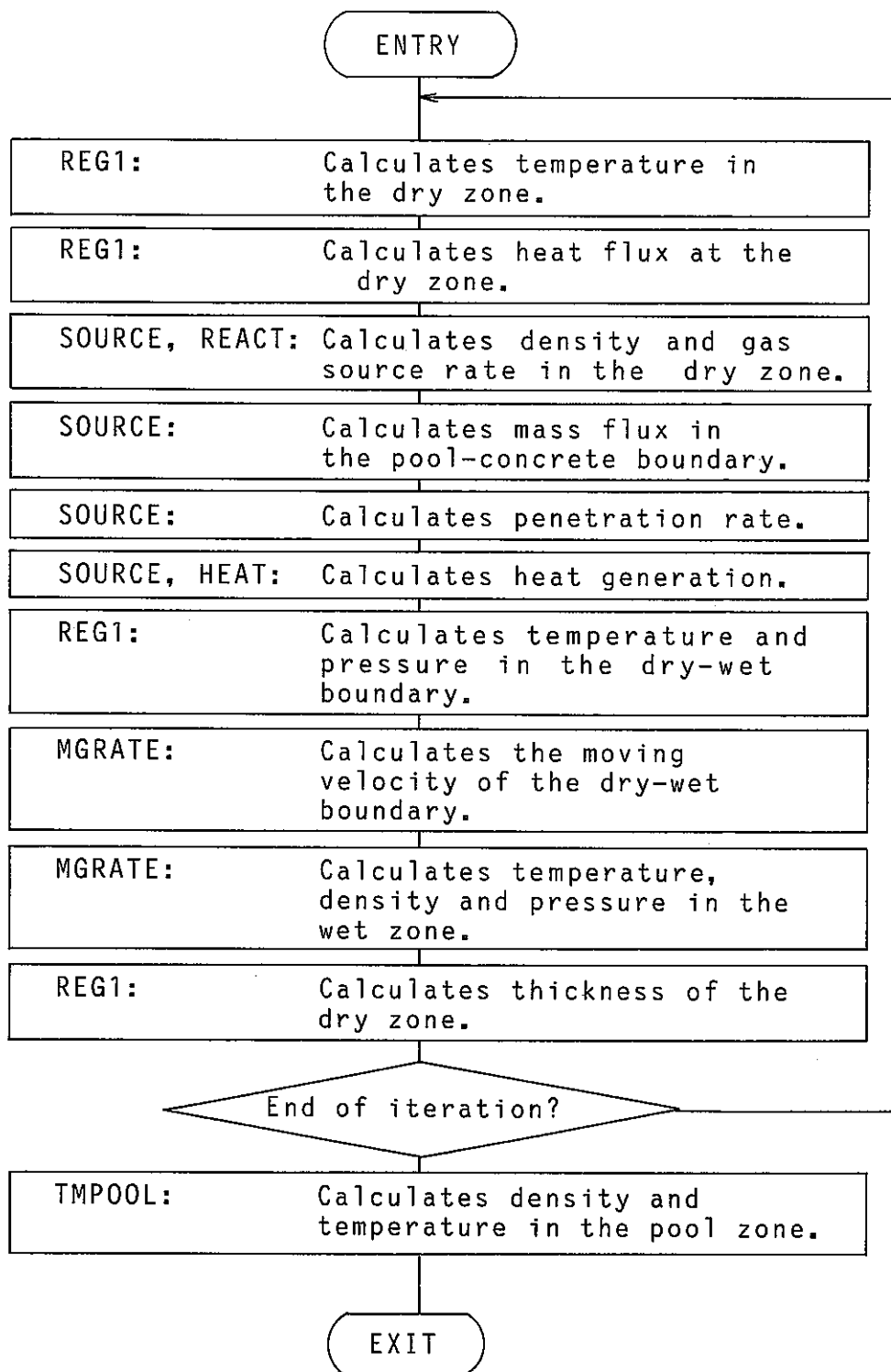


Figure 3.2 Calculation Flow Diagram of Original SLAM

4 Incorporation of SLAM into CONTAIN

4-1. Design Philosophy

The concrete model for DEBCON differs from that for SLAM, as presented in table 2.1, in the way of constructing a coordinate system and solving the conservation equations.

Especially with respect to the way of solving the conservation equations, subprogram CONCRE uses the ASWR method and original SLAM uses the the Galerkin method using the polynomial approximations.

The following philosophy was, therefore, decided to merge the original SLAM code :

- (1) When the calculation of water release or sodium-concrete reaction is not required, the current concrete module CONCRE is used.
- (2) When the calculation of water release or sodium-concrete reaction is required, the concrete model of the original SLAM is adopted as a new subprogram in stead of CONCRE. The new subprogram is named CONCRM that exists in parallel with CONCRE.
- (3) Masses and energy calculated by CONCRM are transported into the pool, the interm, or the atmosphere module in CONTAIN. Then, the sodium pool and the atmosphere models of the original SLAM were eliminated.
- (4) The water release calculation is consistent with the sodium-concrete reaction calculation.
- (5) As interim treatment, almost all of the properties and variables are used in the same manner as the original SLAM code to minimize the change of the module.

It is thus necessary to modify some of the CONTAIN modules, by adding the new subprograms and be replacing or removing the subprograms of the original SLAM. Figure 4.1 shows the flow diagram of the new Lower-Cell control routine, where LAYMOD and CONCRM are newly added.

4-2. Input/Output

Input data for the water release and the sodium-concrete reaction are read by SLINPT, SLCHEM, SLCOOR and SLPROP routines. Output is done by using the subroutines of CONTAIN. Here, a slight modification was added to IRCON, IRCTYP, and INACON routines.

4-3. Storage Array Variable ICH, CH

Since a considerable amount of data are required to execute SLAM, a new Storage Array ICH, CH are created for the calculation besides the Storage Array (H, IH, LH, OH) of CONTAIN.

Thus, a concrete pointer setting routine (CONCPT), and routines to store or restore variables into or from the Storage Array variables (TRANA, TRANB) are newly established. In order to transfer the calculational results in the SLAM routine to CONTAIN, TRANC is also newly created.

4-4. Time Step Calculation Routine (STIME)

The subroutine LIMSTN of the original SLAM code is modified to form STIME in order to establish a compatibility in the manner of time step between CONTAIN and SLAM.

4-5. Concrete Module

CONEQS: Calculates the temperature distribution in the concrete and the amount of gas source.

NATCON: Calculates the sodium-concrete reaction in the boundary layer and the ablation velocity.

WTRRLS: Suppresses the reaction and calculates only dry zone densities.

A calculation flow diagram for the concrete module is presented in Figure 4.2. A newly created CONCRM controls whole calculation procedure for the concrete layer and calls the SLAM routine which is a driver routine for the sodium-concrete reaction or the water migration. SLAM consists of the routines described in the next chapter, which have been constructed after disassembling the main part of the original SLAM code.

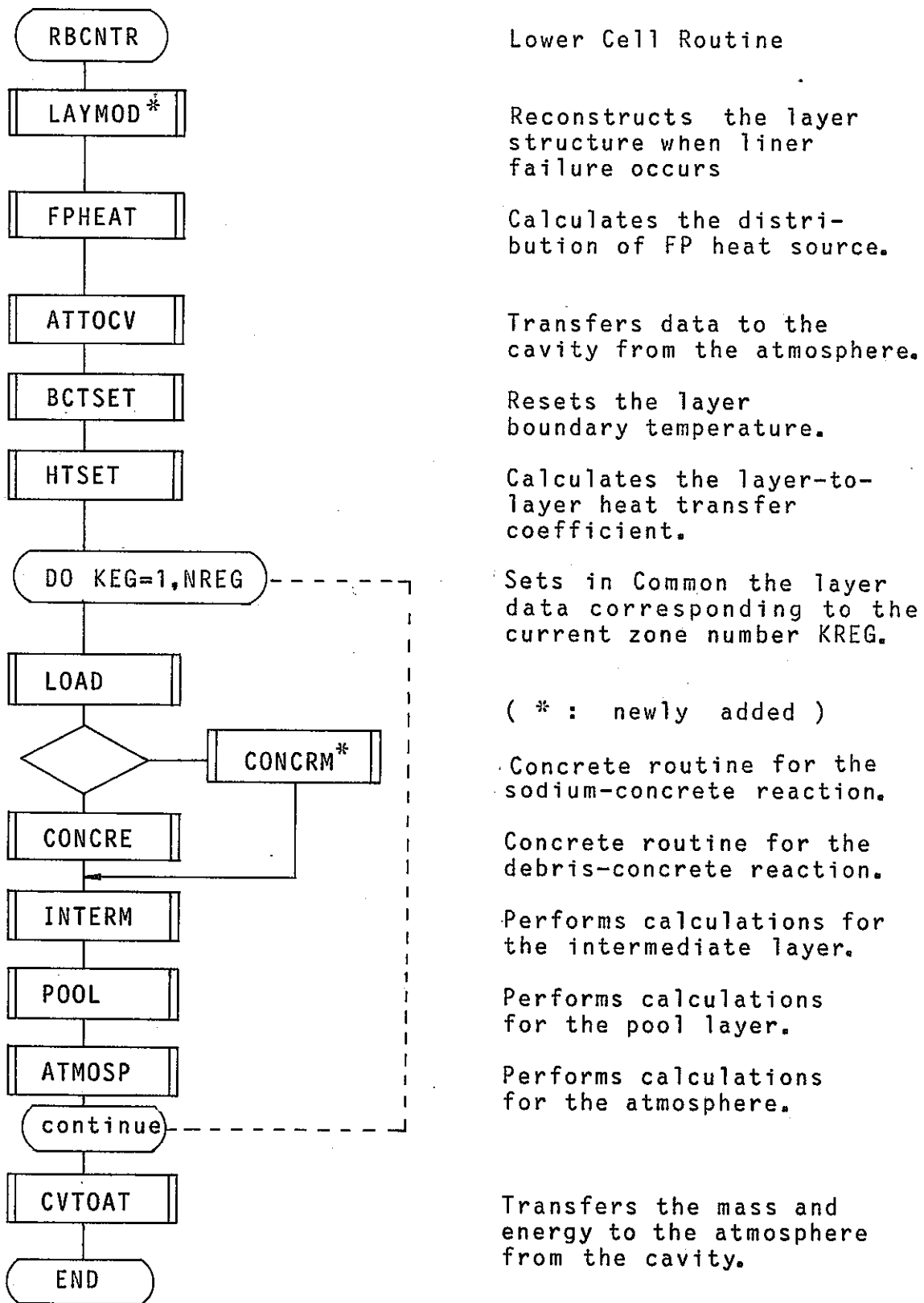


Figure 4.1 Flow Diagram of Lower Cell Control Routine

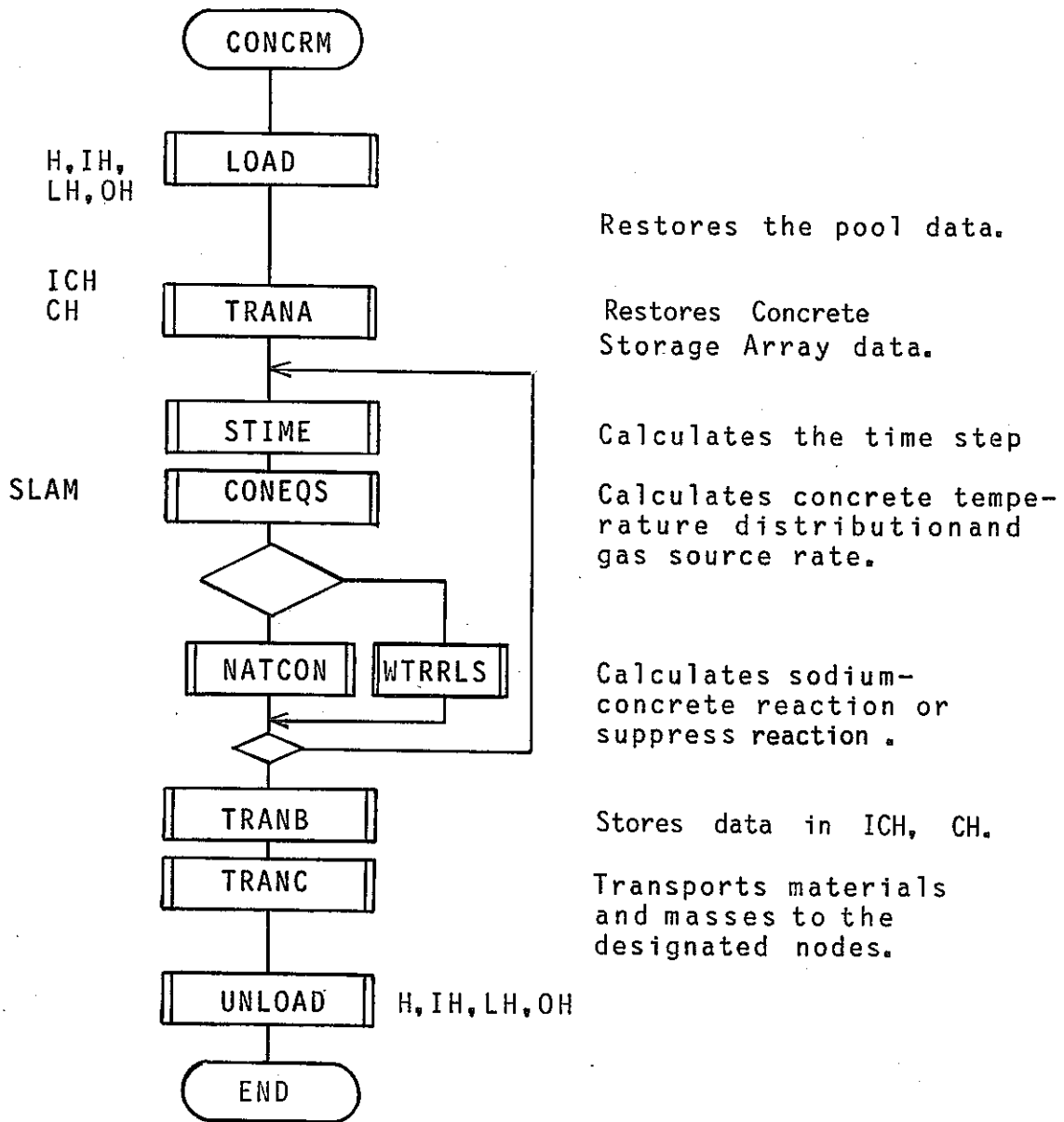


Figure 4.2 Concrete Module Calculation Flow

5. Design of Subroutines5-1. Modification of the Original SLAM Code for CONTAIN Module

For incorporation into the CONTAIN code, it is necessary to modify or remove some routines of the SLAM code, as presented in Table 5.1.

Table 5.1 Modifications of SLAM Subroutines

Original SLAM	Modular SLAM	Modification for CONTAIN
1. INPUT	SLINPT	Addition of some new functions
	SLCHEM	Partial modifications
2. COORDS	SLCOORD	No modification
3. PROP	SLPROP	No modification
4. LIMSTN	STIME	Partial modifications
5. REG1	CONEQS	Partial modifications
6. GAUSS	GAUSS	No modification
7. DIFCOF	SLDIFC	No modification
8. SOURCE	SORGAS, NATCON	Partial modifications
	WTRRLS	Partial modifications
9. REACT	REACT	No modification
10. HEAT	HEAT	No modification
11. ENTHAL	ENTHAL	No modification
12. MGRATE	MGRATE	No modification

Table 5.1 (continued)

SLAM	Modular SLAM	Modification for CONTAIN
13. TRISLV	SLTRIS	No modification
14. TMPOOL		(Removed)
15. THRMOC		(Removed)
16. DSTTRIB		(Removed)
17. PRINT	SLOUT	Partial modifications

5-2. Newly Established Subroutines

- 1 CONCRM
- 2 CONCPT
- 3 LAYMOD
- 4 TRANA
- 5 TRANB
- 6 TRANC
- 7 SLAM

Subroutines which need to be newly established or which require additional functions are described in the following section. The results of coding are presented in Chapter 8.

5-3. Description of Subroutines

5-3.1 SLINPT and SLCHEM

Reads input data related to concrete cell by cell, and sets up initial and parameter values.

SLCHEM:

(1) Setting of Chemical Reaction Data (The original SLAM code TAPE 9 data are included as data sentences.)

SAK ; reaction rate coefficient
EK ; activation energy
AKF ; maximum value of reaction rate coefficient
COFR ; stoichichemical coefficient of reactants
COFP ; stoichichemical coefficient of products
IPHASR ; phase of reactants

IPHASP ; phase of products
NRCT ; number of reactants related to reaction
NPROD ; number of reaction products
MRR ; reactant material number
MPP ; product material number

SLINPT:

- (1) Sets up ISRC, NDRY and NWET (Reads and calculates the number of concrete nodes NNP.)
ISRC ; number of nodes of the boundary layer + 1
NDRY ; number of nodes in the dry zone + 1
NWET ; number of nodes in the wet zone + 1
- (2) Sets up pointers for Concrete Storage Array
- (3) Sets up initial values of SDT through H2FLXT
- (4) Sets up initial values of dry zone density RODRY (i,j)
- (5) Sets up coordinates of the dry and wet zones
- (6) Sets up TDRY (i), TKDRY (i) and TKWET
- (7) Sets up concrete thickness (Calculates from given mass, cross section area and density)
- (8) Sets up TWET (i) through ALPH (i)
- (9) Sets up SIGN (i) and AM (i)
- (10) Sets up ZCOSI (i,j)

5-3.2 STIME

Calculates a time step to be used in the SLAM model or the water release calculation on the basis of the cell time step designated by the input of CONTAIN.

 Interface
 SUBROUTINE STIME (SLTIME, WDT, ISLAF, WTEMP, TSO, WDTDT)

Name	Type	I/O	Content	Unit
SLTIME	R	I/O	time after passing of SDT.	sec
WDT	R	I/O	time used to determine SDT.	sec
ISLAF	I	I	flag to determine the end of SLAM calculation	-
WTEMP	R	I	boundary temperature	K
TSO	R	I	previous dry/wet boundary temperature	K
WDTDT	R	I	boundary temperature changing rate.	K/sec

The SLAM time step SDT is determined on the basis of the lower cell time step PDT as follows;

- (1) Finds SDT exactly in the same manner as SLAM.
- (2) (a) When SDT is greater than, or equal to PDT, SDT shall be equal to PDT. In that case, it will be exactly the same as the pool time step.
- (b) When $(n-1)*SDT$ is less than, or equal to PDT and PDT is less than $n*SDT$ (n is an integer number)
 - i) If $PDT - (n - 1)*SDT$ is less than, or equal to $SDT/4$, calculates $(n - 2)$ times using SDT, and the next time step shall be $PDT - (n - 2)*SDT$.
 - ii) If $PDT - (n - 1)*SDT$ is greater than $SDT/4$, calculates $(n - 1)$ times using SDT. The next time step shall be $PDT - (n - 1)*SDT$.

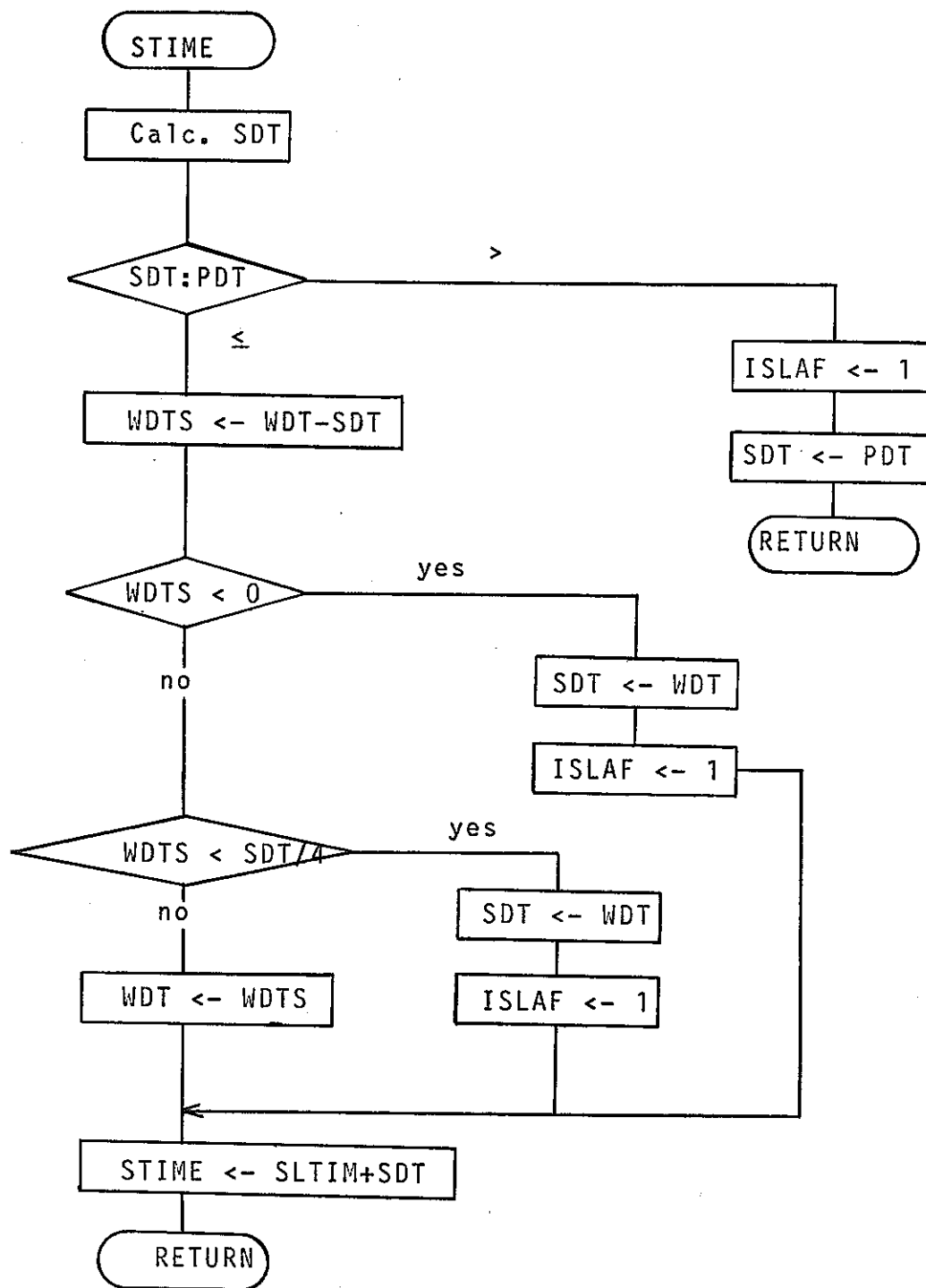


Figure 5.1 Flow Diagram of STIME

(Note: The subroutine LIMSTN of the original SLAM code has been modified.)

5-3.3 CONCRM

This is a concrete layer control routine.

Interface

SUBROUTINE CONCRM (ATA, ATF, DIAG, FJAC, FVEC, XSOL,WRK,
 IWRK, CKM, CCP, HQ, KFLPL, KSPPL, ONAMES,
 OMMTPL, OTYP, CQFP, CQ3P, CA, CDZ, CDZO,
 CMA, CMAO, CRM, CRMO, TBCL, TBCU, CT,
 CTA, CTAO, VCON, PMA, CTBX, CTBY, OCTBNX,
 OCTBNY, OCTBNM, ICTBFL, ICTBNP, KZO, NNP,
 ASRE, ASRM, PSRE, PSRM, RISRE, RISRM,
 RIMA, RIQ3P, CSRE, CSRM, MW, HL, NIND,
 NMAT, N, PT, PRM, PTA, PTAO, PDZ, PQ3P,
 AQ3P, RIT, RITA, RITAO, RIDZ, RIRM)

Name	Type	I/O	Content	Unit
ATA(N,*)	R	I	$D^{-1} * A = ATA(N,N)$	-
ATF(*)	R	I	$D^{-1} * F = ATF(N)$	-
DIAG(*)	R	I	work area	-
FJAC(N,*)	R	I	A(N,N)	-
FVEC(*)	R	I	F(N)	-
XSOL(*)	R	O	work area	-
WRK(*)	R	I	do	-
IWRK	I	I	do	-
CKM(*)	R	O	node thermal conductivity	J/m2/s/K

(continued)

Name	Type	I/O	Content	Unit
CCP(*)	R	0	node average specific heat	J/kg/K
HQ(*)	R	0	thermal conductivity	J/m ² /s/K
KFLPL(*)	I	I	pointer	-
KSPPL(*)	I	I	do	-
ONAMES(*)	C	I	character	-
ONMTPL(*)	C	I	do	-
OTYP(*)	C	I	do	-
CQFP(*)	R	0	heat generation	J/sec
CQ3P(*)	R	0	heat generation	J/sec
CA(*)	R	I	node cross section	m ²
CDZ(*)	R	0	node thickness	m
CDZO(*)	R	I	CDZO = CDZ(t-dt)	m
CMA(NMAT,*)	R	0	mass of material J in the KZ node	kg
CMAO(NMAT,*)	R	I	CMAO = CMA(t-dt)	kg
CRM(*)	R	0	node mass density	kg/m ³
CRMO(*)	R	I	CRMO = CRM(t-dt)	kg/m ³
TBCL(*)	R	I/O	layer boundary temperature (bottom)	K
TBCU(*)	R	I/O	layer boundary temperature (top)	K
CT(*)	R	0	temperature	K
CTA(*)	R	0	average temperature	K
CTAO(*)	R	I	CTAO = CTA(t-dt)	K

(continued)

Name	Type	I/O	Content	Unit
VCON(*)	R	O	effective velocity	m/sec
PMA(NMAT,*)	R	I	mass of the pool node	kg
CTBX(*)	R	I	table (independent variable)	-
CTBY(*)	R	I	table (dependent variable)	-
OCTBNX(*)	C	I	character	-
OCTBNI(*)	C	I	do	-
OCTBNM(*)	C	I	do	-
ICTBFL(*)	I	I	Interpolation info. for cell tables	-
ICTBNP(*)	I	I	No. of data point for cell tables	-
KZO(*)	I	I	Beginning of node by layer	-
NNP(*)	I	I	number of nodes in the layer	-
ASRE (NIND, NMAT, *)	R	I/O	energy rate (atomos- phere)	J/sec
ASRM (NIND, NMAT, *)	R	I/O	mass rate (atomos- phere)	kg/sec
PSRE (NIND, NMAT, *)	R	I/O	pool energy rate	J/sec
PSRM (NIND, NMAT, *)	R	I/O	pool mass rate	kg/sec
RISRE (NIND, NMAT, *)	R	I/O	intermediate energy rate	J/sec

(continued)

Name	Type	I/O	Content	Unit
RISRM (NIND, NMAT, *)	R	I/O	intermediate mass rate	kg/sec
RIMA(NMAT, *)	R	I/O	intermediate mass	kg
RIQ3P(*)	R	I/O	intermediate heat source rate	J/sec
CSRE (NIND, NMAT, *)	R	I/O	concrete energy rate	J/sec
CSRM (NIND, NMAT, *)	R	I/O	concrete mass rate	kg/sec
MW(*)	R	I	molecular mass	kg
HL(*)	R	0	thermal conductivity coefficient	W/m ² /K
NIND	I	I	= 2	-
NMAT	I	I	number of materials	-
PT(*)	R	I	node temperature in the pool layer (three points per node)	K
PRM(*)	R	I	pool layer density	kg/m ³
PTA(*)	R	I	node average tempera- ture in the pool layer	K
PTA(*)	R	I	node average tempera- ture in the pool layer of the preceding time step	K
PDZ(*)	R	I	node thickness in the pool layer	m

(continued)

Name	Type	I/O	Content	Unit
PQ3P(*)	R	I	volume heat source for nodes in the pool layer	J/sec
AQ3P(*)	R	I	volume heat source for nodes in the atmosphere	J/sec
RIT(*)	R	I	node temperature in the interm layer	K
RITA(*)	R	I	node average temperature	K
RITAO(*)	R	I	node average tempera- ture in the interm layer of the preceding time step	K
RIDZ(*)	R	I	node thickness in the interm layer	m
RIRM(*)	R	I	interm layer density	kg/m ³

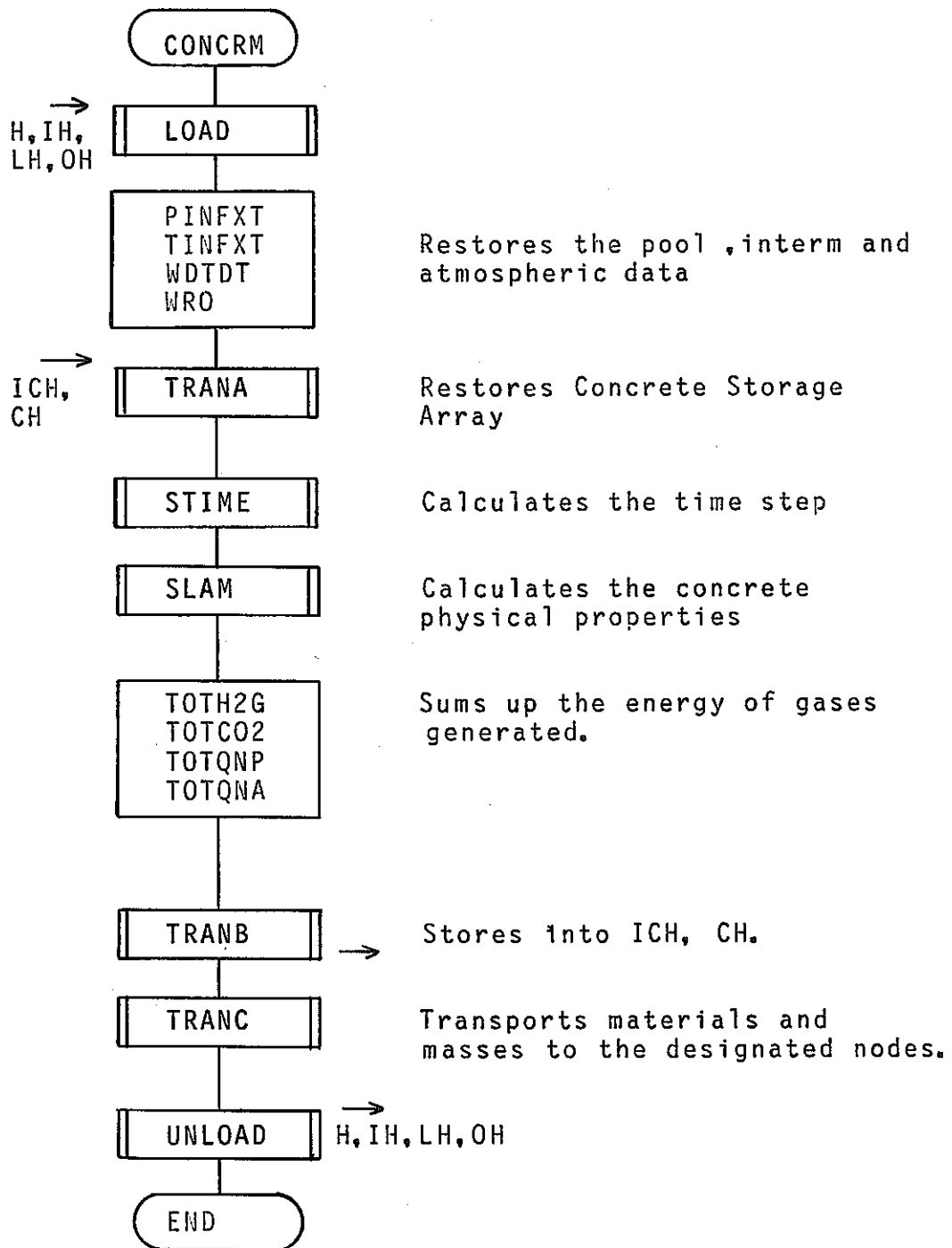


Figure 5.2 Flow Diagram of CONCRM

(Note: The energy of the gasses generated (H_2, CO_2) will be given to the pool, and the masses will be sent to the atmosphere.)

5-3.4 SLAM

Controls the subroutines CONEQS and NATCON.

 Interface

SUBROUTINE SLAM (TIME, DT, WTEMP, WRO, PINFXT,
TINFXT, WDTDT, RLIQ, PFLUX)

Name	Type	I/O	Content	Unit
TIME	R	I	system time	sec
DT	R	I	time step	sec
WTEMP	R	I	boundary temperature	K
WRO(17)	R	I	pool or interm density	kg/m ³
PINFXT	R	I	atmospheric pressure	Pa
WDTDT	R	I	upper boundary temperature changing rate	K/sec
RLIQ(51,8)	R	O	reaction rate	1/sec
PFLUX(17)	R	O	mass flux	kg/m ² /sec

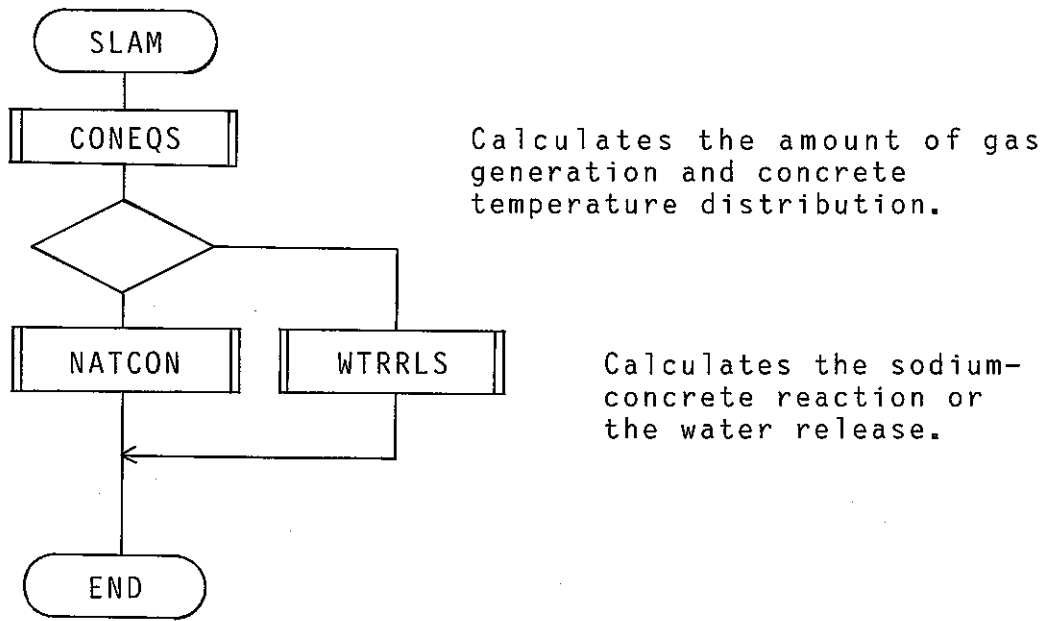


Figure 5.3 Flow Diagram of SLAM

5-3.5 CONEQS

- (1) Calculates the concrete (dry zone) temperature distribution.
- (2) Calculates the generation of carbon dioxide and bound water in accordance with the calculated temperature distribution in the dry zone.
- (3) Calls MGRATE and calculates the moving velocity of the dry-wet boundary and temperatures and densities in the wet zone.

Interface

SUBROUTINE CONEQS (TIME, DT, WTEMP, PINFXT, TINFXT,
 WDTDT, DEL10, TRO, H2OFLO, EPP, ERR1,
 COMP1, TOK, SUMTRM, FX, AMAT, CCOL, XVAR,
 AMO, SINT, VELOLD)

Name	Type	I/O	Content	Unit
TIME	R	I	system time	sec
DT	R	I	system time step	sec
WTEMP	R	I	upper boundary temperature	K
PINFXT	R	I	atmospheric pressure	Pa
TINFXT	R	I	atmospheric temperature	K
WDTDT	R	I	upper boundary temperature changing rate	K/sec

(continued)

Name	Type	I/O	Content	Unit
DEL10	R	I	DEL1 of the preceding step	m
TRO	R	I	TR of the preceding step	K
H2OFLO	R	I	H2O flux of the preceding step	kg/m ² /s
EPP	R		(not used)	-
EPR1	R	I	conditions for judging convergence	-
COMP1(51)	R	O	work area	-
TOK(51)	R	O	work area	-
SUMTRM(51)	R	O	work area	kg/m ³ /s
FX(51)	R	O	(not used)	-
AMAT(12,12)	R	O	work area	-
CCOL(12)	R	O	work area	-
XVAR(22)	R	O	work area	W/m ²
AMO(12)	R	O	polynomial expansion coefficient	-
SINT(51)	R	O	integral of source terms	kg/m ³ /s
VELOLD	R	O	VELOLD = VEL(t-dt)	m/sec

Note: This routine has been constructed by partially modifying the subroutine REG1 of the SLAM code.

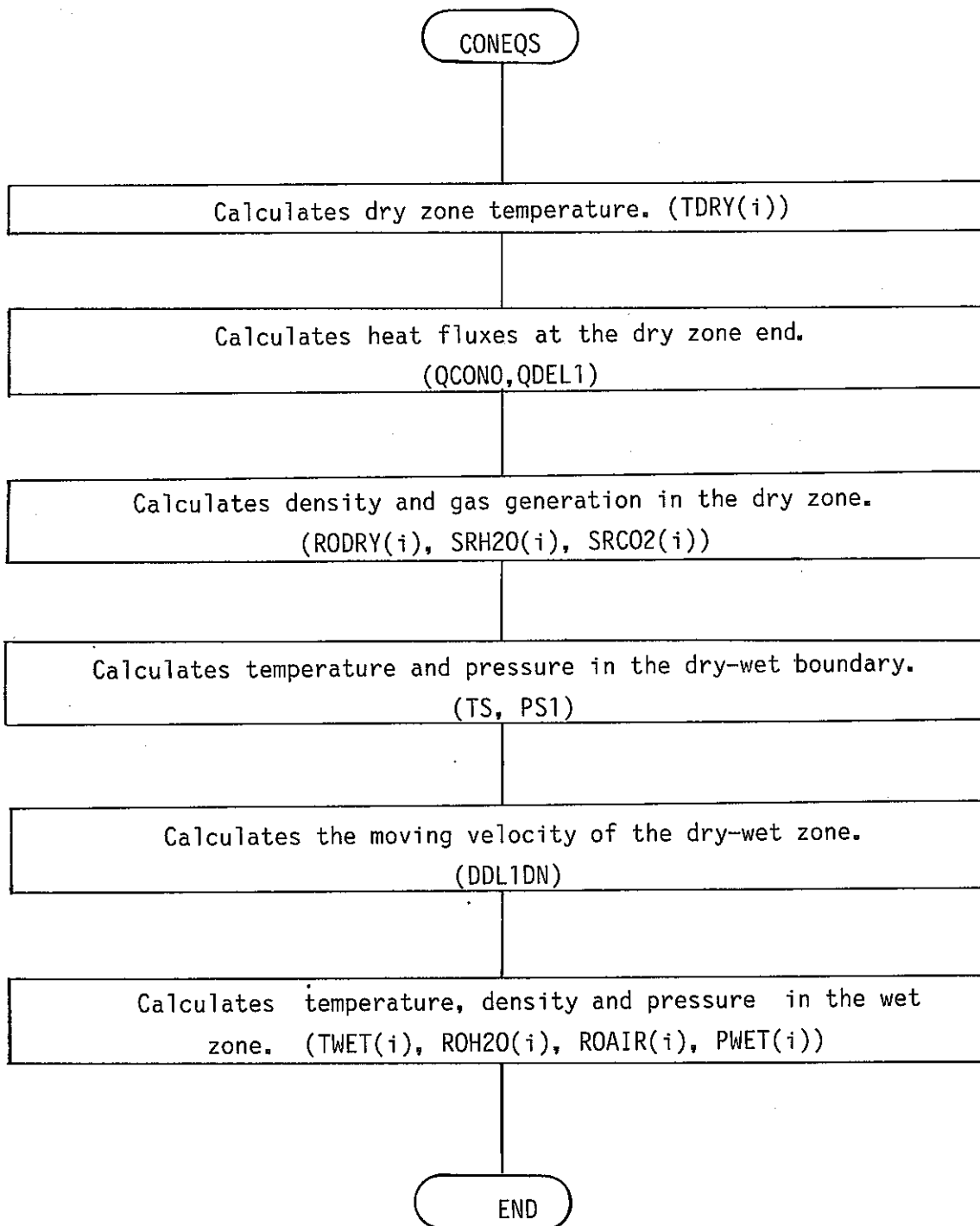


Figure 5.4 Flow Diagram of CONEQS

5-3.6 NATCON

Calculates the sodium-concrete reaction.

Interface

SUBROUTINE NATCON (TIME, DT, TPOOL, ROPOOL, RLIQ, DIFO,
PFLUX, PINFXT, TINFXT)

Name	Type	I/O	Content	Unit
TIME	R	I	system time	sec
DT	R	I	system time step	sec
TPOOL	R	I	pool surface temperature	K
ROPOOL(17)	R	I	density in the pool layer	kg/m ³
RLIQ(51,8)	R	O	reaction rate	1/sec
DIFO	R	O	diffusion coef. parm.	-
PFLUX(17)	R	O	pool-concrete boundary mass flux	kg/m ² /s
PINFXT	R	I	atmospheric pressure	Pa
TINFXT	R	I	atmospheric temperature	K

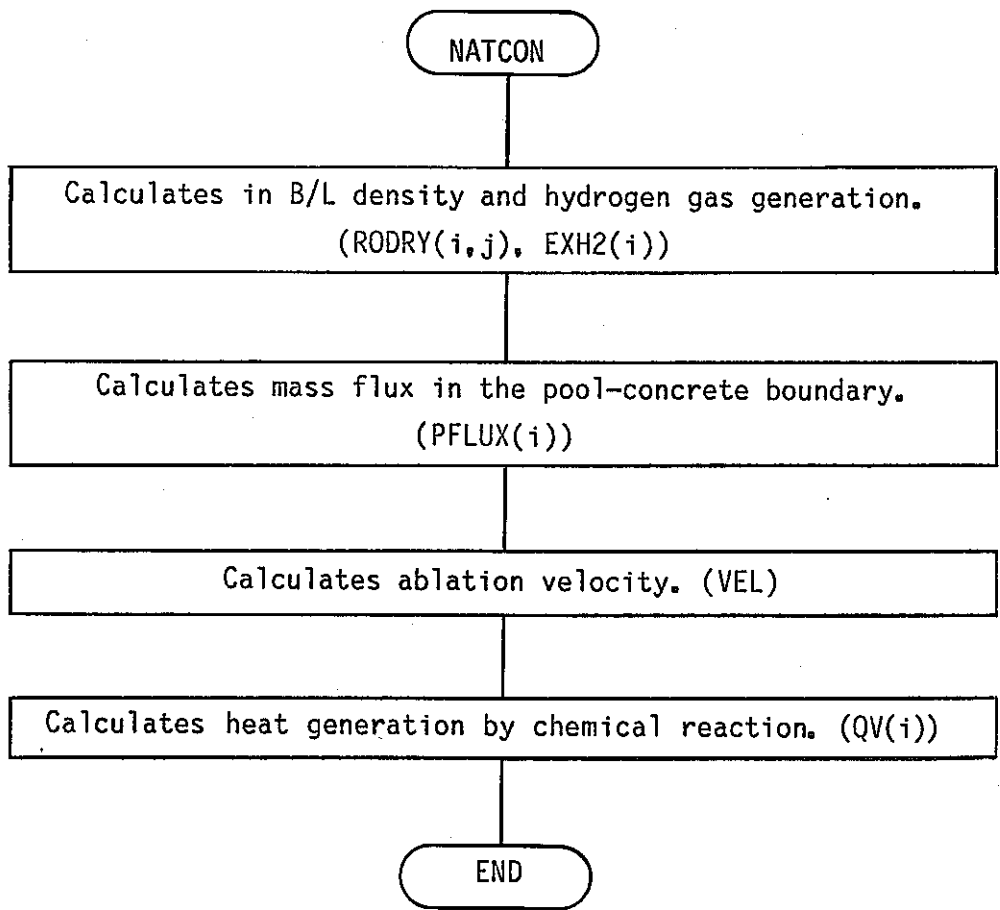


Figure 5.5 Flow Diagram of NATCON

5-3.7 WTRRLS

Calculates the dry zone densities suppressing the calculation of the chemical reaction and the ablation velocity.

Interface

SUBROUTINE WTRRLS (TIME, DT, WTEMP, WRO, RLIQ, DIFO, PFLUX,
PINFXT, TINFXT)

Name	Type	I/O	Content	Unit
TIME	R	I	system time	sec
DT	R	I	system time step	sec
WTEMP	R	I	interm surface temperature	K
WRO(17)	R	I	density in the interm layer	kg/m3
RLIQ(51,8)	R	O	reaction rate	1/sec
DIFO	R	O	diffusion coef. parm.	-
PFLUX(17)	R	O	interm-concrete boundary mass flux	kg/m2/s
PINFXT	R	I	atmospheric pressure	Pa
TINFXT	R	I	atmospheric temperature	K

5-3.8 SORGAS

Calculates the bound water and carbon dioxide generation rate from the dry zone in accordance with the temperature distribution calculated by CONEQS.

Interface

SUBROUTINE SORGAS (TIME, DT)

Name	Type	I/O	Content	Unit
TIME	R	I	system time	sec
DT	R	I	system time step	sec

5-3.9 CONCPT

Sets up the pointers for Concrete Storage Array variables ICH, CH.

5-3.10 LAYMOD

Reconstructs the layer structure when a liner failure occurs

Interface

SUBROUTINE LAYMOD (OTYP,KZO,NNP)

Name	Type	I/O	Content	Unit
OTYP(*)	0	I	layer name	-
KZO(*)	I	I	beginning of node by layer	-
NNP(*)	I	I	number of node by layer number	-

5-3.11 TRANA

Retrieves data into the common variables from Storage Array for the SLAM calculation.

5-3.12 TRANB

Stores data into Storage Array of SLAM from the common variables.

5-3.13 TRANC

Transfers the energy and mass generated by the sodium-concrete reaction to the pool, and intermediate layer, or atmosphere.

Interface

SUBROUTINE TRANC (CA, CDZ, CMA, CT, CTA, CRM, KZO, NNP, ISLCN, PSRM, PSRE, ASRM, ASRE, TOT2G, TOTCO2, TOTQNP, TOTQNA, TOTMS, PQ3P, AQ3P, RISRM, RIQ3P, NIND, NMAT)

Name	Type	I/O	Content	Unit
CA(*)	R	I	cross section area of the concrete layer	m ²
CDZ(*)	R	I	thickness of a node in the concrete layer	m
CMA(NMAT,*)	R	I	mass of material in the node	kg
CT(*)	R	I	concrete layer temperature	K
CTA(*)	R	I	concrete average layer temperature	K
CRM(*)	R	I	node density in the concrete layer	kg/m ³
KZO(*)	I	I	initial node number of the designated layer	-

(continued)

Name	Type	I/O	Content	Unit
NNP(*)	I	I	number of nodes in the designated layer	-
ISLCN(*)	I	I	array for conversion to CONTAIN from SLAM	-
PSRM (NIND, NMAT, *)	R	I/O	mass transfer rate to the pool	kg/sec
PSRE (NIND, NMAT, *)	R	I/O	energy transfer rate to the pool	J/sec
ASRM (NIND, NMAT, *)	R	I/O	mass transfer rate to the atmosphere	kg/sec
ASRE (NIND, NMAT, *)	R	I/O	energy transfer rate to the atmosphere	J/sec
TOTH2G	R	I/O	amount of hydrogen gas generation	kg
TOTCO2	R	I/O	amount of carbon dioxide generation	kg
TOTQNP	R	I/O	energy transfer rate to the pool	J/sec
TOTQNA	R	I/O	energy transfer rate to the atmosphere	J/sec
TOTMS(*)	R	I/O	mass transfer rate to the pool	kg/sec
PQ3P(*)	R	I/O	node volume heat source in the pool layer	J/sec
AQ3P(*)	R	I/O	node volume heat source in the atmospheric layer	J/sec

(continued)

Name	Type	I/O	Content	Unit
RISRM (NIND, NMAT, *)	R	I/O	mass transfer rate to the interm layer	kg/sec
RIQ3P(*)	R	I/O	node volume heat source in the interm layer	J/sec
NIND	I	I	I = 2	-
NMAT	R	I/O	number of materials	-

6. Table of Common Variables

When the common variables of the SLAM code overlap with those of the CONTAIN code, those of the CONTAIN code were given priority. The pool-related common variables of the SLAM code were removed.

COMDECK was arranged as shown in the following page.

