PNCT N952 85-30

NOT FOR PUBLICATION

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

November, 1985

OARAI ENGINEERING CENTER

POWER REACTOR AND NUCLEAR FUEL DEVELOPMENT CORPORATION

NOT FOR PUBLICATION
PNC TN952 85-30
Nov. 1985

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

List of Figures and Tables

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.

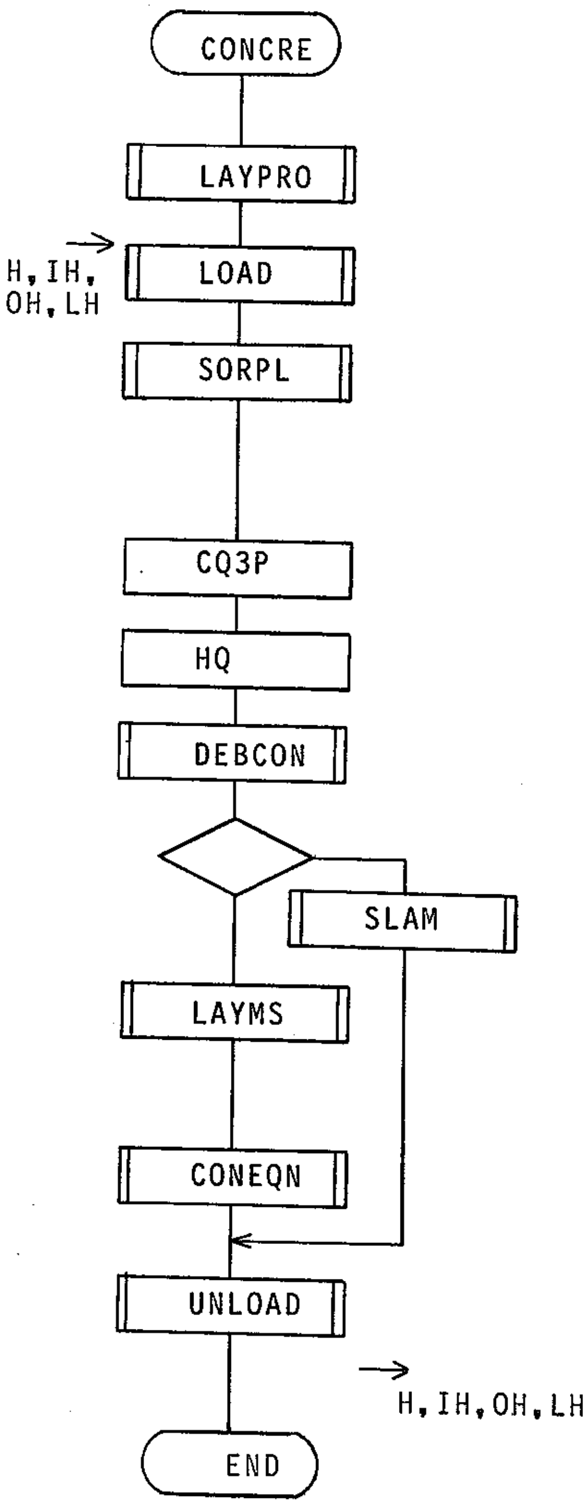
Actually, the sodium-concrete reaction code SLAM is incorporated in the current CONTAIN code. But, since it is perfectly isolated from the other modules, its interaction with the other calculation modules can not be evaluated. A full integration of SLAM with other modules is needed to utilize CONTAIN 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.



Defines thermal conductivity and specific heat.

Restores data from Storage Array

Allocates source for material and/or energy, if any, to nodes, in accordance with the volume ratio.

Calculates heat generation for each node.

Calculates heat transfer coefficients.

Calculates the debris-concrete reaction.

Calculates the sodium-concrete reaction.

Calculates the mass, average specific heat, thickness and average density of the node.

Calculates the temperature distribution in the non-reacted concrete zone.

Stores the calculation results in Storage Array.