(1) COMMON /CPROP/

Variable	Unit	Description
W(17)	kg	mol weight
RGAS(17)	m ² /s ² /K	gas constant/molecular weight
RHO(17,3)	kg/m ³	material density
RHOL(17)	kg/m ³	liquid phase micro density
ROMH2O	kg/m ³	water micro density
SP(17,3)	J/kg/K	specific heat
HF(17)	J/kg	enthalpy
HFG(17)	J/kg	latent heat
TKMAT(17)	W/m/K	thermal conductivity
TSTAR(17)	K	PS = PSTAR*exp(-TSTAR/TS)
PSTAR(17)	Pa	
TCRIT(17)	K	HFG = HSTAR*(1-T/TCRIT)**ZETA
ZETA(17)	-	
HSTAR(17)	J/kg	
PSAT(17)	Pa	sodium saturation temperature
TSAT(17)	J/kg	saturation temperature
HFUS(17)	J/kg	dissolving heat
TMLT(17)	K	dissolving temperature
EPSRAD	-	coef. of radiation from the pool
GAMMA(17)	-	ratio of specific heats
XMUL(17)	kg/m/s	liquid phase viscosity coefficient
IGAS(17)	-	phase judging flag
MIG	-	step interval to calculate water movement

(2) COMMON /CCHEM/

Variable	Unit	Description
SAK(8)	1/sec	reaction rate coefficient
EK(8)	cal/mol	activation energy
AKF(8,3)	1/sec	the maximum value of reaction rate coefficient. (default is 1.0)
COFR(8,17)	-	coefficient of reaction substance in the reaction equation
COFP(8,17)	-	coefficient of product in the reaction equation
NREACT	-	number of reaction equations
IPHASR(8,17)	-	phase of reaction substance
IPHASP(8,17)	-	phase of product
MRR(8,3)	-	reaction substance material number
MPP(8,3)	-	product material number
NRCT(8)	-	number of reactants involved in reaction
NPROD(8)	-	number of reaction products
SRH20(51)	kg/m ³ /s	H ₂ O gas generation rate per volume at the dry zone node point
EXH2(51)	m ³ /s	hydrogen generation rate

(3) COMMON /CNOD/

Variable	Unit	Description
ISRC	-	number of nodes in the dry zone boundary layer
ISRCP1	-	ISRC+1
ISRCP3	-	ISRC+3
ISRCP5	-	ISRC+5
NDRY	-	number of nodes in the dry zone
NWET	-	number of nodes in the wet zone
NPOOL	-	number of nodes in the pool zone
NTERM	-	degree of temperature approximation equation for the dry zone
NSLM	-	number of materials handled with SLAM
IMPL	-	frequency of iteration (dry zone)
IMPL2	-	frequency of iteration (wet zone)

(4) COMMON /CCORD/

Variable	Unit	Description
XZI(51)	-	coordinates of node point in the dry zone
WTZ(51)	-	weighting factor of node point in the dry zone
XETA(51)	m	coordinates of node point in the wet zone
ETA(51)	-	coefficient to calculate WT (51)
WT(51)	-	weighting factor of node point in the dry zone
SINX(12,51)	-	coefficient of approximation equation for temperature distribution in the zone
COSX(12,51)	-	the same as above
SIGN(12)	-	=-1.,1., - - - -1.,1. coefficient
ZCOSI(12,12)	-	coefficient used in calculating AM

(5) COMMON /SLCEL1/

Variable	Unit	Description
SDT	sec	time step in sodium-concrete reaction calculation
DEL1	m	dry zone thickness (initial 0.003)
DELM	m	concrete thickness
DDL1DT	m/s	dry zone thickness changing rate
DIST	m	amount of penetration (length)
DDL1DN	m/s	$-Mv/Pw + V2$
VEL	m/s	penetration rate
VELN	m/s	penetration rate
ROCP	J/m ³ /K	RHOCON*CPCON
TC	K	concrete initial temperature
TS	K	dry-wet boundary temperature
PS1	Pa	dry-wet zone boundary pressure
TDIF	W/m/K	concrete thermal conductivity (at average temperature)
COURNT	-	courandt number
EPSM	-	concrete total void
EPSO	-	concrete initial void
TAV	K	dry zone average temperature
QCV	W/m ²	integrated value of heat generation by chemical reaction
QCONO	W/m ²	heat flux from the pool
QCON	W/m ²	heat flux from the wet zone
TKWET	W/m/K	thermal conductivity of the wet zone
QVSUM	W/m ²	integral of heat source and heat sink in the dry zone

(continued)

QDEL1	W/m ²	heat flux in the dry-wet boundary
H2OFLX	kg/m ² /s	mass flux of evaporating water
H2OSRC	kg/m ² /s	H ₂ O generation rate in the dry zone
WFLXDT	kg/m ²	integral of H2OFLX and H2OSRC
CO2SRC	kg/m ² /s	CO ₂ generation rate in the dry zone
XNTCO2	kg/m ²	amount of CO ₂ generated in the dry zone
VGAS	m/s	converted rate of H ₂ generation
XMH2XT	kg/m ² /s	amount of hydrogen gas generation
H2FLXT	kg/m ²	sum of hydrogen gas generation rates

(6) COMMON /SLCEL2/

Variable	Unit	Description
PDRY(51)	Pa	pressure in the dry zone
PWET(51)	Pa	pressure in the wet zone
ROH2OL(51)	kg/m ³	H ₂ O density at a node point in the wet zone
VBUB(51)	m/s	bubble velocity
AM(12)	-	coefficient of temperature function of the dry zone
ALPHAL(51)	-	liquid volume ratio
RODRY (51,17)	kg/m ³	dry zone density
TDRY(51)	K	temperature at a node point in the dry zone
TKDRY(51)	W/m/K	thermal conductivity at a node point in the dry zone
TWET(51)	K	temperature in the wet zone
ROAIR(51)	kg/m ³	concrete gaseous phase density
ALPH(51)	-	air volume ratio
RODDYI (51,17)	kg/m ³	density in the dry zone
RODRYO (51,17)	kg/m ³	density in the dry zone
QV(52)	W/m ³	heat generation by chemical reaction
QVP(51)	W/m ³	heat generation by chemical reaction in the pool
SRCO2(51)	kg/m ³ /s	carbon dioxide gas generation rate per volume at a dry zone node point

(7) COMMON /CPOINT/

Variable	Description
LC1	initial position to store COMMON SLCEL2 in CH
LC2	initial position to store RODDY (51,17) in CH
LC3	initial position to store TDRY (51) in CH
LC4	initial position to store TKDRY (51) in CH
LC5	initial position to store ALPHAL (51) in CH
LC6	initial position to store XETA (51) in CH
LC7	initial position to store TWET (51) in CH
LC8	initial position to store PWET (51) in CH
LC9	initial position to store ROAIR (51) in CH
LC10	initial position to store ROH2OL (51) in CH
LC11	initial position to store ALPH (51) in CH
LC12	initial position to store QV (52) in CH
LC13	initial position to store AM (12) in CH
LC14	initial position to store SIGN (12) in CH
LC15	initial position to store SINX (12,51) in CH
LC16	initial position to store COSX (12,51) in CH
LC17	initial position to store ZCOSI (12,12) in CH
LC18	initial position to store XZI (51) in CH
LC19	initial position to store WTZ (51) in CH
LC20	initial position to store XETA (51) in CH
LC21	initial position to store ETA (51) in CH
LC22	initial position to store WT (51) in CH

7. Program Configuration

The program configuration of CONTAIN after incorporating the sodium-concrete reaction module and the water release module is presented in the following pages.

In the table, each incorporated routine is provided with a suffix '*', and each partially modified routine is provided with a suffix 'o' on the right shoulder.

```

MAIN  ----*ERRSET
      +---INPUT ----SETMA ----->(SETCOM)
      I      I      +---ECHO
      I      I      +---->(NEXT )
      I      I      +---->(ABORT )
      I      I      +---IMAGH----->(ABORT )
      I      I      +---->(VALUE )
      I      I      +---->(FINDAT)
      I      I      +---GLOSET----->(SETCOM)
      I      I      +---BANNER
      I      I      +---->(RESTAR)
      I      I      +---REDEF ----->(NEXT )
      I      I      +---->(ABORT )
      I      I      +---->(VALUE )
      I      I      +---->(RESTAR)
      I      I      +---RESPLT
      I      I      +---->(IPROPS)
      I      I      +---->(NXTCEL)
      I      I      +---->(SAVCEL)
      I      I      +---->(ISOURC)
      I      I      +---->(IAERCL)
      I      I      +---->(SECOND)
      I      I      +---ISPACE
      I      I      +---IGLOBL----->(NEXT )
      I      I      +---->(ABORT )
      I      I      +---->(VALUE )
      I      I      +---->(IPROPS)
      I      I      +---PTRGLD-----IZERO
      I      I      I      +---->(NEXT )
      I      I      I      +---->(CONVER)
      I      I      I      +---->(IARRAY)
      I      I      I      +---->(ABORT )
      I      I      I      +---->(VALUE )
      I      I      I      +---->(ICPERM)
      I      I      I      +---->(IIPERM)
      I      I      I      +---->(ILPERM)
      I      I      I      +---->(IRPERM)
      I      I      I      +---->(PTRQUT)
      I      I      +---INMATN----->(NEXT )
      I      I      I      +---->(ABORT )
      I      I      +---NAMVEC
      I      I      +---IPROP ----->(ABORT )
      I      I      I      +---->(ENTHX )
      I      I      +---IFISSO----->(ABORT )
      I      I      I      +---->(VALUE )
      I      I      I      +---->(NEXT )
      I      I      I      +---->(ICTEMP)
      I      I      I      +---->(IARRAY)
      I      I      I      +---->(ARRAY )
      I      I      I      +---ZERO
      I      I      I      +---->(CONVER)
      I      I      I      +---->(IRTEMP)
      I      I      I      +---->(PRDF )
      I      I      +---->(ICTEMP)
      I      I      +---INAERO----->(NEXT )
      I      I      I      +---->(CONVER)
      I      I      I      +---->(VALUE )
      I      I      I      +---INCOEF

```

```

MAIN  ----INPUT ----IGLOBL----INAERO----->(UNLOAD)
      I      I      I      +---FLGNOR
      I      I      I      +---IFLOW ----->(NEXT )
      I      I      I      +---->(VALUE )
      I      I      I      +---->(ITABLE)
      I      I      I      +---->(IENGSY)
      I      I      I      +---->(DECAY )
      I      I      I      +---CELSET----->(SETCOM)
      I      I      I      +---ICELL ----->(VALUE )
      I      I      I      +---->(NEXT )
      I      I      I      +---PTRCELo----->(TIMDAT)
      I      I      I      +---IZERO
      I      I      I      +---->(NEXT )
      I      I      I      +---->(CONVER)
      I      I      I      +---->(IARRAY)
      I      I      I      +---->(ABORT )
      I      I      I      +---->(VALUE )
      I      I      I      +---->(ICPERM)
      I      I      I      +---->(IIPERM)
      I      I      I      +---->(ILPERM)
      I      I      I      +---->(IRPERM)
      I      I      I      +---->(PTROUT)
      I      I      I      +---ICAM ----->(NEXT )
      I      I      I      +---IATMOS----->(VALUE )
      I      I      I      I      +---->(PSATX )
      I      I      I      I      +---->(ENTHX )
      I      I      I      I      +---->(ICTEMP)
      I      I      I      I      +---->(IRTEMP)
      I      I      I      I      +---->(NEXT )
      I      I      I      I      +---->(ABORT )
      I      I      I      I      +---->(SHX )
      I      I      I      +---->(IAERCL)
      I      I      I      +---ISPRAY----->(NEXT )
      I      I      I      I      +---->(VALUE )
      I      I      I      I      +---->(ISOURC)
      I      I      I      I      +---->(ABORT )
      I      I      I      +---->(ICTEMP)
      I      I      I      +---IHEATX----->(ABORT )
      I      I      I      I      +---->(NEXT )
      I      I      I      I      +---->(VALUE )
      I      I      I      I      +---->(ARRAY )
      I      I      I      I      +---->(RHOX )
      I      I      I      +---ICHEM ----->(ABORT )
      I      I      I      +---IBURN ----->(NEXT )
      I      I      I      I      +---->(VALUE )
      I      I      I      +---->(ISOURC)
      I      I      I      +---->(ABORT )
      I      I      I      +---ICCOND----->(NEXT )
      I      I      I      +---->(ITEMP)
      I      I      I      +---->(VALUE )
      I      I      I      +---->(ITABLE)
      I      I      I      +---->(ABORT )
      I      I      I      +---IZERO
      I      I      I      +---IREBELo----->(NEXT )
      I      I      I      +---->(SETCOM)
      I      I      I      +---->(ABORT )
      I      I      I      +---IRCONo----->(NEXT )
      I      I      I      I      +---->(VALUE )

```

```

MAIN  ----INPUT ----ICELL ----IREBELo----IRCONo----IRCTYPo---->(NEXT )
      I      I      I      I      I      I      +---->(ENTHX )
      I      I      I      I      I      I      +---->(RHOX )
      I      I      I      I      I      I      +---->(VALUE )
      I      I      I      I      I      I      +---->(ABORT )
      I      I      I      I      I      I      +---INACONo---->(NEXT )
      I      I      I      I      I      I      +---SLPROP*
      I      I      I      I      I      I      +---->(VALUE )
      I      I      I      I      I      I      +---SLPROP*
      I      I      I      I      I      I      +---->(ISOURC)
      I      I      I      I      I      I      +---->(ITABLE)
      I      I      I      I      I      I      +---->(ABORT )
      I      I      I      I      I      I      +---->(RHOX )
      I      I      I      I      I      I      +---->(UNLOAD)
      I      I      I      I      I      I      +---IRINTo---->(NEXT )
      I      I      I      I      I      I      +---->(VALUE )
      I      I      I      I      I      I      +---->(ISOURC)
      I      I      I      I      I      I      +---->(ITABLE)
      I      I      I      I      I      I      +---->(ABORT )
      I      I      I      I      I      I      +---->(RHOX )
      I      I      I      I      I      I      +---IRPOOo---->(NEXT )
      I      I      I      I      I      I      +---->(VALUE )
      I      I      I      I      I      I      +---->(ISOURC)
      I      I      I      I      I      I      +---IPFIRE---->(NEXT )
      I      I      I      I      I      I      I      +---->(VALUE )
      I      I      I      I      I      I      +---->(ITABLE)
      I      I      I      I      I      I      +---->(ABORT )
      I      I      I      I      I      I      +---->(RHOX )
      I      I      I      I      I      I      +---->(VALUE )
      I      I      I      I      I      I      +---IRATM ---->(RHOX )
      I      I      I      I      I      I      +---->(NEXT )
      I      I      I      I      I      I      +---->(ITABLE)
      I      I      I      I      I      I      +---->(ISOURC)
      I      I      I      I      I      I      +---->(ABORT )
      I      I      I      I      I      I      +---IANSQ ---->(VALUE )
      I      I      I      I      I      I      +---->(ABORT )
      I      I      I      I      I      I      +---->(NEXT )
      I      I      I      I      I      I      +---->(LOAD )
      I      I      I      I      I      I      +---SLINPT*---SLCHEM*
      I      I      I      I      I      I      +---CONCPT*---IMOVE
      I      I      I      I      I      I      +---SLCOORD*
      I      I      I      I      I      I      +---->(RHOX )
      I      I      I      I      I      I      +---->(TRAND*)
      I      I      I      I      I      I      +---->(RBOU )
      I      I      I      I      I      I      +---->(LOAD )
      I      I      I      I      I      I      +---REBPLT
      I      I      I      I      I      I      +---->(IENGSY)
      I      I      I      I      I      I      +---->(SAVCEL)
      I      I      I      I      I      I      +---->(ABORT )
      I      I      I      I      I      I      +---INITL
      I      I      I      I      I      I      +---RMOVE
      I      I      I      I      I      I      +---->(ABORT )
+---CONTRL----OUTPUT----EDITIN---->(TIMDAT)
      I      I      I      I      I      I      +---->(NXTCEL)
      I      I      I      I      I      I      +---->(LOAD )
      I      I      I      I      I      I      +---->(TIMDAT)
      I      I      I      I      I      I      +---->(SECOND)
      I      I      I      I      I      I      +---->(NXTCEL)

```

```

MAIN  -----CONTRL-----OUTPUT-----PRCELL----->(SETGAS)
      I          I          I      +--CPLOTo----->(LOAD  )
      I          I          I      +--REBPLT
      I          I          I      +---->(PRATM )
      I          I          I      +---->(RBOUo)
      I          I          I      +---->(IRTEMP)
      I          I          I      +--PRAERO
      I          I          I      +--HTMOUT
      I          I          I      +--PNSPRY
      I          I          I      +--PRENGS----->(LOAD  )
      I          I          I      +---->(UNLOAD)
      I          I      +--PRGLBL-----GLOT ----->(SECOND)
      I          I          I      +--USERO ----->(NXTCEL)
      I          I          I      +--PRFLOW
      I          I          I      +---->(PRDF  )
      I          I      +--RESOUo----->(UNLOAD)
      I          I          I      +---->(ABORT  )
      I          I          I      +---->(SECOND)
      I          I          I      +---->(NXTCEL)
      +---->(ABORT  )
      +--BTIME
      +--CHOZDT----->(SECOND)
      +--GLREST-----ZERO
      I          I      +--RMOVE
      +--CLCNTR----->(NXTCEL)
      I          I      +---->(SETGAS)
      I          I      +--PRCEDT----->(TSATU )
      I          I          I      +---->(PSATX )
      I          I          I      +---->(SECOND)
      I          I      +--CELLDT-----CELLCK
      I          I      +--CLREST-----ZERO
      I          I      +--RBCNTRo-----PHYDT
      I          I          I      +--LAYMOD*----->(LOAD  )
      I          I          I      +---->(UNLOAD)
      I          I          I      +---->(IRTEMP)
      I          I          I      +---->(IITEMP)
      I          I          I      +--FPHEAT-----ANSQ  -----FXPFIT
      I          I          I          I      +--INTERP
      I          I          I          I      +---->(IRTEMP)
      I          I          I          I      +---->(LOAD  )
      I          I          I      +--ZERO
      I          I          I      +--ATTQCV-----ATMPOL----->(PSATX )
      I          I          I          I      +---->(SHX  )
      I          I          I          I      +--CNVNS
      I          I          I          I      +---->(EVACON)
      I          I          I          I      +---->(ENTHX )
      I          I          I          I      +---->(CONDX )
      I          I          I          I      +---->(RHOX  )
      I          I          I          I      +--RADIAT
      I          I          I      +--BCTSET----->(LOAD  )
      I          I          I      +--HTSET  ----->(LOAD  )
      I          I          I          I      +---->(CONVX )
      I          I          I          I      +---->(SETCOM)
      I          I          I          I      +--ZERO
      I          I          I          I      +---->(ABORT  )
      I          I          I          I      +---->(UNLOAD)
      I          I          I          I      +---->(LOAD  )
      I          I          I      +--CONCRE----->(LAYPRO)
  
```

```

MAIN  -----CONTRL-----CLCENTR-----RBCNTRo-----CONCRE----->(LOAD )
      I          I          I          I          +---->(SORPL )
      I          I          I          I          +---TABLE
      I          I          I          I          +---->(RHOX )
      I          I          I          I          +---->(CONVX )
      I          I          I          I          +---DEBCON----->(RHOX )
      I          I          I          I          I          +---->(ENTHX )
      I          I          I          I          I          +---CHMLO ----->(ENTHX )
      I          I          I          I          I          +---->(ABORT )
      I          I          I          I          +---->(SLAM )
      I          I          I          I          +---->(LAYMS )
      I          I          I          I          +---CONABL-----POLY
      I          I          I          I          I          +---->(CONDX )
      I          I          I          I          I          +---->(RHOX )
      I          I          I          I          I          +---->(SHX )
      I          I          I          I          +---->(RBNLS )
      I          I          I          I          +---->(UNLOAD)
      I          I          +---CONCRM*----->(LOAD )
      I          I          I          I          +---->(TRANB*)
      I          I          I          I          +---TABLE
      I          I          I          I          +---->(TRANA*)
      I          I          I          I          +---STIME*
      I          I          I          I          +---->(SLAM*)
      I          I          I          I          +---TRANC*
      I          I          I          I          +---->(UNLOAD)
      I          I          +---INTERM----->(LAYPRO)
      I          I          I          I          +---->(SORPL )
      I          I          I          I          +---TABLE
      I          I          I          I          +---->(LAYMS )
      I          I          I          I          +---->(SHX )
      I          I          I          I          +---->(RBNLS )
      I          I          +---POOL ----->(LAYPRO)
      I          I          I          I          +---->(SORPL )
      I          I          I          I          +---TABLE
      I          I          I          I          +---PFIRE ----->(VISCX )
      I          I          I          I          +---->(LAYMS )
      I          I          I          I          +---->(RBNLS )
      I          I          +---ATMOSP----->(LOAD )
      I          I          I          I          +---->(SORPL )
      I          I          I          I          +---TABLE
      I          I          I          I          +---->(LAYMS )
      I          I          I          I          +---MEDBOL----->(ENTHX )
      I          I          I          I          +---->(RHOX )
      I          I          I          I          +---->(TPROP )
      I          I          I          I          +---MFLOW -----BOSOL ----->(HFP )
      I          I          I          I          I          +---->(TPROP )
      I          I          I          I          I          +---->(IRTEMP)
      I          I          I          I          I          +---->(KUTTA1)
      I          I          I          I          I          +---->(PSATX )
      I          I          I          I          I          +---->(RHOX )
      I          I          I          I          I          +---->(ENTHX )
      I          I          I          I          +---->(TPROP )
      I          I          I          I          +---NBOSOL----->(IRTEMP)
      I          I          I          I          +---->(KUTTA1)
      I          I          I          I          +---->(TPROP )
      I          I          +---CVTOAT----->(LOAD )
      I          +---CCNTRL-----SORATM----->(FLUX )
      I          I          I          +---->(ABORT )

```

```

MAIN  -----CNTRL-----CLCNTR-----CCNTRL-----SORATM----->(ENTHX )
      I           I           +---ENGCTL----->(LOAD )
      I           I           +---ZERO
      I           I           +--->(ENTHX )
      I           I           +--->(VISCX )
      I           I           +--->(RHOX )
      I           I           +---ENGORF-----EST
      I           I           +--->(IRTEMP)
      I           I           +---SORENG----->(FLUX )
      I           I           I           +--->(ABORT )
      I           I           I           +--->(ENTHX )
      I           I           +--->(TENEN )
      I           I           +---ENGSPR----->(ENTHX )
      I           I           I           +--->(RHOX )
      I           I           I           +--->(PSATX )
      I           I           I           +--->(TSATU )
      I           I           I           +---VELT
      I           I           I           +---ENGSPW----->(ENTHX )
      I           I           I           +--->(TENEN )
      I           I           I           +---SPREFF-----RHODD
      I           I           I           I           +---CHI
      I           I           I           I           +---FGAMMA
      I           I           I           +--->(VISCX )
      I           I           I           +---ENGHEX----->(SHX )
      I           I           I           +---ENGICE----->(CONDX )
      I           I           I           I           +--->(VISCX )
      I           I           I           I           +--->(SHX )
      I           I           I           I           +--->(PSATX )
      I           I           I           I           +---CNVNS
      I           I           I           I           +--->(EVACON)
      I           I           I           I           +--->(ENTHX )
      I           I           I           I           +--->(IRTEMP)
      I           I           I           I           +---DCONTH
      I           I           I           +---ENGFCL----->(SHX )
      I           I           I           I           +--->(PSATX )
      I           I           I           I           +--->(VISCX )
      I           I           I           I           +--->(CONDX )
      I           I           I           I           +--->(ENTHX )
      I           I           I           +--->(UNLOAD)
      I           I           +---FPSURF
      I           I           +---HTSURF-----HAS
      I           I           I           +--->(PSATX )
      I           I           I           +--->(RHOX )
      I           I           I           +--->(SHX )
      I           I           I           +---TABLE
      I           I           I           +---CNVNS
      I           I           I           +--->(CONDX )
      I           I           I           +--->(EVACON)
      I           I           I           +--->(ENTHX )
      I           I           +---CHENRE-----AERREA----->(IRTEMP)
      I           I           +---HBURN -----VFLAME
      I           I           I           +---UNBRNF
      I           I           I           +---RBURN
      I           I           I           +---FORCE ----->(PRATH )
      I           I           +---QXLOS ----->(IRTEMP)
      I           I           I           +---HSETUP----->(CONDX )
      I           I           I           I           +--->(SHX )
      I           I           I           I           +--->(DEBOUT)

```



```

MAIN  -----CONTRL-----CLCNTR-----CENTRL-----QXLOS  -----TRIDAG----->(DEBOUT)
      I          I          +-SPRAY ----->(SHX  )
      I          I          +---->(CONDX )
      I          I          +---->(VISCX )
      I          I          +-SORSR----->(FLUX  )
      I          I          +---->(RIIOX  )
      I          I          +-VELT
      I          I          +---->(ENTHX )
      I          +-SOR  -----SORAER----->(FLUX  )
      I          I          +---->(ABORT  )
      I          +---->(SAVCEL)
      +-GLCNTR-----ACNTRL----->(IRTEMP)
      I          +-AERSL -----RHODD
      I          I          +-COEF  -----GAUSBT----#FUN
      I          I          I          +-BF
      I          I          +---->(GAUS2 )
      I          I          +-GROWTH-----RHODD
      I          +-ADCOEF----->(TIMDAT)
      I          +---->(ENTHX )
      I          +---->(PSATX )
      I          +---->(MAGIC )
      I          +---->(IRTEMP)
      I          +---->(KUTTA1)
      I          +-CHI
      +-GSOURC----->(DEBOUT)
      +---->(DECAY )
      +-HMPWR
      +-FPNOVE-----FRELAC
      +-FLOW  -----FLOSET-----TABLE
      I          I          +---->(IITEMP)
      I          I          +-CLSTR
      I          +---->(IRTEMP)
      I          +---->(KUTTA1)
      I          +-RMOVE
      I          +-FLWAER----->(IRTEMP)
      I          I          +-RMOVE
      I          I          +-ZERO
      I          +-FLWFP -----RMOVE
      +-BALANC
  
```

GAUS2 ----#FUN
 +-@F

MAGIC ----FLGNOR

EVACON----->(TSATU)
 +---->(ENTHX)

ABORT ----DUMPAL----->(PTROUT)
 +--EREXIT

DEBOUT----->(SECOND)

LOAD ----CMOVE
 +--IMOVE
 +--LMOVE
 +--RMOVE

NXTCEL-----RDLCH
 +--SHFIN
 +---->(LOAD)

IRPERM-----ZERO
 +-->(ABORT)

IIPERM-----IZERO
 +-->(ABORT)

ILPERM-----LZERO
 +-->(ABORT)

ICPERM-----CZERO
 +-->(ABORT)

PTROUT-----COUT
 +--IOUT
 +--LOUT
 +--ROUT

SAVCEL----->(UNLOAD)
 +--RTLCH
 +--SHFOUT

SETCOM-----CZERO
+--IZERO
+--LZERO
+--ZERO

TINDAT-----TDCDC
+--TDCRA ----*CLOCK
I +--DATE
+--TDIBM ----YTIME ----*TIME
I +--NUMCHR
+--YDATE ----*DATE

SECOND-----*CLOCKM

IRTEMP----->(ABORT)

IITEMP----->(ABORT)

ICTEMP----->(ABORT)

UNLOAD-----CMOVE
+--IMOVE
+--LMOVE
+--RMOVE

DECAY -----RMOVE
+---->(IRTEMP)

IAERCL----->(NEXT)
+---->(CONVER)
+---->(ICTEMP)
+---->(IRTEMP)
+---->(VALUE)
+---->(MAGIC)
+---->(ISOURC)

IENGSY----->(UNLOAD)
+---->(SETCOM)
+---->(NEXT)
+---->(ABORT)
+---->(VALUE)
+---->(ISOURC)
+--IENTNK----->(VALUE)
+--IENPMP----->(VALUE)
+--IENORF----->(VALUE)
+--IENPIP----->(VALUE)
+--IENVAL----->(NEXT)
I +---->(VALUE)
I +---->(ABORT)

```

IENGSY-----IENHEX----->(NEXT )
      I      +---->(VALUE )
      +---IENSPR----->(NEXT )
      I      +---->(VALUE )
      I      +---->(ABORT )
      +---IENFCL----->(NEXT )
      I      +---->(VALUE )
      I      +---->(ABORT )
      +---IENICE----->(NEXT )
      I      +---->(VALUE )
      I      +---->(ABORT )
      +---IENSUP
      +---IENFLT
  
```

```

-----
I PROPS----->(NEXT )
      +---->(VALUE )
      +---->(ENTHX )
      +---->(ABORT )
  
```

```

-----
ISOURC----->(ABORT )
      +---->(VALUE )
      +---->(NEXT )
      +---->(CONVER)
  
```

```

-----
ITABLE----->(ABORT )
      +---->(NEXT )
      +---->(VALUE )
      +---->(ARRAY )
  
```

```

-----
RESTARo----->(LOAD )
      +---->(NXTCEL)
      +---->(TRANA*)
      +---SLPROP*
      +---SLCHEM*
      +---->(SAVCEL)
  
```

```

-----
PRATH ----->(TSATU )
      +---->(PSATX )
  
```

```

-----
PROF ----->(IITEMP)
  
```

```

-----
RBOUoT----->(IRTEMP)
      +---->(LOAD )
      +---SLOUT
  
```

```

-----
CONDXo----->(ABORT )
  
```

```

-----
CONVX ----->(ABORT )
      +---->(RIIOX )
  
```

CONVX ----->(CONDX)
+---->(SHX)
+---->(VISCX)

ENTHX^o ----->(SHX^o)
+---->(ABORT)

PRX ----->(ABORT)
+---->(CONDX)
+---->(VISCX)
+---->(SHX)

PSATX ----->(ABORT)

RHOX^o ----->(ABORT)

SETGAS----->(CONDX)
+---->(SHX)
+---->(VISCX)
+---->(ABORT)

SHX^o ----->(ABORT)

TEMEN ----->(ENTHX)

VISCX ----->(ABORT)

TSATU ----#ALNPV
+ -#ALNPT
+---->(PSATX)

ARRAY ----->(VALUE)

CONVER-----CPBYTE
+---->(NEXT)

IARRAY----->(VALUE)

NEXT -----SCAN

VALUE ----->(NEXT)
+---->(CONVER)
+---->(ABORT)

HFP ----->(PSATX)
+---->(ENTHX)
+---->(RHOX)

TPROP ----->(PSATX)
+---->(RHOX)
+---->(ENTHX)
+---->(SHX)

SLAM* ----CONEQS*----GAUSS*
I +--SLDIFC*
I +--SORGAS*
I +--MGRATE*
+--NATCON*----REACT*
I +--SLTRIS*
I +--HEAT*-----SLENTH*
+--WTRRLS*

TRANA* ----IMOVE
+--RMOVE

TRANB* --RMOVE

LAYNS ----->(SHX)
+---->(RHOX)

LAYPRO----->(CONDX)
+---->(RHOX)
+---->(SHX)

RBNLS ----@RBEQNS
+---->(DEBOUT)
+---->(ABORT)
+--SGEFS ----SGECO ----SASUM
I +--SGEFA ----ISANAX
I I +--SSCAL
I I +--SAXPY
I +--SSCAL
I +--SDOT
I +--SAXPY
+--SGESL ----SAXPY
I +--SDOT
+---->(ABORT)

SORPL ----->(FLUX)
+---->(ABORT)
+---->(ENTHX)

FLUX ----#TYM

COPYSF

DEPOST-----RHODD
+--CHI

DIFFUN---#CFUN

BETCAL-----BETA -----RHODD
I +--CHI
I +--FGAMMA
+---->(GAUS2)

ATMPAS

POLATH

FLODIR

FLODI2

FUNC1

FUNC2

ILTEMP----->(ABORT)

ENGPMP

ENGSPX-----CHI
+--FGAMMA
+---->(PRX)

EXEQNS----->(IRTEMP)
+--EXEQNX----->(DEBOUT)
+--THERMF----->(PSATX)
+---->(ENTHX)

IPCHEM----->(NEXT)
+---->(ABORT)
+---->(VALUE)

COND1

COND2

COND3 ----->(RHOX)
+---->(VISCX)
+---->(PRX)
+---->(CONDX)

HJC ----->(ENTHX)
+---->(RHOX)
+---->(VISCX)
+---->(PRX)
+---->(CONDX)
+---->(SHX)

ATMEQN

FP3

FP2 ----->(HFP)

CONEQN-----ZERO
+---->(DEBOUT)
+---->(ENTHX)

INTEQN----->(DEBOUT)
+---->(ENTHX)

POOEQN----->(DEBOUT)
+---->(ENTHX)

ERRPRT----->(ABORT)

ERSTGT----->(ABORT)

GAUS8 ----#G8
+-@FUN
+---->(ERRCHK)
+---->(ONECHK)

QNC7 ----->(ERRCHK)
+-@FUN
+---->(ONECHK)

8. Correction Set

The correction set to CONTAIN for incorporation of the sodium-concrete reaction module and the water release module is presented in the following pages.

This correction set contains all information, including modules to be incorporated in CONTAIN, comments for the modifications, and modifications becoming necessary as a result of incorporation of the modules.

```

*ID,SLAMCONV                                00000100
*/ *****                                00000200
*/ *                                       * 00000300
*/ * CORRECTION SETS FOR INCORPORATION SODIUM-CONCRETE INTERACTION * 00000400
*/ *      MODULE AND WATER-RELEASE MODULE INTO CONTAIN 1.03 * 00000500
*/ *                                       * 00000600
*/ * SLAMCONV CONSISTS OF THE FOLLOWING CORRECTION GROUPS * 00000700
*/ *                                       * 00000800
*/ *-----* 00000900
*/ ***      MODIFICATION OF ORIGINAL SLAM CODE *** * 00001000
*/ * SLOA : MODIFICATION OF ORIGINAL SLAM COMMON BLOCKS * 00001100
*/ * SLOB : MODIFICATION OF ORIGINAL SLAM SUBROUTINES * 00001200
*/ ***      ADDITION OF SUBROUTINES *** * 00001300
*/ * SLOC : NEWLY ESTABLISHED SUBROUTINES * 00001400
*/ *-----* 00001500
*/ *                                       * 00001600
*/ ***      THE FOLLOWING CORRECTION GROUPS ARE MISCELLANEOUS * 00001700
*/ *      MODIFICATIONS OF CONTAIN 1.03 *** * 00001800
*/ *                                       * 00001900
*/ ***      RELOCATION OF SUBROUTINES *** * 00002000
*/ * SLOD01 : CREATED SUBROUTINE 'LAYMOD' IS CALLED WHEN LINER * 00002100
*/ *      FAILURE OCCURS * 00002200
*/ * SLOD02 : CREATED SUBROUTINE 'CONCRN' IS CALLED IN CASE OF SLAM * 00002300
*/ * SLOD03 : REDUCE COMMENT STATEMENT AND SLAM DUMMY ROUTINE * 00002400
*/ *      AND RELOCATE SLAM ROUTINE * 00002500
*/ * SLOD04 : RELOCATION OF SLAM SUBROUTINE * 00002600
*/ ***      MODIFICATION OF AREA SIZE *** * 00002700
*/ * SLOE01 : ALLOCATE STORAGE ARRAY FOR SLAM CALCULATION * 00002800
*/ * SLOE02 : EXPAND DIMENSION OF REAL STORAGE ARRAY * 00002900
*/ * SLOE03 : REDUCE REGION WHICH IS NOT USED BY SLAM CALCULATION * 00003000
*/ ***      ADDITION OF COMMON VARIABLE *** * 00003100
*/ * SLOF01 : ADD NEW COMMON VARIABLES TO COMMON BLOCK * 00003200
*/ * SLOF02 : INITIAL SET OF SLAM COMMON VARIABLE * 00003300
*/ ***      ADDITION OF MATERIAL FOR SLAM CALCULATION *** * 00003400
*/ * SLOG01 : ADD MATERIAL PROPERTIES OF SLAM TO CONTAIN * 00003500
*/ *      FOR SLAM CALCULATION * 00003600
*/ * SLOG02 : MODIFICATION OF MASS FRACTION OF CONCRETE COMPONENTS * 00003700
*/ *      AND EXTENSION OF ARRAY DIMENSION ABOUT CONCRETE * 00003800
*/ ***      ADDITION OF FLAG FOR SLAM CALCULATION *** * 00003900
*/ * SLOH01 : ADD NEW FLAG 'SLMOUT' , 'WATOUT' TO COMMON BLOCK * 00004000
*/ * SLOH02 : CELL FLAG 'LDUMNC' IS 'TRUE' WHEN SLAM MODEL OR * 00004100
*/ *      WATER MIGRATION IS SELECTED * 00004200
*/ * SLOH03 : CELL FLAG 'SLMOUT' IS 'TRUE' WHEN SODIUM CONCRETE * 00004300
*/ *      INTERACTION OCCURS * 00004400
*/ * SLOH04 : CELL FLAG 'WATOUT' IS 'TRUE' AS WATER RELEASE OCCURS * 00004500
*/ * SLOI01 : LINER FAILURE TEMPERATURE 'TFAIL' IS SET TO DEFAULT * 00004600
*/ * SLOI02 : LINER FAILURE FLAG 'LNFAIL' IS 'TRUE' WHEN LINER FAILURE * 00004700
*/ *      OCCURS * 00004800
*/ ***      INITIAL SET OF 'PTAO(KZ)' AND 'RITAO(KZ)' * 00004900
*/ *      WHICH IS USED AT FIRST CELL TIME STEP IN SUBROUTINE 'CONCRN' * 00005000
*/ * SLOJ01 : ADD A ARGUMENT TO SUBROUTINE * 00005100
*/ * SLOJ02 : SAVE POOL AVERAGE TEMP FOR CURRENT TIME STEP * 00005200
*/ * SLOJ03 : SAVE INTERM AVERAGE TEMP FOR CURRENT TIME STEP * 00005300
*/ ***      ADDITION OF SLAM OUTPUT ROUTINE *** * 00005400
*/ * SLOK01 : ADD SLAM OUTPUT PRINT * 00005500
*/ * SLOK02 : WRITE SLAM CALCULATIONAL RESULTS TO PLOT FILE * 00005600
*/ * SLOK03 : READ CONCRETE STORAGE ARRAY FROM RESTART FILE * 00005700
*/ * SLOK04 : WRITE RESTART FILE FROM CONCRETE STORAGE ARRAY * 00005800
*/ *****                                00005900
*ID,CSLAM                                    00006000
*/ *****                                00006100
*/ * SLOA : MODIFICATION OF ORIGINAL SLAM COMMON BLOCKS * 00006200
*/ *****                                00006300
* I /REBLAY/.28                              00006400
*COMDECK /SLCOM1/                            00006500
COMMON /CNOD/ ISRC,ISRCP1,ISRCP3,ISRCP5,NDRY,NWET,NPOOL,NTERM, 00006600
1 NSLM,IMPL,IMPL2 00006700
COMMON /CPOINT/ LC1,LC2,LC3,LC4,LC5,LC6,LC7,LC8,LC9,LC10,LC11, 00006800
1 LC12,LC13,LC14,LC15,LC16,LC17,LC18,LC19,LC20, 00006900
2 LC21,LC22,LC23 00007000
INTEGER LPOINT(23) 00007100
EQUIVALENCE (LPOINT(1),LC1) 00007200
COMMON /CCORD/XZ1(51),WT2(51),XETA(51),ETA(51),WT(51),SINX(12,51), 00007300
2 COSX(12,51),SIGN(12),ZCOS1(12,12) 00007400
COMMON /SLCEL1/ SDT,DEL1,DELM,DDL1DT,DIST,DDL1DN,VEL,VELN,ROCP,TC, 00007500
1 TS,PS1,DTSDT,TDIF,COURNT,EPSM,EPSO,TAV,QCV,QCONO,QCON,TKWET, 00007600
2 QVSUM,QDEL1,H2OFLX,H2OSRC,WFLXDT,CO2SRC,XNTCO2,VGAS, 00007700
3 XMH2XT,H2FLXT 00007800

```

```

PARAMETER (JSLAMR = 32)
REAL RSLCEL (JSLAMR)
EQUIVALENCE (RSLCEL(1),SDT)
COMMON /SLCEL2/ PDRY(51),PWET(51),ROH2OL(51),VBUB(51),AM(12),
1 ALPHAL(51),RODRY(51,17),TDRY(51),TKDRY(51),TWET(51),
2 ROAIR(51),ALPH(51),RODRYI(51,17),RODRYO(51,17),
3 QV(52),QVP(51),SRCO2(51),SRH2O(51),EXH2(51)
*COMDECK /SLCOM2/
COMMON /CPRP/ W(17),RGAS(17),RHO(17,3),RHOL(17),ROMH2O,
1 SP(17,3),HF(17),HFG(17),TKMAT(17),TSTAR(17),
2 PSTAR(17),TCRIT(17),ZETA(17),HSTAR(17),PSAT,
3 TSAT(17),HFUS(17),TMLT(17),EPSRAD,GAMMA(17),
4 XMUL(17),IGAS(17),MIG
COMMON /CCHEM/ SAK(8),EK(8),AKF(8,3),COFR(8,17),COFP(8,17),
1 NREACT,IPHASR(8,17),IPHASP(8,17),MRR(8,3),
2 MPP(8,3),NRCT(8),NPROD(8)
*IDENT,SLAM
*/ *****
*/ * CORRECTION SET FOR INCORPORATION OF SODIUM-CONCRETE *
*/ * INTERACTION AND WATER-RELEASE MODULE INTO CONTAIN 1.03 *
*/ *****
*I,I-SETNA.166
*/ *****
*/ * SLOB : MODIFICATION OF ORIGINAL SLAM SUBROUTINES *
*/ *
*/ * 1. SLINPT 2. SLCHEM 3. SLCOORD 4. SLPROP *
*/ * 5. SLOUT 6. STIME 7. CONEQS 8. SORGAS *
*/ * 9. MGRATE 10. SLENTH 11. SLDIFC 12. SLTRIS *
*/ * 13. GAUSS 14. NATCON 15. REACT 16. HEAT *
*/ *****
*DECK,I-SLAM
C *****
*/ *****
*/ * SUBROUTINE SLINPT *
*/ *****
SUBROUTINE SLINPT
C
C SPECIES DEFINITIONS
C SLAM CONTAIN
C ----
C 1 - CONCRETE(SILICA)- 25
C 2 - WATER - 7,8,9
C 3 - SODIUM - 10,11,12
C 4 - HYDROGEN - 4
C 5 - NA OH - 21
C 6 - NA2 SID3 - 22
C 7 - NA2 CO3 - 47
C 8 - NA2O - 19
C 9 - CAO - 41
C 10 - CACO3 - 52
C 11 - CO2 - 6
C 12 - C - 26
C 13 - MGCO3 - 53
C 14 - MGO - 40
C 15 - INERT - 54
C 16 - STEEL - 50
C 17 - UO2 - 15
C
C PHASES 1-SOLID 2-LIQUID 3-GAS
C
C * READ INPUT; CALC INITIAL DATA AND PRINT INPUT IMAGE
C
*CALL,/IMPLIC/
*CALL,/SPACE/
*CALL,/PTRS/
*CALL,/MATER/
*CALL,/REBLAY/
*CALL,/REBCOM/
*CALL,/SLCOM1/
*CALL,/SLCOM2/
C
C READ CHEMICAL REACTION DATA
C
C CALL SLCHEM
C
C KREG = JCONC
C IMPL = 0
C IMPL2 = 0

```

```

ISRCP1 = 0
ISRC = IH(LRNNP+KREG)/4
NDRY = ISRC*2.3
NWET = IH(LRNNP+KREG)-NDRY+2
NTPERM = 12
NPDOL = 1
NSLM = 15
C
IF(NFCONC) ISRC=0
C
C
C DETERMINES POINTERS OF CONCRETE STORAGE ARRAY ICH( )
C
CALL CONCP1
C
C INITIAL SET OF CONCRETE COMMON VARIABLE
C
SDT = 1.0E -6
DEL1 = 0.003
DIST = 0.0
DBL1DT = 0.0
DBL1DN = 0.0
VEL = 0.0
VELN = 0.0
VGAS = 0.0
ROCP = RHOCON*CPCON
TS = 400.
PS1 = H(LPGAS+NCELL)
DTSDT = 0.0
TDIF = 0.0
COURNT = 0.0
EPSM = 0.16
EPS0 = RHOCON*FH2OE/ROMH2O
TAV = 400.
QCV = 0.0
QCON0 = 0.0
QCON = 0.0
QVSUM = 0.0
QDEL1 = 0.0
H2OFLX = 0.0
H2OSRC = 0.0
WFLXDT = DEL1*RHOCON*FH2OE
CQ2SRC = 0.0
XNTCO2 = 0.0
XNH2XT = 0.0
H2FLXT = 0.0
DO 100 I = 1,52
100 QV(I) = 0.0
NM1 = NDRY - 1
ISRCP1 = ISRC + 1
IDRY = ISRCP1
DO 200 I = 1,NM1
DO 200 J = 1,NSLM
RODRY(I,J) = 0.0
200 CONTINUE
C
C INITIAL SET OF DRY ZONE DENSITY
C
C
IF (NFCONC) THEN
TNATO = 0.
TNATN = 1.0E+30
ELSE
DO 300 I = 1,ISRC
DO 300 J = 1,NSLM
IF(J.EQ.3) RODRY(I,J)=RHOL(3)
300 CONTINUE
ENDIF
C
DO 400 I = IDRY,NDRY
RODRY(I,10) = IDRYCON * FCAC03
RODRY(I,13) = IDRYCON * FMGC03
RODRY(I,2) = 1.E-06
IF (I.GE.NDRY-3) RODRY(I,2)=RHOCON*FH2OB
RODRY(I,5) = 0.0
RODRY(I,1) = IDRYCON * FS102
RODRY(I,15) = IDRYCON * (1.-FCAC03-FMGC03-FS102)
400 CONTINUE

```

```

00015700
00015800
00015900
00016000
00016100
00016200
00016300
00016400
00016500
00016600
00016700
00016800
00016900
00017000
00017100
00017200
00017300
00017400
00017500
00017600
00017700
00017800
00017900
00018000
00018100
00018200
00018300
00018400
00018500
00018600
00018700
00018800
00018900
00019000
00019100
00019200
00019300
00019400
00019500
00019600
00019700
00019800
00019900
00020000
00020100
00020200
00020300
00020400
00020500
00020600
00020700
00020800
00020900
00021000
00021100
00021200
00021300
00021400
00021500
00021600
00021700
00021800
00021900
00022000
00022100
00022200
00022300
00022400
00022500
00022600
00022700
00022800
00022900
00023000
00023100
00023200
00023300
00023400

```

```

C          00023500
C  INITIAL SET OF SLAM COODINATE SYSTEM          00023600
C          00023700
C  CALL  SLCOORD                                00023800
C          00023900
C  KZ = IH(LRKZO+KREG)                          00024000
C  J = 3*(KZ-1) + 2                             00024100
C  TC = H(LRCT + J)                             00024200
C  IF( NFNACO ) THEN                            00024300
C    KPOOL = JCONC + JINT + 1                  00024400
C    KP = IH(LRKZO+KPOOL)                      00024500
C    K = 3*(KP-1)+1                            00024600
C    TPOOL = H(LRPT+K)                         00024700
C    WTEMP = TPOOL                             00024800
C  ELSE                                          00024900
C    KINT = JCONC + 1                          00025000
C    KI = IH(LRKZO+KINT)                      00025100
C    K = 3*(KI-1)+1                            00025200
C    TINT = H(LRIT+K)                         00025300
C    WTEMP = TINT                             00025400
C  ENDIF                                        00025500
C  DO 500 I = 1,NDRY                            00025600
C    TDRY(I) = WTEMP + (TS-WTEMP)*XZI(I)       00025700
C    TKDRY(I) = TKA*TDRY(I) + TKB            00025800
500 CONTINUE                                   00025900
C    TKWET = TKA * TC + TKB                  00026000
C          00026100
C  DO 700 I = 1,NDRY                            00026200
C    ALPHAI(I)=1.0                             00026300
C    QV(I) =0.0                                00026400
700 CONTINUE                                   00026500
C          00026600
C  THIS ROUTINE DETERMINES A CONCRETE DEPTH    00026700
C          00026800
C  DELM=0.0                                     00026900
C  KZO=IH(LRKZO+KREG)                          00027000
C  KZ1=KZO+IH(LRNNP+KREG)-1                   00027100
C  DO 710 KZ =KZO,KZ1                          00027200
C  DO 720 J=2,NM                               00027300
C    CMA = H(LRCMA+(KZ-1)*NM+J)              00027400
C    CMB = RHOX(J,H(LRCTA+KZ),H(LPGAS+NCELL))*H(LRCA+KZ) 00027500
C    DELM = DELM + CMA/CMB                   00027600
720 CONTINUE                                   00027700
710 CONTINUE                                   00027800
C          00027900
C  INITIAL SET OF TEMPERATURE ,PRESSURE ,MASS DENSITY IN WET ZONE 00028000
C          00028100
C          00028200
C  DO 750 I = 1,NWET                            00028300
C    TWET(I) = TC                             00028400
C    PWET(I)=PBOT                             00028500
C    ROAIR(I)=(PWET(I)/RGAS(2)*TC)*(EPSM-EPSO) 00028600
C    ROMH2OL(I)=ROMH2O*EPSO                 00028700
C    XETA(I)=XETA(I)*(DELM-DEL1)            00028800
C    ALPH(I)=EPSM - EPSO                    00028900
750 CONTINUE                                   00029000
C          00029100
C  INITIAL SET OF COODINATE VARIABLE          00029200
C          00029300
C  AM(1) = 0.0                                  00029400
C  SIGN(1) = -1.0                              00029500
C  DO 800 M = 2,NTERM                          00029600
C    AM(M) = 0.0                              00029700
C    SIGN(M) = -SIGN(M-1)                    00029800
800 CONTINUE                                   00029900
C  DO 900 N = 1,NTERM                          00030000
C  DO 900 M = 1,NTERM                          00030100
C    SUM=0.0                                  00030200
C  DO 880 I = 1,NDRY                            00030300
C    SUM=SUM+XZI(I)*COSX(M,I)*SINX(N,I)*WTZ(I) 00030400
880 CONTINUE                                   00030500
C    ZCOSI(N,M) = SUM                        00030600
900 CONTINUE                                   00030700
C          00030800
C  STORE CONCRETE COMMON BLOCK WITH THE CONCRETE STORAGE ARRAY CH(*) 00030900
C          00031000
C  CALL TRANB                                  00031100
C  RETURN                                      00031200