Figure 2.1 Calculation Flow Diagram for CONTAIN Concrete Reaction Module

Table2.1 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	$Cp = \sum Wi\ Cpi$	Constant
Thermal Conductivity	$Tk = \sum Wi\ Tki$ (Wi=weight ratio)	Tk = Tka*T+Tkb (Tka,Tkb;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.

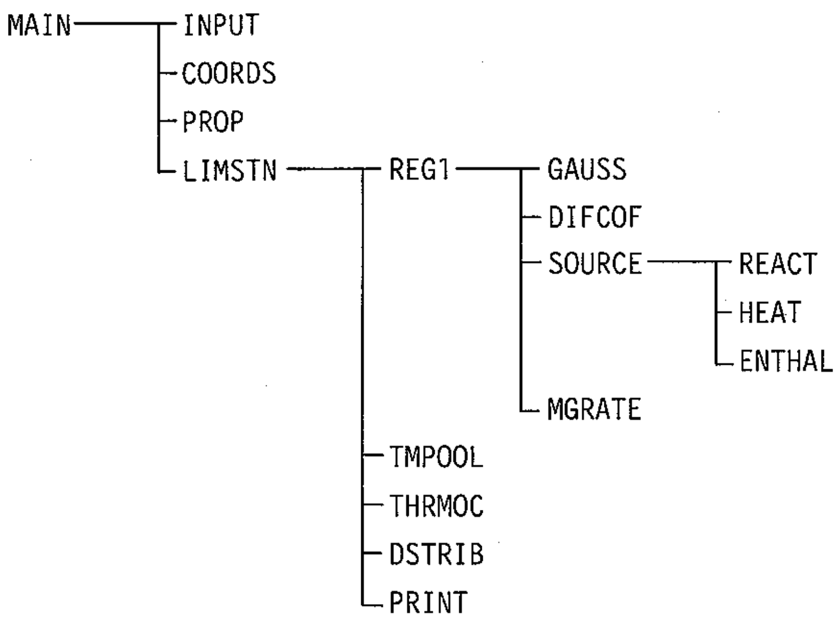


Figure 3.1 Configuration of the Original SLAM code

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 (TMPOOL), 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)

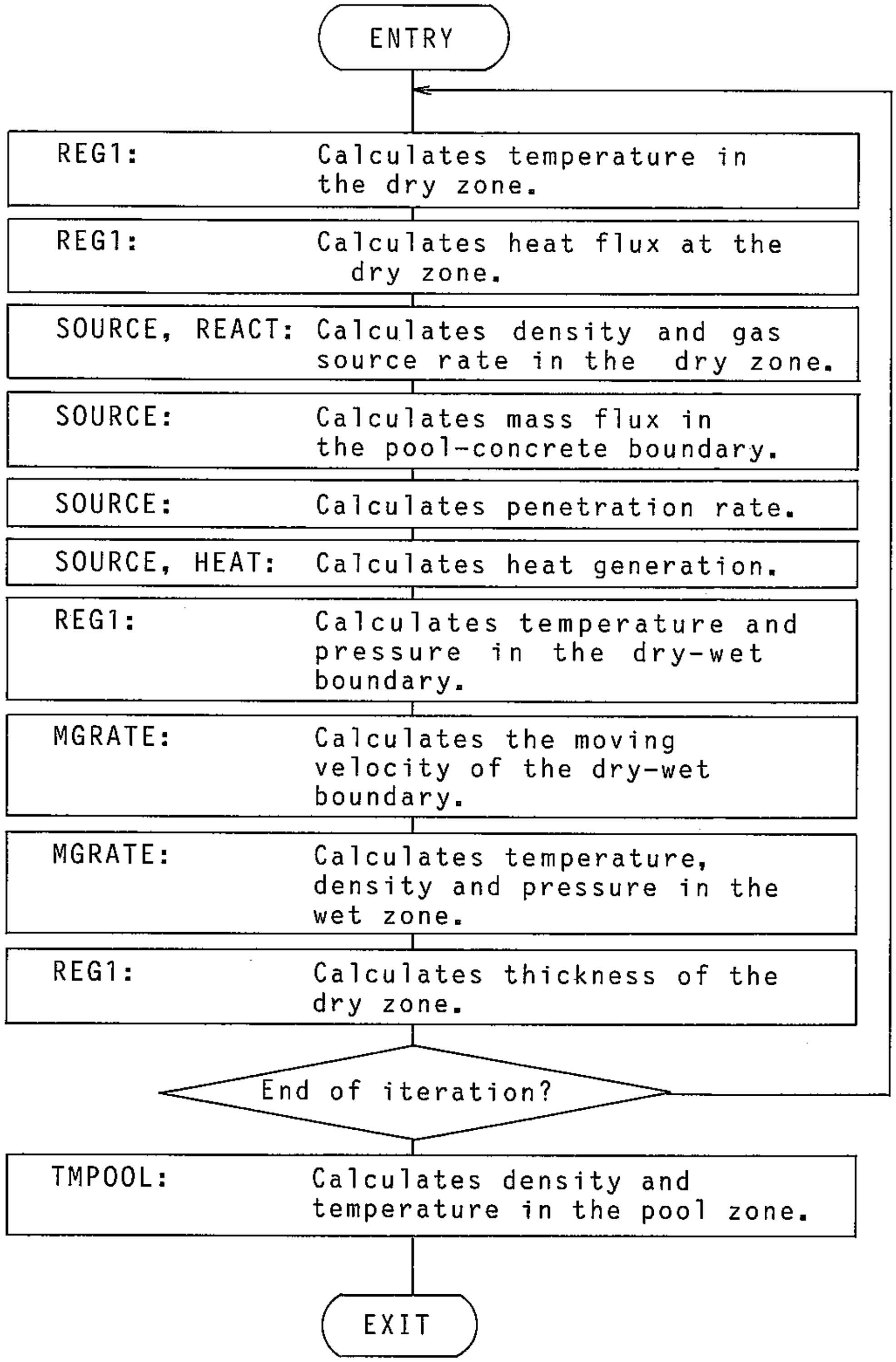


Figure 3.2 Calculation Flow Diagram of Original SLAM

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.