```

```

END 00031300
*/ *****00031400
*/ * SUBROUTINE SLCHEM *00031500
*/ *****00031600
SUBROUTINE SLCHEM 00031700
C 00031800
C THIS ROUTINES SETS CHEMICAL REACTION DATA 00031900
C 00032000
*CA,/IMPLIC/ 00032100
*CA,/SLCOM1/ 00032200
*CA,/SLCOM2/ 00032300
C 00032400
DATA NREACT/6/ 00032500
DATA (SAK(I),I=1,8)/99.0E+05,1.0E+05,8.85E+12,8.85E+12, 00032600
1 40.0E+05,40.0E+05,0.0,0.0/ 00032700
DATA (EK(I),I=1,8)/1.8E+03,1.8E+03,6.0E+04,6.0E+04, 00032800
1 3.4E+04,3.4E+04,0.0,0.0/ 00032900
DATA (AKF(I,1),I=1,8)/0.3,0.3,4*0.1,0.0,0.0/ 00033000
DATA (AKF(I,2),I=1,8)/1.,1.,2*0.1,1.,1.,0.,0./ 00033100
DATA (AKF(I,3),I=1,8)/1.0,1.0,2*0.6,1.0,1.0,0.,0./ 00033200
DATA (COFR(1,I),I=1,17)/0.,1.,1.,14*0./ 00033300
DATA (COFR(2,I),I=1,17)/0.,0.,4.,7*0.,1.,6*0./ 00033400
DATA (COFR(3,I),I=1,17)/0.,0.,4.,6*0.,3.,7*0./ 00033500
DATA (COFR(4,I),I=1,17)/0.,0.,4.,9*0.,3.,4*0./ 00033600
DATA (COFR(5,I),I=1,17)/4*0.,2.,4*0.,1.,7*0./ 00033700
DATA (COFR(6,I),I=1,17)/1.,3*0.,2.,12*0./ 00033800
C 00033900
DATA (COFP(1,I),I=1,17)/3*0,0.5,1.,12*0./ 00034000
DATA (COFP(2,I),I=1,17)/7*0,2.0,3*0,1.,5*0./ 00034100
DATA (COFP(3,I),I=1,17)/6*0,2.0,0.,3.,2*0.,1.,5*0./ 00034200
DATA (COFP(4,I),I=1,17)/6*0,2.0,4*0.,1.,0.,3.,3*0./ 00034300
DATA (COFP(5,I),I=1,17)/0.,1.,4*0.,1.,0.,1.,8*0./ 00034400
DATA (COFP(6,I),I=1,17)/0.,1.,3*0.,1.,11*0./ 00034500
DATA (IPHASR(1,I),I=1,17)/0,3,2,14*0/ 00034600
DATA (IPHASR(2,I),I=1,17)/0,0,2,7*0,3,6*0/ 00034700
DATA (IPHASR(3,I),I=1,17)/0,0,2,6*0,1,7*0/ 00034800
DATA (IPHASR(4,I),I=1,17)/0,0,2,9*0,1,4*0/ 00034900
DATA (IPHASR(5,I),I=1,17)/4*0,2,4*0,1,7*0/ 00035000
DATA (IPHASR(6,I),I=1,17)/1,3*0,2,12*0/ 00035100
DATA (IPHASP(1,I),I=1,17)/3*0,3,2,12*0/ 00035200
DATA (IPHASP(2,I),I=1,17)/7*0,1,3*0,1,5*0/ 00035300
DATA (IPHASP(3,I),I=1,17)/6*0,2,0,1,2*0,1,5*0/ 00035400
DATA (IPHASP(4,I),I=1,17)/6*0,2,4*0,1,0,1,3*0/ 00035500
DATA (IPHASP(5,I),I=1,17)/0,3,4*0,2,0,1,8*0/ 00035600
DATA (IPHASP(6,I),I=1,17)/0,3,3*0,1,11*0/ 00035700
DATA (NRCT(I),I=1,8)/ 6*2,0,0/ 00035800
DATA (NPRDD(I),I=1,8)/ 2,2,3*3,2,0,0/ 00035900
DATA (MRR(I,1),I=1,8)/ 2,11,10,13,10,5,0,0/ 00036000
DATA (MRR(I,2),I=1,8)/ 4*3,5,1,0,0/ 00036100
DATA (MRR(I,3),I=1,8)/ 8*0/ 00036200
DATA (MPP(I,1),I=1,8)/ 5,8,7,7,9,6,0,0/ 00036300
DATA (MPP(I,2),I=1,8)/ 4,12,9,14,2,2,0,0/ 00036400
DATA (MPP(I,3),I=1,8)/ 0,0,12,12,7,3*0/ 00036500
RETURN 00036600
END 00036700
C *****00036800
*/ *****00036900
*/ * SUBROUTINE SLCOOR *00037000
*/ *****00037100
SUBROUTINE SLCOOR 00037200
C 00037300
C INITIAL SET OF MATERIAL PROPERTIES OF CONCRETE 00037400
C 00037500
*CALL,/IMPLIC/ 00037600
*CALL,/SLCOM1/ 00037700
*CALL,/SLCOM2/ 00037800
*CALL,/CONST/ 00037900
C 00038000
PI=3.1415926535 00038100
NTERM=12 00038200
XNTH1=NDRY-1 00038300
ETA(1)=0. 00038400
SUM=0.0 00038500
FR=NDRY*0.2 00038600
TRMX=FR**6 00038700
DO 15 I=2,NDRY 00038800
TERH=((NDRY-I)+1.)*6 00038900
IF(TERM.LT.TRMX) TERM=TRMX 00039000

```

```

DETA=1./TERM                                00039100
ETA(I)=ETA(I-1)+DETA                        00039200
SUM=SUM+DETA                                00039300
15 CONTINUE                                  00039400
DO 20 I=1,NDRY                               00039500
ETA(I)=ETA(I)/SUM                            00039600
20 CONTINUE                                  00039700
C                                             00039800
C WATER MIGRATION COORDINATES                00039900
C                                             00040000
XETA(1)=0.                                   00040100
SUMX=0.0                                     00040200
DO 28 I=2,NWET                               00040300
TERM=(NWET-I+1.)                            00040400
DXETA=1./TERM                                00040500
XETA(I)=XETA(I-1)+DXETA                     00040600
SUMX=SUMX+DXETA                              00040700
28 CONTINUE                                  00040800
DO 29 I=1,NWET                               00040900
XETA(I)=XETA(I)/SUMX                         00041000
29 CONTINUE                                  00041100
DO 30 I=2,NDRY                               00041200
WT(I)=ETA(I)-ETA(I-1)                       00041300
30 CONTINUE                                  00041400
WT(1)=ETA(1)/2.                              00041500
WT(NDRY)=WT(NDRY)/2.                        00041600
C                                             00041700
C LINEAR FUNCTIONS AND WEIGHTING             00041800
C                                             00041900
DXZI=1./XNTM1                                00042000
WTZ(1)=0.3333333*DXZI                       00042100
XZI(1)=0.0                                   00042200
DO 40 I=2,NDRY                               00042300
XZI(I)=XZI(I-1)+DXZI                       00042400
WTZ(I)=1.3333333*DXZI                      00042500
IF((I/2)*2.NE.I) WTZ(I)=0.6666666*DXZI     00042600
DO 50 M=1,NTERM                              00042700
TERM=M*PI*XZI(I)                            00042800
SINX(M,I)=SIN(TERM)                         00042900
COSX(M,I)=COS(TERM)                         00043000
50 CONTINUE                                  00043100
DO 52 M=1,NTERM                              00043200
SINX(M,1)=0.0                               00043300
COSX(M,1)=1.0                               00043400
52 CONTINUE                                  00043500
40 CONTINUE                                  00043600
WTZ(NDRY)=WTZ(1)                            00043700
RETURN                                       00043800
END                                           00043900
C *****                                00044000
*/ *****                                00044100
*/ * SUBROUTINE SLPROP *                    *00044200
*/ *****                                00044300
SUBROUTINE SLPROP                            00044400
C                                             00044500
*CALL,/IMPLIC/                              00044600
*CALL,/SLCOM1/                              00044700
*CALL,/SLCOM2/                              00044800
*CALL,/REBLAY/                              00044900
C                                             00045000
C                                             00045100
C 0.0 ALL DATA                             00045200
C                                             00045300
DO 100 I=1,17                               00045400
W(I)=0.0                                     00045500
GAMMA(I)=0.0                                00045600
ZETA(I)=0.0                                 00045700
TCRIT(I)=0.0                                00045800
PSTAR(I)=0.0                                00045900
TSTAR(I)=0.0                                00046000
HSTAR(I)=0.0                                00046100
XMUL(I)=0.0                                 00046200
RHOL(I)=1200.                               00046300
RGAS(I)=8314.                               00046400
TMLT(I)=298.                                00046500
TSAT(I)=298.                                00046600
HF(I)=0.0                                    00046700
HFUS(I)=0.0                                 00046800

```

PNC SN952 85-30

	HFG(1)=0.0	00046900
	TKMAT(1)=.5	00047000
	IGAS(1)=0	00047100
	DO 200 J=1,3	00047200
	SP(I,J)=0.0	00047300
	IF(I,NE.2) RHO(1,J)=0.0	00047400
200	CONTINUE	00047500
100	CONTINUE	00047600
C		00047700
C	CONCRETE THERMAL CONDUCTIVITY FUNCTION TK=TKA*TEMP(KELVIN)+TKB	00047800
C		00047900
C		00048000
C	SPECIES 1 IS SI-O2 PHASE1 ONLY = SOLID	00048100
C		00048200
	W(1)=60.	00048300
	SP(1,1)=1100.	00048400
	RHO(1,1)=RHDCON	00048500
	HF(1)=-205.	00048600
	TMLT(1)=1743.	00048700
300	CONTINUE	00048800
C		00048900
C	SPECIES 2 IS WATER PHASE 1 IS LIQUID PHASE 2 IS GAS	00049000
C		00049100
	SP(2,2)=4184.	00049200
	HF(2)=-68.317	00049300
	SP(2,3)=1860.	00049400
	RHO(2,3)=1.013E5/(8314.*800./18)	00049500
	W(2)=18.	00049600
	GAMMA(2)=1.33	00049700
	ZETA(2)=0.33	00049800
	TCRIT(2)=647.	00049900
	PSTAR(2)=1.758E10	00050000
	TSTAR(2)=4500.	00050100
	HSTAR(2)=3.1E6	00050200
	TSAT(2)=373.	00050300
	HFG(2)=HSTAR(2)*(1.-373./TCRIT(2))*ZETA(2)	00050400
	XHUL(2)=2.3E-4	00050500
	RHOL(2)=600.	00050600
	ROMH2O=1000.	00050700
	TKMAT(2)=0.11	00050800
	RGAS(2)=8314./18.	00050900
	IGAS(2)=2	00051000
C		00051100
C	SPECIES 3 IS SODIUM PROPERTIES: RATIO OF SPEC HEATS, DIETERICI	00051200
C	EQN OF STATE, PRESS FUNCTION=RT/(V-B),TEMP FUNCTION=A/(VR),CRITIC	00051300
C	TEMP, SAT TEMP, HEAT OF VAPORIZATION, CONDUCTIVITY	00051400
C		00051500
	SP(3,2)=1296.	00051600
	SP(3,1)=SP(3,2)	00051700
	HFUS(3)=1.14E05	00051800
	RHO(3,2)=800.	00051900
	RHOL(3)=800.	00052000
	TMLT(3)=370.9	00052100
	HF(3)=-.575	00052200
	SP(3,3)=4481.*4184/(23*902)	00052300
	RHO(3,3)=1.013E5/(8314*1000./23.)	00052400
	W(3)=23.	00052500
	GAMMA(3)=1.666	00052600
	ZETA(3)=0.341	00052700
	HSTAR(3)=4.81E6	00052800
	PSTAR(3)=3.27E9	00052900
	TSTAR(3)=1.202E4	00053000
	TCRIT(3)=2509.	00053100
C	TSAT(3)=-TSTAR(3)/ALOG(P1NFXT/(PSTAR(3)*XMOLNA))	00053200
C	HFG(3)=HSTAR(3)*(1.-TSAT(3)/TCRIT(3))*ZETA(3)	00053300
	TKMAT(3)=50.	00053400
C		00053500
C	SPECIES 4 IS HYDROGEN GAS 1 PHASE ONLY: SPECIFIC HEAT,	00053600
C	DENSITY, MOLECULAR WEIGHT, CONDUCTIVITY	00053700
C		00053800
	SP(4,3)=(6404.*4184.)/(902.*2.)	00053900
	RHO(4,3)=1.013E5/(8314.*1000/2.)	00054000
	W(4)=2.	00054100
	TKMAT(4)=0.05	00054200
	RHOL(4)=3000.	00054300
	IGAS(4)=4	00054400
C		00054500
C	SPECIES 5 IS NAOH	00054600

C	W(5)=40.	00054700
	RHOL(5)=850.	00054800
	SP(5,2)=3614.	00054900
	HF(5)=-100.0	00055000
	SP(5,1)=3827.	00055100
	HFUS(5)=2.76E5	00055200
	TMLT(5)=595.	00055300
	PSTAR(5)=4.002E9	00055400
	TSTAR(5)=1.748E4	00055500
C		00055600
C		00055700
C	SPECIES 6 IS NA2 SIO3 SOLID PHASE ONLY	00055800
C		00055900
C		00056000
	W(6)=122.	00056100
	SP(6,1)=1485.	00056200
	HF(6)=-372.	00056300
	HFUS(6)=3.53E5	00056400
	TMLT(6)=1360.	00056500
		00056600
		00056700
C		00056800
C	SPECIES 7 NA2CO3	00056900
C		00057000
	W(7)=106.	00057100
	HF(7)=-270.	00057200
	SP(7,2)=1776.	00057300
	SP(7,1)=1578.	00057400
	HFUS(7)=2.79E5	00057500
	TMLT(7)=1123.	00057600
		00057700
C		00057800
C	SPECIES 8 NA2O	00057900
C		00058000
	W(8)=62.	00058100
	HF(8)=-99.9	00058200
	SP(8,2)=1635.	00058300
	SP(8,1)=1649.	00058400
	HFUS(8)=7.57E5	00058500
	TMLT(8)=1405.	00058600
		00058700
C		00058800
C	SPECIES 9 CAO	00058900
C		00059000
	W(9)= 56.	00059100
	HF(9)=-151.6	00059200
	SP(9,1)=877.	00059300
	TMLT(9)=2888.	00059400
	RHO(9,1)=1540.	00059500
		00059600
C		00059700
C	SPECIES 10 CaCO3	00059800
C		00059900
	W(10)=100.	00060000
	HF(10)=-288.4	00060100
	SP(10,1)=1173.	00060200
	TMLT(10)=3000.	00060300
	RHO(10,1)=2200.	00060400
	HFUS(11)=2.E05	00060500
		00060600
C		00060700
C	SPECIES 11 CO2	00060800
C		00060900
	W(11)=44.	00061000
	HF(11)=-94.	00061100
	RHOL(11)=3000.	00061200
	TKMAT(11)=0.075	00061300
	SP(11,3)=1378.	00061400
	RHO(11,1)=RHO(10,1)-RHO(9,1)	00061500
	IGAS(11)=11	00061600
		00061700
C		00061800
C	SPECIES 12 C - GRAPHITE	00061900
C		00062000
	W(12)=12.	00062100
	HF(12)=0.	00062200
	SP(12,1)=1243.	00062300
		00062400
C		
C	MGC03	
C		
	W(13)=84.3	
	HF(13)=-267.6	
	SP(13,1)=1173.	
	RHOL(13)=1400.	

```

C                                     00062500
C      H60                             00062600
C                                     00062700
      W(14)=40.3                       00062800
      HF(14)=-143.8                    00062900
      SP(14,1)=877.                     00063000
      RHOL(14)=1400.                   00063100
C                                     00063200
C      INERTS                           00063300
C                                     00063400
      W(15)=100.                        00063500
C                                     00063600
C      SPECIES 16 STEEL                 00063700
C                                     00063800
      SP(16,2)=750.                     00063900
      SP(16,1)=639.                     00064000
      HFUS(16)=2.7E5                    00064100
      RHO(16,2)=6979.                   00064200
      RHOL(16)=6979.                    00064300
      TMLT(16)=1700.                   00064400
      HF(16)=-.575                      00064500
      SP(16,3)=492.                     00064600
      RHO(16,3)=1.0116E5/(81614*3000./56.) 00064700
      W(16)=56.                          00064800
      GAMMA(16)=1.26                    00064900
      ZETA(16)=0.36                     00065000
      HSTAR(16)=8.17E6                  00065100
      PSTAR(16)=1.33E11                 00065200
      TSTAR(16)=4.337E4                 00065300
      TCRIT(16)=1E4                     00065400
C      TSAT(16)=-TSTAR(16)/ALOG(H(LPGAS+NCELL)/PSTAR(16)) 00065500
      HFG(16)=HSTAR(16)*(1.-TSAT(16)/TCRIT(16))*ZETA(16) 00065600
      TKMAT(16)=20.                      00065700
      XMUL(16)=5.36E-3                  00065800
C                                     00065900
C      SPECIES 17 UO2                   00066000
C                                     00066100
      SP(17,2)=504.                     00066200
      SP(17,1)=638.                     00066300
      HFUS(17)=2.08E5                    00066400
      RHO(17,2)=8699.                   00066500
      RHOL(17)=8699.                    00066600
      TMLT(17)=3100.                    00066700
      HF(17)=-272.                       00066800
      SP(17,3)=511.                      00066900
      RHO(17,3)=1.0116E5/(81614*3000./270.) 00067000
      W(17)=270.                          00067100
      GAMMA(17)=1.05                     00067200
      ZETA(17)=.597                      00067300
      HSTAR(17)=2.62E6                   00067400
      PSTAR(17)=1.44E11                  00067500
      TSTAR(17)=5.17E4                   00067600
      TCRIT(17)=8.4E3                    00067700
C      TSAT(17)=-TSTAR(17)/ALOG(H(LPGAS+NCELL)/PSTAR(17)) 00067800
      HFG(17)=HSTAR(17)*(1.-TSAT(17)/TCRIT(17))*ZETA(17) 00067900
      TKMAT(17)=5.0                      00068000
      XMUL(17)=4.3E-3                    00068100
C                                     00068200
C                                     00068300
C      CONVERT HEAT OF FORMATION TO JOULES/SI-HOLE 00068400
C                                     00068500
C                                     00068600
      DO 270 I=1,17                       00068700
      HF(I)=HF(I)*418400.                 00068800
      IF(W(I).NE.0.0) RGAS(I)=8314./W(I) 00068900
270  CONTINUE                             00069000
      RETURN                               00069100
      END                                  00069200
C *****                                00069300
* I,O-RESOUT.147                          00069400
* DECK,D-SLAM                             00069500
*/ *****                                00069600
*/ *      SUBROUTINE SLOUT                  *00069700
*/ *****                                00069800
      SUBROUTINE SLOUT                    00069900
C                                     00070000
C *** PRINTS THE CALCULATIONAL SUMMARY 00070100
C                                     00070200

```

```

C
*CALL,/IMPLIC/                                00070300
*CALL,/FLAGS/                                  00070400
*CALL,/SLCOM1/                                 00070500
*CALL,/SLCOM1/                                 00070600
*CALL,/SLCOM2/                                 00070700
*CALL,/REBLAY/                                 00070800
*CALL,/TIME/                                   00070900
CHARACTER*51 WORDA(18)                        00071000
C
DATA WORDA/'DUMMY',                            00071100
*'AVERAGE DRY ZONE TEMPERATURE, TDRY (K) = ', 00071200
*'WET-DRY INTERFACE TEMPERATURE, TS (K) = ', 00071300
*'CONCRETE REACTION HEAT, QCV (W/M**2) = ', 00071400
*'CONCRETE SURFACE HEAT FLUX, QCONO (W/M**2) = ', 00071500
*'HEAT FLUX INTO WET ZONE, QCON (W/M**2) = ', 00071600
*'PENETRATION DEPTH, DIST (M) = ', 00071700
*'ABLATION VELOCITY, VEL (M/S) = ', 00071800
*'DRY ZONE DEPTH, DEL1 (M) = ', 00071900
*'DRY ZONE GROWTH RATE, DDL1DT (M/S) = ', 00072000
*'DRY ZONE HEAT SOURCES, QVSUM (W/M**2) = ', 00072100
*'WET-DRY INTERFACE HEAT FLUX, QDEL1 (W/M**2) = ', 00072200
*'WET-DRY INTERFACE H2O PRESS., PS1 (PA) = ', 00072300
*'INTERFACE H2O EVAP. FLUX, H2OFLX (KG/M**2-S)= ', 00072400
*'DRY ZONE BOUND H2O FLUX, H2OSRC (KG/M**2-S)= ', 00072500
*'INTEGRAL OF BOUND + EVAP H2O, WFLXDT (KG/M**2) = ', 00072600
*'DRY ZONE BOUND CO2 FLUX, CO2SRC (KG/M**2-S)= ', 00072700
*'INTEGRAL OF BOUND CO2, XNTCO2 (KG/M**2) = '/ 00072800
C
TDR1=TDRY*(NDRY/2+1)                          00072900
IF (SLMOUT) WRITE (6,80) PTPDT,PDT            00073000
IF (WATOUT) WRITE (6,85) PTPDT,PDT            00073100
80  FORMAT(/// SLAM SODIUM-CONCRETE INTERACTION:',20X,'PTPDT = ', 00073200
*'G13.6,16X,'TIMESTEP = ',G13.6//)           00073300
85  FORMAT(/// WATER MIGRATION CALCULATION ',20X,'PTPDT = ', 00073400
*'G13.6,16X,'TIMESTEP = ',G13.6//)           00073500
WRITE (6,90) WORDA(2), TDR1, WORDA(3),TS      00073600
WRITE (6,90) WORDA(4), QCV, WORDA(5), QCONO   00073700
IF (SLMOUT) WRITE (6,90) WORDA(6), QCON, WORDA(7), DIST 00073800
IF (SLMOUT) WRITE (6,90) WORDA(8), VEL, WORDA(9), DEL1   00073900
IF (SLMOUT) WRITE (6,90) WORDA(10), DDL1DT,WORDA(11), QVSUM 00074000
WRITE (6,90) WORDA(12), QDEL1,WORDA(13), PS1 00074100
IF (WATOUT) WRITE (6,90) WORDA(9), DEL1,WORDA(10), DDL1DT 00074200
WRITE (6,90) WORDA(14), H2OFLX,WORDA(15),H2OSRC 00074300
WRITE (6,90) WORDA(16), WFLXDT,WORDA(17),CO2SRC 00074400
WRITE (6,90) WORDA(18),XNTCO2                 00074500
90  FORMAT(1X,A51,G12.5,5X,A51,G12.5)          00074600
RETURN                                         00074700
END                                             00074800
C *****                                     00074900
*I,U-TRIDAG.62                               00075000
*DECK,RB-SLAM                                00075100
*/ *****                                     00075200
*/ * SUBROUTINE STIME                          *00075300
*/ *****                                     00075400
SUBROUTINE STIME (SLTIME,WDT,ISLAF,TR,TSN,WDTDT) 00075500
C
C ** THIS ROUTINE DETERMINES TIME STEP FOR SODIUM CONCRETE INTERACTION 00075600
C
*CALL,/IMPLIC/                                00075700
*CALL,/TIME/                                  00075800
*CALL,/NACON/                                 00075900
*CALL,/SLCOM1/                                 00076000
*CALL,/SLCOM2/                                 00076100
*CALL,/REBLAY/                                 00076200
C
C *-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-* 00076300
C TIME STEP POINT                             00076400
C I SAVE SYSTEM TIME STEP DT13 BEFORE ENTERING LOOP 00076500
DETEX=PDT                                     00076600
C I GET 2ND FROM LAST VALUE OF SDT TO CALC DT1 IN NEXT LOOP 00076700
100 CONTINUE                                  00076800
DT1=SDT*1.2                                  00076900
IF (SLTIME.LT.TNATO+1.) DT1=SDT*2.          00077000
DT2=1.E3                                     00077100
DT3=1.E3                                     00077200
DT4=1.E3                                     00077300
DT5=1.E3                                     00077400
DT6=1.E3                                     00077500

```

```

DT7=1.E3                                00078100
DT8=1.E3                                00078200
DT9=1.E3                                00078300
DT10=1.E3                               00078400
DT11=1.E03                              00078500
DT12=1.E03                              00078600
IF(DDL1DT.NE.0.0) DT2=(ABS(1.E-2*DEL1/DDL1DT)) 00078700
IF(DDL1DT.NE.0.0) DT3=(0.5*TDIF/DDL1DT**2)    00078800
IF(VEL.NE.0.0) DT4=(ABS(0.15*DEL1/(VEL*NDRY))) 00078900
DTPA=AMAX1(ABS(TR-TS),1.,0.01*ABS(TS-TSN))    00079000
IF(DTSDT.NE.0.0) DT5=(ABS(DTPA/DTSDT))        00079100
IF(WDTDT.NE.0.0) DT6=(ABS(DTPA/WDTDT))        00079200
IF(COURNT.GT.0.5) DT7=(SDT/COURNT*0.5)       00079300
DTIME=2.+ABS(SLTIME-TNATO)                   00079400
DTMAX=3.*ALOG(DTIME)                         00079500
DT8=(DTMAX)                                  00079600
IF(IMPL2.NE.0) DT9=SDT*(20./FLOAT(IMPL2))    00079700
IF(IMPL.NE.0) DT10=SDT*(20./FLOAT(IMPL))     00079800
TERM=QVSUM+QCONO-QDEL1+VEL*ROCP*(TR-TS)     00079900
DTDT=TERM/(RHOCON*DEL1*CPCON)                00080000
IF(DTDT.GT.0.0) DT12=100./DTDT               00080100
DTINF=SDT                                     00080200
C      I SAVE 2ND FROM LAST VALUE OF DT TO CALC DT1 IN NEXT LOOP 00080300
C      THE LAST VALUE OF DT IN THE LOOP WILL USUALLY BE SMALLER 00080400
C      I DT CANNOT BE GEATER THAN REMAINDER OF SYSTEM TIME STEP 00080500
SDT=AMIN1(DT1,DT2,DT3,DT4,DT5,DT6,DT7,DT8,DT9,DT10,DT11,DT12 00080600
*,DETEX)                                     00080700
C      I REMAINDER OF ORIG SYSTEM TIME STEP 00080800
DETEX=DETEX-SDT                              00080900
IF(SDT.EQ.DT1) ISTEP=1                       00081000
IF(SDT.EQ.DT2) ISTEP=2                       00081100
IF(SDT.EQ.DT3) ISTEP=3                       00081200
IF(SDT.EQ.DT4) ISTEP=4                       00081300
IF(SDT.EQ.DT5) ISTEP=5                       00081400
IF(SDT.EQ.DT6) ISTEP=6                       00081500
IF(SDT.EQ.DT7) ISTEP=7                       00081600
IF(SDT.EQ.DT8) ISTEP=8                       00081700
IF(SDT.EQ.DT9) ISTEP=9                       00081800
IF(SDT.EQ.DT10) ISTEP=10                     00081900
IF(SDT.EQ.DT11) ISTEP=11                     00082000
IF(SDT.EQ.DT12) ISTEP=12                     00082100
IF(SDT.EQ.DETEX) ISTEP=13                    00082200
IF(SDT.LT.1.E-6) PRINT 9823, SDT,ISTEP      00082300
C                                             00082400
9823 FORMAT(' *****/' SMALL TIME STEP IN SLAM ',E10.3, 00082500
*, ISTEP= ',I4,/' *****')                00082600
C                                             00082700
C      COMPARISON SLAM TIME STEP WITH CONTAIN TIME STEP 00082800
C                                             00082900
PDTX = PDT - 0.0000001                       00083000
IF (SDT .GT. PDTX) THEN                       00083100
ISLAF = 1                                     00083200
SDT = PDT                                     00083300
RETURN                                        00083400
ELSE                                           00083500
WDTS = WDT - SDT                             00083600
ENDIF                                         00083700
IF (WDTS.LE.-0.000001) THEN                   00083800
SDT = WDT                                     00083900
ISLAF = 1                                     00084000
ELSE                                           00084100
IF (WDTS.LE.SDT*0.25) THEN                   00084200
SDT = WDT                                     00084300
ISLAF = 1                                     00084400
ELSE                                           00084500
WDT = WDTS                                    00084600
ENDIF                                         00084700
ENDIF                                         00084800
SLTIME = SLTIME + SDT                        00084900
C                                             00085000
C                                             00085100
C      NSTEPO=NSTEP                             00085200
C      NSTEP=NSTEP+1                             00085300
RETURN                                        00085400
END                                            00085500
C                                             00085600
C                                             00085700
*/ *****00085800

```

```

*/ *          SUBROUTINE  CONEQS                      *00085900
*/ *****                                *****00086000
SUBROUTINE CONEQS(TIME,DT,WTEMP,PINFXT,TJNFXT,WDTDT,DEL10,TRO,
1 H2OFLO,EPP,ERR1,COMP1,TOK,SUMTRM,FX,AMAT,CCOL,XVAR,AMD,SINT,
2 VELOLD)
C
*CALL,/IMPLIC/
*CALL,/SLCOM1/
*CALL,/SLCOM2/
*CALL,/NACON/
*CALL,/REBLAY/
*CALL,/CONST/
C
DIMENSION COMP1(51),TOK(51),SUMTRM(51),FX(51),AMAT(12,12)
1          ,CCOL(12),XVAR(22),AMO(12),SINT(51)
C *****
C * THIS SECTION CALCULATES THE WATER RELEASE AND HEAT TRANSFER *
C * IN THE CONCRETE *
C *****
C ITERATION POINT ON THE END OF TIME STEP VALUES
C
MIG = 0
TR = WTEMP
IMPL=IMPL+1
C I DRY REG H2O HEAT OF EVAP
HFG(2)=HSTAR(2)*(1.-TS/TCRIT(2))*ZETA(2)
C
AVE THERMAL COND IN DRY CONC
C
TAVE=0.5*(TDRY(ISRCP5)+TDRY(NDRY))
C I CONDUCTIVITY IN DRY REGION
TK=TKA*TAVE+TKB
IF(TK.LT.0.1) TK=0.05
DO 545 I=1,ISRCP1,ISRCP5
TKDRY(I)=ANAX1(0.05,TKA*TDRY(I)+TKB)
545 CONTINUE
TDIF=TK/(ROCP)
C
DO THE INTEGRATIONS
C
DO 110 N=1,NTERM
DO 120 M=1,NTERM
SUM1=0.0
SUM2=0.0
TERM2=- (M*PI)*DDL1DT/(DEL1)
TERM6=VEL*M*PI/DEL1
IF(M.NE.N .AND. ((M+N)/2)*2 .NE. (M+N) )
* SUM2=(2.*N/((M**2-M**2)*PI))
IF(N.EQ.M) SUM1=0.5
SUM8=SUM1*(M*M*PI*PI/DEL1**2)*TDIF
SUM6=SUM2*TERM6
ZCOSIF=ZCOSI(N,M)*TERM2
TRM=XZ1(2)*M*M*PI*PI/(DEL1*DEL1*ROCP)
TRM2=-M*PI/(DEL1**2*ROCP)
WTT=1.
DO 119 I=2,ISRCP5
IF(I.EQ.ISRCP5) WTT=0.5
TKAVE=0.5*(TKDRY(I-1)+TKDRY(I))-TK
SUM8=SUM8+(TKAVE)*(SINX(N,I)*SINX(N,I))*TRM*WTT
* +(TKDRY(I)-TKDRY(I-1))*(COSX(M,I)*SINX(N,I))*TRM2*WTT
119 CONTINUE
AMAT(N,M)=SUM1+DT*(ZCOSIF+SUM6+SUM8)
120 CONTINUE
SUM3=- (SIGN(N))/(N*PI)
SUM4=- (SIGN(N)-1.)/(N*PI)
TDDL=SUM3*(TR-TS)*DDL1DT/(DEL1)
SUM3=SUM3*(DTSDT-WDTDT)
SUM5=WDTDT*SUM4
SUM7=VEL*(TS-TR)/DEL1*SUM4
SUM9=0.0
TRM=(TS-TR)/(ROCP*DEL1**2)
WTT=1.
DO 112 I=2,NDRY
IF(I.EQ.ISRCP5) WTT=0.5
SUM9=SUM9+(QV(I)*SINX(N,I)+QV(I-1)*SINX(N,I-1))*XZ1(2)*0.5
IF(I.LE.ISRCP5)
* SUM9=SUM9+(TKDRY(I)-TKDRY(I-1))*SINX(N,I)*TRM*WTT

```

```

112 CONTINUE                                00093700
    CCOL(N)=- (SUM3+TDDL+SUM5+SUM7-SUM9)*DT+AM(N)*0.5 00093800
110 CONTINUE                                00093900
C                                            00094000
C      INVERT MATRIX AND CALC SOLUTION          00094100
C                                            00094200
C      CALL GAUSS (AMAT,AM,CCOL,NTERM,12)      00094300
C                                            00094400
C      CALC HEAT FLUX AND ENERGY BALANCE     00094500
C                                            00094600
C      SUM=0.0                                 00094700
C      SUMDEL=0.0                              00094800
C      DO 160 M=1,NTERM                        00094900
C      XM=M                                    00095000
C      SUM=SUM+XM*AM(M)                        00095100
C      SUMDEL=SUMDEL+SIGN(M)*XM*AM(M)         00095200
160 CONTINUE                                00095300
C      I HEAT FLUX AT CONC SURFACE            00095400
C      QCONO=-TKDRY(1)*(SUM*PI+(TS-TR))/DEL1  00095500
C      QDEL1=-TK*(TS-TR)+PI*SUNDEL)/DEL1     00095600
C      QDEL1=-TK*(TDRY(NDRY)-TDRY(NDRY-1))/(XZI(2)*DEL1) 00095700
C      SUMDT1=0.0                              00095800
C      SUMDT2=0.0                              00095900
C      DO 161 M=1,NTERM                        00096000
C      SUMDT1=SUMDT1+(AM(M)-AM(N))/DT*(1./XM-SIGN(M)/XM) 00096100
C      SUMDT2=SUMDT2-AM(M)*(SIGN(M)-1.)/XM   00096200
161 CONTINUE                                00096300
C      TERM1=ROCP*DEL1*SUNDT1/PI              00096400
C      TERM2=ROCP*DDL1DT*SUNDT2/PI           00096500
C      TERM3=ROCP*(TR-TS)*0.5*DDL1DT         00096600
C      TERM4=ROCP*WDTDT*DEL1                 00096700
C      TERM5=ROCP*(DTS-DT)*DEL1*0.5         00096800
C      ENBAL1=TERM1+TERM2+TERM3+TERM4+TERM5  00096900
C      ENBAL2=ROCP*VEL*(TS-TR)               00097000
C      ENBAL3=QCONO-QDEL1                     00097100
C      ENBAL4=QVSUM                           00097200
C      XVAR(9)=ENBAL1                          00097300
C      XVAR(10)=ENBAL2                         00097400
C      XVAR(11)=ENBAL3                         00097500
C      XVAR(12)=ENBAL4                         00097600
C      IF(ENBAL3.EQ.0.0) GO TO 440             00097700
C      IF(ABS(ENBAL4/ENBAL3).LT.1) ENBAL=(ENBAL1+ENBAL2-ENBAL4)/ENBAL3 00097800
C      IF(ABS(ENBAL4/ENBAL3).GT.1.) ENBAL=(ENBAL1+ENBAL2-ENBAL3)/ENBAL4 00097900
C      I HEAT FLUX INTO WET REGION            00098000
440 QCON=-TKWET*2.*(TWET(1)-TS)/XETA(2)     00098100
C                                            00098200
C                                            00098300
C      DARCY EQN NUMERICAL INTEGRALS          00098400
C                                            00098500
C      AP=0.3949E-18                          00098600
C      BP=0.01416                             00098700
C      TAVE=(TR+TS)/2.                         00098800
C      CALL SLDIFFC(TAVE,PINFXT,DIFF,2,4,XMU)  00098900
C      COMP1(1)=0.0                            00099000
C      DO 85 I=1,NDRY                          00099100
C      SUM1=0.0                                00099200
C      DO 86 M=1,NTERM                          00099300
C      SUM1=SUM1+AM(M)*SINX(M,I)              00099400
86 CONTINUE                                00099500
C      TDRY(I)=TR+(TS-TR)*XZI(I)+SUM1        00099600
C      IF(TDRY(I).LE.0.0) WRITE(6,9445) I,VEL,TDRY(I),QV(I) 00099700
9445 FORMAT(' I,VEL,TDRY(I),QV(I)= ',I6,4E12.5) 00099800
C      PRMDRY=AP*EXP(BP*TDRY(I))              00099900
C      TOK(I)=TDRY(I)/PRMDRY                  00100000
C      COMP1(I+1)=COMP1(I)+TOK(I)*WTZ(I)     00100100
85 CONTINUE                                00100200
C                                            00100300
C      GAS SOURCE INTEGRALS                    00100400
C                                            00100500
C      CALL SORGAS(TIME,DT)                    00100600
C                                            00100700
C      SINT(1)=0.0                             00100800
C      DO 285 I=2,NDRY                         00100900
C      SRCGAS=0.5*(SRH20(I)+SRCO2(I)+SRH20(I-1)+SRCO2(I-1)) 00101000
C      SINT(I)=SINT(I-1)+SRCGAS*(XZI(I)-XZI(I-1)) 00101100
285 CONTINUE                                00101200
C      SUMTRM(1)=0.0                          00101300
C      DO 280 I=2,NDRY                         00101400

```

```

SUMTRM(I)=SUMTRM(I-1)+TOK(I)*(SINT(NDRY)-SINT(I))*WTZ(I) 00101500
280 CONTINUE 00101600
C 00101700
C SOLVE FOR TS AND PS1 SIMULTANEOUSLY BY NEWT RAPH 00101800
C 00101900
TRMO=2.*XMU*RGAS(2)*DEL1 00102000
TRM1=TRMO*COMP1(NDRY) 00102100
TRM2=TRMO*DEL1*SUMTRM(NDRY) 00102200
C I USES H2O HEAT OF VAPORIZATION 00102300
TRM3=2.*TKWET/(XETA(2)*HFG(2)) 00102400
ITNR=0 00102500
TSN=TS 00102600
ERM=1.E-05 00102700
EPNR=0.25 00102800
54 CONTINUE 00102900
ITNR=ITNR+1 00103000
IF(ITNR.GT.15) EPNR=0.5 00103100
PS1=PSTAR(2)*EXP(-TSTAR(2)/TSN) 00103200
C I H2O SAT PRESS IN WET REGION 00103300
HH = QDEL1/HFG(2)+TRM3*(TWET(1)-TSN) 00103400
IF (HH .LT. 0.0) HH = 0.0 00103500
F=PS1*PS1-TRM1*HH-TRM2 00103600
* -PINFXT*PINFXT 00103700
DFDTS=2.*PS1*PS1*(TSTAR(2)/(TSN*TSN))+TRM1*TRM3 00103800
TSNR=TSN-F/DFDTS 00103900
ERR=ABS((TSN-TSNR)/TSN) 00104000
IF(ITNR.GT.16) WRITE(6,99) ITNR,TSN,TSNR,ERR,PS1,PINFXT 00104100
99 FORMAT(1X,I3,5E12.4) 00104200
TSNR=EPNR*TSN+(1.-EPNR)*TSNR 00104300
IF(ITNR.GT.15) ERM=1.E-03 00104400
IF(TSNR.LT.TWET(1)) TSNR=0.5*(TWET(1)+TSN) 00104500
TSN=TSNR 00104600
IF(ITNR.LT.20.AND.ERR.GT.ERM) GO TO 54 00104700
ERR1=ABS((TS-TSN)/TSN) 00104800
TS=TSN 00104900
C I USES H2O HEAT OF VAP 00105000
H2OFLX=QDEL1/HFG(2)+TRM3*(TWET(1)-TS) 00105100
IF (H2OFLX.LE.0.0) H2OFLX = 0.0 00105200
C 00105300
C CALC PRESSURES AND FLUXES BY THE RATE AT WHICH AIR IS PUSHED 00105400
C FROM THE SATURATION REGION 00105500
C 00105600
CALL SLDIFC(TC,PINFXT,DIFF,2,4,XMU) 00105700
CALL MGRATE(TIME,DT,WTEMP,EPSMI,LAST,PINFXT,TINFXT,XMU) 00105800
C 00105900
C 00106000
C ERROR CHECK AND DEBUG PRINTS 00106100
C 00106200
IF(IMPL.GT.20) WRITE(6,92) IMPL,NSTEP,TIME,DT,ERR1 00106300
92 FORMAT(6H IMPL=,I4,7H NSTEP=,I4,6H TIME=,F10.3,4H DT=, 00106400
* E10.3,5H ERR=,E10.3) 00106500
IF(IMPL.GT.20) WRITE(6,901) PS1,H2OFLX,DDL1DT,DEL1,TS,PS1 00106600
*,TR,VEL 00106700
901 FORMAT(5H PS1=,E10.3,8H H2OFLX=,E10.3,8H DDL1DT=,E10.3 00106800
*,6H DEL1=,E10.3,/,4H TS=,E10.3,5H PS1=,E10.3 00106900
*,4H TR=,E10.3,5H VEL=,E10.3) 00107000
IF(IMPL.GT.21) ERR1=0.0 00107100
C 00107200
RETURN 00107300
END 00107400
*/ *****00107500
*/ * SUBROUTINE SORGAS *00107600
*/ *****00107700
SUBROUTINE SORGAS (TIME,DT) 00107800
C 00107900
*CALL,/IMPLIC/ 00108000
*CALL,/SLCOM1/ 00108100
*CALL,/SLCOM2/ 00108200
*CALL,/NACON/ 00108300
*CALL,/REBLAY/ 00108400
C 00108500
C 00108600
C 00108700
C 00108800
RELEASE GAS CALCULATIONS FROM DRY REGION 00108900
C 00109000
NM1=NDRY-1 00109100
DD 102 I=1,NM1 00109200
DD 102 J=1,NSLM 00109300

```

```

RODRYD(I,J)=RODRY(I,J)                                00109300
102 CONTINUE                                           00109400
103 CONTINUE                                           00109500
C                                                       00109600
C   THE SOURCES TERMS FOR GAS RELEASE                 00109700
C                                                       00109800
ROMIN=1.E-5                                           00109900
DO 300 I=ISRCP1,NM1                                   00110000
SRH2O(I)=0.0                                         00110100
SRCO2(I)=0.0                                         00110200
C                                                       00110300
C   BOUND H2O                                         00110400
C                                                       00110500
SH2O=0.0                                             00110600
RODRY(I,2)=RODRYD(I,2)                               00110700
IF(RODRY(I,2).GT.ROMIN)SH2O=-3.3E10*EXP(-20560./TDRY(I)) 00110800
IF(SH2O.NE.0.0)RODRY(I,2)=RODRYD(I,2)*EXP(SH2O*DT) 00110900
C                                                       00111000
C   BOUND CO2                                         00111100
C                                                       00111200
SCO2=0.0                                             00111300
SCO2D=0.0                                            00111400
RODRY(I,10)=RODRYD(I,10)                            00111500
RODRY(I,13)=RODRYD(I,13)                            00111600
TDAVE=0.5*(TDRY(I)+TDRY(I+1))                      00111700
IF(RODRY(I,10).GT.ROMIN)SCO2=-3.6E5*EXP(-19362./TDAVE) 00111800
IF(RODRY(I,13).GT.ROMIN)SCO2D=-4.2E8*EXP(-19362./TDAVE) 00111900
IF(SCO2.LT.-0.15)SCO2=-0.15                         00112000
IF(SCO2D.LT.-0.15)SCO2D=-0.15                     00112100
IF(SCO2.NE.0.0)RODRY(I,10)=RODRYD(I,10)*EXP(SCO2*DT) 00112200
IF(SCO2D.NE.0.0)RODRY(I,13)=RODRYD(I,13)*EXP(SCO2D*DT) 00112300
RODRY(I,9)=(FCACO3*RHOCON-RODRY(I,10))*W(9)/W(10) 00112400
IF(RODRY(I,9).LT.0.0)RODRY(I,9)=0.0                00112500
RODRY(I,14)=(FMGCO3*RHOCON-RODRY(I,13))*W(14)/W(13) 00112600
IF(RODRY(I,14).LT.0.0)RODRY(I,14)=0.0              00112700
C                                                       00112800
C   THE GAS SOURCE TERMS                             00112900
C                                                       00113000
SRH2O(I)=- (RODRY(I,2)-RODRYD(I,2))/DT              00113100
SRCO2(I)=- (RODRY(I,10)-RODRYD(I,10))*W(11)/(W(10)*DT) 00113200
* - (RODRY(I,13)-RODRYD(I,13))*W(11)/(W(13)*DT) 00113300
300 CONTINUE                                          00113400
H2OSRC=0.0                                           00113500
CO2SRC=0.0                                           00113600
DO 115 I=ISRCP1,NM1                                  00113700
H2OSRC=H2OSRC+SRH2O(I)*WTZ(I)*DEL1                  00113800
CO2SRC=CO2SRC+SRCO2(I)*WTZ(I)*DEL1                  00113900
DO 115 J=1,NSLM                                       00114000
RODRY(I,J)=RODRYD(I,J)                              00114100
115 CONTINUE                                          00114200
RETURN                                               00114300
END                                                  00114400
*/ *****00114500
*/ * SUBROUTINE MGRATE *00114600
*/ *****00114700
SUBROUTINE MGRATE(TIME,DT,WTEMP,EPSMI,LAST,PINFXT,TINF,XMU) 00114800
C                                                       00114900
*CALL,/IMPLIC/                                       00115000
*CALL,/SLCOM1/                                       00115100
*CALL,/SLCOM2/                                       00115200
*CALL,/NACON/                                       00115300
*CALL,/REBLAY/                                       00115400
*CALL,/REBCOM/                                       00115500
C                                                       00115600
DIMENSION PP(51),RH2OLD(51),ROAIRO(51),TWETO(51) 00115700
DIMENSION DRG(51),DRI(51),FPRS(51,2),POLD(51,2) 00115800
DIMENSION PERML(51),PERMG(51)                       00115900
DATA PERMG/51*0.0/                                   00116000
C                                                       00116100
C   BEFORE FIRST TIME STEP PRESET ALL THE VARIABLES FOR 00116200
C   THE WATER MIGRATION MODEL                         00116300
C                                                       00116400
LAST=NWET                                           00116500
SEAL=0.                                              00116600
ALINC=0.006                                         00116700
ISATO=0                                             00116800
ISATL=1                                             00116900
C                                                       00117000