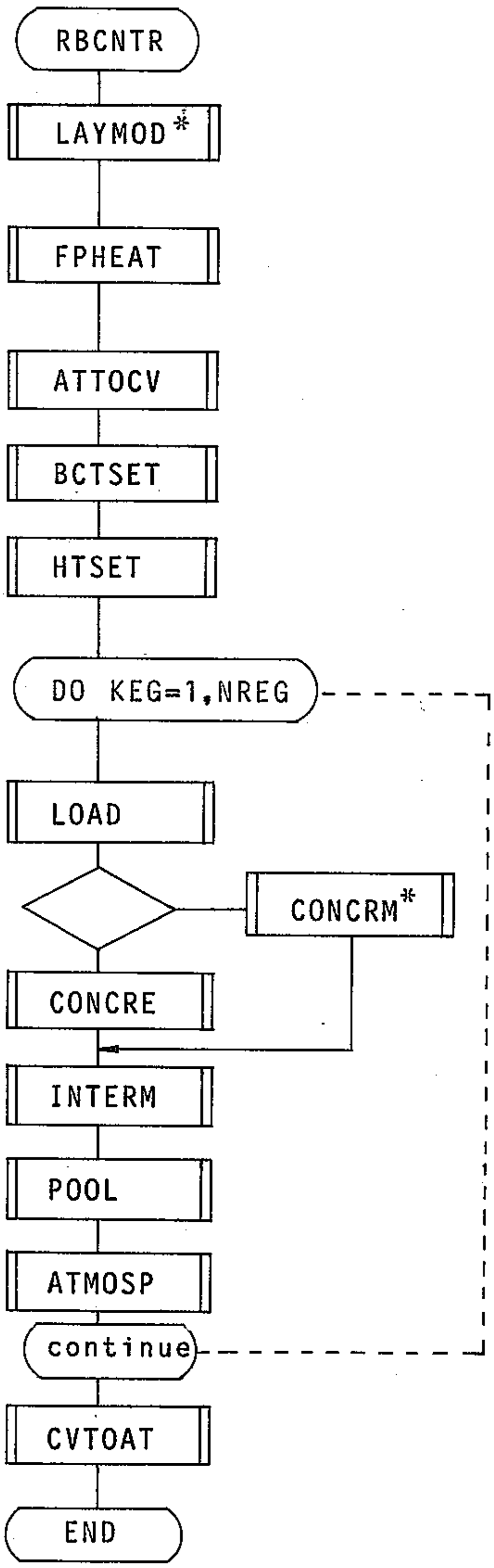
4-5. Concrete Module

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.