```



```

C      UPDATE BEGINNING OF TIME STEP VALUES                                00117100
C                                                                                   00117200
      DO 102 I=1, LAST                                                         00117300
      ROAIRO(I)=ROAIR(I)                                                         00117400
      RH2OLO(I)=RHH2OL(I)                                                       00117500
      TWETO(I)=TWET(I)                                                           00117600
      PWET(I)=PBOT                                                                00117700
      APW=2.94E-28                                                                00117800
      BPW=0.069                                                                    00117900
      IF(TWET(I).GT.383.) APW=2.27E-22                                          00118000
      IF(TWET(I).GT.383.) BPW=0.0336                                           00118100
      PERML(I)=APW*EXP(BPW*TWET(I))                                             00118200
102    PERMG(I)=PERML(I)*2.                                                       00118300
1011  CONTINUE                                                                    00118400
C                                                                                   00118500
C      FIND LAST NODE                                                            00118600
C                                                                                   00118700
      LASTN=LAST                                                                  00118800
      XLEN=DELM-DELI-DIST                                                         00118900
      DO 330 I=1, LAST                                                            00119000
      IF(XETA(I).LT.XLEN) GO TO 330                                             00119100
      LASTN=I                                                                      00119200
      XETA(I)=XLEN                                                                00119300
      IF(XETA(I)-XETA(I-1) .LT. XETA(2) ) THEN                                00119400
      LASTN = I - 1                                                                00119500
      NWET = LASTN                                                                00119600
      ENDIF                                                                        00119700
      GO TO 340                                                                    00119800
330   CONTINUE                                                                    00119900
340   LAST=LASTN                                                                  00120000
      IF(LAST.EQ.4) TWFIN=TIME                                                    00120100
      NM1=LAST-1                                                                  00120200
      IMPL2=0                                                                      00120300
      EPSMI=0.5                                                                    00120400
C                                                                                   00120500
C      IMPLICIT ITERATION POINT ON THE END OF TIME STEP TEMP AND PRESS        00120600
C                                                                                   00120700
50    CONTINUE                                                                    00120800
      IMPL2=IMPL2+1                                                                00120900
C                                                                                   00121000
C      THE COORD SYSTEM VELOCITY DDL1DN                                       00121100
C                                                                                   00121200
      IF(MIG.EQ.0) GO TO 220                                                       00121300
      IF(ALPH(1).GT.ALINC) GO TO 200                                             00121400
      ISATO=1                                                                      00121500
      DO 210 I=2, LAST                                                            00121600
      IF(ALPH(I).LT.ALINC) GO TO 210                                           00121700
      ISATL=I                                                                      00121800
      GO TO 220                                                                    00121900
210   CONTINUE                                                                    00122000
C                                                                                   00122100
C      INCOMPRESIBLE LIQUID WATER FLOW                                         00122200
C                                                                                   00122300
C                                                                                   00122400
220   CONTINUE                                                                    00122500
      IF (MIG.EQ.0) ISATL=LAST                                                    00122600
      DXSAT=XETA(ISATL)                                                           00122700
      DPOXO=(PS1-PWET(ISATL))/DXSAT                                             00122800
      SUM=0.0                                                                      00122900
      LST=ISATL-1                                                                  00123000
      DO 295 I=1, LST                                                             00123100
      SUM=SUM+(XETA(I+1)-XETA(I))/PERML(I)                                       00123200
295   CONTINUE                                                                    00123300
      PERNT=XETA(ISATL)/SUM                                                       00123400
      DRL(1)=PERNT*DPOXO/XMUL(2)                                                 00123500
      GO TO 230                                                                    00123600
200   CONTINUE                                                                    00123700
      DPOXO=2.*(PS1-PWET(1))/(XETA(2)-XETA(1))                                  00123800
      XKRL=(1.-ALPH(1)/EPSM)**3                                                  00123900
      DRL(1)=PERML(1)*XKRL*DPOXO/XMUL(2)                                        00124000
230   CONTINUE                                                                    00124100
      XKRM=ALPH(1)/EPSM                                                           00124200
      IF(XKRM.LT.1.E-10) XKRM=1.E-10                                           00124300
      DRG(1)=PERMG(1)*XKRM*DPOXO/XMU                                           00124400
      DDL1DO=DDL1DN                                                                00124500
      DDL1DN=H2OFLX/RHH2OL(1)+DRL(1)                                           00124600
642  CONTINUE                                                                    00124700
C                                                                                   00124800

```

```

C      END OF TIME STEP TEMP                                00124900
C                                                                 00125000
      TWAVE=0.5*(TWET(1)+TWET(LAST))                       00125100
C      I AVG CONDUCTIVITY IN WET REGION                    00125200
      TKWET=TKA*TWAVE+TKB                                  00125300
      TDIFW=TKWET/(ROCP)                                  00125400
      RFDX2=1./(2.*XETA(2)*XETA(2))                      00125500
      RFDX1=1./((XETA(3)*XETA(2)))                       00125600
      TSP=2.*TS-TWET(1)                                   00125700
      DXID=XETA(2)*0.5+0.5*(XETA(3)-XETA(2))            00125800
      TID=TS                                              00125900
      IF(DDL1DN.GT.0.0) TID=TWET(2)                      00126000
      VV=-DDL1DN                                          00126100
      DENOM=(1./DT-VV/DXID)+2.*TDIFW*(RFDX1+RFDX2)      00126200
      XNUM=TWETO(1)/DT-VV*TID/DXID+2.*TDIFW*(TWET(2)*RFBX1
*      +TSP*RFDX2)                                       00126400
      TWET(1)=XNUM/DENOM                                  00126500
      PWET(1)=PS1                                         00126600
      RFBX2=1./((XETA(LAST)-XETA(LAST-2))*(XETA(LAST)-XETA(NM1)))
      DXID=XETA(LAST)-XETA(NM1)                          00126800
      TID=TWET(NM1-1)                                    00126900
      IF(DDL1DN.GT.0.0) TID=TWET(NM1)                   00127000
      DENOM=(1./DT-VV/DXID+4.*TDIFW*RFDX2)              00127100
      XNUM=TWETO(NM1)/DT-VV*TID/DXID+2.*TDIFW*TWET(NM1-1)*RFDX2
      XNUM=XNUM+2.*TDIFW*TXL*RFDX2                      00127300
      TWET(NM1)=XNUM/DENOM                              00127400
      NM2=NM1-1                                          00127500
      DO 199 I=2,NM2                                     00127600
      RFDX1=1./((XETA(I+2)-XETA(I))*(XETA(I+1)-XETA(I)))
      RFDX2=1./((XETA(I+1)-XETA(I-1))*(XETA(I+1)-XETA(I)))
      DXID=0.5*(XETA(I+1)-XETA(I-1))                    00127900
      IF(DDL1DN.GT.0.0) DXID=0.5*(XETA(I+2)-XETA(I))
      TID=TWET(I-1)                                      00128100
      IF(DDL1DN.GT.0.0) TID=TWET(I+1)                   00128200
      DENOM=(1./DT-VV/DXID)+2.*TDIFW*(RFDX1+RFDX2)
      XNUM=TWETO(I)/DT-VV*TID/DXID+2.*TDIFW*
*      ( TWET(I+1)*RFDX1+TWET(I-1)*RFDX2 )              00128500
      TWET(I)=XNUM/DENOM                                 00128600
      IF(IMPL2.GT.1) GO TO 199                            00128700
C      I H2O SAT PRESS IN WET REGION                      00128800
      PWVAP=PSSTAR(2)*EXP(-TSTAR(2)/TWET(I))            00128900
      PWET(I)=ROAIR(1)*RGAS(2)*TWET(I)/(ALPH(I))+PWVAP
199     CONTINUE                                         00129000
      IF(MIG.EQ.0) RETURN                                 00129200
C      IF((NSTEP/MIG)*MIG.NE.NSTEP .AND. NSTEP.NE.1) RETURN
C                                                                 00129300
C                                                                 00129400
C      END OF TIME STEP DENSITIES AND PRESSURE           00129500
C                                                                 00129600
      PWET(LAST)=PINFXT                                    00129700
      RDH2DL(LAST)=RDH2DL(NM1)                          00129800
      ROAIR(LAST)=ROAIR(NM1)                             00129900
      IF(ISATO.NE.1) GO TO 260                            00130000
      DO 270 I=ISATO,ISATL                               00130100
      DRL(I)=DRL(1)                                      00130200
      DRG(I)=0.0                                         00130300
      IF(I.EQ.ISATL) GO TO 270                            00130400
      DXP=XETA(I+1)-0.5*(XETA(I+1)-XETA(I))              00130500
      PWET(I)=PS1-(PS1-PWET(ISATL))*DXP/XETA(ISATL)
270     CONTINUE                                         00130700
260     CONTINUE                                         00130800
      DO 31 I=ISATL,NM1                                  00130900
      ALPHL=EPSM-ALPH(I)                                 00131000
      DDPX=(PWET(I)-PWET(I+1))/(XETA(I+1)-XETA(I))
      XKRL=(1.-ALPH(I)/EPSM)**3                          00131200
      DRL(I+1)=PERML(I)*DPDX*XKRL/(XMUL(2))            00131300
      IF(ALPH(I).GT.ALINC) GO TO 280                    00131400
280     CONTINUE                                         00131500
      IF(I.EQ.NM1) DRL(I+1)=AMAX1(0.0,DRL(I+1))
      XKRM=ALPH(I)/EPSM                                  00131700
      IF(XKRM.LT.1.E-10) XKRM=1.E-10                   00131800
      DRG(I+1)=PERMG(I)*DPDX *XKRM / (XNU)             00131900
C                                                                 00132000
C      LIQUID                                             00132100
C                                                                 00132200
      VELI=DRL(I)-DDL1DN                                 00132300
      VELIP1=DRL(I+1)-DDL1DN                            00132400
      IF(I.EQ.NM1 .AND. SEAL.EQ.1) VELIP1=-DDL1DN
      DLI=0.0                                             00132500
      DLI=0.0                                             00132600

```

```

DRI=1.0 00132700
IF(VELI.GT.0.0) GO TO 5 00132800
DLI=-1.0 00132900
DRI=0.0 00133000
5 DLIP1=0.0 00133100
DRIP1=1.0 00133200
IF(VELIP1.GT.0.0) GO TO 6 00133300
DLIP1=-1.0 00133400
DRIP1=0.0 00133500
6 CONTINUE 00133600
XNUM=ROH2OL(I-1)*VELI*DRI+ROH2OL(I+1)*VELIP1*DLIP1 00133700
DXETA=XETA(I+1)-XETA(I) 00133800
XNUM=XNUM/DXETA+RH2OLO(I)/DT 00133900
DENOM=1./DT+(VELI*DLI+VELIP1*DRIP1)/DXETA 00134000
ROH2OL(I)=XNUM/DENOM 00134100
C 00134200
C GAS 00134300
C 00134400
VELI=DRG(I)-DDL1DN 00134500
VELIP1=DRG(I+1)-DDL1DN 00134600
IF(I.EQ.NM1 .AND. SEAL.EQ.1) VELIP1=-DDL1DN 00134700
IF(I.EQ.1 .AND. VELI.GT.0.0) VELI=0.0 00134800
DLI=0.0 00134900
DRI=1.0 00135000
IF(VELI.GT.0.0) GO TO 7 00135100
DLI=-1.0 00135200
DRI=0.0 00135300
7 DLIP1=0.0 00135400
DRIP1=1.0 00135500
IF(VELIP1.GT.0.0) GO TO 8 00135600
DLIP1=-1.0 00135700
DRIP1=0.0 00135800
8 CONTINUE 00135900
XNUM=ROAIR(I-1)*VELI*DRI+ROAIR(I+1)*VELIP1*DLIP1 00136000
DXETA=XETA(I+1)-XETA(I) 00136100
XNUM=XNUM/DXETA+ROAIR(I)/DT 00136200
DENOM=1./DT+(VELI*DLI+VELIP1*DRIP1)/DXETA 00136300
ROAIR(I)=XNUM/DENOM 00136400
ALPH(I)=EPSM-ROM2OL(I)/ROMH2O 00136500
DXDT=ABS(DDL1DN) 00136600
C I H2O SAT PRESS IN WET REG 00136700
PWVAP=PSTAR(2)*EXP(-TSTAR(2)/TWET(I)) 00136800
PP(I)=ROAIR(I)*RGAS(2)*TWET(I)/(ALPH(I))+PWVAP 00136900
31 CONTINUE 00137000
C 00137100
C UPDATE AND CHECK FOR CONVERGENCE 00137200
C 00137300
ERRMX=0.0 00137400
DO 40 I=ISATL,NM1 00137500
ERR=ABS((PP(I)-PWET(I))/PWET(I)) 00137600
IF(ERR.GT.ERRMX) INX=I 00137700
ERRMX=AMAX1(ERRMX,ERR) 00137800
40 CONTINUE 00137900
C 00138000
C NEWTON RAPHSON OR WIEGHTED ITERATION UPDATE ON PRESSURE 00138100
C 00138200
DO 145 I=ISATL,NM1 00138300
116 IF(IMPL2.GT.20 .AND. I.EQ.INX) WRITE(6,909)IMPL2,I,PWET(I) 00138400
* ,PP(I),ERRMX 00138500
909 FORMAT(2I5,5E12.4) 00138600
FPRS(I,2)=FPRS(I,1) 00138700
FPRS(I,1)=PP(I)-PWET(I) 00138800
POLD(I,2)=POLD(I,1) 00138900
POLD(I,1)=PWET(I) 00139000
IF(ABS(FPRS(I,1)/PP(I)).LT.1.E-08) GO TO 115 00139100
IF(IMPL2.EQ.1) GO TO 115 00139200
IF(FPRS(I,2)/FPRS(I,1).GT.0.0) GO TO 115 00139300
IF(PWET(I)-POLD(I,2).EQ.0.0) GO TO 115 00139400
DFDP=(FPRS(I,1)-FPRS(I,2))/(PWET(I)-POLD(I,2)) 00139500
IF(DFDP.EQ.0.0) GO TO 115 00139600
PWET(I)=PWET(I)-FPRS(I,1)/DFDP 00139700
GO TO 145 00139800
115 PWET(I)=EPSMI*PP(I)+(1.-EPSMI)*PWET(I) 00139900
145 CONTINUE 00140000
IF((IMPL2/10)*10.EQ.IMPL2) EPSMI=EPSMI*0.9 00140100
IF(IMPL2.EQ.50) STOP 00140200
EPSMI=AMAX1(EPSMI,0.2) 00140300
IF(ERRMX.GT.5.E-04) GO TO 50 00140400

```

```

C                                     00140500
C   CALCULATE EXPLICIT PARAMETERS AFTER CONVERGENCE                       00140600
C                                     00140700
C   PPWM=DRL(1)/DDL1DN                                                    00140800
C   COURNT=0.0                                                             00140900
C   DO 80 I=1, LAST                                                         00141000
C   CRNT=( (DRG(I)-DDL1DN)*DT)/(XETA(I+1)-XETA(I))                       00141100
C   IF(ROAIR(I).LT.1.E-6) CRNT=0.0                                        00141200
C   COURNT=AMAX1(COURNT,ABS(CRNT))                                        00141300
80  CONTINUE                                                                00141400
C   RMIG=AMIN1(DDL1DN*ROH2OL(1)/H2OFLX,5.0)                             00141500
C   DRL10=DRL(1)                                                           00141600
C   RETURN                                                                  00141700
90  FORMAT(6E11.4)                                                         00141800
C   END                                                                     00141900
*/ *****00142000
*/ *   SUBROUTINE SLENTH                                                    *00142100
*/ *****00142200
C   SUBROUTINE SLENTH(TEMP,ENTH,MAT,IPHS)                                  00142300
*CALL,/IMPLIC/                                                            00142400
*CALL,/SLCOM1/                                                            00142500
*CALL,/SLCOM2/                                                            00142600
C                                     00142700
C   TREF=298.                                                                00142800
C   IF(IPHS.EQ.2) GO TO 10                                                  00142900
C   IF(IPHS.EQ.3) GO TO 20                                                  00143000
C   I ENTHALPY OF SOLID- PROP                                              00143100
C   ENTH=HF(MAT)+SP(MAT,1)*(TEMP-TREF)*W(MAT)                             00143200
C   RETURN                                                                  00143300
10  CONTINUE                                                                00143400
C   SENS=SP(MAT,1)*(TMT(MAT)-TREF)+SP(MAT,2)*(TEMP-TMT(MAT))            00143500
C   J ENTHALPY OF LIQUID- PROP                                            00143600
C   ENTH=HF(MAT)+(SENS+HFUS(MAT))*W(MAT)                                  00143700
C   RETURN                                                                  00143800
20  CONTINUE                                                                00143900
C   SENS=SP(MAT,1)*(TMT(MAT)-TREF)+SP(MAT,2)*(TSAT(MAT)-TMT(MAT))      00144000
C   * +SP(MAT,3)*(TEMP-TSAT(MAT))                                          00144100
C   I ENTHALPY OF VAPOR- PROP                                             00144200
C   ENTH=HF(MAT)+(SENS+HFUS(MAT)+HFG(MAT))*W(MAT)                       00144300
C   RETURN                                                                  00144400
C   END                                                                     00144500
[*****00144600
*/ *****00144700
*/ *   SUBROUTINE SLDIFC                                                    *00144800
*/ *****00144900
C   SUBROUTINE SLDIFC(TAVE,PRESS,DIFF,I,J,XHU)                            00145000
C                                     00145100
C   THIS SUBROUTINE CALCULATES THE AVERAGE DIFFUSION COF                00145200
C   FOR SPECIES I AND J AT THE AVERAGE TEMP TAVE                        00145300
C                                     00145400
*CALL,/IMPLIC/                                                            00145500
*CALL,/SLCOM1/                                                            00145600
*CALL,/SLCOM2/                                                            00145700
C                                     00145800
C   DIMENSION EPSOK(17),SIGMA(17)                                         00145900
C                                     00146000
C   PATH=PRESS/1.013E05                                                    00146100
C   EPSCON=0.20                                                            00146200
C   EPSOK(2)=32.                                                            00146300
C   EPSOK(3)=1375.                                                         00146400
C   EPSOK(4)=60.                                                           00146500
C   EPSOK(5)=1962.                                                         00146600
C   SIGMA(2)=3.737                                                         00146700
C   SIGMA(3)=3.567                                                         00146800
C   SIGMA(4)=2.827                                                         00146900
C   SIGMA(5)=3.804                                                         00147000
C   TERM1=(TAVE)**3*(1./W(I)+1./W(J))                                       00147100
C   SIG=(SIGMA(I)+SIGMA(J))/2.                                             00147200
C   EPS=(EPSOK(I)*EPSOK(J))*0.5                                           00147300
C   TEPS=TAVE/EPS                                                          00147400
C   XMD=0.82515+1.9066E-3*TEPS+0.66448*TEPS**(-1.0676)-0.050246*      00147500
C   * SQRT(TEPS)                                                           00147600
C   DIFF=0.018768*(TERM1)**0.5/(SIG*SIG*XMD*PRESS)                       00147700
C   TERM1=2.67E-6*SQRT(W(I)*TAVE)/SIGMA(I)**2                             00147800
C   TEPS=TAVE/EPSON(I)                                                     00147900
C   XMD=0.92495+2.07368E-3*TEPS+0.719288*TEPS**(-1.151049)-5.46452E-2 00148000
C   * *SQRT(TEPS)                                                         00148100
C   XMU=TERM1/XMD                                                         00148200

```

```

RETURN 00148300
END 00148400
*/ *****00148500
*/ * SUBROUTINE SLTRIS *00148600
*/ *****00148700
SUBROUTINE SLTRIS(A,B,C,D,N) 00148800
C 00148900
*CA /IMPLIC/ 00149000
DIMENSION A(1),B(1),C(1),D(1) 00149100
C 00149200
C A=LOWER DIAGONAL 00149300
C B=DIAGONAL 00149400
C C=UPPER DIAGONAL 00149500
C D=RESULTANT 00149600
C N=NUMBER OF UNKNOWNNS 00149700
C 00149800
C 00149900
DO 10 I=2,N 00150000
FCTR=A(I)/B(I-1) 00150100
B(I)=B(I)-FCTR*C(I-1) 00150200
D(I)=D(I)-FCTR*D(I-1) 00150300
10 CONTINUE 00150400
C 00150500
D(N)=D(N)/B(N) 00150600
K=N 00150700
DO 20 I=2,N 00150800
K=K-1 00150900
D(K)=(D(K)-C(K)*D(K+1))/B(K) 00151000
20 CONTINUE 00151100
C 00151200
RETURN 00151300
END 00151400
*/ *****00151500
*/ * SUBROUTINE GAUSS *00151600
*/ *****00151700
SUBROUTINE GAUSS(A,X,B,M,NDIM) 00151800
*CA /IMPLIC/ 00151900
C 00152000
DIMENSION A(NDIM,1),X(1),B(1),IS(30),JS(30),LOC(30) 00152100
N=0 00152200
DO 13 K=1,M 00152300
IF(A(K,K).EQ.0.0)GO TO 13 00152400
N=N+1 00152500
LOC(N)=K 00152600
13 CONTINUE 00152700
IF(N.EQ.0) RETURN 00152800
IF (N .GT. 1) GO TO 14 00152900
K=LOC(1) 00153000
X(K)=B(K)/A(K,K) 00153100
RETURN 00153200
14 N1=N-1 00153300
DO 4 KK=1,N1 00153400
K=LOC(KK) 00153500
CMAX=0.0 00153600
DO 1 II=KK,N 00153700
I=LOC(II) 00153800
DO 2 JJ=KK,N 00153900
J=LOC(JJ) 00154000
IF(ABS(A(I,J)).LE.CMAX) GO TO 2 00154100
IM=1 00154200
JM=J 00154300
IS(KK)=II 00154400
JS(KK)=JJ 00154500
CMAX=ABS(A(I,J)) 00154600
2 CONTINUE 00154700
1 CONTINUE 00154800
DO 11 JJ=KK,N 00154900
J=LOC(JJ) 00155000
TEMP=A(K,J) 00155100
A(K,J)=A(IM,J) 00155200
A(IM,J)=TEMP 00155300
11 CONTINUE 00155400
DO 12 II=1,N 00155500
I=LOC(II) 00155600
TEMP=A(I,K) 00155700
A(I,K)=A(I,JM) 00155800
A(I,JM)=TEMP 00155900
12 CONTINUE 00156000

```

```

      K1=KK+1
      DO 3 I1=K1,N
      I=LOC(I1)
      A(I,K)=A(I,K)/A(K,K)
3     CONTINUE
      DO 4 I1=K1,N
      I=LOC(I1)
      DO 4 JJ=K1,N
      J=LOC(JJ)
      A(I,J)=A(I,J)-A(I,K)*A(K,J)
4     CONTINUE
      DO 5 KK=1,N
      K=LOC(KK)
      X(K)=B(K)
5     CONTINUE
      DO 6 KK=1,N1
      K1=KK+1
      IM=IS(KK)
      K=LOC(KK)
      IM=LOC(IM)
      TEMP=X(K)
      X(K)=X(IM)
      X(IM)=TEMP
      DO 6 I1=K1,N
      I=LOC(I1)
      X(I)=X(I)-A(I,K)*X(K)
6     CONTINUE
      DO 7 I=1,N
      K=N-I+1
      K1=K+1
      K=LOC(K)
      TEMP=0.0
      IF(K1.GT.N) GO TO 8
      DO 9 JJ=K1,N
      J=LOC(JJ)
      TEMP=TEMP+X(J)*A(K,J)
9     CONTINUE
8     X(K)=(X(K)-TEMP)/A(K,K)
7     CONTINUE
      DO 10 I=1,N1
      K=N-I
      JM=JS(K)
      JM=LOC(JM)
      K=LOC(K)
      TEMP=X(K)
      X(K)=X(JM)
      X(JM)=TEMP
10    CONTINUE
      RETURN
      END
*/ *****
*/ *          SUBROUTINE  NATCON          *00161200
*/ *****00161300
      SUBROUTINE NATCON (TIME,DT,TPOOL,ROPOOL,RLIQ,DIFO,PFLUX,PINFXT,
1          TINFXT)
C
*CALL,/IMPLIC/
*CALL,/NACON/
*CALL,/SLCON1/
*CALL,/SLCON2/
*CALL,/REBLAY/
C
      DIMENSION GAMBL(51),ROPOOL(17),RLIQ(51,8),PFLUX(17)
1          ,A(51),B(51),C(51),D(51),V(51)
      DATA FRAC /0.500/
C
      SOLVE THE SPECIES CONTINUITY EQN BY A THREE STEP PROCEDURE
C      STEP 1 CHEMICAL REACTION
C      STEP 2 MASS DIFFUSION
C      STEP 3 ADVECTION AND MASS CONSERVATION
C
      DO 10 I =1,52
10     QV(I) = 0.0
C
      THE CHEMICAL REACTIONS
C
      NM1=NDRY-1
      DO 510 I=1,ISRC

```

```

DO 515 J=1,NSLM
515 RODRYI(I,J)=RODRYO(I,J) 00163900
DO 510 IR=1,8 00164000
RLIQ(I,IR)=0.0 00164100
510 CONTINUE 00164200
SUM = 0.0 00164300
DO 520 I=1,ISRC 00164400
CALL REACT (TIME,DT,TPOOL,I,1,RLIQ) 00164500
SUM = SUM + EXH2(I)*DEL1*XZI(2) 00164600
520 CONTINUE 00164700
XMH2XT = SUM 00164800
H2FLXT = XMH2XT*DT 00164900
C 00165000
C THE DIFFUSION 00165100
C 1 DENSITY OF H2O(V) IN DRY REGION 00165200
RHO2=PIFXT/(RGAS(2)*TDRY(ISRC)) 00165300
C 1 DENSITY OF CO2 IN DRY REGION 00165400
RHO11=PIFXT/(RGAS(11)*TDRY(ISRC)) 00165500
VGAS=(H2OFLX+H2OSRC)/RHO2 00165600
DXXI=DEL1*XZI(2) 00165700
GAMV=GAMO*VGAS*DXXI 00165800
DO 500 I=1,ISRCP1 00165900
AAM=GAM2*DEL1*(XZI(ISRCP1)-XZI(I)) 00166000
BB=0.0 00166100
IF(AAM.LT.14.) BB=EXP(-AAM) 00166200
GAMBL(I)= (GAMV*(1.+GAM1*(1.-BB))+GAMINF*DXXI) 00166300
500 CONTINUE 00166400
DO 560 J=1,NSLM 00166500
DO 600 I=1,ISRC 00166600
A(I)=-GAMBL(I) 00166700
B(I)=DXXI**2/DT+GAMBL(I)+GAMBL(I+1) 00166800
C(I)=-GAMBL(I+1) 00166900
D(I)=RODRYI(I,J)*DXXI**2/DT 00167000
600 CONTINUE 00167100
D(1)=D(1)+ROPOOL(J)*GAMBL(1) 00167200
B(ISRC)=B(ISRC)-GAMBL(ISRC+1) 00167300
A(1)=0.0 00167400
C(ISRC)=0.0 00167500
CALL SLTRIS(A,B,C,D,ISRC) 00167600
DO 550 I=1,ISRC 00167700
RODRYI(I,J)=D(I) 00167800
550 CONTINUE 00167900
560 CONTINUE 00168000
C 00168100
C LAST THE ADVECTION 00168200
C 00168300
DO 335 I=ISRCP1,NDRY 00168400
V(I)=VEL 00168500
335 CONTINUE 00168600
DO 33 K=1,NM1 00168700
I=K 00168800
ITNR=0 00168900
IF(K.LE.ISRC) I=ISRC-K+1 00169000
IF(K.LE.ISRC) V(I)=V(I+1) 00169100
205 CONTINUE 00169200
ITNR=ITNR+1 00169300
XXI= (XZI(I)-XZI(ISRCP1)) 00169400
XXIP1= (XZI(I+1)-XZI(ISRCP1)) 00169500
VELI=V(I)-DDL1DT*XXI 00169600
VELIP1=V(I+1)-DDL1DT*XXIP1 00169700
IF(I.EQ.ISRC) VELIP1=0.0 00169800
DENOM=1./DT+DDL1DT/DEL1 00169900
COR1=0.0 00170000
TKBL=0.0 00170100
C 00170200
C MATERIAL LOOP 00170300
C 00170400
DO 31 J=1,NSLM 00170500
IF(I.GT.ISRC .AND. RODRY(I,J).EQ.0.0) GO TO 31 00170600
VB=0.0 00170700
IF(IGAS(J).NE.0) VB=0.0 00170800
VIPM=AMIN1(0.0,VELIP1-VB) 00170900
VIPP=AMAX1(0.0,VELIP1-VB) 00171000
VIP=AMAX1(0.0,VELI-VB) 00171100
VIM=AMIN1(0.0,VELI-VB) 00171200
DENOMT=DENOM+(VIPP-VIM)/DXXI 00171300
ROIN1=RODRY(I-1,J) 00171400
ROI1P1=RODRY(I+1,J) 00171500
00171600

```

```

IF(I.EQ.ISRCP1) ROIM1=0.0                                00171700
IF(I.EQ.1) ROIM1=AMAX1(0.,ROPOOL(J))                   00171800
IF(I.EQ.ISRC) DENOMT=DENOMT-VIPP/DXXI                 00171900
ROVE=(ROIM1*VIP-VIPM*ROIPI)/DXXI                     00172000
C                                                       00172100
C SOURCES IN THE INTERFACE NODE DUE TO ABLATION        00172200
C                                                       00172300
SRC=0.0                                                00172400
IF(I.NE.ISRC) GO TO 838                                00172500
SRC=-VEL*RODRY(I+1,J)                                  00172600
SRC=AMAX1(0.0,SRC/DXXI)                                00172700
838 CONTINUE                                           00172800
XNUM=ROVE+RODRY(I,J)/DT + SRC                          00172900
RODRY(I,J)=XNUM/(DENOMT)                                00173000
IF (RODRY(I,J) .LE. 0.0) RODRY(I,J)=0.0               00173100
IF(I.EQ.1) PFLUX(J)=-VIM*RODRY(I,J)-VIP*ROIM1         00173200
* + GAMBL(1)*(RODRY(I,J)-ROIM1)/DXXI                 00173300
IF(I.GT.ISRC) GO TO 31                                 00173400
VOLFR=RODRY(I,J)/RHOL(J)                              00173500
COR1=COR1+VOLFR                                        00173600
CPBL=SP(J,2)                                           00173700
IF(CPBL.EQ.0.0) CPBL=CPCOM                             00173800
IF(J.EQ.3 .AND. VOLFR.LT.0.25) VOLFR=0.0              00173900
TKBL=TKBL+VOLFR*(TKMAT(J)+GAMBL(I+1)*RHOL(J)*CPBL)  00174000
IF(I.EQ.ISRC) DIFO=GAMBL(I+1)                          00174100
31 CONTINUE                                           00174200
C                                                       00174300
C END OF MATRL DO LOOP                                 00174400
C                                                       00174500
C                                                       00174600
IF(I.GT.ISRC) GO TO 32                                  00174700
TKDRY(I)=(ALPHAL(I)*TKBL+(1.-ALPHAL(I))*0.03)*ALPHAL(I) 00174800
F=1.-COR1                                              00174900
IF(ABS(F).LT.0.00002) GO TO 32                         00175000
IF(ITNR.EQ.1) GO TO 360                                00175100
DFDV=(F-FOLD)/(V(I)-VOLD)                             00175200
FOLD=F                                                 00175300
VNEW=V(I)-F/DFDV                                       00175400
CHG=1.                                                 00175500
IF(VNEW.NE.0.0) CHG=ABS((VNEW-V(I))/VNEW)             00175600
IF(CHG.GT.0.5) V(I)=(V(I)+VNEW)*0.5                  00175700
IF(CHG.LE.0.5) V(I)=VNEW                              00175800
IF(ITNR.GT.10) V(I)=(VOLD+VNEW)*0.5                  00175900
GO TO 205                                              00176000
360 FOLD=F                                             00176100
VOLD=V(I)                                              00176200
V(I)=V(I)*1.05-1.E-05                                00176300
GO TO 205                                              00176400
32 CONTINUE                                           00176500
33 CONTINUE                                           00176600
C                                                       00176700
C CALCULATE THE LOCAL BUBBLE VELOCITY BASED UPON STOKES THEOREM 00176800
C                                                       00176900
DO 710 I=1,ISRC                                       00177000
RHOBL=0.0                                              00177100
DO 711 J=1,NSLM                                       00177200
RHOBL=RHOBL+RODRY(I,J)                                00177300
711 CONTINUE                                           00177400
ANAL=RODRY(I,3)/RHOL(3)                                00177500
ANAOH=RODRY(I,5)/RHOL(5)                              00177600
ALRE=1.-ANAL-ANAOH                                    00177700
AMUL=ANAL*1.86E-4+ANAOH*1.E-3                       00177800
XHUBL=(AMUL*XVAR1 + ALRE*2.)/(AMUL*XVAR1+ALRE)        00177900
VBUB(I)=AMIN1(0.5,2.*RBUB**2*RHOBL/XHUBL)           00178000
ALPHAL(I)=1.-VGAS/(VBUB(I)+VGAS)                     00178100
710 CONTINUE                                           00178200
C                                                       00178300
C CALCULATE ABLATION VELOCITY BY CHEMICAL REACTION    00178400
C                                                       00178500
IR=3                                                   00178600
AKK=AMIN1(AKF(IR,1),SAK(IR)*EXP(-EK(IR)/(1.98*TDRY(ISRCP1)))) 00178700
IF(RODRY(I,3)/RHOL(3).GT.AKF(IR,3)) AKK=AKK*AKF(IR,2) 00178800
DRIVE=RODRY(ISRC,3) / (COFR(IR,3)*W(3))              00178900
VELN=0.0                                               00179000
IF(TIME.GT.TNATO) VELN=-AKK*DRIVE*FRAC*DEL1*XZI(2)   00179100
VELN=AMIN1(0.0,VELN)                                  00179200
89 CONTINUE                                           00179300
C                                                       00179400

```



```

C
C      HEAT SOURCES DUE TO CHEMISTRY
C
      CALL HEAT (1,RLIQ)
      HBH20=3.E6
      HBCO2=4.E6
      QVSUM=0.0
      TERM=0.5*DEL1*ROCP*XZI(2)
      NTM1=NDRY-1
      DO 122 I=1,NTM1
      QV(I+1)=(QV(I+1)-(SRH20(I)*HBH20+SRCO2(I)*HBCO2))/ROCP
      IF(I.GE.2) QVSUM=QVSUM+(QV(I)+QV(I-1))*TERM
122  CONTINUE
      RETURN
90  FORMAT(1H ,6E11.4)
      END
*/ *****00179500
*/ *      SUBROUTINE WTRRLS *00179600
*/ *****00179700
      SUBROUTINE WTRRLS (TIME,DT,WTEMP,WRD,RLIQ,DIFO,PFLUX,PINFXT,
1      TINFXT)
C
C      *CALL,/IMPLIC/
C      *CALL,/NACON/
C      *CALL,/SLCOM1/
C      *CALL,/SLCOM2/
C      *CALL,/REBLAY/
C
C      DIMENSION GAMBL(51),WRD(17),RLIQ(51,8),PFLUX(17)
1      ,A(51),B(51),C(51),D(51),V(51)
      DATA FRAC /0.500/
C
C      SOLVE THE SPECIES CONTINUITY EQN BY A TWO STEP PROCEDURE
C      STEP 1 MASS DIFFUSION
C      STEP 2 ADVECTION AND MASS CONSERVATION
C
      DO 10 I =1,52
10  QV(I) = 0.0
C
C      NM1=NDRY-1
      DO 510 IR=1,8
      RLIQ(I,IR)=0.0
510  CONTINUE
      SUM = 0.0
C
      DXXI=DEL1*XZI(2)
C
C      LAST THE ADVECTION
C
      DO 335 I=ISRCP1,NDRY
      V(I)=VEL
335  CONTINUE
      DO 33 K=1,NM1
      I=K
      ITNR=0
      IF(K.LE.ISRC) I=ISRC-K+1
      IF(K.LE.ISRC) V(I)=V(I+1)
205  CONTINUE
      ITNR=ITNR+1
      XXI= (XZI(I)-XZI(ISRCP1))
      XXIP1= (XZI(I+1)-XZI(ISRCP1))
      VELI=V(I)-DDL1DT*XXI
      VELIP1=V(I+1)-DDL1DT*XXIP1
      IF(I.EQ.ISRC) VELIP1=0.0
      DENOM=1./DT+DDL1DT/DEL1
      COR1=0.0
      TKBL=0.0
C
C      MATERIAL LOOP
C
      DO 31 J=1,NSLM
      IF(I,GT.ISRC .AND. RODRY(I,J).EQ.0.0) GO TO 31
      VB=0.0
      IF(IGAS(J).NE.0) VB=0.0
      VIPH=AMIN1(0.0,VELIP1-VB)
      VIPP=AMAX1(0.0,VELIP1-VB)

```

```

VIP=AMAX1(0.0,VEL1-VB) 00187300
VIM=AMIN1(0.0,VEL1-VB) 00187400
DENOMT=DENOM+(VIPP-VIM)/DXXI 00187500
ROIM1=RODRY(I-1,J) 00187600
ROIP1=RODRY(I+1,J) 00187700
IF(I.EQ.ISRCP1) ROIM1=0.0 00187800
C 00187900
IF(I.EQ.1) THEN 00188000
  IF((J.EQ.2).OR.(J.EQ.4).OR.(J.EQ.11)) THEN 00188100
    ROIM1 = AMAX1(0.,WRO(J)) 00188200
  ELSE 00188300
    ROIM1 = 0. 00188400
  ENDIF 00188500
ENDIF 00188600
C 00188700
IF(I.EQ.ISRC) DENOMT=DENOMT-VIPP/DXXI 00188800
ROVE=(ROIM1*VIP-VIP*ROIP1)/DXXI 00188900
C 00189000
SOURCES IN THE INTERFACE NODE DUE TO ABLATION 00189100
C 00189200
SRC=0.0 00189300
IF(I.NE.ISRC) GO TO 838 00189400
SRC=-VEL*RODRY(I+1,J) 00189500
SRC=AMAX1(0.0,SRC/DXXI) 00189600
838 CONTINUE 00189700
XNUM=ROVE+RODRY(I,J)/DT + SRC 00189800
RODRY(I,J)=XNUM/(DENOMT) 00189900
IF (RODRY(I,J) .LE. 0.0) RODRY(I,J)=0.0 00190000
IF(I.EQ.1) PFLUX(J)=-VIM*RODRY(I,J)-VIP*ROIM1 00190100
C * + GAMBL(1)*(RODRY(I,J)-ROIM1)/DXXI 00190200
IF(I.GT.ISRC) GO TO 31 00190300
VOLFR=RODRY(I,J)/RHOL(J) 00190400
COR1=COR1+VOLFR 00190500
CPBL=SP(J,2) 00190600
IF(CPBL.EQ.0.0) CPBL=CPCON 00190700
IF(J.EQ.3 .AND. VOLFR.LT.0.25) VOLFR=0.0 00190800
TKBL=TKBL+VOLFR*(TKMAT(J)+GAMBL(I+1)*RHOL(J)*CPBL) 00190900
IF(I.EQ.ISRC) DIFO=GAMBL(I+1) 00191000
31 CONTINUE 00191100
C 00191200
C 00191300
C END OF MATRL DO LOOP 00191400
C 00191500
IF(I.GT.ISRC) GO TO 32 00191600
TKDRY(I)=(ALPHAL(I)*TKBL+(1.-ALPHAL(I))*0.03)*ALPHAL(I) 00191700
F=1.-COR1 00191800
IF(ABS(F).LT.0.00002) GO TO 32 00191900
IF(ITNR.EQ.1) GO TO 360 00192000
DFDV=(F-FOLD)/(V(I)-VOLD) 00192100
FOLD=F 00192200
VOLD=V(I) 00192300
VNEW=V(I)-F/DFDV 00192400
CHG=1. 00192500
IF(VNEW.NE.0.0) CHG=ABS((VNEW-V(I))/VNEW) 00192600
IF(CHG.GT.0.5) V(I)=(V(I)+VNEW)*0.5 00192700
IF(CHG.LE.0.5) V(I)=VNEW 00192800
IF(ITNR.GT.10) V(I)=(VOLD+VNEW)*0.5 00192900
GO TO 205 00193000
360 FOLD=F 00193100
VOLD=V(I) 00193200
V(I)=V(I)*1.05-1.E-05 00193300
GO TO 205 00193400
32 CONTINUE 00193500
33 CONTINUE 00193600
RETURN 00193700
90 FORMAT(1H ,6E11.4) 00193800
END 00193900
*/ *****00193900
*/ * SUBROUTINE REACT *00194000
*/ *****00194100
SUBROUTINE REACT(TIME,DT,WTEMP,I,ICALC,RLIQ) 00194200
C 00194300
*CALL,/IMPLIC/ 00194400
*CALL,/SLCOM1/ 00194500
*CALL,/SLCOM2/ 00194600
*CALL,/NACON/ 00194700
C 00194800
DIMENSION RLIQ(51,8) 00194900
C 00195000

```

```

C      LIQUID NAOH - SLURRY REACTIONS (SLAM)                                00195100
C                                                                                   00195200
C      SOURCES DUE TO GAS RELEASE                                             00195300
C                                                                                   00195400
      EXH2(I)=0.0                                                                00195500
      IF(ICALC.EQ.0 .OR. I.NE.ISRC) GO TO 70                                   00195600
      DO 60 JJ=1,2                                                                00195700
      IF(JJ.EQ.1) J=2                                                             00195800
      IF(JJ.EQ.2) J=11                                                            00195900
      IF(J.EQ.2) SRC=H2OFLX+H2OSRC                                              00196000
      IF(J.EQ.11) SRC=CO2SRC                                                      00196100
      SRC=AMAX1(0.0,SRC/(DEL1*(XZ1(I+1)-XZ1(I))))                             00196200
      XNUM=RODRYI(I,J)/DT+SRC                                                    00196300
      DENOM=1./DT                                                                  00196400
      RODRYI(I,J)=XNUM/DENOM                                                      00196500
60    CONTINUE                                                                    00196600
70    CONTINUE                                                                    00196700
      IF(ICALC.EQ.0) TREAT=WTMP                                                  00196800
      IF(ICALC.NE.0) TREAT=TDRY(I)                                               00196900
C                                                                                   00197000
C      NEXT THE CHEMICAL REACTIONS                                             00197100
C                                                                                   00197200
C                                                                                   00197300
      DO 121 IR=1,NREACT                                                           00197400
      IF(SAK(IR).EQ.0.0) GO TO 121                                                00197500
      ICNTP=NPROD(IR)                                                             00197600
      IF(NRCT(IR).GT.2) GO TO 100                                                00197700
C                                                                                   00197800
C      2 REACTANTS                                                             00197900
C                                                                                   00198000
      XX1=RODRYI(I,MRR(IR,1))/(W(MRR(IR,1))*COFR(IR,MRR(IR,1)))                00198100
      XX2=RODRYI(I,MRR(IR,2))/(W(MRR(IR,2))*COFR(IR,MRR(IR,2)))                00198200
      FMIN=AMIN1(XX2,XX1)                                                         00198300
      IF(FMIN.LE.1.E-3) GO TO 121                                                00198400
      FMAX=AMAX1(XX2,XX1)                                                         00198500
      AKK=AMIN1(AKF(IR,1),SAK(IR)*EXP(-EK(IR)/(1.98*TREAT)))                   00198600
      IF(RODRYI(I,3)/RHOL(3).GT.AKF(IR,3)) AKK=AKK*AKF(IR,2)                  00198700
      IF(AKK.LT.1.E-03) GO TO 121                                                00198800
      BB=- (XX1+XX2+1./ (AKK*DT))                                                 00198900
      CC=XX2*XX1                                                                  00199000
      TERM=BB*BB-4.*CC                                                            00199100
      XX=0.5*(-BB+SQRT(TERM))                                                      00199200
      XX2=0.5*(-BB-SQRT(TERM))                                                    00199300
      IF(XX.GT.FMIN) XX=XX2                                                       00199400
      IF(FMAX/FMIN.GT.100.) XX=FMIN*(1.-EXP(-AKK*DT*FMAX))                     00199500
      RLIQ(I,IR)=XX/DT                                                            00199600
      DO 8855 IK=1,2                                                              00199700
      RODRYI(I,MRR(IR,IK))=RODRYI(I,MRR(IR,IK))-XX*W(MRR(IR,IK))              00199800
      * *COFR(IR,MRR(IR,IK))                                                      00199900
8855  CONTINUE                                                                    00200000
      DO 30 IP=1,ICNTP                                                            00200100
      MM=MPP(IR,IP)                                                                00200200
      RODRYI(I,MM)=RODRYI(I,MM)+XX*W(MM)*COFP(IR,MM)                           00200300
30    CONTINUE                                                                    00200400
      EXH2(I)=EXH2(I)+XX*W(4)*COFP(IR,4)/DT                                     00200500
335  CONTINUE                                                                    00200600
      GO TO 121                                                                    00200700
C                                                                                   00200800
C      3 REACTANTS                                                             00200900
C                                                                                   00201000
100  CONTINUE                                                                    00201100
      IF(NRCT(IR).GT.3) WRITE(6,966) NRCT(IR)                                    00201200
966  FORMAT('  CODE NOT SET UP FOR ',I5,'  REACTANTS')                          00201300
      XA=RODRYI(I,MRR(IR,1))/(COFR(IR,MRR(IR,1))*W(MRR(IR,1)))                  00201400
      XB=RODRYI(I,MRR(IR,2))/(COFR(IR,MRR(IR,2))*W(MRR(IR,2)))                  00201500
      XC=RODRYI(I,MRR(IR,3))/(COFR(IR,MRR(IR,3))*W(MRR(IR,3)))                  00201600
      FMIN=AMIN1(XA,XB,XC)                                                         00201700
      FMAX=AMAX1(XA,XB,XC)                                                         00201800
      IF(FMIN.LE.1.E-3) GO TO 121                                                00201900
      IF(XA.GT.FMIN .AND. XA.LT.FMAX) FMID=XA                                     00202000
      IF(XB.GT.FMIN .AND. XB.LT.FMAX) FMID=XB                                     00202100
      IF(XC.GT.FMIN .AND. XC.LT.FMAX) FMID=XC                                     00202200
      AKK=AMIN1(AKF(IR,1),SAK(IR)*EXP(-EK(IR)/(1.98*TREAT))*FMAX)               00202300
      IF(RODRYI(I,3)/RHOL(3).GT.AKF(IR,3)) AKK=AKK*AKF(IR,2)                  00202400
      IF(AKK.LT.1.E-03) GO TO 121                                                00202500
      BB=- (FMIN+FMID+1./ (AKK*DT))                                               00202600
      CC=FMID*FMIN                                                                 00202700
      TERM=BB*BB-4.*CC                                                            00202800

```

```

XX=0.5*(-BB+SQRT(TERM))                                00202900
XX2=0.5*(-BB-SQRT(TERM))                               00203000
IF(XX.GT.FMIN) XX=XX2                                  00203100
IF(FMAX/FMIN.GT.100.) XX=FMIN*(1.-EXP(-AKK*DT*FMAX))   00203200
DMODT=(XX)/DT                                          00203300
RLIQ(I,IR)=DMODT                                       00203400
DO 5599 IK=1,3                                          00203500
RODRYI(I,MRR(IR,IK))=RODRYI(I,MRR(IR,IK))-XX*W(MRR(IR,IK)) 00203600
*   *COFR(IR,MRR(IR,IK))                               00203700
5599 CONTINUE                                           00203800
DO 50 IP=1,ICNTP                                       00203900
MM=MPP(IR,IP)                                          00204000
RODRYI(I,MM)=RODRYI(I,MM)+XX*W(MM)*COFP(IR,MM)       00204100
50 CONTINUE                                             00204200
EXH2(I)=EXH2(I)+COFP(IR,4)*W(4)*DMODT                00204300
121 CONTINUE                                           00204400
RETURN                                                  00204500
END                                                      00204600
*/ *****00204700
*/ * SUBROUTINE HEAT *00204800
*/ *****00204900
SUBROUTINE HEAT (ICALC,RLIQ)                            00205000
C                                                         00205100
*CALL,/IMPLIC/                                         00205200
*CALL,/SICOM1/                                         00205300
*CALL,/SICOM2/                                         00205400
*CALL,/NACON/                                          00205500
C                                                         00205600
DIMENSION RLIQ(51,8)                                   00205700
C                                                         00205800
C   DEFINE COFS IN THE CHEMICAL REACTION EQN          00205900
C                                                         00206000
QV(1)=0.0                                              00206100
IF(ICALC.NE.0) QCV=0.0                                 00206200
IMAX=ISRC                                              00206300
IF(ICALC.EQ.0) IMAX=NPOOL                              00206400
DO 100 I=1,IMAX                                       00206500
SUMR=0.0                                               00206600
DO 10 IR=1,NREACT                                     00206800
IF(RLIQ(I,IR).LT.1.E-05) GO TO 10                    00206900
DO 30 M=1,14                                          00207000
IF(COFR(IR,M).EQ.0.0) GO TO 20                       00207100
C   I ENTHALPY OF DRY REGION- PROP                    00207200
CALL SLENTHTDRY(I,ENTH,M,IPHASR(IR,M))                00207300
SUMR=SUMR+ENTH*RLIQ(I,IR)*COFR(IR,M)                  00207400
20 CONTINUE                                           00207500
IF(COFP(IR,M).EQ.0.0) GO TO 30                       00207600
CALL SLENTHTDRY(I,ENTH,M,IPHASP(IR,M))                00207700
SUMP=SUMP+ENTH*RLIQ(I,IR)*COFP(IR,M)                  00207800
30 CONTINUE                                           00207900
10 CONTINUE                                           00208000
IF(ICALC.EQ.0) GO TO 99                               00208100
QV(I+1)=SUMR-SUMP                                     00208200
QCV=QCV+(QV(I+1)+QV(I))*0.5*DEL1*XZI(2)              00208300
GO TO 100                                              00208400
99 QVP(I)=SUMR-SUMP                                    00208500
100 CONTINUE                                           00208600
IF(ICALC.NE.0) QCV=QCV+0.5*QV(ISRCP1)*DEL1*XZI(2)    00208700
RETURN                                                  00208800
END                                                      00208900
*1 RB-CONCR.397                                        00209000
*/ *****00209100
*/ * SLOC : NEWLY ESTABLISHED SUBROUTINES *00209200
*/ * 1. CONCRM 2. SLAM 3. CONCP  *00209300
*/ * 4. TRANA 5. TRANB 6. TRANC *00209400
*/ * 7. LAYMOD *00209500
*/ *****00209600
*DECK,RB-CONCM                                         00209700
*/ *****00209800
*/ * SUBROUTINE CONCRM *00209900
*/ *****00210000
SUBROUTINE CONCRM                                       00210100
1 (ATA,ATF,DIAG,FJAC,FVEC,XSQL,WKR,IWRK,CKM,CCP,HQ,KFLPL,KSPPL, 00210200
2 ONAMES,DMHTPL,DTYP,COFP,CQ3P,CA,CDZ,CDZO,CHA,CHAO,CRM,CRMO, 00210300
3 TBCL,TBCU,CT,CTA,CTAO,VCON,PMA,CTDX,CTDY,OCTBNX,OCTBNY,OCTBNM, 00210400
4 ICTBFL,ICTBNP,KZO,NNP,ASRE,ASRM,PSRE,PSRM,RISRE,RISRM,RIMA, 00210500
5 RIQ3P,CSRE,CSRM,NW,HL,CMAS,NIND,NMAT,N,PT,PRM,PIA,PTAO,PDZ, 00210600

```

```

        6 PQ3P,AQ3P,RIT,RITA,RITAO,RIDZ,RIRM)                                00210700
C ... INCLUDE 'IMPLIC.CMN'                                                00210800
*CA /INPLIC/                                                                00210900
C                                                                            00211000
        REAL ATA(N,*),ATF(*),DIAG(*),FJAC(N,*),FVEC(*),XSOL(*),WRK(*)      00211100
        REAL CKM(*),CCP(*),HQ(*),CQFP(*),CQ3P(*),CA(*),CDZ(*),CDZO(*)     00211200
        REAL CMA(NMAT,*),CMAO(NMAT,*),CRM(*),CRMO(*),TBCL(*),TBCU(*)      00211300
        REAL CT(*),CTA(*),CTAO(*),VCON(*),PHM(NMAT,*),PRM(*),PT(*)        00211400
        REAL CTBX(*),CTBY(*),PTA(*),PTAO(*),PDZ(*),PQ3P(*),AQ3P(*)       00211500
        REAL ASRE(NIND,NMAT,*),ASRM(NIND,NMAT,*),PSRE(NIND,NMAT,*),      00211600
        REAL PSRN(NIND,NMAT,*),RISRE(NIND,NMAT,*),RISRM(NIND,NMAT,*),    00211700
        REAL CSRE(NIND,NMAT,*),CSRM(NIND,NMAT,*),                        00211800
        REAL RIMA(NMAT,*),RIQ3P(*),HL(*),CMAS(*)                          00211900
        REAL HW(*),RIT(*),RITA(*),RITAO(*),RIDZ(*),RIRM(*)               00212000
        INTEGER IWRK(*),KFLPL(*),KSPPL(*),ICTBFL(*),ICTBNP(*)            00212100
        INTEGER KZO(*),NNP(*),NIND,NMAT,N,ISLCN(17),ICNSL(54)            00212200
        REAL ROPOOL(17),PFLUX(17),TOTMS(17),ROINT(17),WRO(17)            00212300
        REAL V(51),RLIQ(51,8)                                             00212400
        CHARACTER*8 ONAMES(*),OMMTPL(*),OTYP(*)                          00212500
        CHARACTER*8 OCTBNX(*),OCTBNY(*),OCTBNM(*)                        00212600
        EXTERNAL CONEQN                                                    00212700
C ... INCLUDE 'ATMOS.CMN'                                                  00212800
C ... INCLUDE 'CNCRT.CMN'                                                  00212900
C ... INCLUDE 'CONST.CMN'                                                 00213000
C ... INCLUDE 'FLAGS.CMN'                                                 00213100
C ... INCLUDE 'HCOEF.CMN'                                                 00213200
C ... INCLUDE 'MATER.CMN'                                                 00213300
C ... INCLUDE 'NACON.CMN'                                                 00213400
C ... INCLUDE 'PTRS.CMN'                                                  00213500
C ... INCLUDE 'REBCOM.CMN'                                                00213600
C ... INCLUDE 'REBLAY.CMN'                                                00213700
C ... INCLUDE 'SOURC.CMN'                                                 00213800
C ... INCLUDE 'SPACE.CMN'                                                 00213900
C ... INCLUDE 'TAPE.CMN'                                                  00214000
C ... INCLUDE 'TIME.CMN'                                                  00214100
C                                                                            00214200
C *** ATMOSPHERIC GAS PROPERTIES/PARAMETERS *** RI=KKM                    00214300
C                                                                            00214400
*CA /ATMOS/                                                                00214500
*CA /CONCRT/                                                                00214600
*CA /CONST/                                                                00214700
*CA /FLAGS/                                                                00214800
*CA /OUTFLG/                                                                00214900
*CA /HCOEF/                                                                00215000
*CA /MATER/                                                                00215100
*CA /NACON/                                                                00215200
*CA /PTRS/                                                                00215300
*CA /REBCOM/                                                                00215400
*CA /REBLAY/                                                                00215500
*CA /SOURCE/                                                                00215600
*CA /SPACE/                                                                00215700
*CA /TAPE/                                                                00215800
*CA /TIME/                                                                00215900
*CA /SLCOM1/                                                                00216000
*CA /SLCOM2/                                                                00216100
C                                                                            00216200
        LOGICAL ACTIV,ACTIVH,ABLAFL                                       00216300
        DATA OROUT/'CONCRM' /                                           00216400
C        CONVERSION ARRAY FROM SLAM MATERIAL NUMBER TO CONTAIN MATERIAL    00216500
C        NUMBER                                                            00216600
        DATA ISLCN/25,8,11,4,21,22,47,19,41,52,6,26,53,40,54,50,15/     00216700
C        CONVERSION ARRAY FROM CONTAIN MATERIAL NUMBER TO SLAM MATERIAL    00216800
C        NUMBER                                                            00216900
        DATA ICNSL/3*0,4,0,11,0,2,2*0,3,4*0,50,15,0,8,0,5,6,0,0,1,12,13*0 00217000
        1,14,9,5*0,7,4*0,10,13,15/                                       00217100
C ... INCLUDE 'DEBUGO.CMN'                                                00217200
*CA /DEBUGO/                                                                00217300
C                                                                            00217400
C                                                                            00217500
C                                                                            00217600
C *                                                                           *00217700
C *           THE PHYSICS CONTROLLING ROUTINE FOR A CONCRETE LAYER        *00217800
C *                                                                           *00217900
C *                                                                           *00218000
C                                                                            00218100
C                                                                            00218200
C        OBLOC = 'CONCRT'                                                 00218300
        CALL LOAD ( OBLOC )                                               00218400

```

```

C                                     00218500
C   RESET   OUTPUT FLAGS                                     00218600
C                                     00218700
C   SLMOUT = .FALSE.                                       00218800
C   WATOUT = .FALSE.                                       00218900
C                                     00219000
C   IF (JINT .EQ. 0) LNFAIL = .TRUE.                       00219100
C                                     00219200
C                                     00219300
C   SWITCH   SODIUM CONCRETE INTERACTION OR WATER  RELEASE 00219400
C                                     00219500
C   IF (LNFAIL.AND.NFCONC) THEN                             00219600
C       NFNCONC = .FALSE.                                   00219700
C       NFNACO = .TRUE.                                    00219800
C       ISRC = IH(LRNNP+KREG)/4                             00219900
C       DO 10 I = 1,ISRC                                   00220000
C           DO 10 J = 1,NSLM                               00220100
C               RODRY(I,J)=0.0                             00220200
C               IF (J .EQ. 3) RODRY(I,J)=RHOL(3)          00220300
C               CONTINUE                                   00220400
10          CALL TRANB                                     00220500
C   ENDIF                                                  00220600
C                                     00220700
C                                     00220800
C                                     00220900
C       TABLE MODIFICATIONS TO THE HEAT TRANSFER COEFFICIENT 00221000
C                                     00221100
C                                     00221200
C                                     00221300
C   IF( HTTBFL ) THEN                                       00221400
C                                     00221500
C       IF( OCTBNX(IHTTBL).EQ.'TIME' ) THEN                00221600
C           TIME= PTIME + PDT/2.0                          00221700
C           CALL TABLE                                     00221800
C       1 ( IHTTBL,CTBX,CTBY,ICTBFL,ICTBNP, ACTIVH,TIME,HTVAL ) 00221900
C                                     00222000
C       ELSE IF( OCTBNX(IHTTBL).EQ.'TEMP' ) THEN          00222100
C           TEMP= ( TBCU(KREG) + CT( (KZO(KREG)+NMP(KREG)-1)*3 ) ) *0.5 00222200
C           CALL TABLE                                     00222300
C       1 ( IHTTBL,CTBX,CTBY,ICTBFL,ICTBNP, ACTIVH,TEMP,HTVAL ) 00222400
C                                     00222500
C   TERMINATE : IF( OCTBNX(IHTTBL).EQ.'XXXXXXX' ) THEN   00222600
C                                     00222700
C       ENDIF                                             00222800
C                                     00222900
C   TERMINATE : IF( HTTBFL ) THEN                          00223000
C                                     00223100
C       ENDIF                                             00223200
C                                     00223300
C   IF THE OVERLYING STRUCTURE MATCHES THE CHARACTER NAME OF THE TABEL 00223400
C       THEN RESET THE DEFAULT HEAT TRANSFER COEFFICIENT 00223500
C                                     00223600
C       IF( ACTIVH .AND. OCTBNM(IHTTBL).EQ.OTYP(KRUP) ) THEN 00223700
C                                     00223800
C           HQ(KREG+1)= HTVAL                              00223900
C                                     00224000
C       ELSE                                              00224100
C                                     00224200
C           IF( KREG.EQ.(NREG-1) ) THEN                    00224300
C               LFCOEF= HTUPLY                             00224400
C           ELSE                                           00224500
C               LFCOEF= HTLAY                              00224600
C           ENDIF                                          00224700
C           IF( LFCOEF ) THEN                               00224800
C               HQ(KREG+1)= HL(2*KREG)*HL(2*KREG+1) / (HL(2*KREG)+HL(2*KREG+1)) 00224900
C           ELSE                                           00225000
C               HQ(KREG+1)= 0.0                            00225100
C           ENDIF                                          00225200
C                                     00225300
C   TERMINATE : IF( ACTIVH .AND. OCTBNM(IHTTBL).EQ.OTYP(KREG+1) ) THEN 00225400
C                                     00225500
C       ENDIF                                             00225600
C                                     00225700
C   ACTIV = .TRUE.                                         00225800
C   KPOOL = JCONC + JINT + 1                               00225900
C   KINT = JCONC + 1                                       00226000
C *****00226100
C *   SODIUM   CONCRETE INTERACTIONS                      *00226200

```

```

C *****00226300
C      IF( LNFAIL .AND. NFNACO .AND. NRFAS) THEN
C      00226400
C      00226500
C CHECK ON INTERACTION CONSTRAINTS : SODIUM H > 0.0
C      00226600
C      PTIME > TNATO
C      00226700
C      PTIME < TNATN
C      00226800
C      00226900
C      KPOOL= JCONC + JINT + 1
C      00227000
C      00227100
C      IF( PMA(LNAL,KZO(KPOOL)).GT.0.0 ) THEN
C      00227200
C      ACTIV= .TRUE.
C      00227300
C      ELSE
C      00227400
C      ACTIV= .FALSE.
C      00227500
C      ENDIF
C      00227600
C      00227700
C TERMINATE :      IF( LNFAIL .AND. NFNACO .AND. NRFAS ) THEN
C      00227800
C      00227900
C      00228000
C      00228100
C      00228200
C      00228300
C      00228400
C      00228500
C      IF (.NOT.ACTIV) RETURN
C      00228600
C      00228700
C      00228800
C      KZ=KZO(KREG)
C      00228900
C      AREA = H(LRCA+KZ)
C      00229000
C      PINFXT=H(LPGAS+NCELL)
C      00229100
C      TINFXT=H(LTGAS+NCELL)
C      00229200
C      TOTMAS=0.0
C      00229300
C *
C * DETERMINATION OF DENSITY IN LAYER
C *
C * 00229400
C * 00229500
C * 00229600
C      IF ( NFNACO ) THEN
C      00229700
C      KZ = KZO(KPOOL)
C      00229800
C      J = 3*(KZ-1) + 1
C      00229900
C      TPOOL = PT(J)
C      00230000
C      DTRDT=(PTA(KZO(KPOOL))-PTAD(KZO(KPOOL)))/PDT
C      00230100
C      WDTDT=DTRDT
C      00230200
C      DO 55 J = 1, NMAT
C      00230300
C      TOTMAS = TOTMAS + PMA(J,KZ)
C      00230400
C      CONTINUE
C      00230500
C      PRM(KZ) = TOTMAS/(AREA*PDZ(KZ))
C      00230600
C      ELSE
C      00230700
C      KZ = KZO(KINT)
C      00230800
C      J = 3*(KZ-1) + 1
C      00230900
C      TINT = RIT(J)
C      00231000
C      DTIDT=(RITA(KZO(KINT))-RITAD(KZO(KINT)))/PDT
C      00231100
C      WDTDT=DTIDT
C      00231200
C      DO 56 J = 1, NMAT
C      00231300
C      TOTMAS = TOTMAS + RIMA(J,KZ)
C      00231400
C      CONTINUE
C      00231500
C      RIRM(KZ) = TOTMAS/(AREA*RIDZ(KZ))
C      00231600
C      00231700
C      ENDIF
C      00231800
C      SUM=0.0
C      00231900
C      DO 50 I=1,NSLM
C      00232000
C      IF(I.NE.15) THEN
C      00232100
C      II = IMATER(ISLCN(I)+1)
C      00232200
C      IF( NFNACO ) THEN
C      00232300
C      ROPOOL(I)=PMA(II,KZO(KPOOL))/(AREA*PDZ(KZO(KPOOL)))
C      00232400
C      WRO (I) = ROPOOL(I)
C      00232500
C      SUM = SUM + ROPOOL(I)
C      00232600
C      ELSE
C      00232700
C      ROINT(I)=RIMA(II,KZO(KINT))/(AREA*RIDZ(KZO(KINT)))
C      00232800
C      WRO (I) = ROINT(I)
C      00232900
C      SUM = SUM + ROINT(I)
C      00233000
C      ENDIF
C      00233100
C      ENDIF
C      00233200
C 50 CONTINUE
C      00233300
C      IF( NFNACO ) THEN
C      00233400
C      II = KZO(KPOOL)
C      00233500
C      ROPOOL(15)=PRM(II)-SUM
C      00233600
C      WRO(15)=ROPOOL(15)
C      00233700
C      WATOUT=.FALSE.
C      00233800
C      SLNDOUT=.TRUE.
C      00233900
C      WTEMP = TPOOL
C      00234000
C      ELSE

```

```

        II = KZ0(KINT)                                00234100
        ROINT(15)=RIRM(II)-SUM                         00234200
        WRO(15)=ROINT(15)                             00234300
        WATOUT=.TRUE.                                 00234400
        SLMOUT=.FALSE.                               00234500
        WTEMP = TINT                                  00234600
    ENDIF                                             00234700
C                                                    00234800
C *****00234900
C * LOAD STORAGE ARRAY CH,ICH WITH COMMON BLOCK      *00235000
C *****00235100
    CALL TRANA                                       00235200
C                                                    00235300
    TSO = TS                                         00235400
    TOTN20 = 0.0                                     00235500
    TOTCO2 = 0.0                                     00235600
    TOTN2G = 0.0                                     00235700
    TOTQNP = 0.0                                     00235800
    TOTQNI = 0.0                                     00235900
    TOTQNA = 0.0                                     00236000
    DO 60 I = 1,NSLM                                 00236100
    60 TOTMS(I) = 0.0                                 00236200
C                                                    00236300
    SLTIME = PTIME                                   00236400
    ISLAF = 0                                         00236500
    WDT = PDT                                         00236600
100 CONTINUE                                       00236700
C                                                    00236800
C DETERMINATION OF TIME STEP                       00236900
C                                                    00237000
    CALL STIME (SLTIME,WDT,ISLAF,WTEMP,TSO,WDTDT)    00237100
C *****00237200
C * *00237300
C * SOLVE THE CONSERVATION EQUATIONS FOR THE CONCRETE LAYER *00237400
C * *00237500
C *****00237600
C                                                    00237700
    TOTM= 0.0                                        00237800
    DO 300 KZ=KZ0(KREG),(KZ0(KREG)+MNP(KREG)-1)      00237900
    DO 455 J=2,NMAT                                  00238000
    TOTM= TOTM + CMA(J,KZ)                          00238100
455 CONTINUE                                       00238200
300 CONTINUE                                       00238300
C                                                    00238400
    KZ = KZ0(KPOOL)                                  00238500
    J = 3*(KZ-1) +1                                 00238600
    TPOOL = PT(J)                                    00238700
C                                                    00238800
    IF( TOTM.GT.0.0 ) CALL SLAM                      00238900
1 ( SLTIME,SDT,WTEMP,WRO,PINFXT,TINFXT,WDTDT,RLIQ,PFLUX) 00239000
C                                                    00239100
    DTSDT = (TS-TSO)/SDT                            00239200
    TSO = TS                                         00239300
C                                                    00239400
C CALCULATION OF MASS AND ENERGY WHICH IS RELEASED TO POOL OR INTERM 00239500
C *00239600
C *00239700
C *00239800
    IF (NFNACO) THEN                                00239900
    TOTN2G = TOTN2G + PFLUX(4) * SDT * AREA          00240000
    TOTCO2 = TOTCO2 + PFLUX(11) * SDT * AREA        00240100
    TOTQNP = TOTQNP - QCONO * SDT * AREA            00240200
    TOTQNA = TOTQNA + SP(4,3)*(WTEMP-TINFXT)*PFLUX(4)*SDT*AREA 00240300
1 + SP(11,3)*(WTEMP-TINFXT)*PFLUX(11)*SDT*AREA    00240400
    DO 320 I = 1,NSLM                                00240500
    IF (I.EQ.4 .OR. I.EQ.11) GO TO 320              00240600
    TOTMS(I) = TOTMS (I) + PFLUX(I)*AREA*SDT        00240700
320 CONTINUE                                       00240800
    ELSE                                             00240900
    TOTQNI = TOTQNI - QCONO * SDT * AREA            00241000
    ENDIF                                           00241100
C                                                    00241200
    IF (ISLAF.EQ.0) GO TO 100                       00241300
    CALL TRANB                                       00241400
C *****00241500
C * *00241600
C * AT THIS POINT THE MASS SOURCES HAVE BEEN DETERMINED, NOW DEFINE *00241700
C * THE NEW MASS TOTALS IN THE LAYER *00241800
C * *00241800

```



```

C *****00241900
C
CALL TRANC(CA,CDZ,CMA,CT,CTA,CRH,KZO,NNP,ISLCN,PSRM,PSRE,ASRM, 00242000
1 ASRE,TOTH2G,TOTH2O,TOTCO2,TOTQNP,TOTQNJ,TOTQNA,TOTMS,PQ3P,AQ3P, 00242100
2 RISRM,RIQ3P,NIND,NMAT) 00242200
C 00242300
350 CONTINUE 00242400
C 00242500
C ***** 00242600
C 00242700
C IH(LRNNP+KREG)=NWET+NDRY-2 00242800
C 00242900
C 00243000
C 00243100
C RESET THE NEXT LAYERS LOWER BOUNDARY CONDITION 00243200
C 00243300
TBCL(KREG+1)= CT( 3*(KZO(KREG)+NNP(KREG)-1) ) 00243400
C 00243500
C TERMINATE : IF( ACTJV .AND. DNACO.EQ.'SLAM' ) THEN 00243600
C 00243700
CALL UNLOAD ( DBLOC ) 00243800
RETURN 00243900
END 00244000
*/ *****00244100
*/ * SUBROUTINE SLAM *00244200
*/ *****00244300
SUBROUTINE SLAM(TIME,DT,WTEMP,WRO,PINFXT,TINFXT,WDTDT, 00244400
1 RLIQ,PFLUX) 00244500
C 00244600
*CALL,/IMPLIC/ 00244700
*CALL,/SLCOM1/ 00244800
*CALL,/SLCOM2/ 00244900
*CALL,/NACON/ 00245000
*CALL,/REBLAY/ 00245100
*CALL,/CONST/ 00245200
C 00245300
DIMENSION COMP1(51),TOK(51),SUMTRM(51),FX(51),AMAT(12,12) 00245400
1 ,CCOL(12),XVAR(22),AMO(12),SINT(51),WRO(*) 00245500
2 ,PFLUX(*),RLIQ(51,8) 00245600
C ***** 00245700
C * 00245800
C * PHYSICS OF THE CONCRETE * 00245900
C * * 00246000
C ***** 00246100
C 00246200
C UPDATE BEGINNING OF TIME STEP VALUES 00246300
C 00246400
DO 55 N=1,NTERM 00246500
55 AMO(N)=AM(N) 00246600
DEL10=DEL1 00246700
TRO=WTEMP 00246800
H2OFLO=H2OFLX 00246900
VELOLD=VEL 00247000
VEL=0.5*(VELOLD+VELN) 00247100
IF(ABS(VEL).LT.1.E-10) VEL=0.0 00247200
DIST=DIST-VEL*DT 00247300
IMPL=0 00247400
EPP=0.5 00247500
66 CONTINUE 00247600
C 00247700
C *****00247800
C * WATER RELEASE AND HEAT TRANSFER CALCULATION *00247900
C *****00248000
C 00248100
CALL CONEQS(TIME,DT,WTEMP,PINFXT,TINFXT,WDTDT,DEL10,TRO,H2OFLO, 00248200
1 EPP,ERR1,COMP1,TOK,SUMTRM,FX,AMAT,CCOL,XVAR,AMO,SINT,VELOLD) 00248300
C 00248400
DBL10T=(DBL10N+VEL+DBL10T)*0.5 00248500
DEL1=DEL10+(DBL10T)*DT 00248600
IF(ERR1.GT.1.E-03) GO TO 66 00248700
C 00248800
C *****00248900
C * WE MUST NOW DECIDE WHICH CONCRETE MODEL TO USE *00249000
C * ACCORDING TO ACCIDENT SCENARIO *00249100
C *****00249200
IF ( NFNACO ) THEN 00249300
C ***** 00249400
C 00249500
C SODIUM CONCRETE INTERACTIONS 00249600

```



```

LC17 = LC16 + NTERN*NDRY          00257500
LC18 = LC17 + NTERN*NTERM        00257600
LC19 = LC18 + 51                  00257700
LC20 = LC19 + 51                  00257800
LC21 = LC20 + 51                  00257900
LC22 = LC21 + 51                  00258000
LC23 = LC22 + 51                  00258100
CALL IMOVE (LPOINT(1),ICH(LICH+9),23) 00258200
RETURN                              00258300
END                                  00258400
C                                    00258500
*/ *****00258600
*/ *      SUBROUTINE TRANA          *00258700
*/ *****00258800
      SUBROUTINE TRANA              00258900
C                                    00259000
C      THIS ROUTINE LOADS STORAGE ARRAY CH,ICH 00259100
C      WITH CONCRETE COMMON BLOCK      00259200
C                                    00259300
*CA /IMPLIC/                        00259400
*CA /SPACE/                          00259500
*CA /PTRS/                            00259600
*CA /SLCOM1/                          00259700
C                                    00259800
C      READ POINTERS                00259900
C                                    00260000
      NTM = 31                       00260100
      LICH = (NCELL-1) * NTM          00260200
      IMPL = ICH(LICH + 1)            00260300
      IMPL2 = ICH(LICH + 2)           00260400
      ISRC = ICH(LICH + 3)            00260500
      NDRY = ICH(LICH + 4)            00260600
      NWET = ICH(LICH + 5)            00260700
      NTERM = ICH(LICH + 6)           00260800
      NPOOL = ICH(LICH + 7)           00260900
      NSLM = ICH(LICH + 8)            00261000
      CALL IMOVE(ICH(LICH+9),LPOINT(1),23) 00261100
C                                    00261200
      ISRCP1 = ISRC + 1                00261300
      ISRCP3 = ISRC + 3                00261400
      ISRCP5 = ISRC + 5                00261500
C                                    00261600
C      LOAD VALUES ESTIMATED FROM CONCRETE STORAGE ARRAY CH( ) 00261700
C                                    00261800
      CALL RMOVE (CH(LC1+1),RSLCEL,JSLAMR) 00261900
C                                    00262000
      DO 200 J=1,NSLM                  00262100
      DO 200 I=1,NDRY                  00262200
      LLL = LC2+I+NDRY*(J-1)          00262300
      RODRY(I,J)=CH(LLL)              00262400
200 CONTINUE                          00262500
C                                    00262600
      DO 400 I=1,NDRY                  00262700
      TDRY(I)=CH(LC3+I)                00262800
      TKDRY(I)=CH(LC4+I)               00262900
      ALPHAL(I)=CH(LC5+I)              00263000
400 CONTINUE                          00263100
C                                    00263200
      DO 600 I=1,NWET                  00263300
      XETA(I)=CH(LC6+I)                00263400
      TWET(I)=CH(LC7+I)                00263500
      PWET(I)=CH(LC8+I)                00263600
      ROAIR(I)=CH(LC9+I)               00263700
      ROH2OL(I)=CH(LC10+I)            00263800
      ALPH(I)=CH(LC11+I)               00263900
600 CONTINUE                          00264000
C                                    00264100
      DO 800 I=1,52                    00264200
      QV(I)=CH(LC12+I)                 00264300
800 CONTINUE                          00264400
C                                    00264500
      DO 1000 I=1,NTERM                 00264600
      AM(I)=CH(LC13+I)                 00264700
      SIGN(I)=CH(LC14+I)               00264800
1000 CONTINUE                         00264900
      DO 1200 I = 1,NTERM                00265000
      DO 1200 J = 1,NDRY                00265100
      LLL = LC15+I+NTERM*(J-1)         00265200

```

```

      KKK = LC16+I+NTERM*(J-1)                                00265300
      SINX(I,J) = CH(LLL)                                     00265400
      COSX(I,J) = CH(KKK)                                    00265500
1200 CONTINUE                                                00265600
      DO 1300 I = 1,NTERM                                    00265700
      DO 1300 J = 1,NTERM                                    00265800
      MMM = LC17+J+NTERM*(J-1)                              00265900
      ZCOSI(I,J)=CH(MMM)                                     00266000
1300 CONTINUE                                                00266100
C
      DO 1400 I = 1,51                                       00266200
      XZI(I) = CH(LC18+I)                                    00266400
      WTZ(I) = CH(LC19+I)                                    00266500
      XETA(I) = CH(LC20+I)                                   00266600
      ETA(I) = CH(LC21+I)                                    00266700
      WT(I) = CH(LC22+I)                                     00266800
1400 CONTINUE                                                00266900
      RETURN                                                 00267000
      END                                                    00267100
*/ *****00267200
*/ *      SUBROUTINE  TRANB                                  *00267300
*/ *****00267400
      SUBROUTINE  TRANB
C
C      THIS ROUTINE STORES VALUES ESTIMATED
C      WITH THE CONCRETE STORAGE ARRAY CH( )
C
*CA /IMPLIC/
*CA /SPACE/
*CA /PTRS/
*CA /SLCOM1/
C
C      READ POINTERS
C
      NTM = 31
      LICH = (NCELL-1) * NTM
C
C      STORE VALUES ESTIMATED WITH THE CONCRETE STORAGE ARRAY CH( )
C
      ICH(LICH+3) = ISRC
      ICH(LICH+5) = NWET
      CALL  RMOVE (RSLCEL,CH(LC1+1),JSLAMR)
C
      DO 200 J=1,NSLM
      DO 200 I=1,NDRY
      LLL = LC2+I+NDRY*(J-1)
      CH(LLL)=RODRY(I,J)
200 CONTINUE
C
      DO 400 I=1,NDRY
      CH(LC3+I)=TDRY(I)
      CH(LC4+I)=TKDRY(I)
      CH(LC5+I)=ALPHAL(I)
400 CONTINUE
C
      DO 600 I=1,NWET
      CH(LC6+I)=XETA(I)
      CH(LC7+I)=TWET(I)
      CH(LC8+I)=PWET(I)
      CH(LC9+I)=ROAIR(I)
      CH(LC10+I)=ROH2OL(I)
      CH(LC11+I)=ALPH(I)
600 CONTINUE
C
      DO 800 I=1,52
      CH(LC12+I)=QV(I)
800 CONTINUE
C
      DO 1000 I=1,NTERM
      CH(LC13+I)=AM(I)
      CH(LC14+I)=SIGN(I)
1000 CONTINUE
C
      DO 1200 I = 1,NTERM
      DO 1200 J = 1,NDRY
      LLL = LC15+I+NTERM*(J-1)
      KKK = LC16+I+NTERM*(J-1)
      CH(LLL) = SINX(I,J)

```

```

      CH(KKK) = COSX(I,J)                                00273100
1200 CONTINUE                                          00273200
      DO 1300 I = 1,NTERM                               00273300
      DO 1300 J = 1,NTERM                               00273400
      MMM = LC17+1+NTERM*(J-1)                         00273500
      CH(MMN) = ZCOSI(I,J)                             00273600
1300 CONTINUE                                          00273700
C                                                       00273800
      DO 1400 I=1,51                                    00273900
      CH(LC18+I) = XZI(I)                              00274000
      CH(LC19+I) = WTZ(I)                              00274100
      CH(LC20+I) = XETA(I)                             00274200
      CH(LC21+I) = ETA(I)                              00274300
      CH(LC22+I) = WT(I)                               00274400
1400 CONTINUE                                          00274500
C                                                       00274600
      RETURN                                           00274700
      END                                               00274800
*/ *****00274900
*/ *          SUBROUTINE TRANC                          *00275000
*/ *****00275100
      SUBROUTINE TRANC (CA,CDZ,CMA,CT,CTA,CRM,KZO,NNP,ISLCN,PSRM,PSRE, 00275200
1  ASRM,ASRE,TOTH2G,TOTH2O,TOTCO2,TOTQNP,TOTQNI,TOTQNA,TOTMS,PQ3P,00275300
2  AQ3P,RISRM,RIQ3P,NIND,NMAT)                        00275400
C                                                       00275500
*CALL,/IMPLIC/                                       00275600
C                                                       00275700
      REAL CA(*),CDZ(*),CMA(NMAT,*),CT(*),CTA(*),CRM(*) 00275800
      REAL PSRM(NIND,NMAT,*),PSRE(NIND,NMAT,*),TOTMS(*),PQ3P(*),AQ3P(*) 00275900
      REAL ASRM(NIND,NMAT,*),ASRE(NIND,NMAT,*),          00276000
      REAL RISRM(NIND,NMAT,*),RIQ3P(*)                   00276100
      INTEGER ISLCN(17),KZO(*),NNP(*)                   00276200
*CALL,/SLCOM1/                                       00276300
*CALL,/SLCOM2/                                       00276400
*CALL,/PTRS/                                         00276500
*CALL,/WATER/                                        00276600
*CALL,/TIME/                                         00276700
*CALL,/REBCOM/                                       00276800
*CALL,/REBLAY/                                       00276900
*CALL,/SPACE/                                        00277000
C *****00277100
C * TRANSPORT SLAM CALCULATION RESULTS TO CONTAIN VARIABLE *00277200
C *****00277300
      AREA = H(LRCA+KZO(KREG))                          00277400
C WET REGION *****00277500
      NM1 = NWET - 1                                    00277600
      DO 200 I = 1,NM1                                  00277700
      J = I + 1                                         00277800
      KZ = KZO(KREG)+NWET-I-1                          00277900
C                                                       00278000
C CALCULATE NEW LAYER THICKNESS                       00278100
C                                                       00278200
      CDZ(KZ)=XETA(J)-XETA(J-1)                        00278300
C                                                       00278400
C NEW MASS IN THE LAYER                               00278500
C                                                       00278600
      DO 100 L = 1,NMAT                                  00278700
100 CMA(L,KZ) = 0.0                                     00278800
      VOL = AREA*CDZ(KZ)                                00278900
C                                                       00279000
      CMA(IMATER(ISLCN(1)+1),KZ)=RHOCN*FSI02*VOL      00279100
      CMA(IMATER(ISLCN(2)+1),KZ)=RHOCN*(FH20E+FH20B)*VOL 00279200
      CMA(IMATER(ISLCN(10)+1),KZ)=RHOCN*FCACO3*VOL    00279300
      CMA(IMATER(ISLCN(13)+1),KZ)=RHOCN*FMGCO3*VOL    00279400
      I1 = IMATER(ISLCN(15)+1)                          00279500
      CMA(I1,KZ)=RHOCN*(1.-FCACO3-FMGCO3-FSI02-FH20E-FH20B)*VOL 00279600
C                                                       00279700
C REDEFINE MIXTURE DENSITY                             00279800
C                                                       00279900
      CRM(KZ)=RHOCN                                    00280000
C                                                       00280100
C DEFINE TEMPERATURE                                  00280200
C                                                       00280300
      II = KZ*3-1                                       00280400
      IF(I .EQ. 1) THEN                                  00280500
      CT(II+1)=TDRY(NDRY)                               00280600
      CT(II)=(TDRY(NDRY)+TWET(I))/2.0                 00280700
      ELSE                                              00280800

```

```

        CT(II+1)=TWET(I-1)                                00280900
        CT(II)=(TWET(I-1)+TWET(I))/2.0                  00281000
    ENDIF                                                00281100
    CT(II-1)=TWET(I)                                    00281200
    CTA(KZ)=CT(II)                                       00281300
200 CONTINUE                                           00281400
C   DRY REGION                                          00281500
    NM1 = NDRY - 1                                       00281600
    DZZ = DEL1*XZI(2)                                    00281700
    DO 500 I = 1,NM1                                    00281800
        J = I + 1                                       00281900
        KZ = KZO(KREG)+NWET-2+NDRY-1                    00282000
C   CALCULATE NEW LAYER THICKNESS                       00282100
C   CDZ(KZ)=DZZ                                         00282200
C   NEW MASS IN THE LAYER                               00282300
C   DO 300 L = 1,NMAT                                   00282400
300 CMA(L,KZ)=0.0                                       00282500
    VOL = AREA*DZZ                                       00282600
    CRM(KZ)=0.0                                         00282700
C   DO 400 L = 1,NSLM                                   00282800
    CMA(IMATER(ISLCN(L)+1),KZ)=RODRY(I,L)*VOL           00282900
400 CRM(KZ)=CRM(KZ)+RODRY(I,L)                         00283000
C   DEFINE TEMPERATURE                                  00283100
C   II = KZ*3-1                                        00283200
    CT(II+1)=TDRY(J-1)                                   00283300
    CT(II)=(TDRY(J-1)+TDRY(J))/2.0                     00283400
    CT(II-1)=TDRY(J)                                    00283500
    CTA(KZ)=CT(II)                                       00283600
500 CONTINUE                                           00283700
C *****00283800
C *   TRANSPORT MASS AND ENERGY                       *00283900
C *****00284000
C   IF ( NFNACO ) THEN                                  00284100
    KPOOL = JCONC + JINT + 1                             00284200
    DO 700 KZ =KZO(KPOOL),KZO(KPOOL)+NNP(KPOOL)-1      00284300
        DO 600 I = 1,NSLM                                00284400
            II = IMATER(ISLCN(I)+1)                     00284500
600 PSRM(1,II,KZ)=PSRM(1,II,KZ) + TOTMS(I)/PDT/NNP(KPOOL) 00284600
700 PQ3P(KZ)=PQ3P(KZ) + TOTQNP /PDT/NNP(KPOOL)        00284700
C   ELSE                                                00284800
    KINT = JCONC + 1                                     00284900
    KPOOL = JCONC + JINT + 1                             00285000
    DO 900 KZ =KZO(KINT),KZO(KINT)+NNP(KINT)-1         00285100
        RIQ3P(KZ)=RIQ3P(KZ) + TOTQNI/PDT/NNP(KINT)    00285200
        PQ3P(KZ(KPOOL))=PQ3P(KZ(KPOOL)) + TOTQNP/PDT/NNP(KINT) 00285300
    ENDIF                                                00285400
    IF (NFCOnc) RETURN                                  00285500
C   KZATM=KZO(NREG)                                    00285600
    II = IMATER(ISLCN(4)+1)                              00285700
    ASRM(1,II,KZATM) = ASRM(1,II,KZATM) + TOT2G/PDT   00285800
    II = IMATER(ISLCN(11)+1)                            00285900
    ASRM(1,II,KZATM) = ASRM(1,II,KZATM) + TOTCO2/PDT  00286000
    AQ3P(KZATM)=AQ3P(KZATM) + TOTQNA /PDT              00286100
    RETURN                                               00286200
    END                                                  00286300
*I I-IREBEL.325                                        00286400
*/ *****00286500
*/ *   SUBROUTINE LAYMOD                               *00286600
*/ *****00286700
    SUBROUTINE LAYMOD (DTYP,KZO,NNP)                   00286800
*CA,/IMPLIC/                                           00286900
    CHARACTER*8 DTYP(*)                                00287000
    INTEGER KZO(*),NNP(*)                              00287100
*CA /NACON/                                             00287200
*CA /PTRS/                                             00287300
*CA /REBCOM/                                          00287400
*CA /REBLAY/                                          00287500
*CA /SPACE/                                           00287600

```

```

DATA OREBLA / 'REBLAY' /
C
C THIS ROUTINE READJUST THE LAYERS INFORMATION WITH LINER FAILURE
C
    II = 0
    MREG = 0
    DO 100 IREG = 1 , NREG
    KREG = IREG
    CALL LOAD ( OREBLA )
    IF ( OLAY .EQ. 'INTERM' ) THEN
        II = II + 1
        IF( II .GT. 2 ) THEN
            MREG = MREG + 1
            OTYP(MREG) = OTYP(IREG)
            KZO(MREG) = KZO(IREG) - 2
            NNP(MREG) = NNP(IREG)
            KREG = MREG
            CALL UNLOAD ( OREBLA )
        ENDIF
    ELSE
        MREG = MREG + 1
        OTYP(MREG) = OTYP(IREG)
        KZO(MREG) = KZO(IREG)
        NNP(MREG) = NNP(IREG)
        KREG = MREG
        CALL UNLOAD ( OREBLA )
    ENDIF
100 CONTINUE
C
    NREG = MREG
    JINT = JINT - 2
    RETURN
END
*/ *****
*/ * *** THE FOLLOWING CORRECTION GROUPS IS MISCELLANEOUS *00292100
*/ * MODIFICATIONS OF CONTAIN 1.03 *** *00292200
*/ *****00292300
*/ *****00292400
*/ * SLOD01 : CREATED SUBROUTINE 'LAYMOD' IS CALLED WHEN LINER FAILURE *00292500
*/ * OCCURS *00292600
*/ *****00292700
*I RB-RBCNT.53 00292800
C 00292900
C RECONSTRUCT THE LAYER STRUCTURE WHEN LINER FAILURE OCCURS 00293000
C 00293100
C IF ((JINT .GE. 2) .AND. LNFAIL .AND. LDUMMC) THEN 00293200
CALL LAYMOD (DH(LROTYP+1),IH(LRKZO+1),IH(LRNNP+1)) 00293300
ENDIF 00293400
*/ *****00293500
*/ * SLOD02 : CREATED SUBROUTINE 'CONCRM' IS CALLED IN CASE OF SLAM *00293600
*/ *****00293700
*I RB-RBCNT.160 00293800
IF ( (.NOT.NFNACO) .AND. (.NOT.NFCONC) ) THEN 00293900
*I RB-RBCNT.185 00294000
ELSE IF ( NFNACO .OR. NFCONC ) THEN 00294100
CALL CONCRM 00294200
1 ( H(LRBTMP+LTATA),H(LRBTMP+LTATF),H(LRBTMP+LTDIAG), 00294300
2 H(LRBTMP+LTFJAC),H(LRBTMP+LTFVEC),H(LRBTMP+LTXSOL), 00294400
3 H(LRBTMP+LTIWRK),IH(LIBTMP+LTIWRK), 00294500
4 H(LRBTMP+LTCKM),H(LRBTMP+LTCCP),H(LRHQ+1), 00294600
5 IH(MFLPL+1),IH(MSPPL+1),OH(INAMES+1),OH(MMTPL+1), 00294700
6 OH(LROTYP+1),H(LRBTMP+LTCQFP),H(LRBTMP+LTCQ3P), 00294800
7 H(LRCA+1),H(LRCDZ+1),H(LRCDZO+1),H(LRCMA+1), 00294900
8 H(LRCMAQ+1),H(LRBTMP+LTCRM),H(LRBTMP+LTCRMO), 00295000
9 H(LRTBCL+1),H(LRTBCU+1),H(LRCT+1),H(LRCTA+1), 00295100
1 H(LRCTAQ+1),H(LRVCON+1),H(LRPHA+1),H(LCTBX+1), 00295200
1 H(LCTBY+1),OH(LCTBNX+1),OH(LCTBNY+1),OH(LCTBNM+1), 00295300
2 IH(LCTBFL+1),IH(LCTBNP+1),IH(LRKZO+1),IH(LRNNP+1), 00295400
3 H(LRBTMP+LTASRE),H(LRBTMP+LTASRM),H(LRBTMP+LTPSRE), 00295500
4 H(LRBTMP+LTPSRM),H(LRBTMP+LTISRE),H(LRBTMP+LTISR), 00295600
5 H(LRIMA+1),H(LRBTMP+LTIQ3P),H(LRBTMP+LTCSRE), 00295700
6 H(LRBTMP+LTCSRM),H(IMW+1),H(LRBTMP+LTHL),H(LRCMAS+1),NIND, 00295800
7 NH,NMAX,H(LRPT+1),H(LRBTMP+LTPRM),H(LRPTA+1),H(LRPTAQ+1), 00295900
8 H(LRPDZ+1),H(LRBTMP+LTPQ3P),H(LRBTMP+LTAQ3P),H(LRIT+1), 00296000
9 H(LRITA+1),H(LRITAO+1),H(LRIDZ+1),H(LRBTMP+LTI RM)) 00296100
ENDIF 00296200
*/ *****00296300
*/ * SLOD03 : REDUCE COMMENT STATEMENT AND SLAM DUMMY SUBROUTINE *00296400

```

```

*/ *          AND RELOCATE SLAM SUBROUTINE          *00296500
*/ *****00296600
*/D,I-INACON.55                                00296700
      ELSE IF (OTEST.EQ.'SLAM') THEN              00296800
*/D I-INACON.61,62                                00296900
*/D I-INACON.65,80                                00297000
*/D C100K.104,115                                00297100
*/ *****00297200
*/ * SLOD04 : RELOCATION OF SLAM SUBROUTINE        *00297300
*/ *****00297400
*/I I-IREBEL.295                                00297500
      KREG = JCONC                                00297600
      CALL LOAD( OREBLA )                          00297700
      IF (LDUMNC) THEN                              00297800
        CALL SLINPT                                00297900
      ENDIF                                         00298000
*/ *****00298100
*/ * SLOE01 : ALLOCATES STORAGE ARRAY FOR SLAM CALCULATION *00298200
*/ *****00298300
*/I /SPACE/.27                                  00298400
C                                                00298500
C      *** SLAM POINTERS ***                      00298600
C                                                00298700
      REAL CH                                       00298800
      INTEGER ICH                                   00298900
      COMMON /CSLAM1/ CH(9000)                      00299000
      COMMON /CSLAM2/ ICH(200)                     00299100
*/ *****00299200
*/ * SLOE02 : EXPAND DIMENSION OF REAL STORAGE ARRAY *00299300
*/ * FOR SLAM CALCULATION                        *00299400
*/ *****00299500
*/D /SPACE/.13                                  00299600
      PARAMETER (JRSPAC=10000)                      00299700
*/ *****00299800
*/ * SLOE03 : REDUCE REGION WHICH IS NOT USED BY SLAM CALCULATION *00299900
*/ *****00300000
*/D PTRCEL.457                                  00300100
*/D PTRCEL.500,515                              00300200
*/I RB-RBCNT.60                                  00300300
      IF (LDUMNC) THEN                              00300400
        NMAX = 4                                    00300500
      ELSE                                           00300600
        NMAX = 3*NDEFLT(1) + 1                     00300700
      ENDIF                                         00300800
C                                                00300900
      LTATA = LTMTOT + NREG                          00301000
      LTATF = LTATA + NMAX*NMAX                     00301100
      LTDIAG = LTATF + NMAX                         00301200
      LTFJAC = LTDIAG + NMAX                        00301300
      LTFVEC = LTFJAC + NMAX*NMAX                  00301400
      LTXSOL = LTFVEC + NMAX                       00301500
      LTRK = LTXSOL + NMAX                         00301600
      NEND = LTRK + NMAX * 5                        00301700
C                                                00301800
      LTHL = NEND                                   00301900
      NEND = LTHL + 2*NREG                          00302000
C                                                00302100
      NTMPR = NEND                                  00302200
      LTIWRK = 1                                    00302300
      NEND = LTIWRK + NMAX + NMAX                  00302400
      NTMPI = NEND                                  00302500
*/ *****00302600
*/ * SLOF01 : ADD NEW COMMON VARIABLES TO COMMON BLOCK *00302700
*/ *****00302800
*/D /REBLAY/.24,26                               00302900
      2 FAL203,FCR203,FCO2,FH2DE,FH2DB,FCAC03,FMGC03,TABLAT,EFUSN, 00303000
      3 RHOC0N,CP0CN,TKA,TKB,AP,BP,RBUB,GAMO,GAN1,GAM2,GAMINF, 00303100
      4 DAVDIA,DMNDEV,TFA1L,XVAR1                 00303200
      PARAMETER (JRREBL=32)                        00303300
*/D /REBCOM/.22,23                               00303400
      1 ,HTXC0N,DNOX,DMNA,DMPER,DMON,QFM,BOILMS,BOILEN,PBOT 00303500
      PARAMETER ( JRCPOL=20 )                      00303600
*/ *****00303700
*/ * SLOF02 : INITIAL SET OF SLAM COMMON VARIABLES *00303800
*/ *****00303900
*/I I-IRCTYP.49                                  00304000
C                                                00304100
C                                                00304200

```



```

RHOCOM = 2250.0                                00304300
CPCOM = 1000.0                                  00304400
XVAR1 = 1.000                                    00304500
TKA = -1.375E-03                                00304600
TKB = 1.6128                                     00304700
AP = 0.3949E-18                                 00304800
BP = 0.01416                                     00304900
RBUB = 0.004                                     00305000
GAHO = 0.07700                                   00305100
GAM1 = 200.0                                     00305200
GAN2 = 1.0E-10                                   00305300
GAMINF = 1.0E-08                                00305400
*/ *****00305500
*/ * SLOGO1 : ADD MATERIAL PROPERTIES OF SLAM TO CONTAIN *00305600
*/ * FOR SLAM CALCULATION *00305700
*/ *****00305800
*D C100K.1                                         00305900
  6          LAR,LNA2CO,LZR,LZRO2,LSS,LSSOX,LCACO3,LMGCO3, 00306000
  7          LINERT,LCOOL                             00306100
*/ /WATER/.16                                     00306200
  PARAMETER (JIMATR=56)                             00306300
*/ /INPUTW/.8                                     00306400
  PARAMETER (JPWORD=54)                             00306500
*/ I-SETUP.241                                    00306600
  DATA (OPWORD(I,52),I=1,2) / 'CACO3', ' ' /        00306700
  DATA (DFALT(I,52),I=1,3), (IDFALT(I2,52),I2=1,3) 00306800
  1 /1.0E+4,1.E20,100.,3,0,1/                       00306900
  DATA (OPWORD(I,53),I=1,2) / 'MGCO3', ' ' /        00307000
  DATA (DFALT(I,53),I=1,3), (IDFALT(I2,53),I2=1,3) 00307100
  1 /1.0E+4,1.E20,84.,3,0,1/                       00307200
  DATA (OPWORD(I,54),I=1,2) / 'INERT', ' ' /        00307300
  DATA (DFALT(I,54),I=1,3), (IDFALT(I2,54),I2=1,3) 00307400
  1 /1.0E+4,1.E20,100.,3,0,1/                       00307500
*/ D PR-RHOX.35                                    00307600
  4          41,42,43,44,45, 46,47,48,49,50,51,52,53,54),I 00307700
*/ I PR-RHOX.260                                   00307800
C                                                    00307900
C CACO3                                             00308000
52 RHOX = 2200.                                     00308100
  RETURN                                           00308200
C                                                    00308300
C MGCO3                                             00308400
53 RHOX = 1400.                                     00308500
  RETURN                                           00308600
C                                                    00308700
C INERT                                             00308800
54 RHOX = 2885.                                     00308900
*/ D PR-SHX.41                                     00309000
  4          41,42,43,44,45, 46,47,48,49,50,51,52,53,54),I 00309100
*/ I PR-SHX.281                                    00309200
C                                                    00309300
C CACO3                                             00309400
52 SHX = 1173.                                       00309500
  RETURN                                           00309600
C                                                    00309700
C MGCO3                                             00309800
53 SHX = 1173.                                       00309900
  RETURN                                           00310000
C                                                    00310100
C INERT                                             00310200
54 SHX = 600.                                       00310300
  RETURN                                           00310400
*/ D PR-ENTHX.23                                   00310500
  4          41,42,43,44,45, 46,47,48,49,50,51,52,53,54),I 00310600
*/ I PR-ENTHX.234                                  00310700
C                                                    00310800
C CACO3                                             00310900
52 GO TO 100                                         00311000
C                                                    00311100
C MGCO3                                             00311200
53 GO TO 100                                         00311300
C                                                    00311400
C INERT                                             00311500
54 GO TO 100                                         00311600
*/ D PR-CONDX.32                                   00311700
  4          41,42,43,44,45, 46,47,48,49,50,51,52,53,54),I 00311800
*/ I PR-CONDX.273                                  00311900
C                                                    00312000