Lower Cell Routine

Reconstructs the layer structure when liner failure occurs

Calculates the distribution of FP heat source.

Transfers data to the cavity from the atmosphere.

Resets the layer boundary temperature.

Calculates the layer-to-layer heat transfer coefficient.

Sets in Common the layer data corresponding to the current zone number KREG.

(* : newly added)

Concrete routine for the sodium-concrete reaction.

Concrete routine for the debris-concrete reaction.

Performs calculations for the intermediate layer.

Performs calculations for the pool layer.

Performs calculations for the atmosphere.

Transfers the mass and energy to the atmosphere from the cavity.

Figure 4.1 Flow Diagram of Lower Cell Control Routine

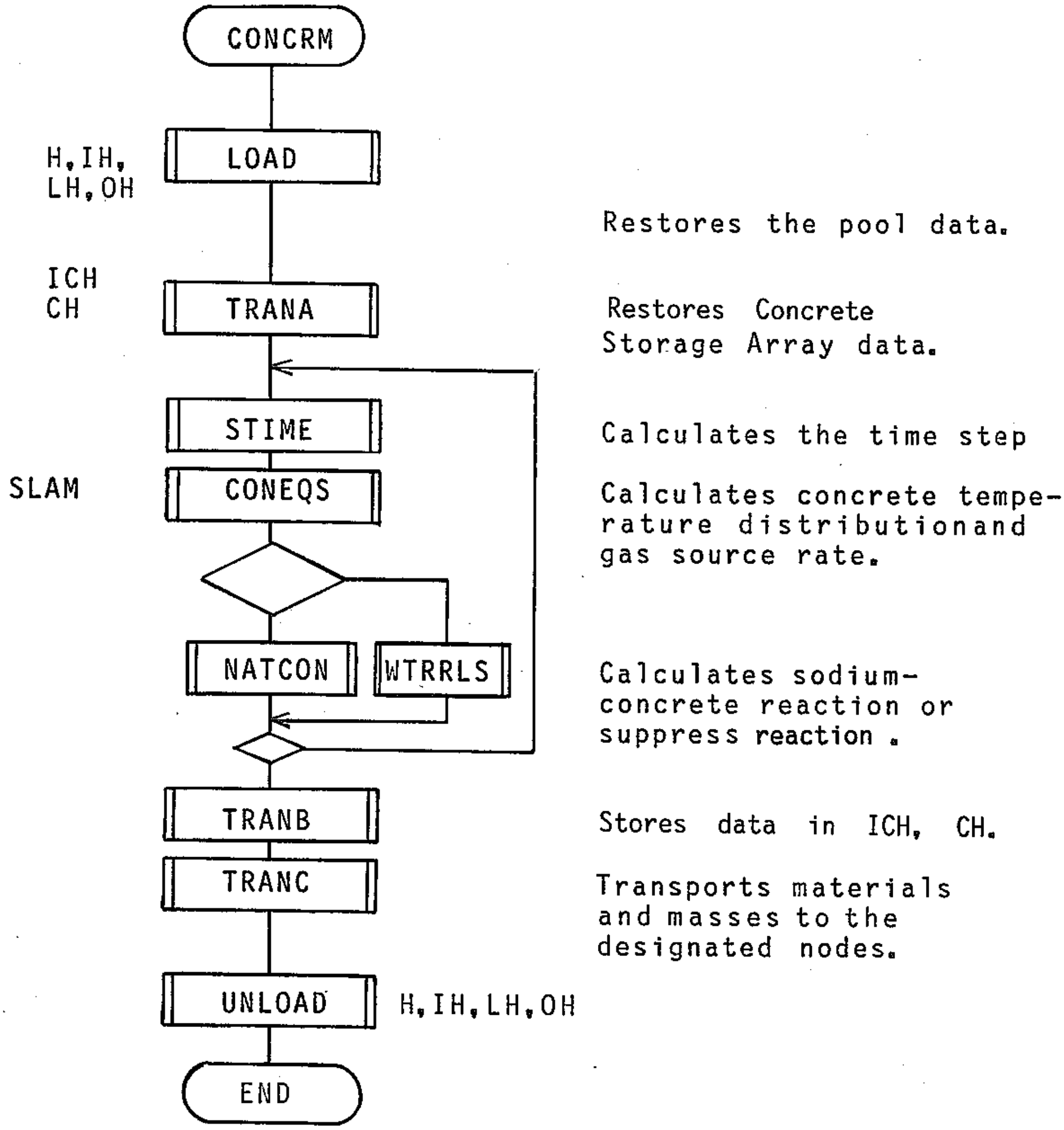


Figure 4.2 Concrete Module Calculation Flow

5. Design of Subroutines

5-1. 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	SLCOOR	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. DSTRIB		(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 ; stoichemical coefficient of reactants
COFP ; stoichemical 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.