```

```

C      CACO3                                00312100
  52  CONDX = 2.0                            00312200
      RETURN                                  00312300
C                                            00312400
C      HGC03                                00312500
  53  CONDX = 2.                              00312600
      RETURN                                  00312700
C                                            00312800
C      INERT                                 00312900
  54  CONDX = 2.                              00313000
      RETURN                                  00313100
*/ *****00313200
*/ * SLOG02 : MODIFICATION OF MASS FRACTION OF CONCRETE COMPONENTS *00313300
*/ * EXTENSION OF ARRAY DIMENSION FOR CONCRETE *00313400
*/ *****00313500
*D C100K.117                                00313600
  REAL BASALT(17),LIME(17),GENRIC(17)        00313700
*D I-IRCTYP.37                               00313800
  1 0.062,0.0832,0.00,0.0150,0.0386,0.020,0.0,0.0,1575.0,5.225E+09/ 00313900
*D I-IRCTYP.38,39                            00314000
  DATA LIME/0.1105,7*0.0,                    00314100
  1 0.0,0.0,0.,0.049,0.0305,0.567,0.243,1575.0,3.46E+09/ 00314200
*D I-IRCTYP.41                               00314300
  1 0.012,0.016,0.00004,0.35698,0.0394,0.02,0.0,0.0,1575.0,3.46E+09/ 00314400
*I I-IRCTYP.68                               00314500
  FCACO3=BASALT(14)                            00314600
  FMGCCO3=BASALT(15)                            00314700
*D I-IRCTYP.69                               00314800
  IF(TABLAT.LT.0.0001) TABLAT=BASALT(16)        00314900
*D I-IRCTYP.72                               00315000
  EL(LCON)=BASALT(17)/RHOX(LCON,TABLAT,PRES)    00315100
*I I-IRCTYP.91                               00315200
  FCACO3=LIME(14)                              00315300
  FMGCCO3=LIME(15)                              00315400
*D I-IRCTYP.92                               00315500
  IF(TABLAT.LT.0.0001) TABLAT=LIME(16)         00315600
*D I-IRCTYP.95                               00315700
  EL(LCON)=LIME(17)/RHOX(LCON,TABLAT,PRES)    00315800
*I C100K.131                                00315900
  FCACO3=GENRIC(14)                            00316000
  FMGCCO3=GENRIC(15)                            00316100
*D C100K.132                                00316200
  IF(TABLAT.LT.0.0001) TABLAT=GENRIC(16)        00316300
*D C100K.133                                00316400
  EL(LCON)=GENRIC(17)/RHOX(LCON,TABLAT,PRES)    00316500
*I I-IRCTYP.137                              00316600
  FCACO3=VALUE(DMY)                            00316700
  FMGCCO3=VALUE(DMY)                            00316800
*/ *****00316900
*/ * SLOH01 : ADD NEW FLAG 'SLMOUT','WATOUT' TO COMMON BLOCK *00317000
*/ *****00317100
*D /FLAGS/.49,50                             00317200
  5   NFZROX,SLMOUT,WATOUT                      00317300
  PARAMETER (JLFLAC=22)                        00317400
*D /FLAGS/.53                                00317500
  3   HTLOLY,HTLAY,HTRAD,NFCHEM,NFENGS,NFSENG,NFSPWD,SLMOUT,WATOUT 00317600
*/ *****00317700
*/ * SLOH02 : CELL FLAG 'LDUMNC' IS 'TRUE' WHEN SLAM MODEL OR WATER *00317800
*/ * MIGRATION MODEL IS SELECTED *00317900
*/ *****00318000
*I I-IRCON.164                               00318100
  LDUMNC = .TRUE.                              00318200
*I I-IRCON.172                               00318300
  CALL SLPROP                                  00318400
  LDUMNC = .TRUE.                              00318500
*/ *****00318600
*/ * SLOH03 : CELL FLAG 'SLMOUT' IS 'TRUE' WHEN SODIUM CONCRETE *00318700
*/ * INTERACTION OCCURS *00318800
*/ *****00318900
*I I-IRCON.164                               00319000
  SLMOUT = .TRUE.                              00319100
*/ *****00319200
*/ * SLOH04 : CELL FLAG 'WATOUT' IS 'TRUE' WHEN WATER RELEASE *00319300
*/ * OCCURS *00319400
*/ *****00319500
*I I-IRCON.171                               00319600
  WATOUT = .TRUE.                              00319700
*/ *****00319800

```

```

*/ * SLO101 : LINER FAILURE TEMPERATURE 'TFAIL' IS SET TO DEFAULT *00319900
*/ *****00320000
*D I-IRINT.59 00320100
    TFAIL = 1.0E+10 00320200
*/ *****00320300
*/ * SLO102 : LINER FAILURE FLAG 'LNFAIL' IS 'TRUE' WHEN LINER *00320400
*/ * FAILURE OCCURS *00320500
*/ *****00320600
*D I-IREBEL.46 00320700
    LNFAIL = .FALSE. 00320800
*D RB-INTER.240,241 00320900
    IF ( RITA( KZO( KREG )) .LT. TFAIL ) THEN 00321000
        LNFAIL = .FALSE. 00321100
    ELSE 00321200
        LNFAIL = .TRUE. 00321300
*/ *****00321400
*/ * SLOJ01 : ADD A ARGUMENT TO SUBROUTINE *00321500
*/ *****00321600
*D I-IREBEL.144 00321700
    4 OH(INAMES+1),OTYP,IH(LRKZO+1),IH(LRNHP+1),NM,H(LRITAO+1)) 00321800
*D I-IREBEL.181 00321900
    4 OH(INAMES+1),IH(LRKZO+1),IH(LRNHP+1),NM,H(LTRANT+1),H(LRPTAO+1)) 00322000
*D I-IRP00.4 00322100
    2 OTBNY,OTBNM,ICTBFL,ICTBNP,ONAMES,KZO,NNP,NMAT,TRNT,PTAO) 00322200
*D I-IRP00.7 00322300
    REAL PA(*),PDZ(*),PMA(NMAT,*),PT(*),PTA(*),PTAO(*),EL(*),EPH(*) 00322400
*D I-IRINT.5 00322500
    3 ONAMES,OTYP,KZO,NNP,NMAT,RITAO) 00322600
*D I-IRINT.9 00322700
    REAL TBCL(*),TBCU(*),CTBX(*),CTBY(*),RITAO(*) 00322800
*/ *****00322900
*/ * SLOJ02 : SAVE POOL AVERAGE TEMP FOR CURRENT TIME STEP *00323000
*/ *****00323100
*I I-IRP00.63 00323200
    PTAO(KZ) = PTA(KZ) 00323300
*I I-IRP00.107 00323400
    PTAO(KZ) = PTA(KZ) 00323500
*/ *****00323600
*/ * SLOJ03 : SAVE INTERM AVERAGE TEMP FOR CURRENT TIME STEP *00323700
*/ *****00323800
*I I-IRINT.64 00323900
    RITAO(KZ)=RITA(KZ) 00324000
*I I-IRINT.107 00324100
    RITAO(KZ)=RITA(KZ) 00324200
*/ *****00324300
*/ * SLOK01 : ADD SLAM OUTPUT PRINT *00324400
*/ *****00324500
*I,O-RBOUT.341 00324600
C ***** 00324700
C SLAM OUTPUT PRINTS 00324800
C ***** 00324900
C 00325000
    IF( SLMOUT .OR. WATOUT) THEN 00325100
        CALL SLOUT 00325200
    ELSE 00325300
C 00325400
C *** END OF "IF( SLMOUT .OR. WATOUT) THEN" *** 00325500
C ENDF 00325600
*/ *****00325700
*/ * SLOK02 : WRITE SLAM CALCULATIONAL RESULTS TO PLOT FILE *00325800
*/ *****00325900
*D PL-CPLOT.155 00326000
*I PL-CPLOT.222 00326100
    400 CONTINUE 00326200
*I PL-CPLDT.9 00326300
*CALL,/SLCOM1/ 00326400
*CALL,/SLCON2/ 00326500
*I PL-CPLOT.129 00326600
    OFLAG= '400' 00326700
    WRITE(N12) OFLAG,OBLANL,NCELL,SS 00326800
    WRITE(N12) OBLANK,OBLANK,JCONC,JINT,JPOOL,JATM,NREG 00326900
*I PL-CPLOT.180 00327000
    IF (NFNACO.OR.NFCONC) THEN 00327100
        TDR1 = TDRY<NDRY/2+1) 00327200
        OFLAG = '335' 00327300
        WRITE(N11) OFLAG,OBLANK,NCELL,SS 00327400
        WRITE(N11) OBLANK,OBLANK,TDR1,TS,QCV,QCONO,QCON,DIST,VEL,DEL1, 00327500
        1 DDL1DT,QVSUM,QDEL1,PS1,H2OFLX,H2OSRC,WFLXDT,CO2SRC,XNTCO2 00327600

```

```

ENDIF                                                    00327700
C
C
*/ PL-CPL0T.220                                         00327900
IF (MFCO.DR.NFCO) THEN                                  00328000
OFLAG = '335'                                          00328100
WRITE(NTM) OFLAG,OBLANK,NCELL,SS                       00328200
WRITE(NTM) OBLANK,OBLANK,TDR1,TS,QCV,QCONO,QCON,DIST,VEL,DEL1, 00328300
1 DDL1DT,QVSUM,QDEL1,PS1,H2OFLX,H2OSRC,WFLXDT,CO2SRC,XNTCO2 00328400
ENDIF                                                    00328500
C
C
*/ *****00328600
*/ * SLOK03 : READ CONCRETE STORAGE ARRAY FROM RESTART FILE *00328800
*/ *****00328900
*/ I-RESTAR.17                                          00329000
*CALL,/NACON/                                           00329100
*CALL,/SLCOM1/                                          00329200
*CALL,/SLCOM2/                                          00329300
*/ I C100H.210                                          00329400
DO 80 NCELL = 1,NCELLS                                 00329500
LCELP = LECS + 12*(NCELL-1)                           00329600
KREG = 1                                                00329700
CALL NXTCEL ( LCELP )                                  00329800
IF ( .NOT. LDUMNC ) GO TO 80                            00329900
READ (NDSK,END=85) OBLANK,OBLANK,                     00330000
1 ( CH(II),II=1,9000),(ICH(II),II=1,200)              00330100
CALL LOAD ('REBLAY')                                    00330200
CALL TRANA                                             00330300
CALL SLPROP                                            00330400
CALL SLCHEM                                            00330500
GO TO 85                                                00330600
80 CONTINUE                                             00330700
85 CONTINUE                                             00330800
*/ *****00330900
*/ * SLOK04 : WRITE RESTART FILE FROM CONCRETE STORAGE ARRAY *00331000
*/ *****00331100
*/ I D-RESOUT.19                                        00331200
*CALL,/NACON/                                           00331300
DATA DROUT / 'RESOUT' /                                00331400
*/ I D-RESOUT.111                                       00331500
DO 81 NCELL = 1,NCELLS                                 00331600
LCELP = LECS + 12*(NCELL-1)                           00331700
CALL NXTCEL ( LCELP )                                  00331800
IF ( .NOT. LDUMNC ) GO TO 81                            00331900
WRITE (NDSK) OBLANK,OBLANK,                           00332000
1 ( CH(II),II=1,9000),(ICH(II),II=1,200)              00332100
GO TO 82                                                00332200
81 CONTINUE                                             00332300
82 CONTINUE                                             00332400

```

9. Test Calculation

9-1. Sodium-Concrete Interaction

9-1.1 System and Conditions of Test Calculation

A system consisting of the concrete, sodium, pool and atmosphere is chosen for the test calculation .

The concrete is limestone , and its chemical compositions are as shown in Table 9.1. The dimensions of the concrete, and the initial conditions for pool and atmosphere are presented in Figure 9.1

Table 9.1 Limestone Concrete Compositions

Composition	Weight Ratio
SiO ₂	0.1105
TiO ₂	0.0
MnO	0.0
MgO	0.0
CaO	0.0
Na ₂ O	0.0
K ₂ O	0.0
Fe ₂ O ₃	0.0
Al ₂ O ₃	0.0
Cr ₂ O ₃	0.0
CO ₂	0.0
H ₂ O(E)	0.049
H ₂ O(B)	0.0305
CaCO ₃	0.567
MgCO ₃	0.243

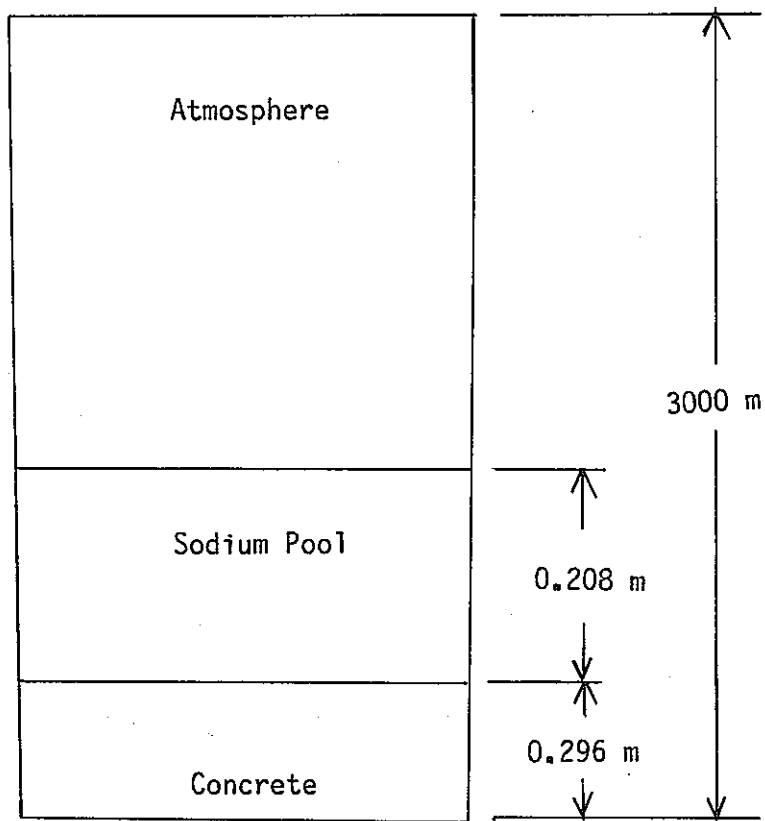


Figure 9.1 System for Sodium-Concrete Interaction
Test Calculation

Initial conditions

- Concrete Temp. = 298 (K)
- Pool Temp. = 873 (K)
- Atmos. Temp. = 298 (K)
- Atmos. Pres. = 0.1 (MPa)

Since the calculational conditions for the SLAM module incorporated into CONTAIN are slightly different from the original SLAM code, it was necessary, to add some modifications to the original SLAM to obtain the identical calculational conditions to conduct the calculation.

Modifications required for SLAM are;

- (1) Chemical reactions in the pool are ignored.
- (2) The sodium density is made equal to that in the CONTAIN.
(RHO (3) = 807).

Inputs to be modified are;

- (1) Radiation heat released from the pool is not considered
(EPSRAD = 0.0).
- (2) Mass fraction of Na in saturated Na-NaOH solution is made smaller (XMOLNA = 0.05).
- (3) Pool power heater is set to be OFF.

Two calculations (Case ID SLAM-1 and SLAM-2) were performed by the original SLAM code to examine the effects of parameters (Table 9.2), and another two calculations (Case ID CONTAIN-1 and CONTAIN-2) were performed by CONTAIN including SLAM module, varying time step size (Table 9.3).

Table 9.2 Calculation Cases for Original SLAM Code

Case ID	Heat Radiation	Mass frac. of Na in sat. Na-NAOH	Chemical Reactions in Pool	Na Density (kg/m3)	Pool Power Heater
SLAM-1	yes	0.8	yes	800	no
SLAM-2	no	0.05	no	807	no

In the table, conditions of Case SLAM-1 is identical to that of the original SLAM sample case except for the pool power heater.

Table 9.3 Calculation Cases for New CONTAIN Code

Time Zone	CONTAIN-1 Time Step (sec)	CONTAIN-2 Time Step (sec)
0-200	4.	2.
200-600	8.	4.
600-1200	16.	8.
1200-2400	20.	10.
2400-4800	40.	20.
4800-9600	80.	40.
9600-16000	160.	80.


```

DEF 1=SI02 2=H2O 3=NA 4=H2 5=NAOH 6=NA2SI03 7=NA2CO3 8=NA2O
9=CAO 10=CAC03 11=CO2 12=C 13=MGC03 14=MGO 15=INERT
10 SPECIES FOR PLOTTING
3 5 7 10 9 3 8 11 2 12
# OF REACTIONS
6
REACTION # 1 - # SPECIES AND RATE COEF (AK) (H2O+NA--NAOH+0.5H2)
4 99.0000E+05 1.8000E 03 0.3
H2O # COEFR IPHASR COEFP IPHASP
2 1.0000E 00 3
NA
3 1.0000E 00 2
NAOH # COEFR IPHASR COEFP IPHASP
5 0.0000E 00 0 1.0000E 00 2
H2
4 0.0000E 00 0 0.5000E 00 3
REACTION # 2 #SPECIES AND AK (CO2+4NA--2NA2O+C)
4 1.0000E+05 1.8000E 03 0.3
CO2
11 1.0000E 00 3
NA
3 4.0000E 00 2
NA2O
8 0.0000E 00 0 2.0000E 00 1
C
12 0.0000E 00 0 1.0000E 00 1
REACTION # 3 4NA + 3CAC03 ---- 2NAC03 + 3CAO + C
5 8.8500E+12 60000.0E 00 0.1000E 00 0.1000E 00 0.6000E 00
CAC03
10 3.0000E 00 1
NA
3 4.0000E 00 2
NAC03
7 0.0000E 00 0 2.0000E 00 2
CAO
9 0.0000E 00 0 3.0000E 00 1
C
12 0.0000E 00 0 1.0000E 00 1
REACTION # 4 4NA + 3MGC03 ---- 2NAC03 + 3MGO + C
5 8.8500E+12 60000.0E 00 0.1000E 00 0.1000E 00 0.6000E 00
MGC03
13 3.0000E 00 1
NA
3 4.0000E 00 2
NAC03
7 0.0000E 00 0 2.0000E 00 2
MGO
14 0.0000E 00 0 3.0000E 00 1
C
12 0.0000E 00 0 1.0000E 00 1
REACTION #5 CAC03+2NAOH----CAO+H2O+NA2CO3
5 40.0000E+05 34000.0E 00 0.1
CAC03
10 1.0000E 00 1
NAOH
5 2.0000E 00 2
CAO
9 0.0000E 00 0 1.0000E 00 1
H2O
2 0.0000E 00 0 1.0000E 00 3
NA2CO3
7 0.0000E 00 0 1.0000E 00 2
REACTION #6 2NAOH + SI02 ----- NA2SI03 + H2O
4 40.0000E+05 34000.0E 00 0.1
NAOH
5 2.0000E 00 2
SI02
1 1.0000E 00 1
NA2SI03
6 0.0000E 00 0 1.0000E 00 1
H2O
2 0.0000E 00 0 1.0000E 00 3

```

List 9.1 SLAM (Case SLAM-1) Test Calculation Input Data (Unit-9)

```

NDRY  NWET  NPRINT  ISRC  NPOOL  ITHRMO  MIG  NMAX      SNL-TEST 28
28    20    0010    12    07      5      0    3000
CTYPE  PINF      TINF      TC      XHOLNA      TPOOL
3.00000E 00 1.00000E 05 298.000E 00 298.000E 00 0.80000E 00 873.000E 00
FMGCO   FCACO      FSIO2      RHOCN      CPCON      EPSRAD
0.24300E 00 0.56700E 00 0.11050E 00 2250.00E 00 1000.00E 00 0.50000E 00
DELH    POWER      DEPTH      TSET      TWFIN      DTHAX
0.30400E 00 00000.0E 00 0.20800E 00 873.000E 00 15000.0E 00 20.0000E 00
FSATOH  TPWR      GAMINF      TKA      TKB      RBUB
0.05000E 00 28800.0E 00 1.00000E-08-1.37500E-03 1.61280E 00 0.00400E 00
RNACA   DTDIST(PLOT)  HINF      GAMO      GAM1      GAM2
0.03000E 00 1800.00E 00 22.0000E 00 0.07700E-00 200.000E 00 1.00000E-10
DIPL0T  TC(P00L)      TC1      TC2      TC3      TC4
30.0000E 00-1.00000E 00 0.01000E 00 0.02540E 00 0.07600E 00 0.15000E 00
HULYI   TIMEO      RO-EVP      RO-BND      XVAR(1)
0.50000E 00 0.00000E 00 110.000E 00 72.0000E 00 1.00000E 00

```

List 9.2 SLAM (Case SLAM-1) Test Calculation Input Data (Unit-10)

```

IBM
&& ----- SC1004-----
&& CONTAIN TEST SC1004 - COMPARISON CONTAIN WITH SLAM STAND ALONE CODE

&& - - - - - < GLOBAL > - - - - -

CONTROL
  NCELLS=1
  NTITL=3
  NTZONE=7
EOI

&& ..... MATERIAL NAMES .....
MATERIAL
  COMPOUND  NAL NAV NA2SIO3 NA2CO3 NA2O CAO MGCO3 CACO3 H2O NAOH
            CO2 GRAPH SIO2 CONC  HGO  INERT

&& ..... EDIT TIMES .....
TIMES 36000. 0.0  4.0 20.0  200.  8.  40.  600. 16.  80. 1200.
       20.0 200.0 2400. 40. 400. 4800. 80. 800. 9600.
       160.0 1600. 16000. 0.5

TITLE
          CONTAIN TEST SC1004 ( POOL POWER HEATER OFF )
          COMPARISON CONTAIN WITH SLAM
          SNL EXPERIMENT NO. 28 - LIMESTONE CONCRETE

&& ..... LIQUID METAL REACTOR .....
FAST

&& ..... ACTIVATE ATMOSPHERIC THERMODYNAMICS .....
THERMO

PRLOW-CL

&& - - - - - < CELL=1 > - - - - -

CELL=1

CONTROL  JCONC=50 JPOOL=1 NUMTBC=1 MAXTBC=2 EOI

&& ..... CELL VOLUME + HEIGHT .....
GEOMETRY 10.E+05 3000.

&& ..... 2 ATMOSPHERE COMPONENTS - N2 AND NAV - TEMP. AND PRESS. ....
          ATMOS=2 1.0E+5 298. N2=0.99 NAV=0.01

LOW-CELL
GEOMETRY .0707  && FLOOR AREA
CONCRETE COMPOS=1  CONCRETE = LIME 60.45
TEMP = 298.0
PHYSICS
  NA-CONC
  MODEL=SLAM EOI
EOI  && TERMINATE THE PHYSICS
EOI  && TERMINATE THE LAYER
POOL TEMP = 873.0
COMPOS 1 NAL = 11.867
PHYSICS
  HT-COEF
      NAME = ATMOS
      VAR-X = TIME,X= 2, 0. 20000.
      VAR-Y = COEF,Y= 2, 22.0 22.0
EOI  && TERMINATES HEAT TRANSFER TABLE
EOI  && TERMINATE THE PHYSICS
EOI
BC 298.0 1.0E+05
EOI  && TERMINATE THE LOW-CELL INPUT
EOF

```

List 9.3 CONTAIN (Case CONTAIN-1) Input Data for Sodium-Concrete Interaction

9-1.2 Results of Calculation and Discussion

The calculational results are plotted in Figures 9.2 through 9.17. Major results are compared in Table 9.4.

As for pool temperature, the following relationship can be drawn from that table.

$$\text{SLAM-1} < \text{SLAM-2} < \text{CONTAIN-2} < \text{CONTAIN-1}$$

It is obvious that the time step of CONTAIN affects the calculation results. If the CONTAIN system time step is made smaller, its results probably approach those of SLAM. It was therefore concluded that the new CONTAIN was correctly implemented and time step should be properly chosen when SLAM module of CONTAIN is used.

Table 9.4 Comparison of Results of SLAM and CONTAIN
(after approx. 1,000 seconds)

Case ID	Time (sec)	Pool Temp. (K)	Penetration Length (m)	Ablation Velocity (m/sec)	Dry Zone Depth (m)
SLAM-1	1060	707	0.484e-3	0.855e-8	0.330e-1
SLAM-2	1050	755	0.102e-2	0.109e-6	0.332e-1
CONTAIN-1	1080	801	0.237e-2	0.924e-6	0.332e-1
CONTAIN-2	1080	790	0.199e-2	0.562e-6	0.335e-1

It was also found that the radiation heat released from the pool has an important effect in the case of the original SLAM code.

POOL TEMPERATURE

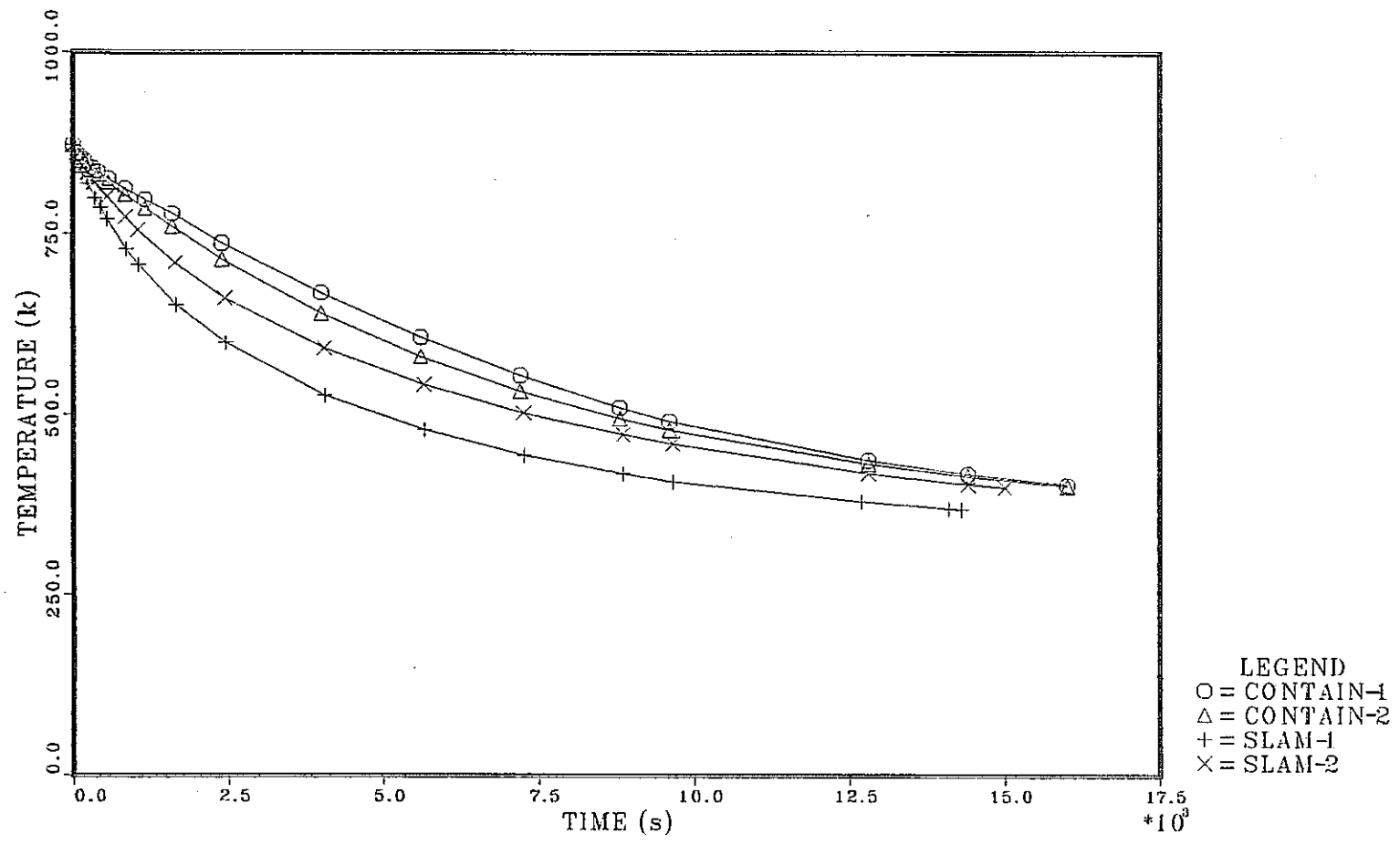


Figure 9.2 Comparison of Original SLAM and CONTAIN Calculations of Pool Temperature during Sodium-Concrete Reaction

DRY ZONE TEMPERATURE

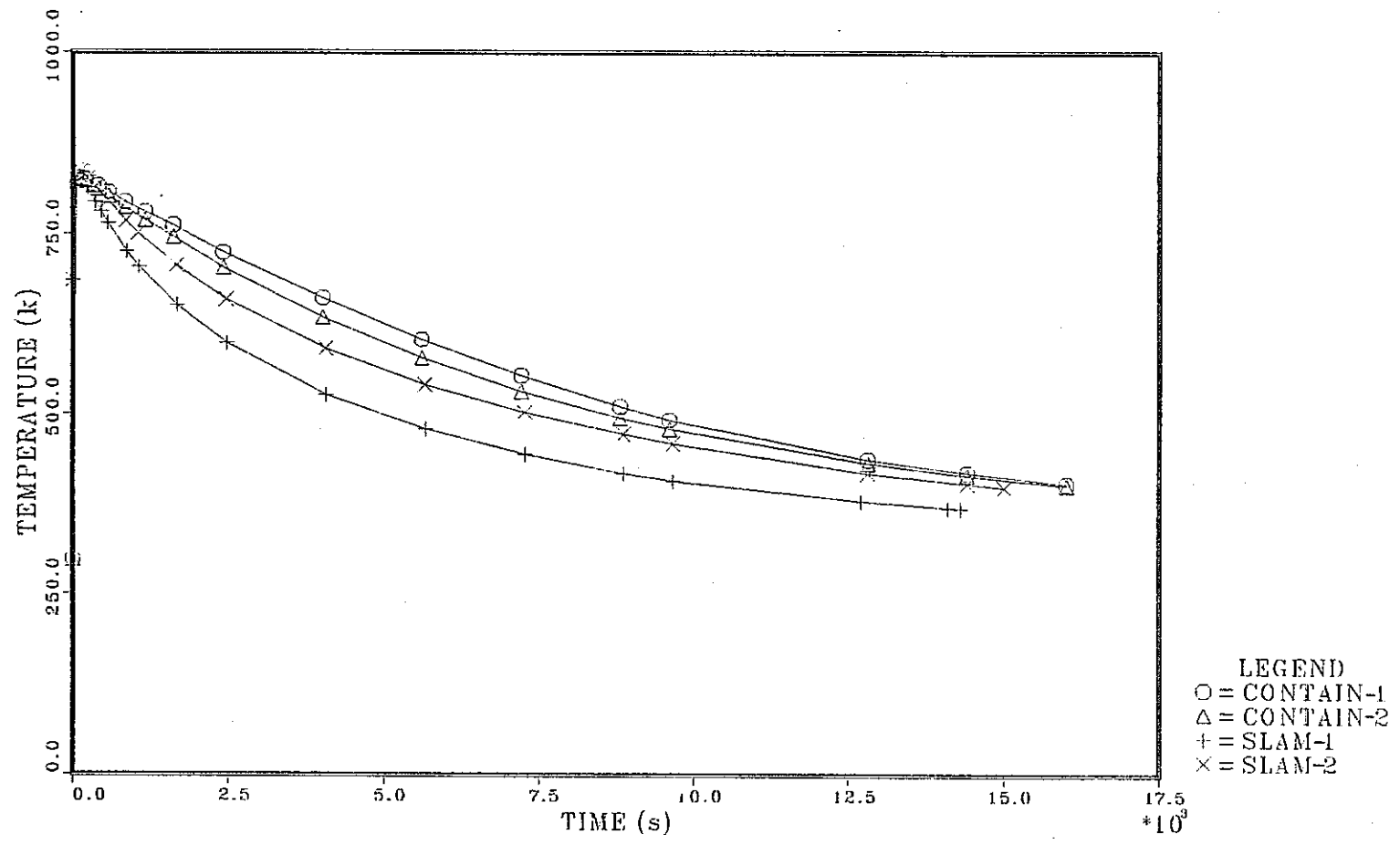


Figure 9.3 Comparison of Original SLAM and CONTAIN Calculations of Concrete DRY Zone Temperature during Sodium-Concrete Reaction

WET-DRY INTERFACE TEMPERATURE

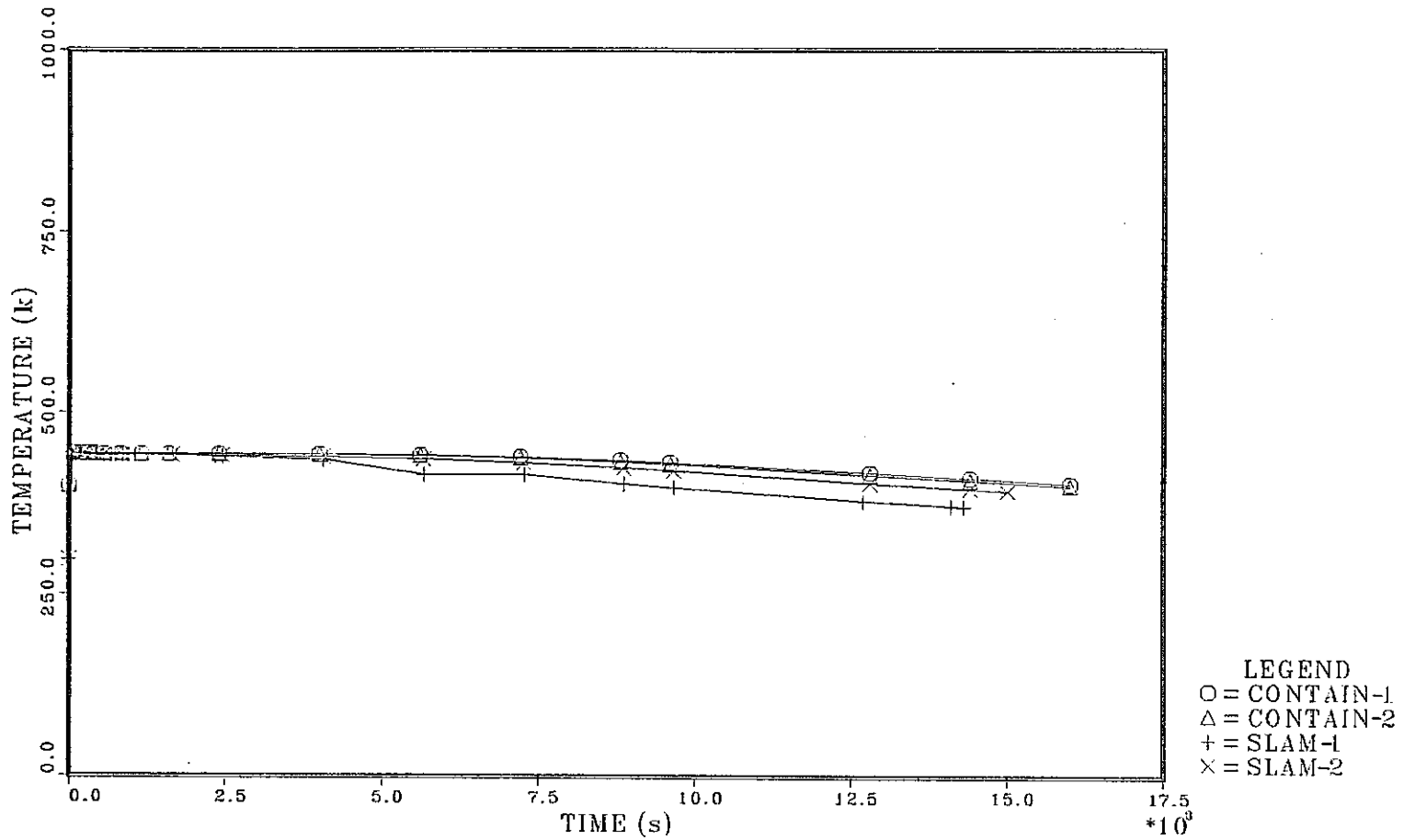


Figure 9.4 Comparison of Original SLAM and CONTAIN Calculations of Wet-Dry Interface Temperature during Sodium-Concrete Reaction

WET-DRY INTERFACE H2O PRESS

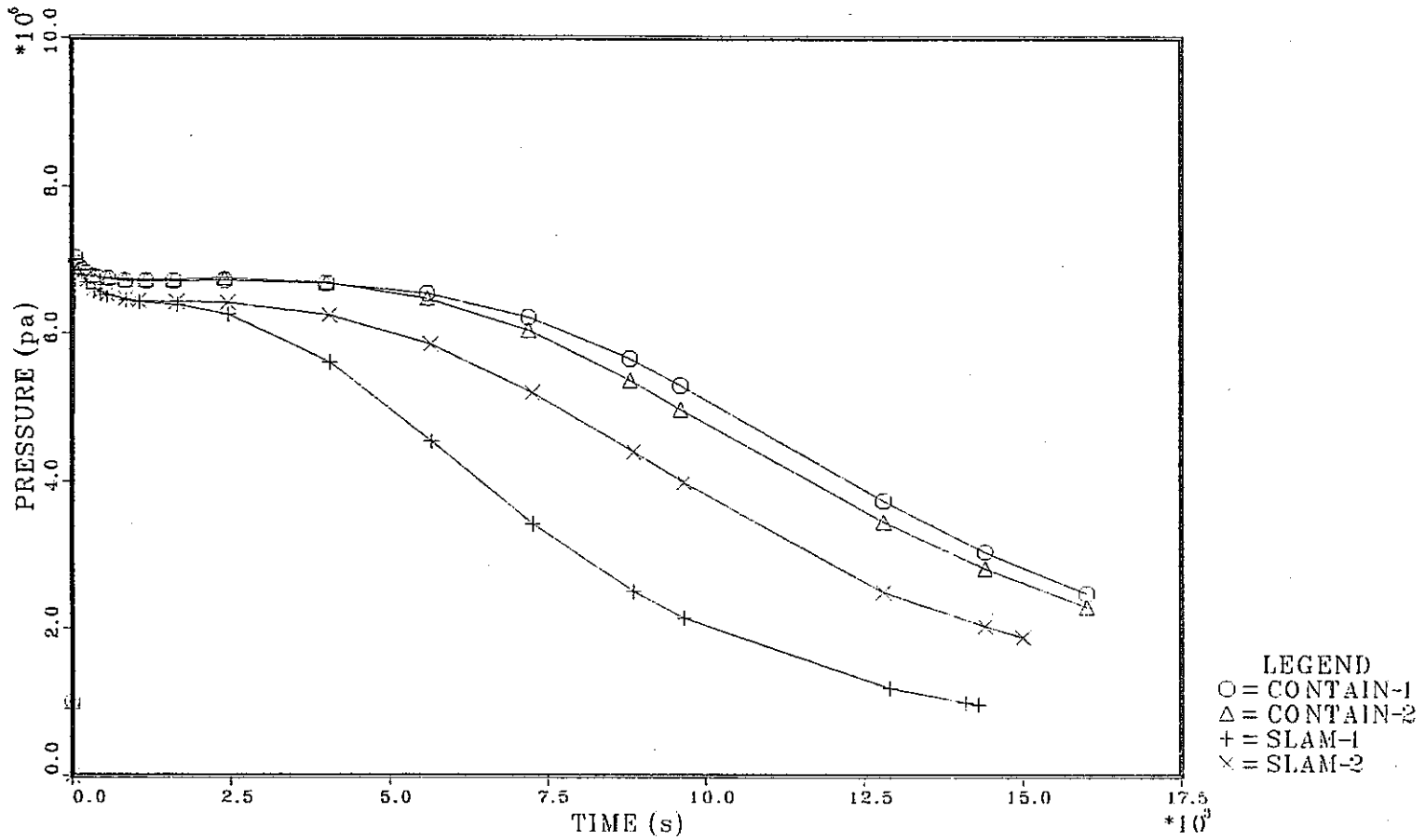


Figure 9.5 Comparison of Original SLAM and CONTAIN Calculations of Wet-Dry Interface H2O Pressure using Sodium-Concrete Reaction

ABLATION VELOCITY

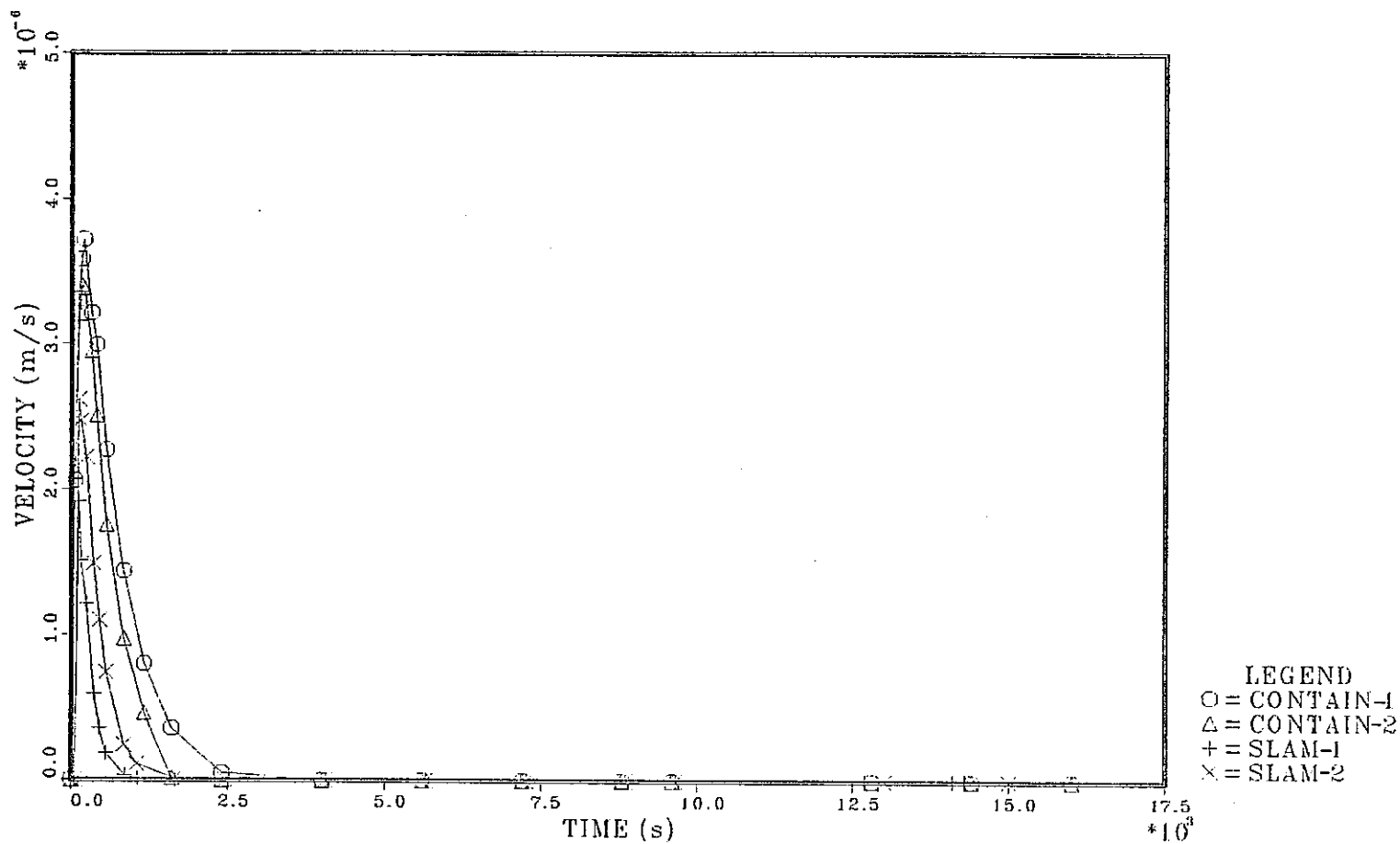


Figure 9.6 Comparison of Original SLAM and CONTAIN Calculations of Ablation Velocity during Sodium-Concrete Reaction

PENETRATION DEPTH

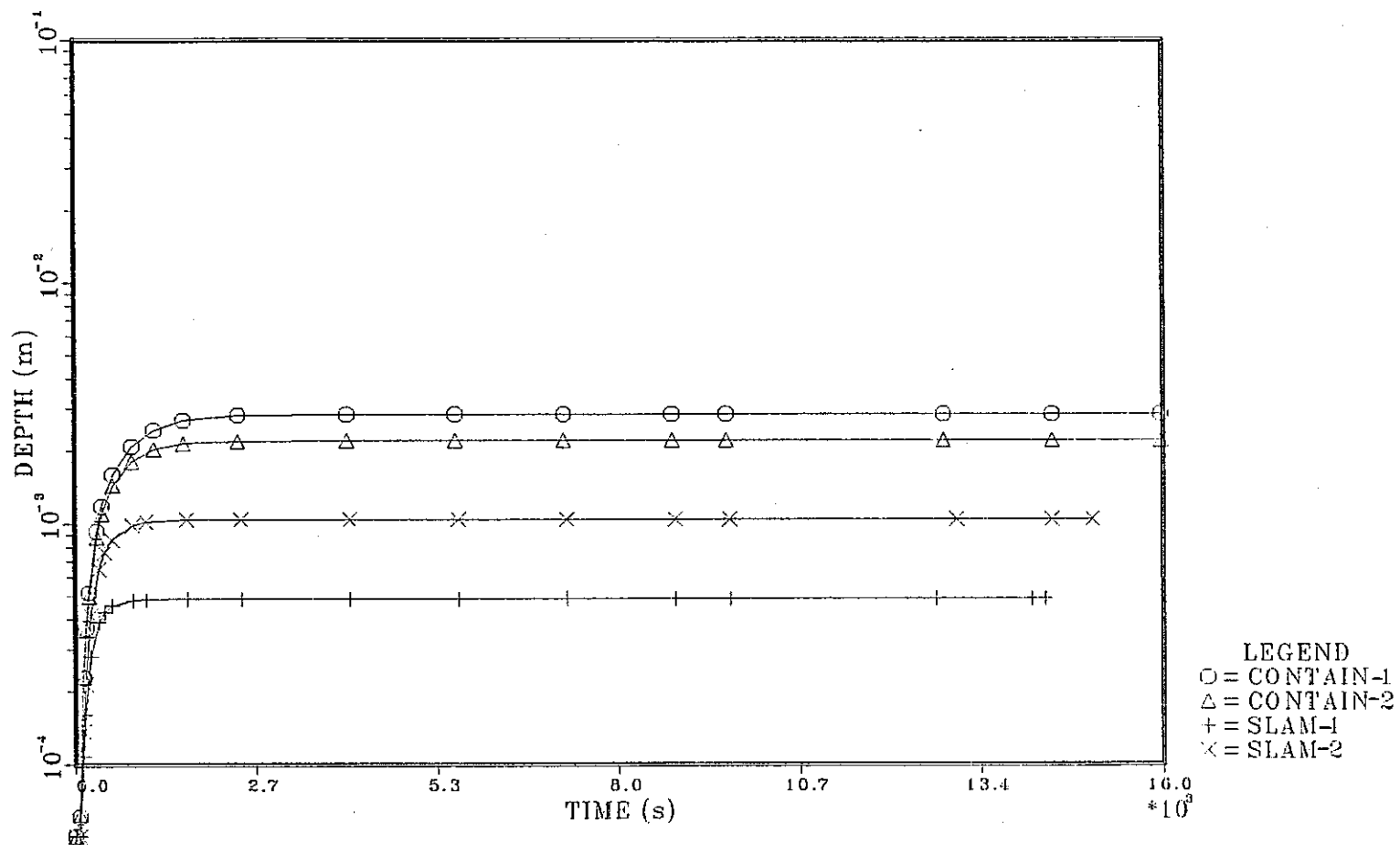


Figure 9.7 Comparison of Original SLAM and CONTAIN Calculations of Penetration Depth during Sodium-Concrete Reaction

DRY ZONE GROWTH RATE

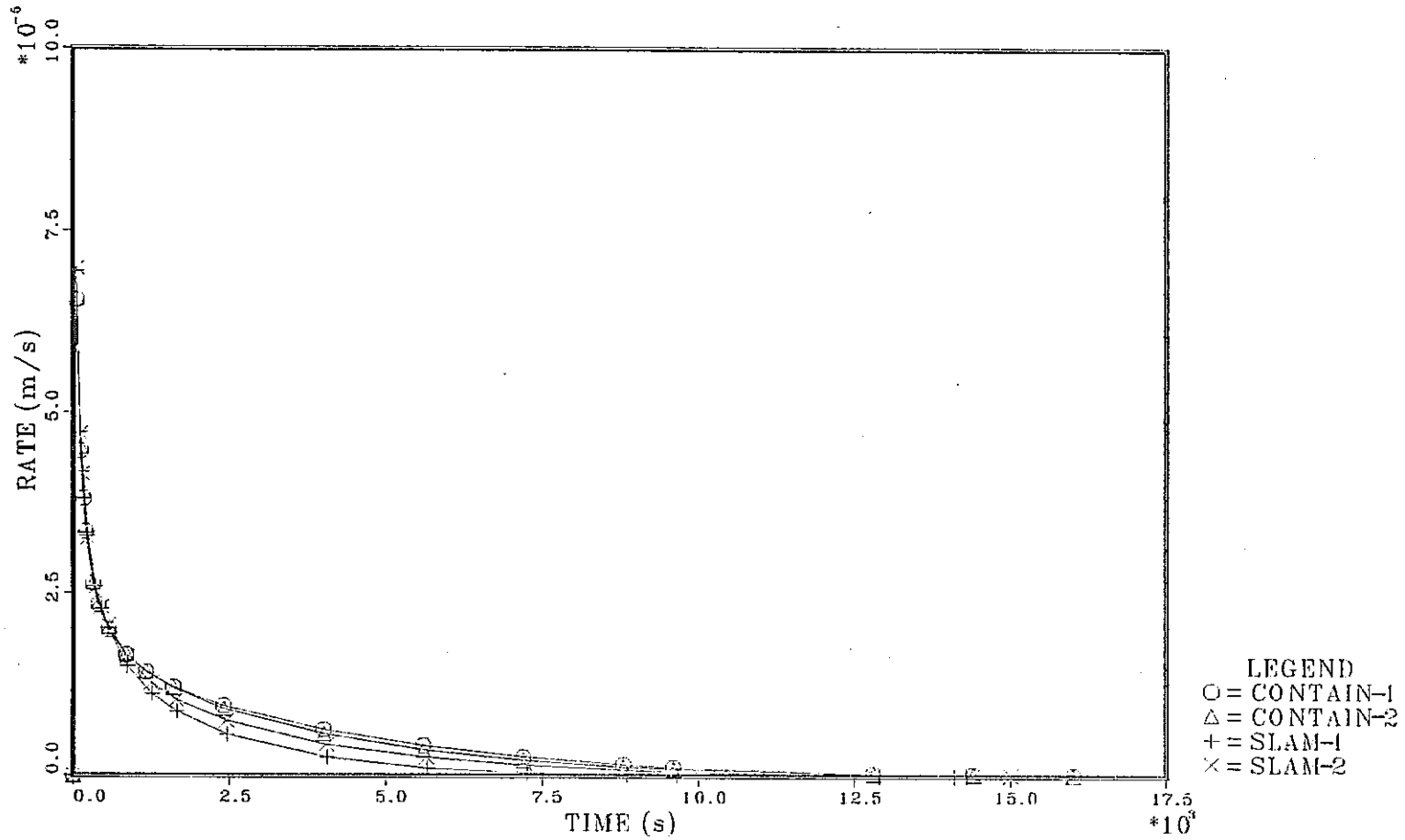


Figure 9.8 Comparison of Original SLAM and CONTAIN Calculations of Dry Zone Growth Rate during Sodium-Concrete Reaction

DRY ZONE DEPTH

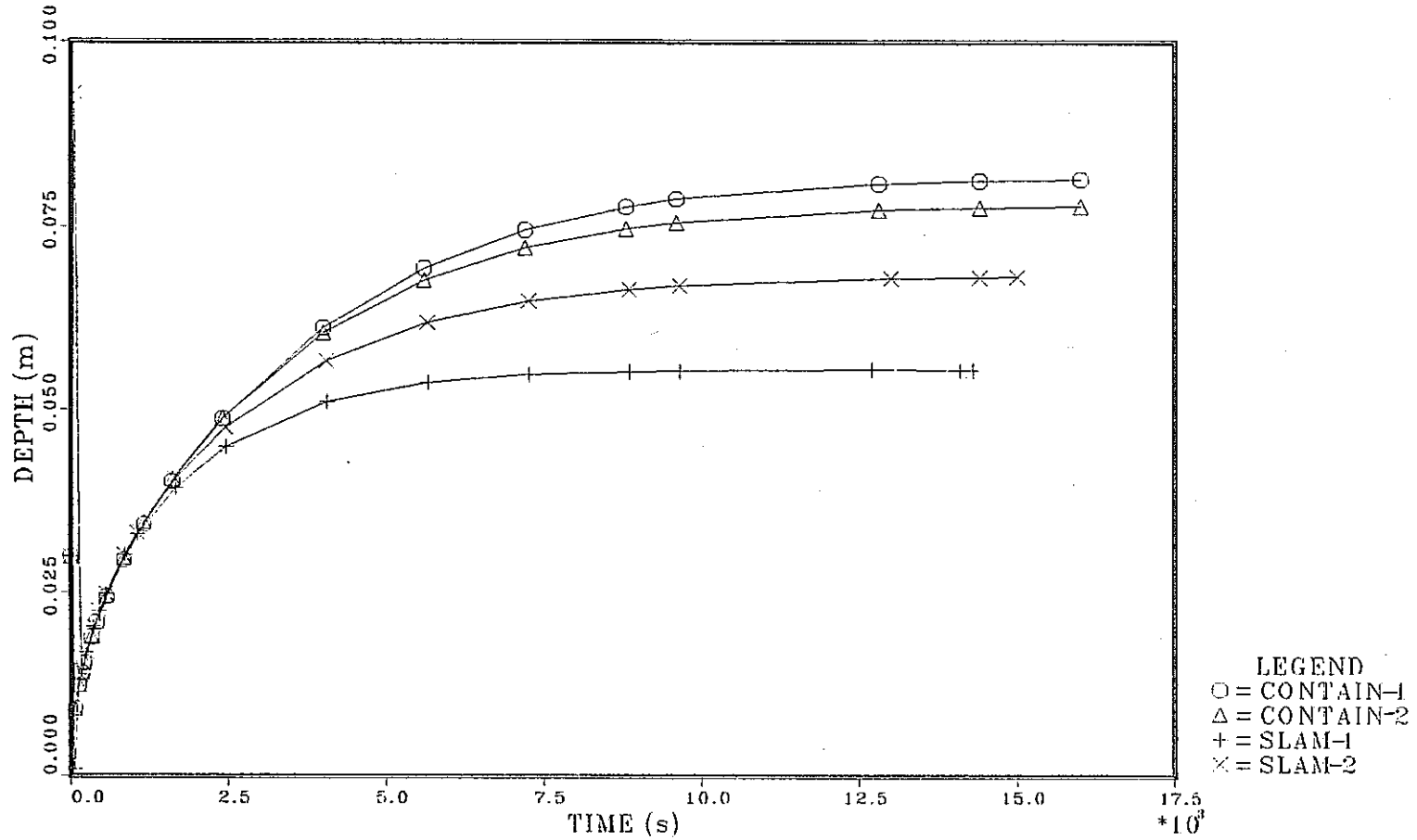


Figure 9.9 Comparison of Original SLAM and CONTAIN Calculations of Dry Zone Depth during Sodium-Concrete Reaction

INTERFACE H2O EVAP. FLUX

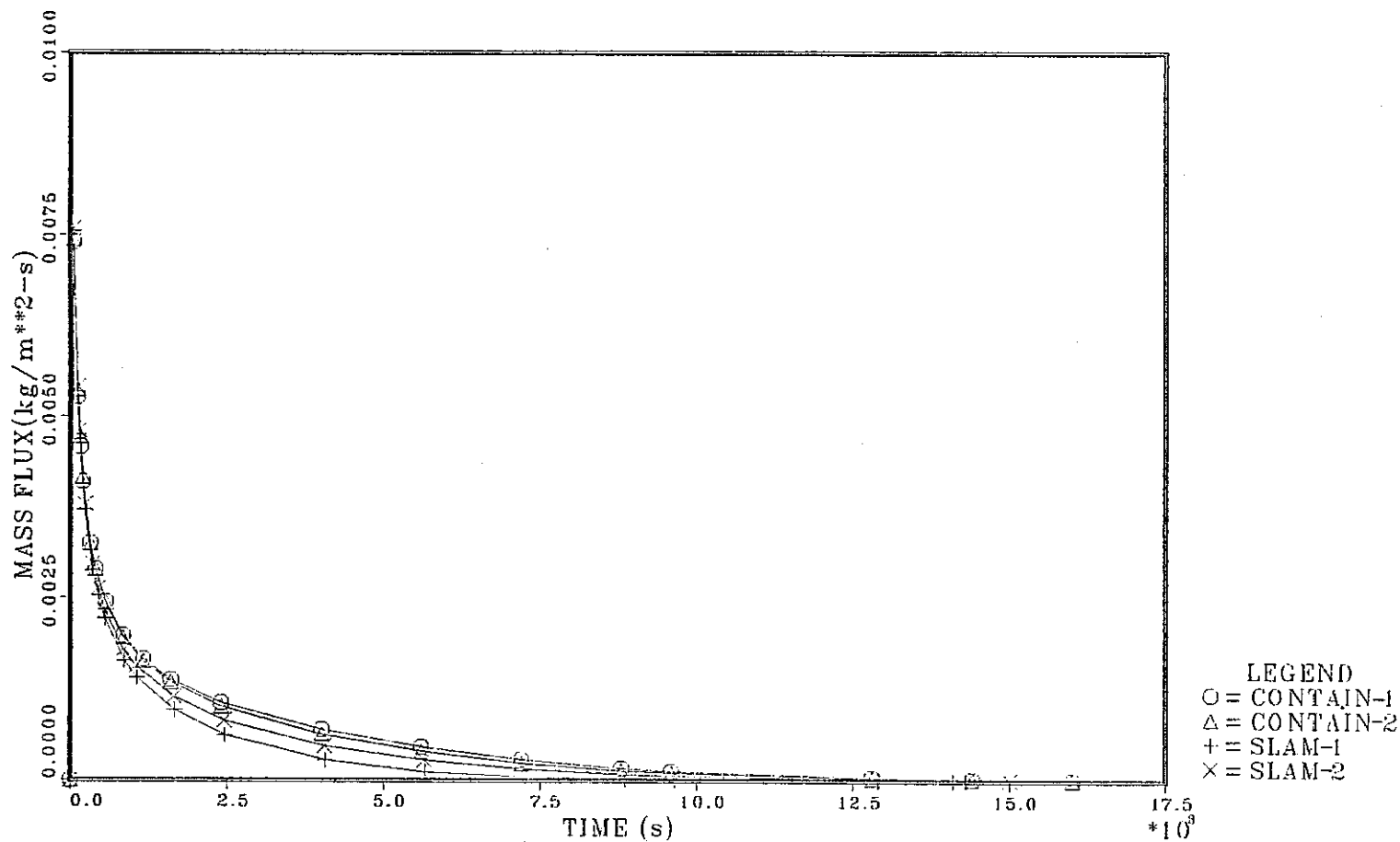


Figure 9.10 Comparison of Original SLAM and CONTAIN Calculations of Interface H2O Evaporated Flux during Sodium-Concrete Reaction

DRY-ZONE BOUND H2O FLUX

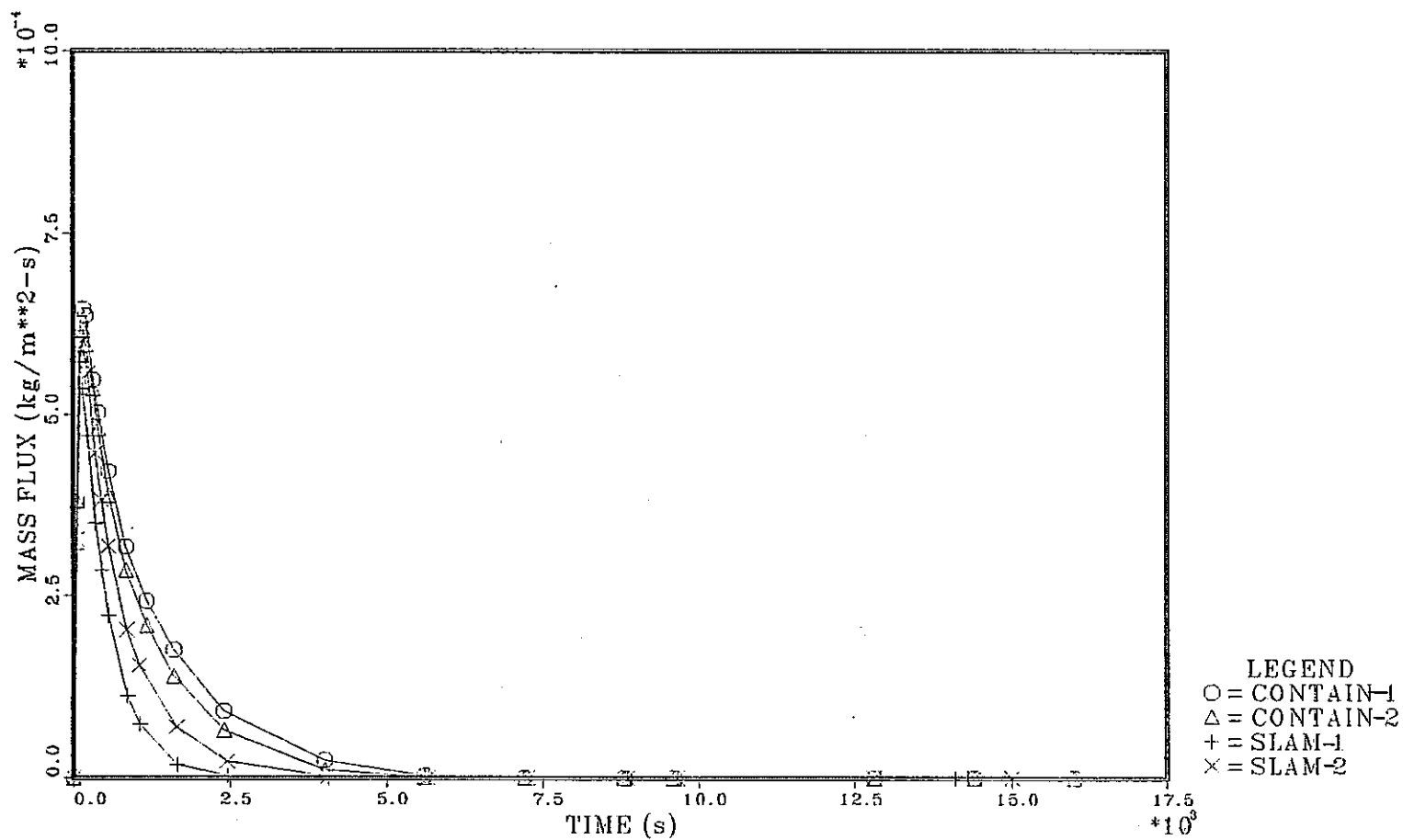


Figure 9.11 Comparison of Original SLAM and CONTAIN Calculations of Dry Zone Bound H2O Flux during Sodium-Concrete Reaction

DRY ZONE BOUND CO2 FLUX

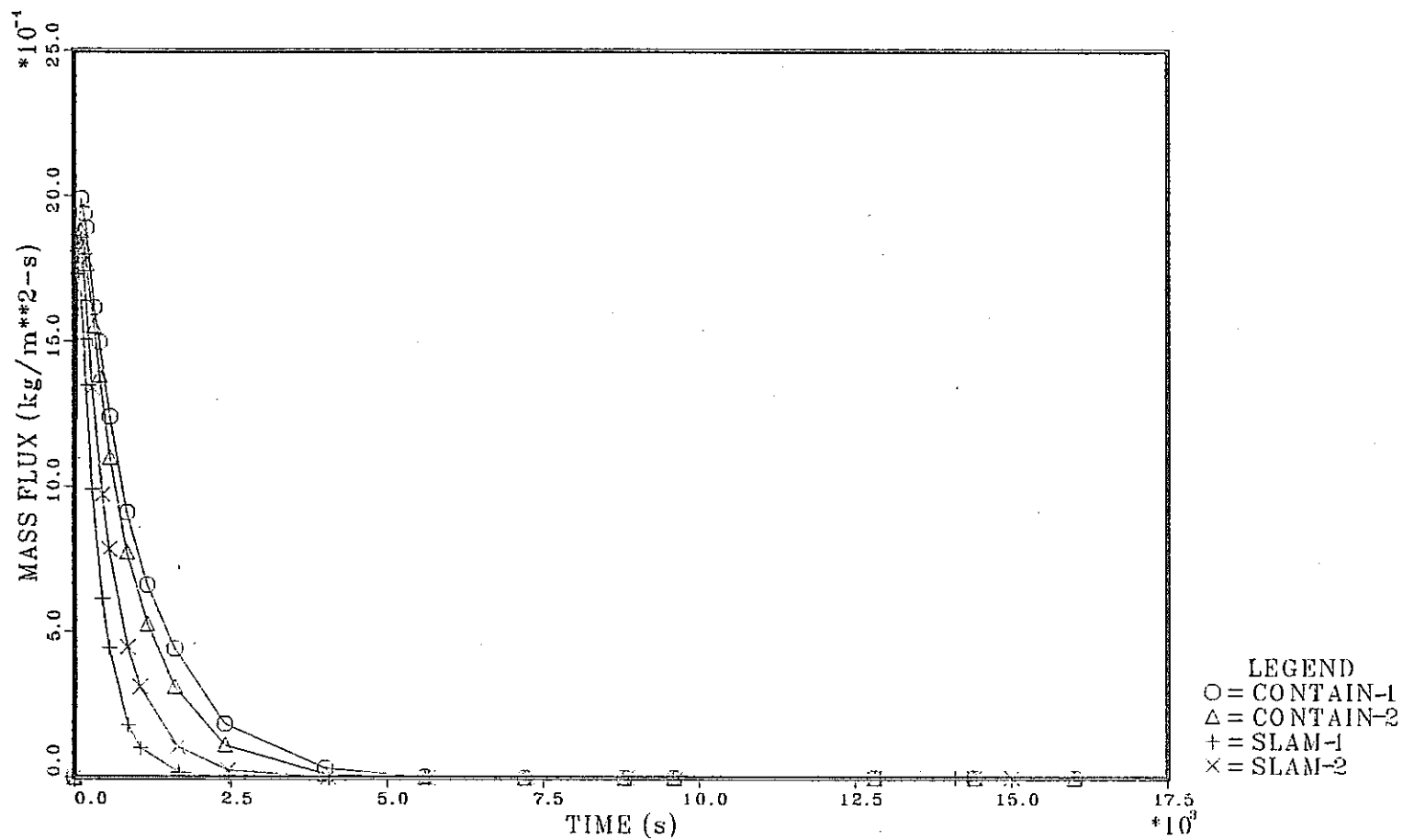


Figure 9.12 Comparison of Original SLAM and CONTAIN Calculation of Dry Zone Bound CO2 Flux during Sodium-Concrete Reaction

CONCRETE REACTION HEAT

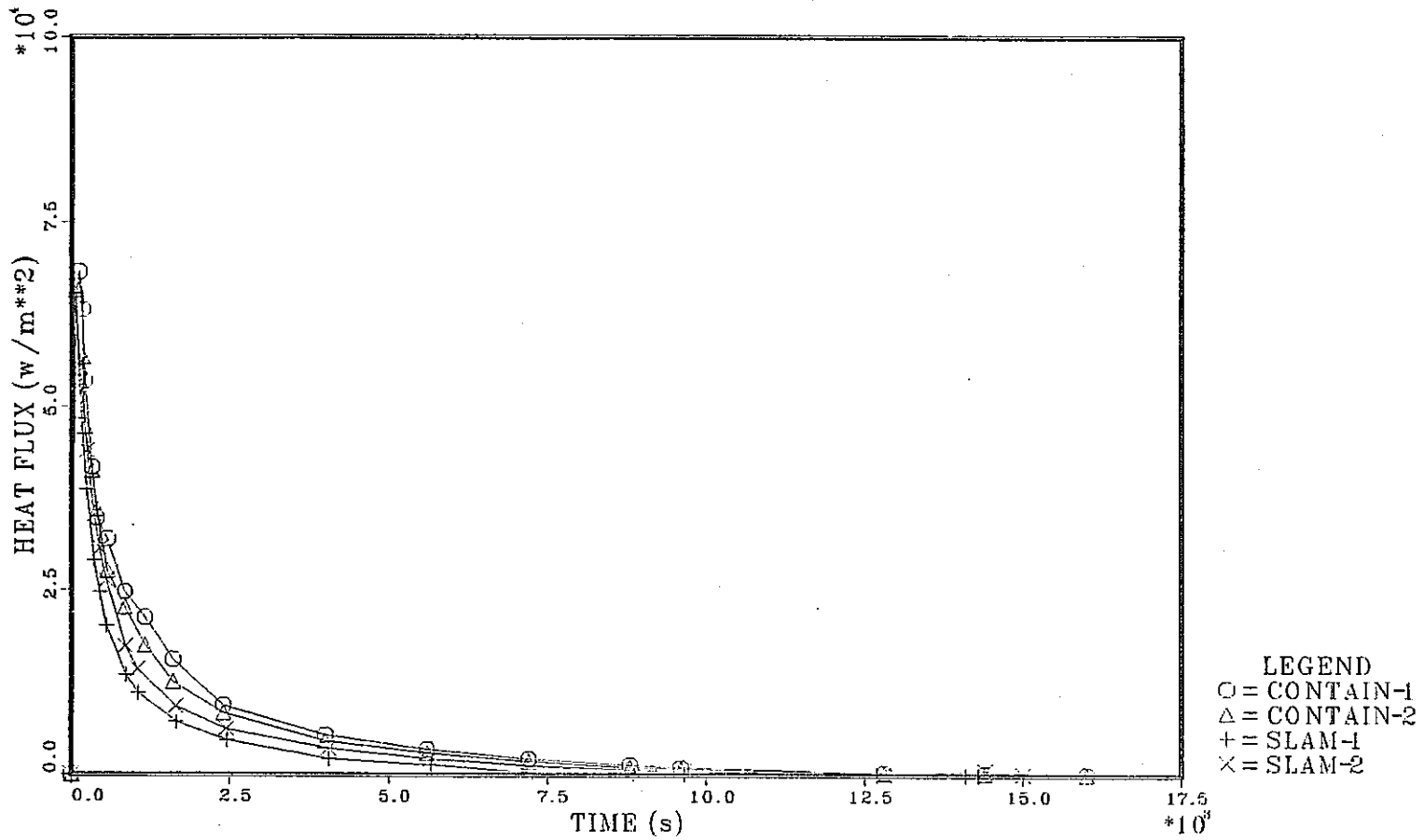


Figure 9.13 Comparison of Original SLAM and CONTAIN Calculations of Concrete Reaction Heat during Sodium-Concrete Reaction

CONCRETE SURFACE HEAT FLUX

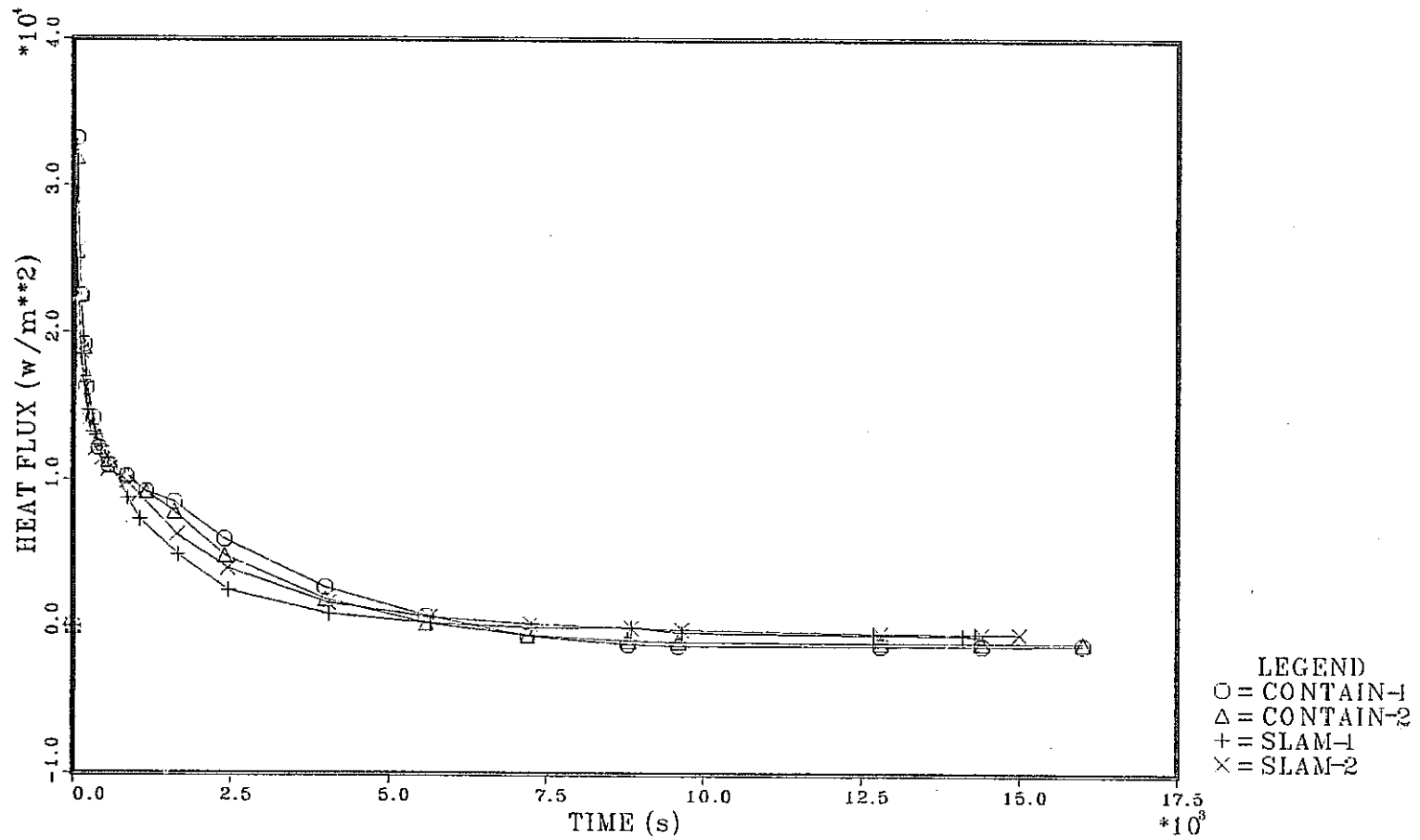


Figure 9.14 Comparison of Original SLAM and CONTAIN Calculation of Concrete Surface Heat Flux during Sodium-Concrete Reaction

WET-DRY INTERFACE HEAT FLUX

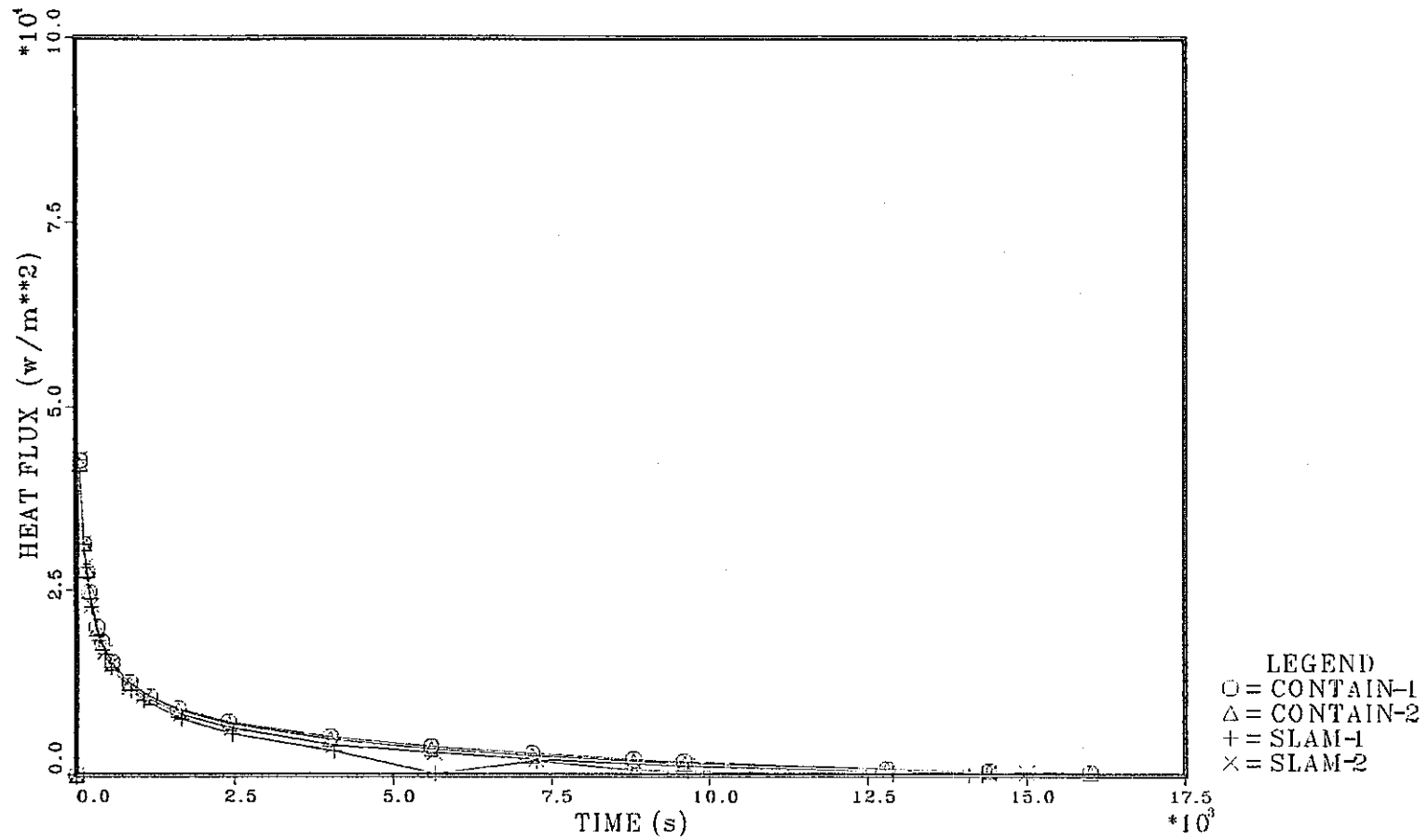


Figure 9.15 Comparison of Original SLAM and CONTAIN Calculations of Wet-Dry Interface Heat Flux during Sodium-Concrete Reaction

DRY ZONE HEAT SOURCES

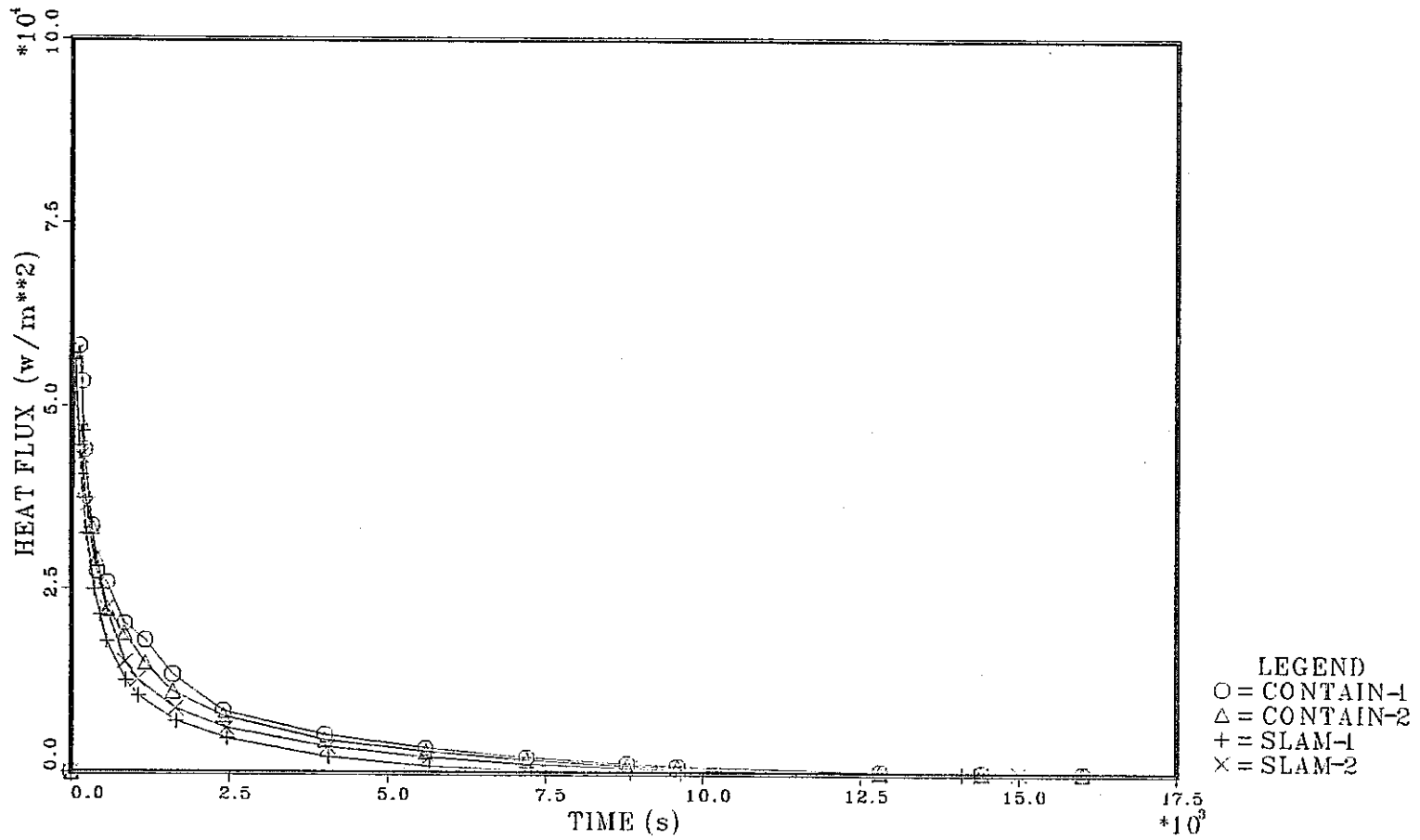


Figure 9.16 Comparison of Original SLAM and CONTAIN Calculations of Dry Zone Heat Sources during Sodium-Concrete Reaction

HEAT FLUX INTO WET ZONE

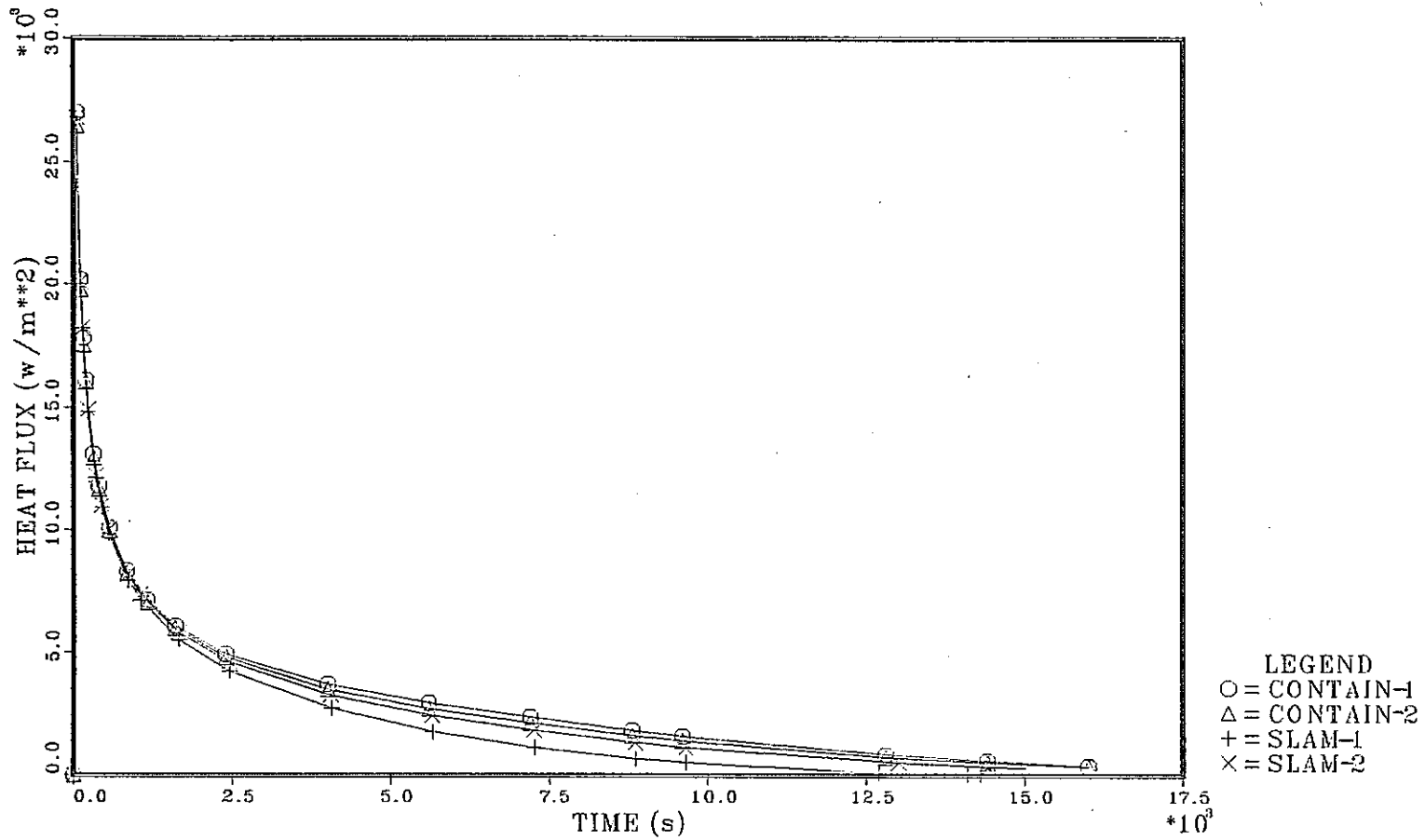


Figure 9.17 Comparison of Original SLAM and CONTAIN Calculations of Heat Flux into Wet Zone during Sodium-Concrete Reaction

9-2. Water Release Calculation

9-2.1 System and Conditions of Test Calculation

A system consisting of concrete, interm, sodium pool and atmosphere is chosen for the test calculation purpose.

The concrete is the limestone concrete, and its compositions are as shown in Table 9.1. The dimensions of the concrete and the initial conditions of interm, pool and atmosphere are presented in Figure 9.18

Two interm layers which are composed of FE are added to separate the sodium pool layer from the concrete layer.

The keyword 'H2O-MIGR' is specified to activate the water release calculation (see Table 9.4 .)

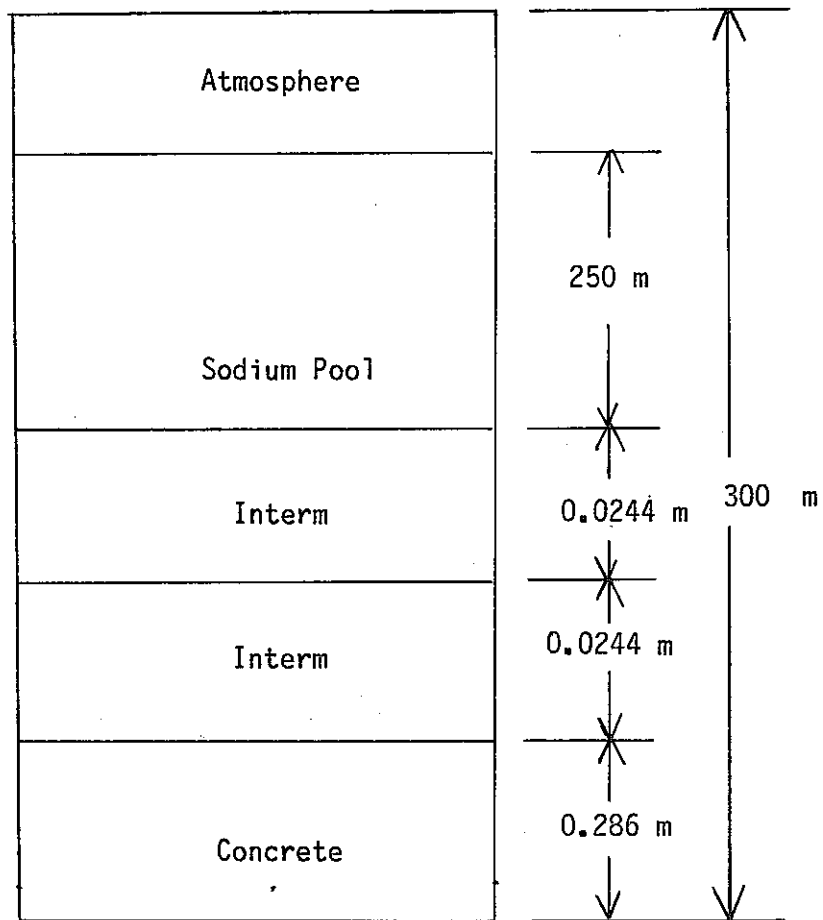


Figure 9.18 System for Water Release Test Calculation

Initial conditions

- Concrete Temp. = 293 (K)
- Interm Temp. = 500 (K)
- Pool Temp. = 923 (K)
- Atmos. Temp. = 500 (K)
- Atmos. Pres. = 0.1 (MPa)

```

INPUT <<<< IBM >>>> ECHO
INPUT <<<< && ----- WRC002----- >>>> ECHO
INPUT <<<< && CONTAIN TEST WRC002 - CHECK CALCULATION OF WATER RELEASE ROUTINE >>>> ECHO
INPUT <<<< && NA/CONCRETE INTERACTION >>>> ECHO
INPUT <<<< && - - - - - < GLOBAL > - - - - - >>>> ECHO
INPUT <<<< >>>> ECHO
INPUT <<<< CONTROL >>>> ECHO
INPUT <<<< NCELLS=1 >>>> ECHO
INPUT <<<< NTITL=3 >>>> ECHO
INPUT <<<< NTZONE=5 >>>> ECHO
INPUT <<<< EOI >>>> ECHO
INPUT <<<< >>>> ECHO
INPUT <<<< && ..... MATERIAL NAMES ..... >>>> ECHO
INPUT <<<< MATERIAL >>>> ECHO
INPUT <<<< COMPOUND NAL NAV NA2S1O3 NA2CO3 NA2O CAO MGCD3 CAC03 H2O NAOH >>>> ECHO
INPUT <<<< CO2 GRAPH SIO2 CONC MGO INERT FE N2 >>>> ECHO
INPUT <<<< >>>> ECHO
INPUT <<<< && ..... EDIT TIMES ..... >>>> ECHO
INPUT <<<< TIMES 360000. 0.0 4. 20. 200. 8.0 40.0 1000. >>>> ECHO
INPUT <<<< 16.0 400.0 4000. 80. 800. 10000. >>>> ECHO
INPUT <<<< 160.0 1600. 16000. 0.5 >>>> ECHO
INPUT <<<< && 640.0 6400. 80000. 1280. 12800. 128000. 0.5 >>>> ECHO
INPUT <<<< TITLE >>>> ECHO
INPUT <<<< CONTAIN TEST WRC002 >>>> ECHO
INPUT <<<< WATER RELEASE ROUTINE >>>> ECHO
INPUT <<<< VERSION 3 MOD-PNC01S CONTAIN 1.03 >>>> ECHO
INPUT <<<< >>>> ECHO
INPUT <<<< && ..... LIQUID METAL REACTOR ..... >>>> ECHO
INPUT <<<< FAST >>>> ECHO
INPUT <<<< >>>> ECHO
INPUT <<<< && ..... ACTIVATE ATMOSPHERIC THERMODYNAMICS ..... >>>> ECHO
INPUT <<<< THERMO >>>> ECHO
INPUT <<<< >>>> ECHO
INPUT <<<< PRLow-CL >>>> ECHO
INPUT <<<< LONGEDT=5 >>>> ECHO
INPUT <<<< >>>> ECHO
INPUT <<<< && - - - - - < CELL=1 > - - - - - >>>> ECHO
INPUT <<<< >>>> ECHO
INPUT <<<< CELL=1 >>>> ECHO
INPUT <<<< >>>> ECHO
INPUT <<<< CONTROL JCONC=30 JPOOL=1 JINT=2 EOI >>>> ECHO
INPUT <<<< >>>> ECHO
INPUT <<<< && ..... CELL VOLUME + HEIGHT ..... >>>> ECHO
INPUT <<<< GEOMETRY 1000. 300. >>>> ECHO
INPUT <<<< >>>> ECHO
INPUT <<<< && ..... 2 ATMOSPHERE COMPONENTS - N2 AND NAV - TEMP. AND PRESS. .... >>>> ECHO
INPUT <<<< ATMOS=1 1.0E+5 500. N2=1.0 >>>> ECHO
INPUT <<<< >>>> ECHO
INPUT <<<< LOW-CELL >>>> ECHO
INPUT <<<< GEOMETRY .29** && FLOOR AREA >>>> ECHO
INPUT <<<< CONCRETE COMPOS=1 CONCRETE = LIME 239.3 >>>> ECHO
INPUT <<<< TEMP = 293.0 >>>> ECHO
INPUT <<<< PHYSICS >>>> ECHO
INPUT <<<< H2O-MIGR >>>> ECHO
INPUT <<<< EOI && TERMINATE THE PHYSICS >>>> ECHO
INPUT <<<< EOI && TERMINATE THE LAYER >>>> ECHO
INPUT <<<< INTERM LAY-NAME = LINER TEMP = 500. >>>> ECHO
INPUT <<<< COMPOS = 1 FE = 55.76 >>>> ECHO
INPUT <<<< EOI && TERMINATES THE LINER LAYER >>>> ECHO
INPUT <<<< INTERM LAY-NAME = LINER TEMP = 500. >>>> ECHO
INPUT <<<< COMPOS = 1 FE = 55.76 >>>> ECHO
INPUT <<<< PHYSICS FAILURE = 1000. >>>> ECHO
INPUT <<<< EOI && TERMINATES THE PHYSICS OPTION >>>> ECHO
INPUT <<<< EOI && TERMINATES THE LINER LAYER >>>> ECHO
INPUT <<<< POOL TEMP = 923.0 >>>> ECHO
INPUT <<<< COMPOS 1 NAL = 57640. >>>> ECHO
INPUT <<<< EOI && TERMINATES THE POOL LAYER >>>> ECHO
INPUT <<<< BC 293.0 1.0E+05 >>>> ECHO
INPUT <<<< EOI && TERMINATE THE LOW-CELL INPUT >>>> ECHO
INPUT <<<< EOF >>>> ECHO

```

List 9.4 CONTAIN Input Data for Water Release Calculation

9-2.2 Results of of Calculation

The calculational results are plotted in Figures 9.19 through 9.31. Although comparison of calculated results with experiments or other calculations is not performed, it is seemed that the results are quite reasonable. Therefore, the merging of the water release module must be successful.

In addition, the another calculation has been conducted considering the liner failure. The capability of transition from the water release module to the sodium-concrete reaction module were found to function well.