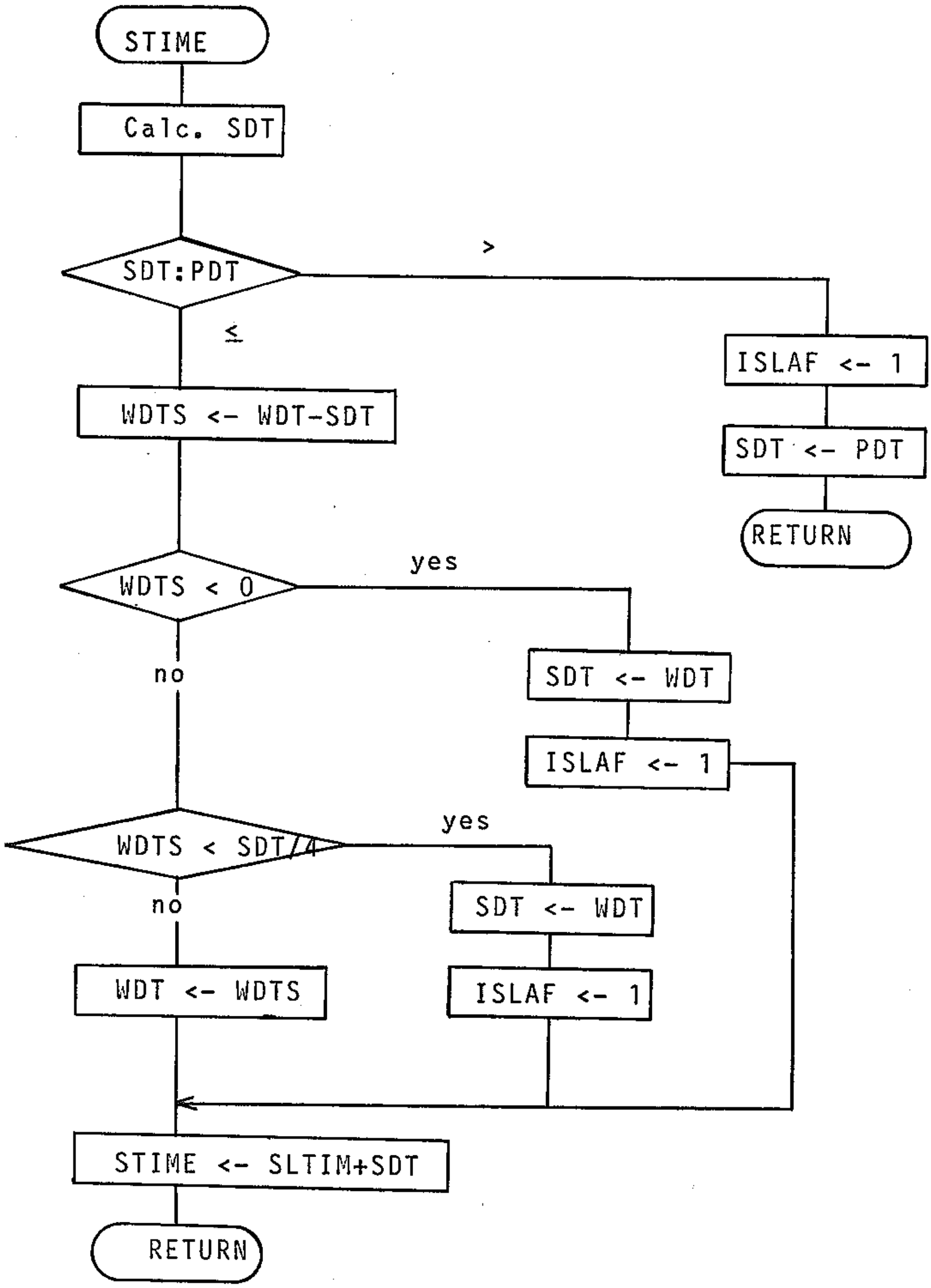


Figure 5.1 Flow Diagram of STIME

(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/m2/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	m2
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/m3
CRMO(*)	R	I	CRMO = CRM(t-dt)	kg/m3
TBCL(*)	R	I/0	layer boundary temperature (bottom)	K
TBCU(*)	R	I/0	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	-
OCTBNY(*)	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	O	thermal conductivity coefficient	W/m2/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/m3
PTA(*)	R	I	node average temperature in the pool layer	K
PTA(*)	R	I	node average temperature 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/m3