CELL 1 VOL. AVG. TEMPS. FOR LOW-CELL

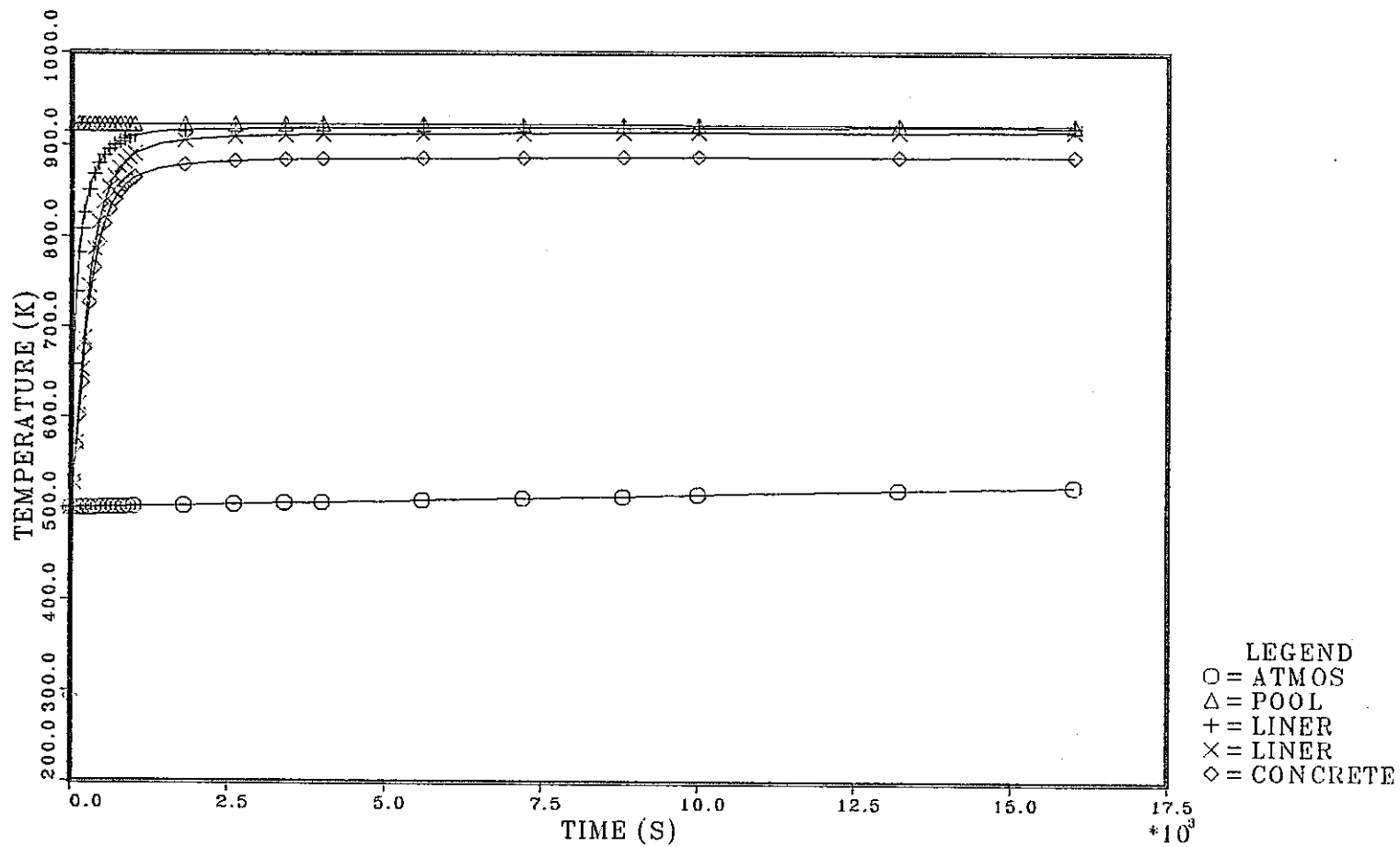


Figure 9.19 Results of Low-Cell Temperature of Water Release Calculation

WET-DRY INTERFACE TEMPERATURE

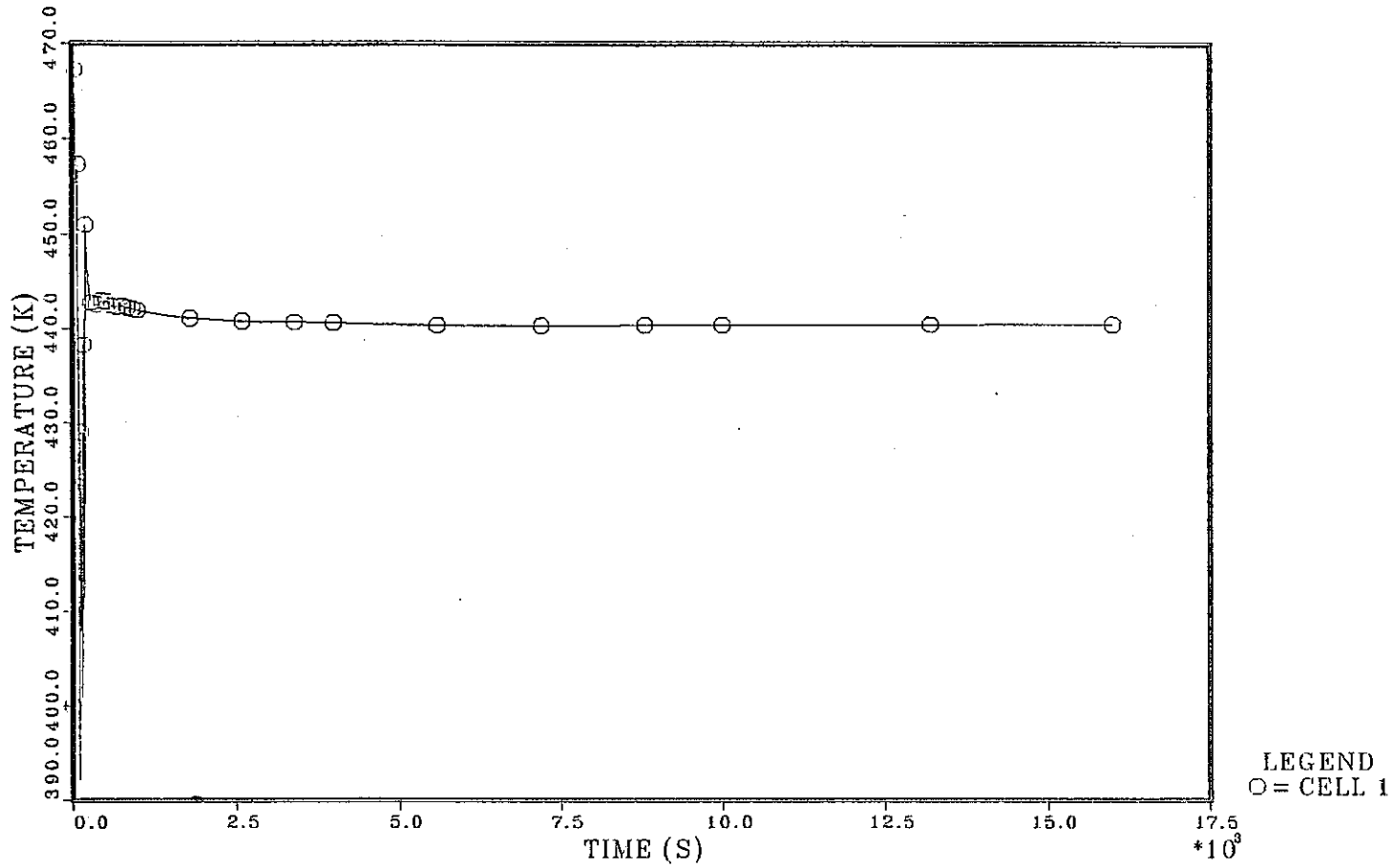


Figure 9.20 Result of Wet-Dry Interface Temperature of Water Release Calculation

WET-DRY INTERFACE H2O PRESS.

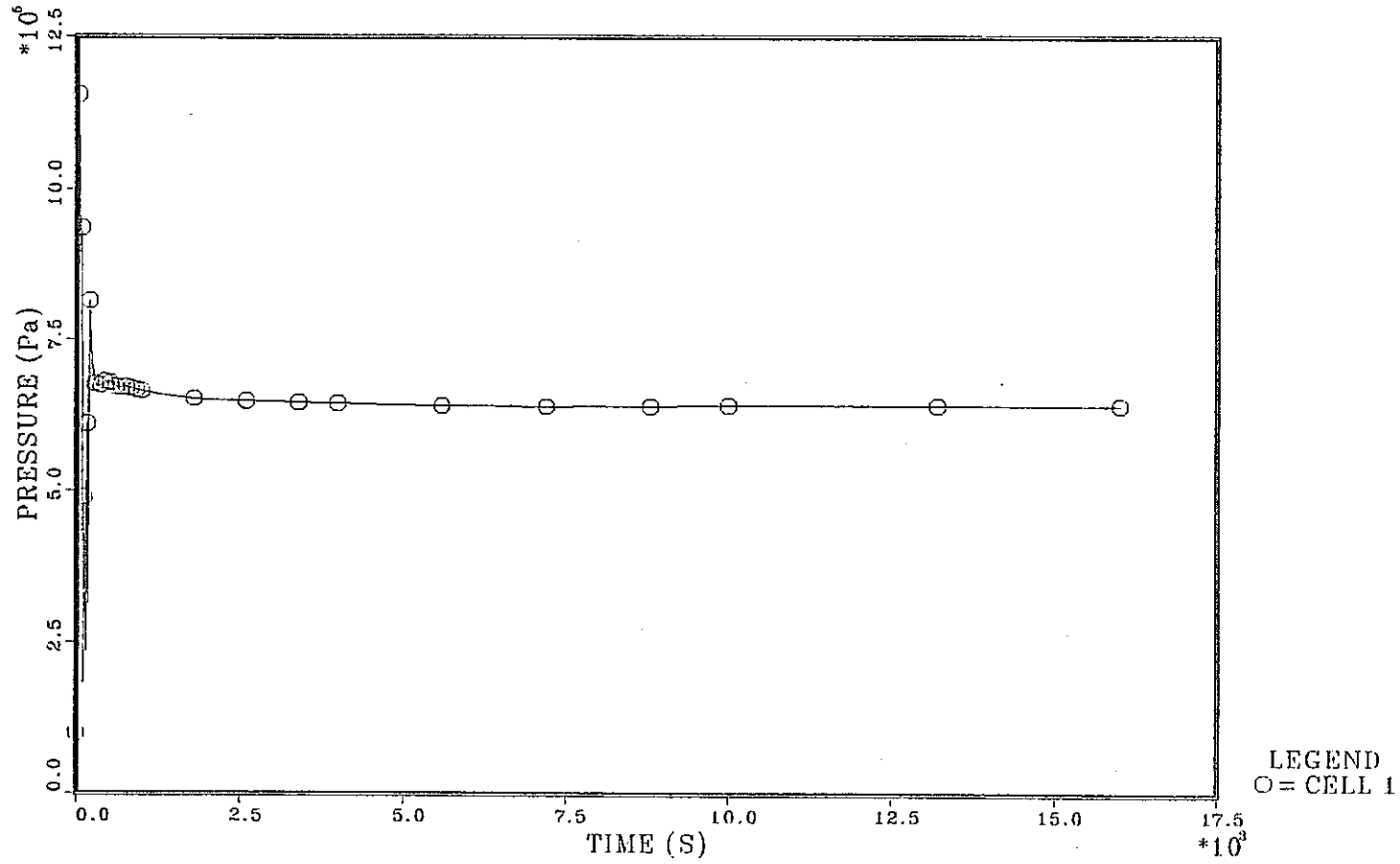


Figure 9.21 Result of Wet-Dry Interface H2O Pressure of Water Release Calculation

DRY ZONE GROWTH RATE

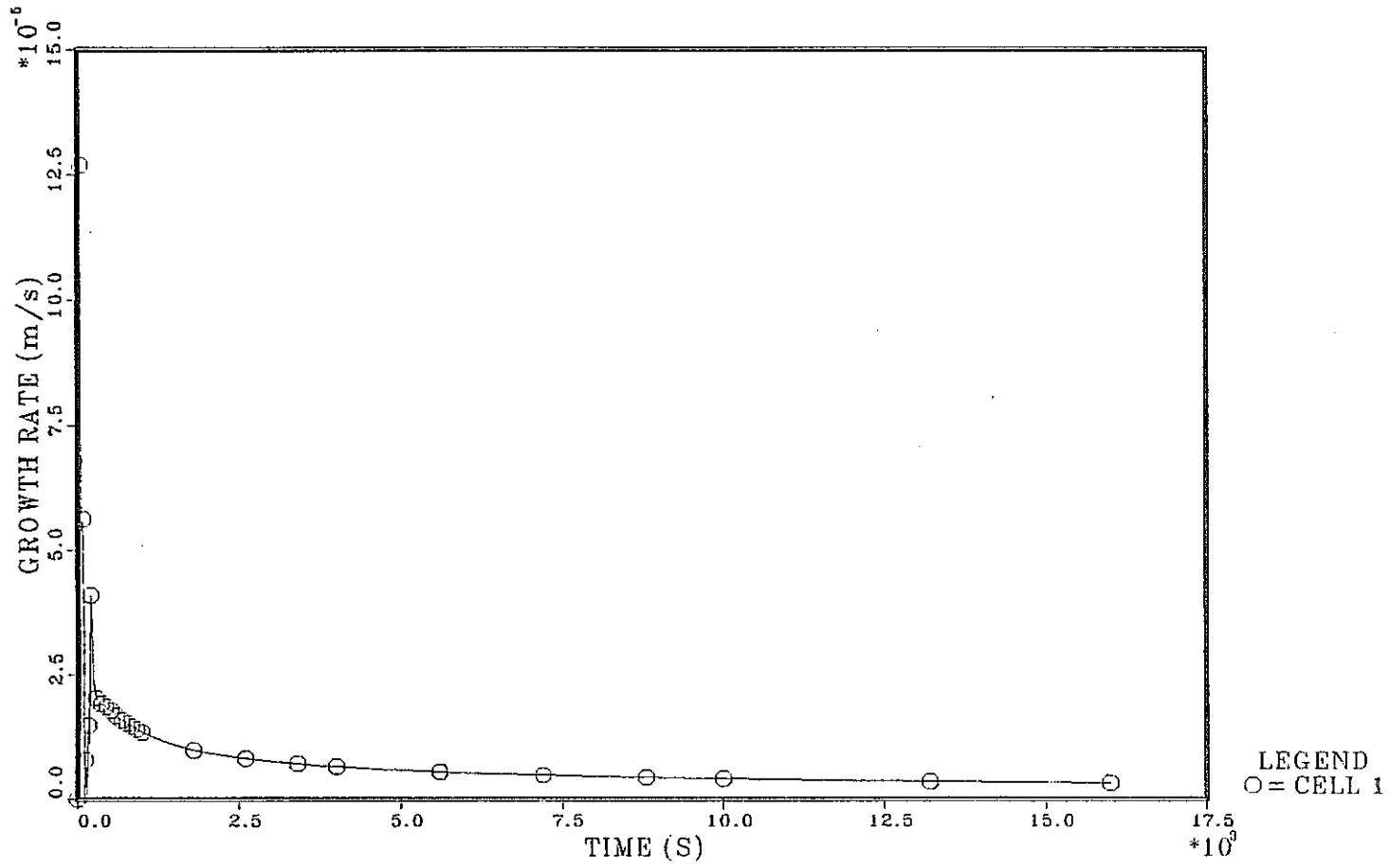


Figure 9.22 Result of Dry Zone Growth Rate of Water Release Calculation

DRY ZONE DEPTH

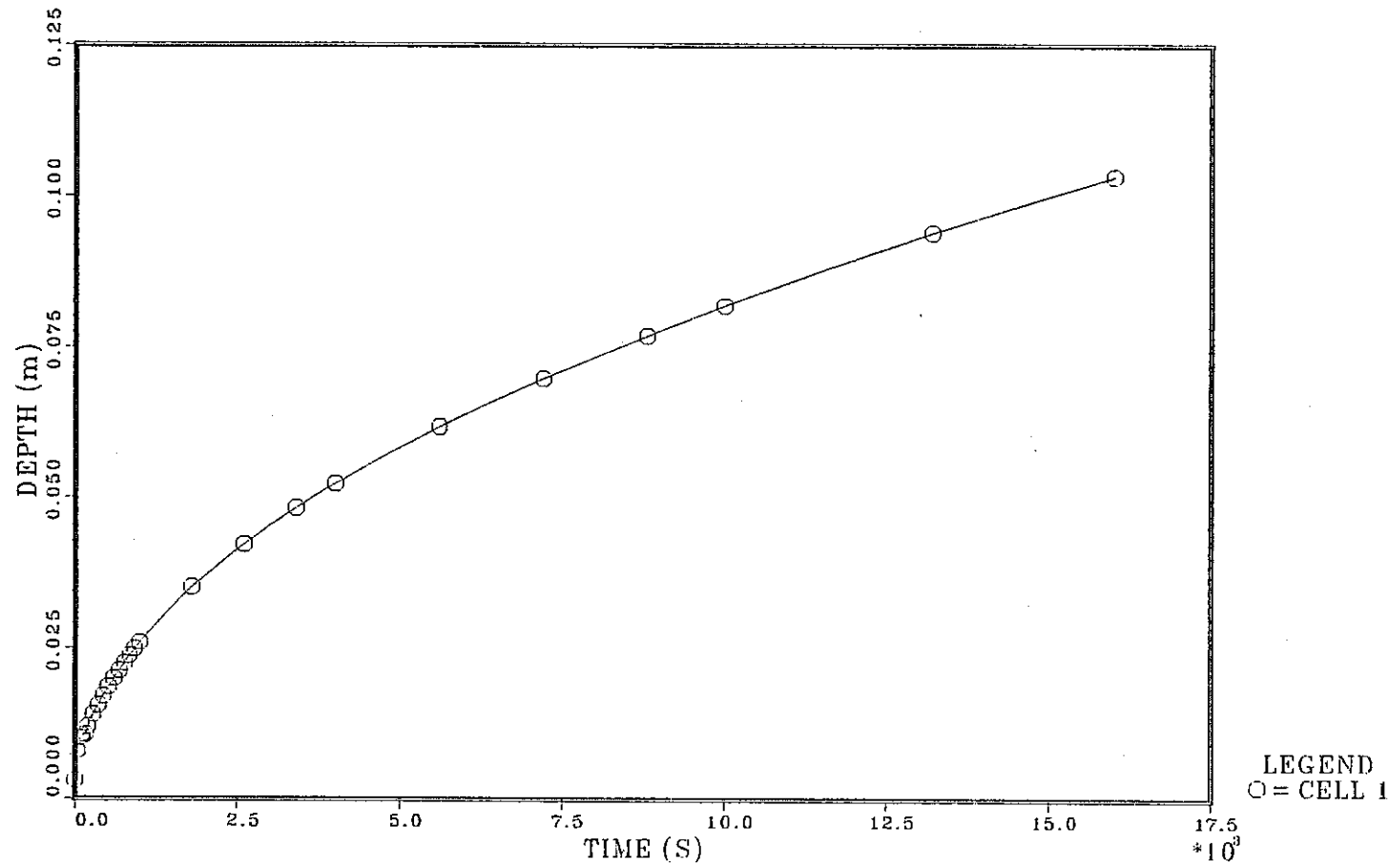


Figure 9.23 Result of Dry Zone Depth of Water Release Calculation

INTERFACE H2O EVAP. FLUX

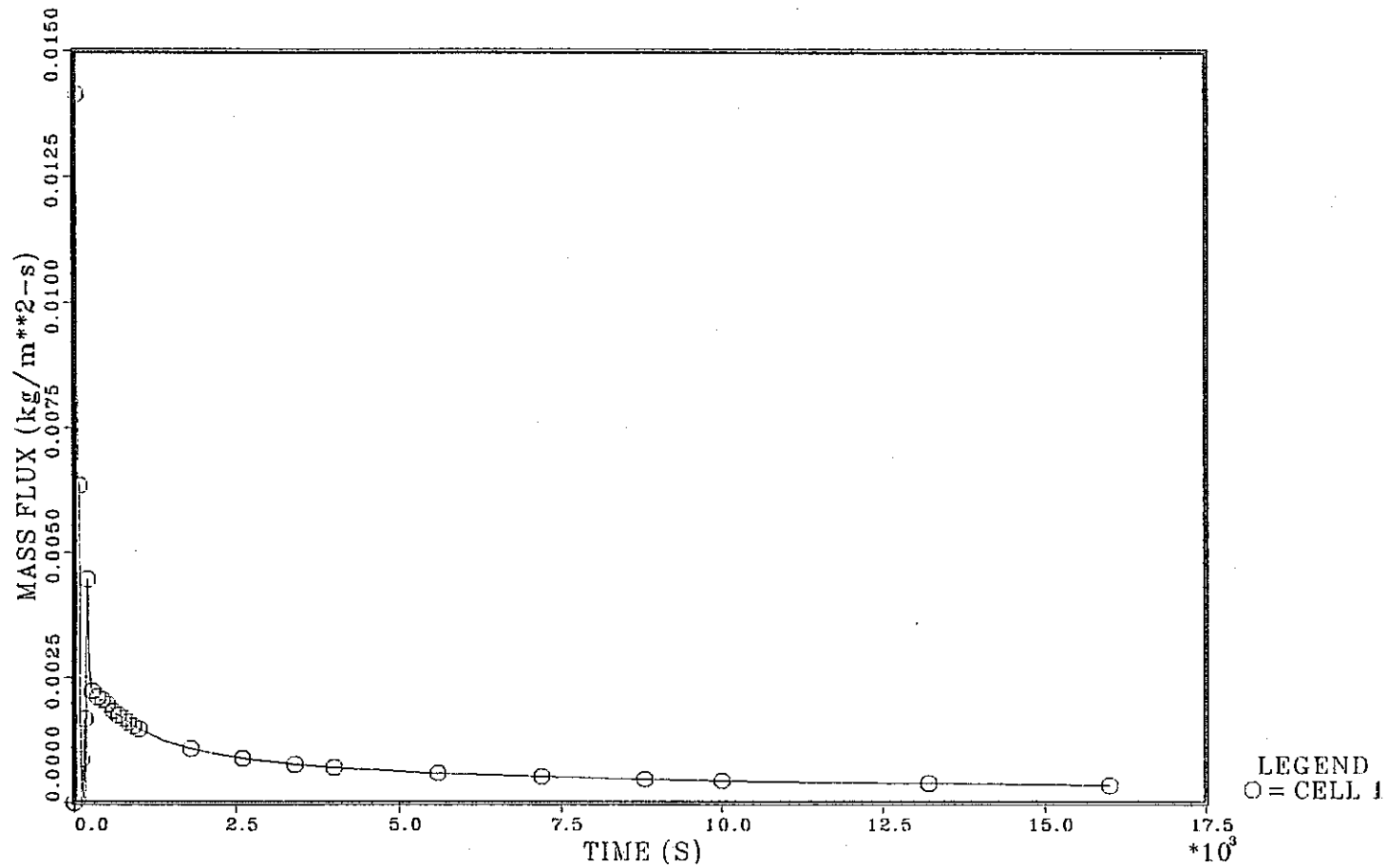


Figure 9.24 Result of Interface H2O Evaporated Flux of Water Release Calculation

DRY ZONE BOUND H2O FLUX

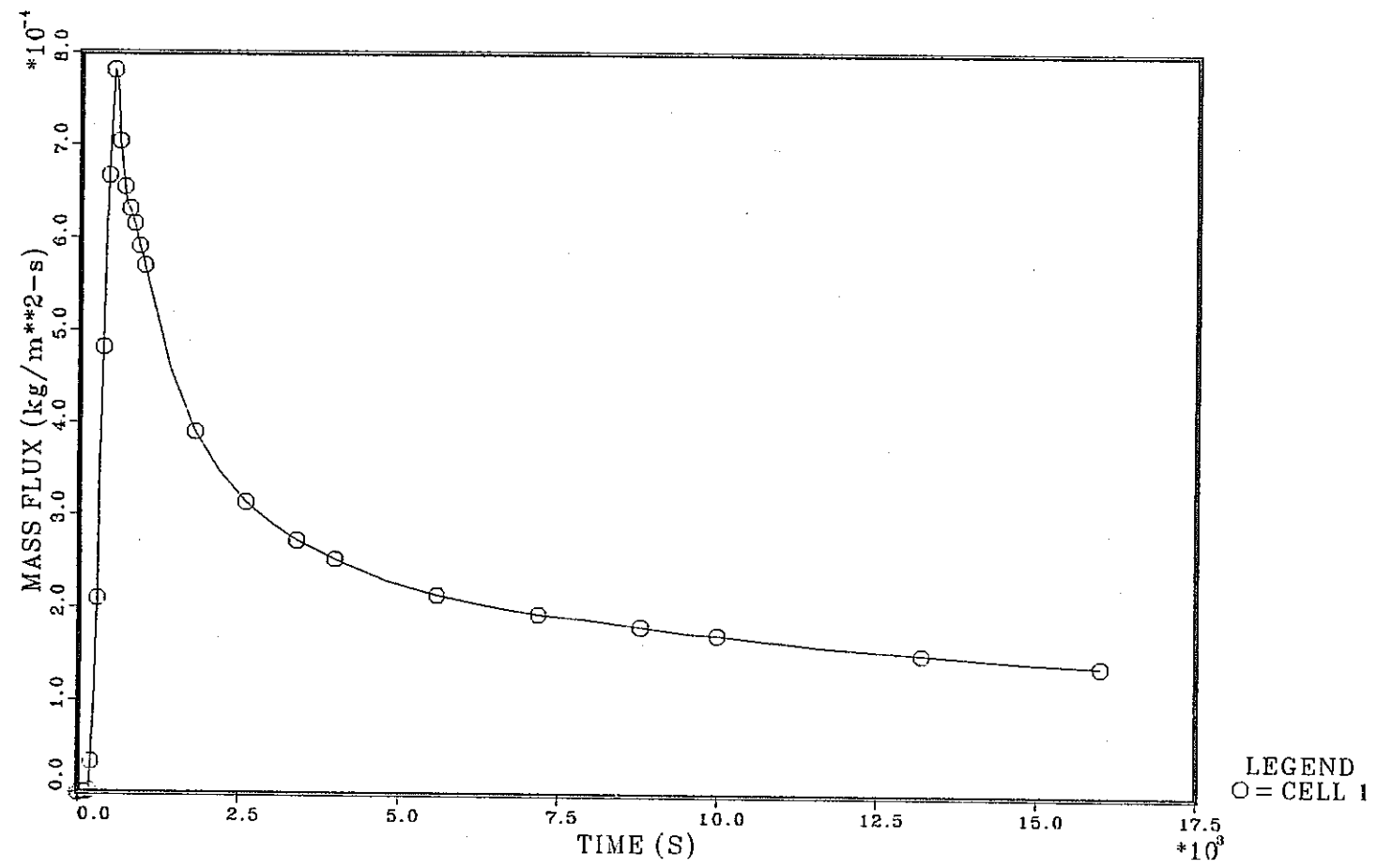


Figure 9.25 Result of Dry Zone Bound H2O Flux of Water Release Calculation

INTEGRAL OF BOUND + EVAP H2O

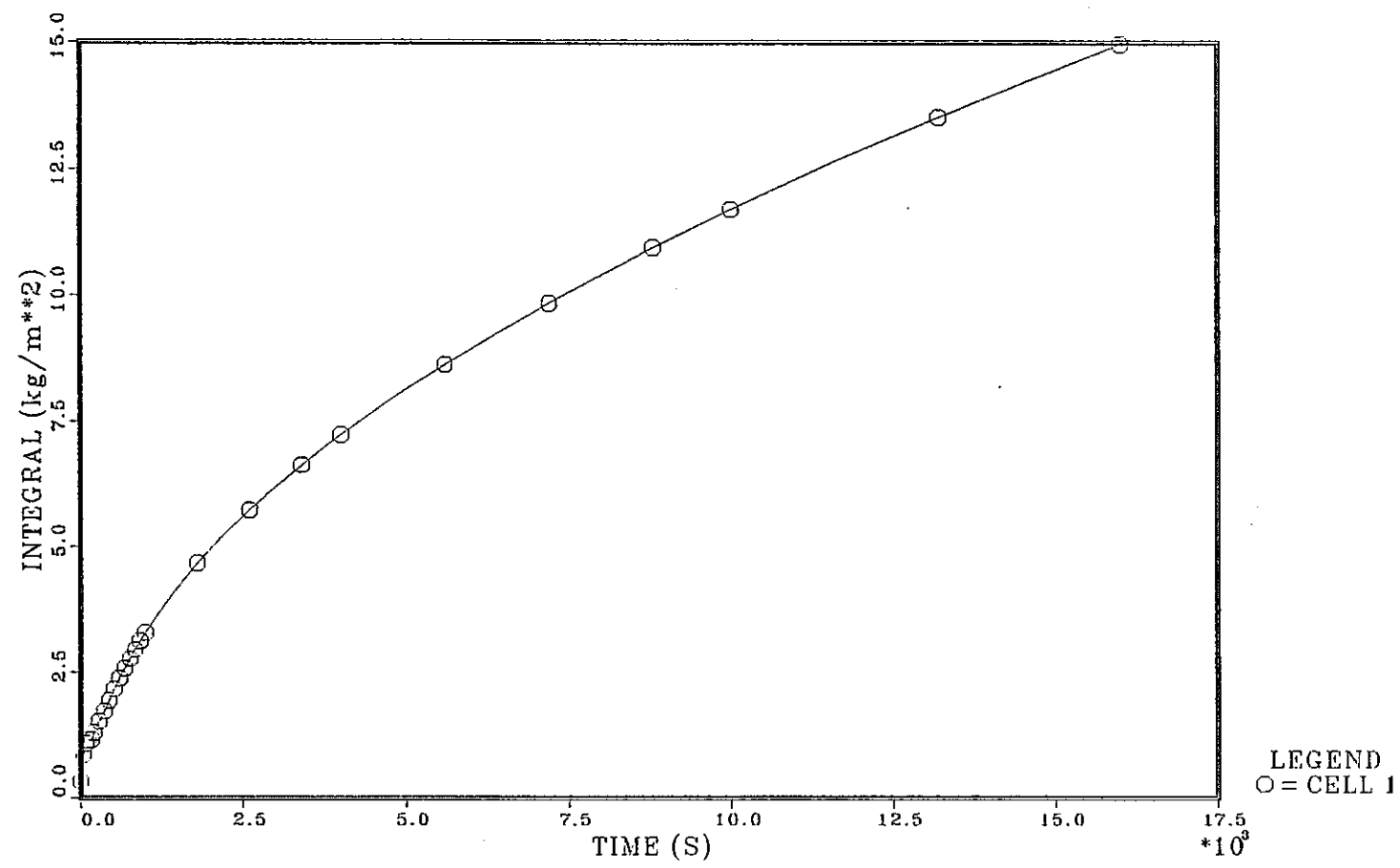


Figure 9.26 Result of Integral of Bound + Evaporated H2O of Water Release Calculation

DRY ZONE BOUND CO2 FLUX

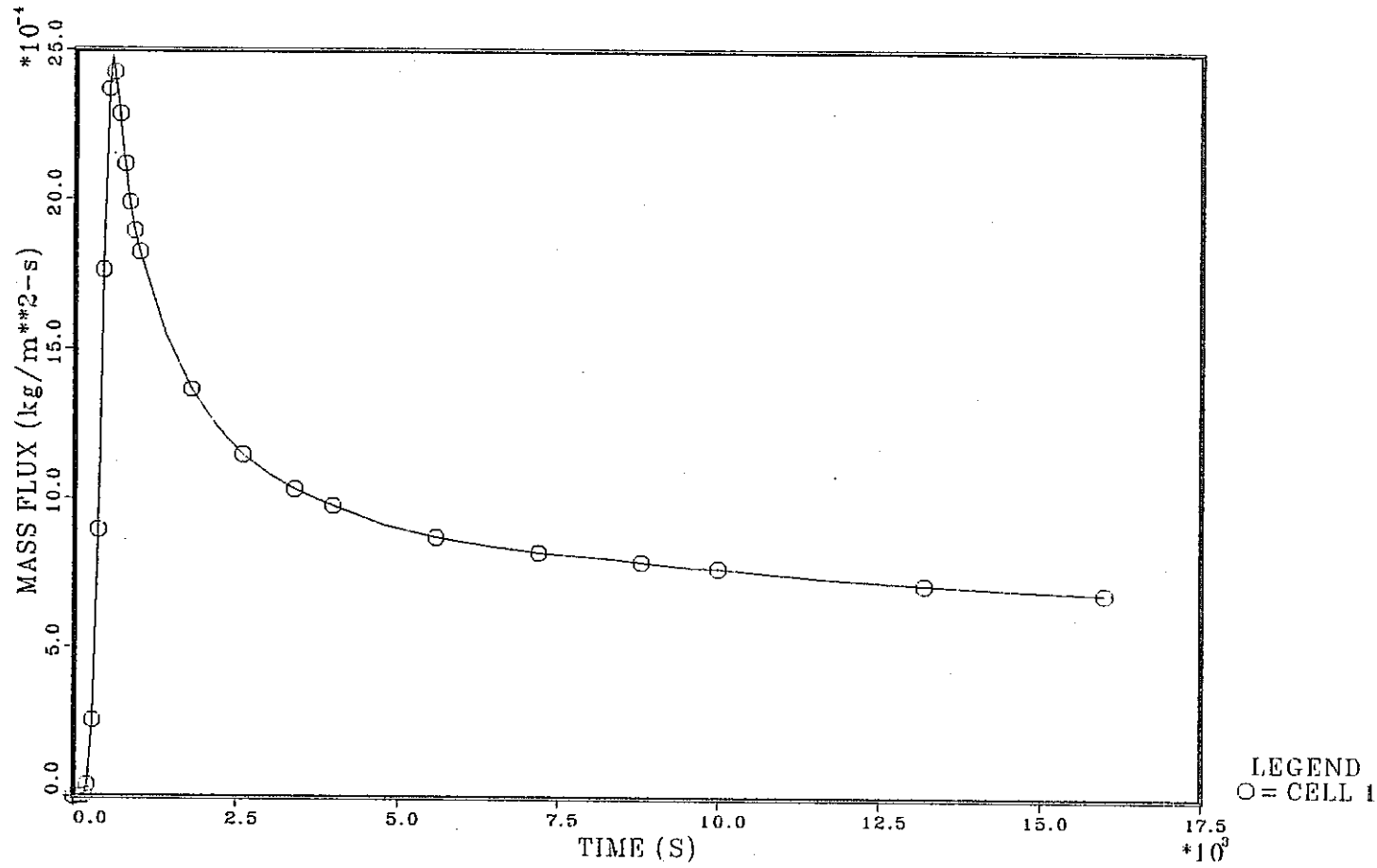


Figure 9.27 Result of Dry Zone Bound CO2 Flux of Water Release Calculation

INTEGRAL OF BOUND CO2

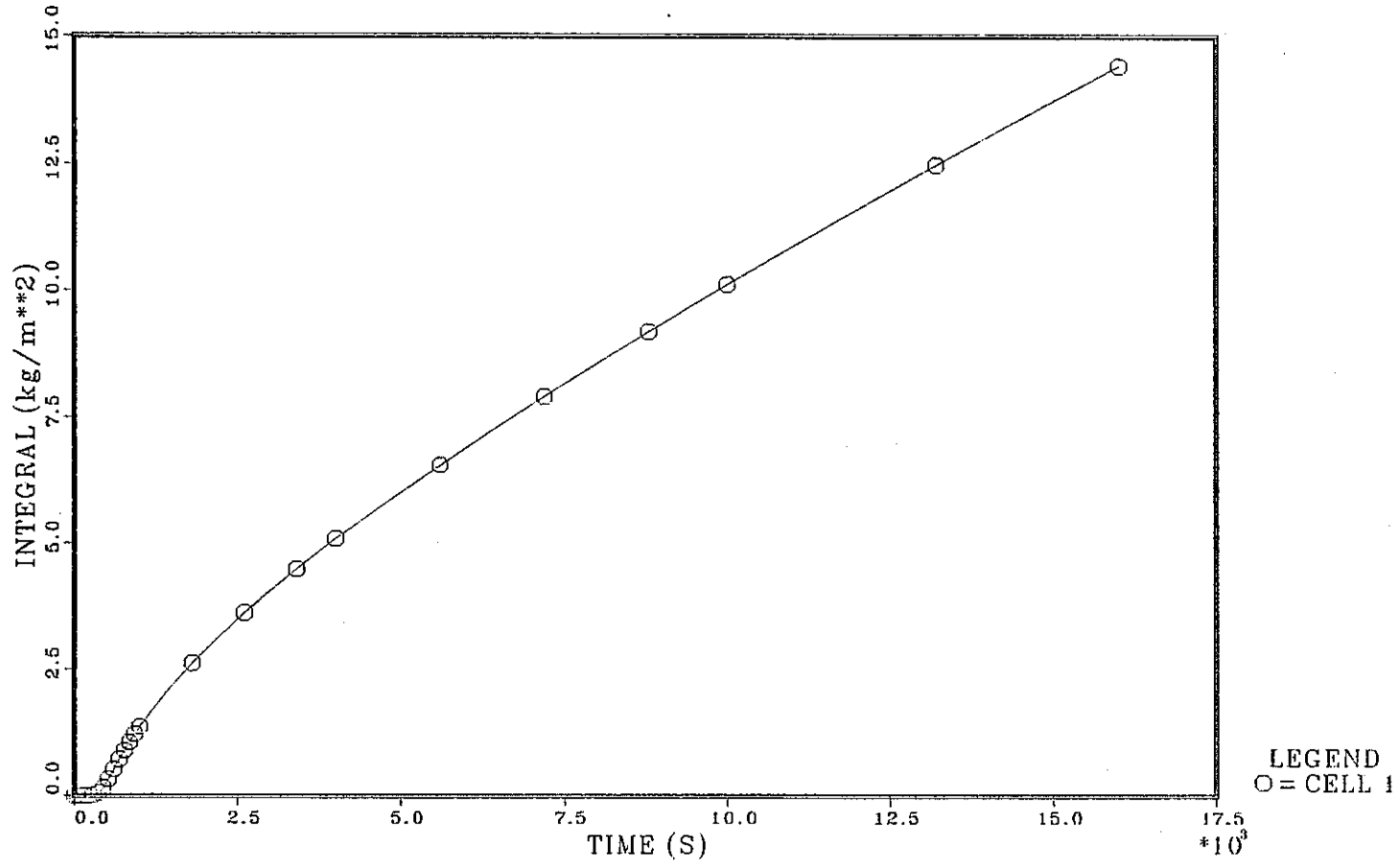


Figure 9.28 Result of Integral of Bound CO2 of Water Release Calculation

CONCRETE SURFACE HEAT FLUX

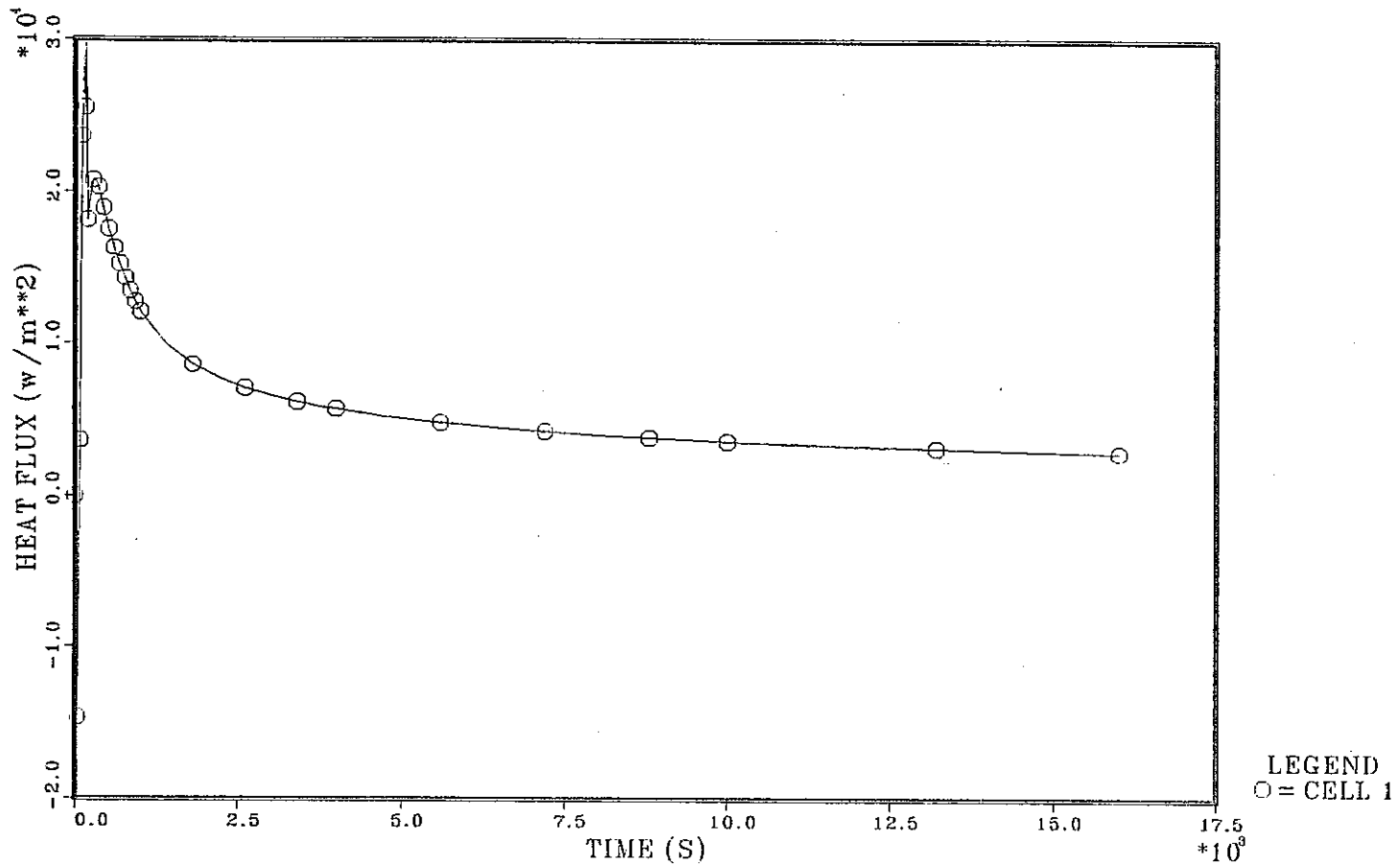


Figure 9.29 Result of Concrete Surface Heat Flux of Water Release Calculation

WET-DRY INTERFACE HEAT FLUX

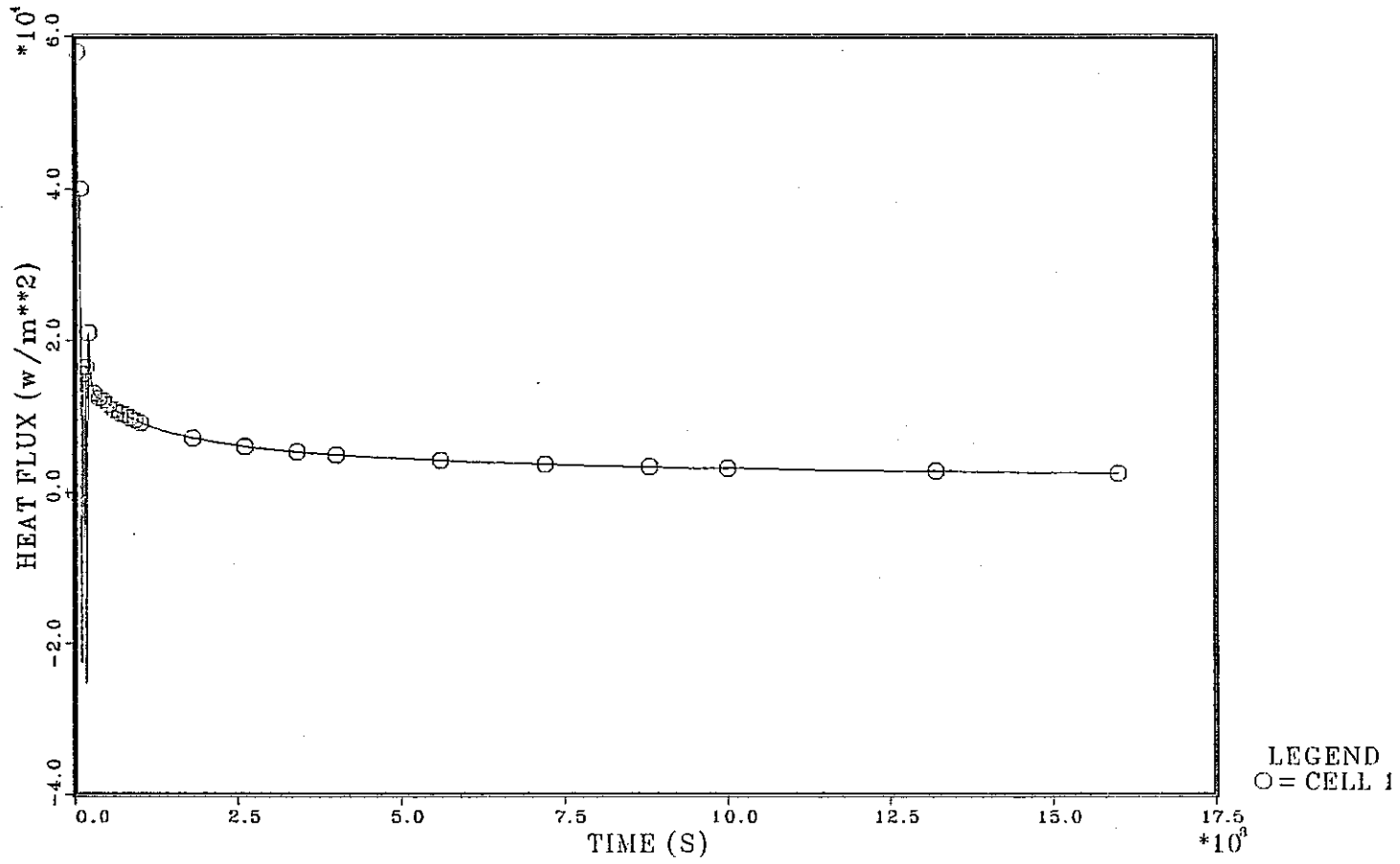


Figure 9.30 Result of Wet-Dry Interface Heat Flux of Water Release Calculation

HEAT FLUX INTO WET ZONE

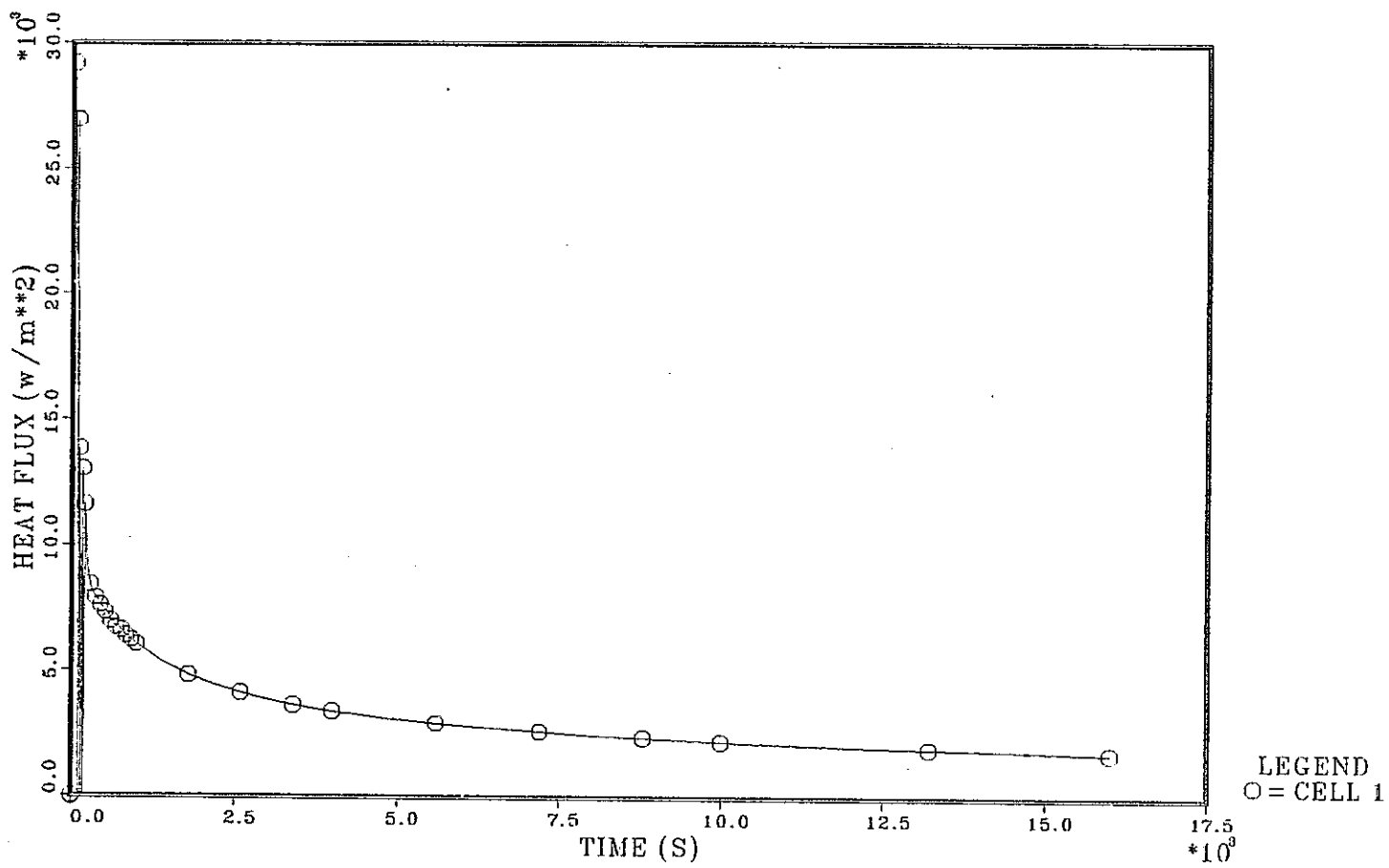


Figure 9.31 Result of Heat Flux into Wet Zone of Water Release Calculation

10. Future Tasks

The work to incorporate the sodium-concrete reaction module and the water release module into the CONTAIN code have been completed. The work should be continued, however, from the following two standpoints:

- 1) To confirm the accuracy of CONTAIN in comparison with experimental data.
- 2) To improve the capability to allow simulations for Japanese concrete.