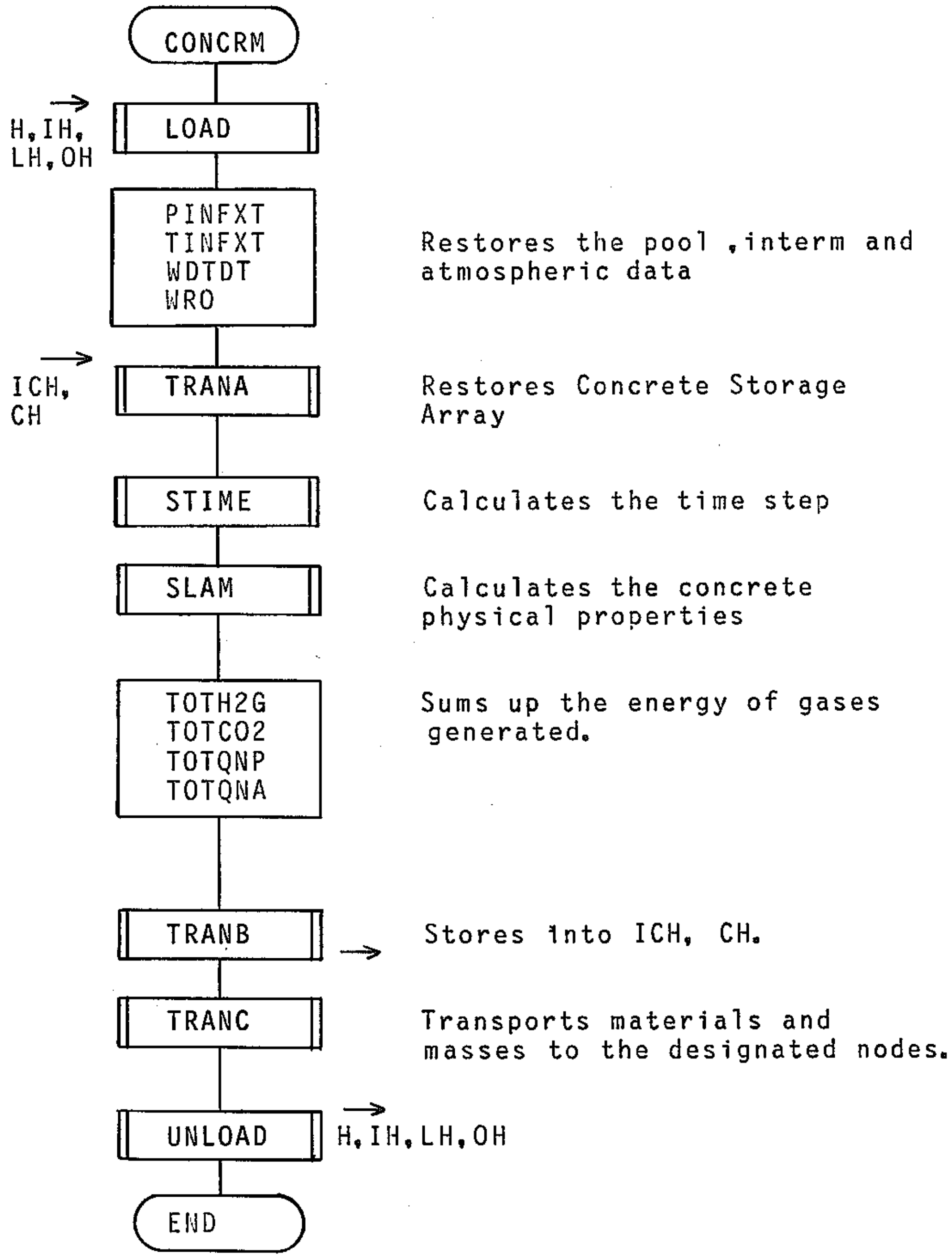


Figure 5.2 Flow Diagram of CONCRM

(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/m3
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/m2/sec

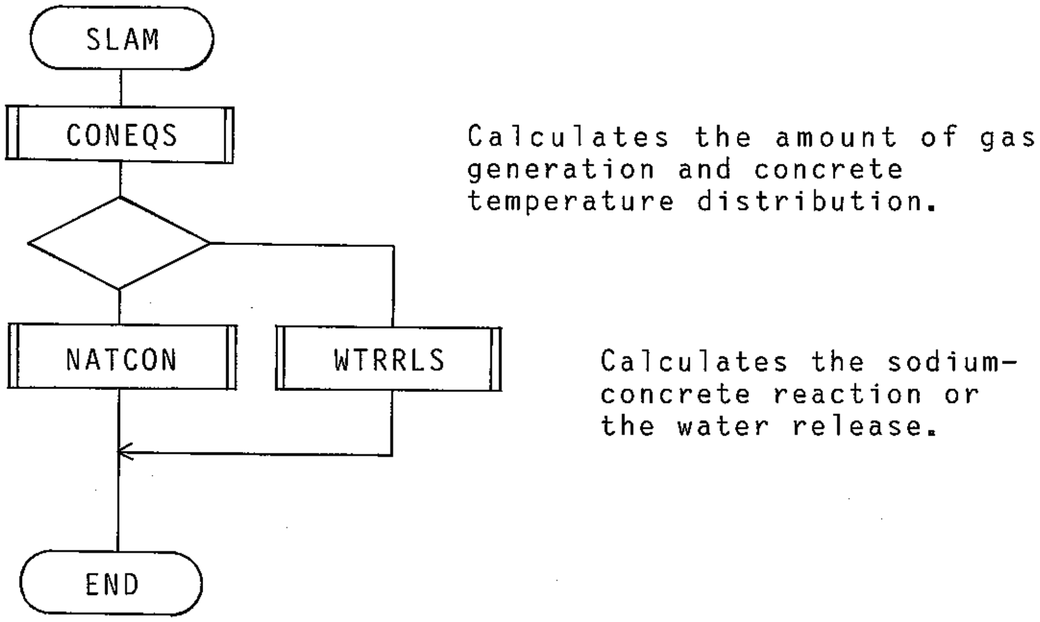


Figure 5.3 Flow Diagram of SLAM

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
DEL1O	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/m2/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/m3/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/m2
AMO(12)	R	O	polynomial expansion coefficient	-
SINT(51)	R	O	integral of source terms	kg/m3/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.

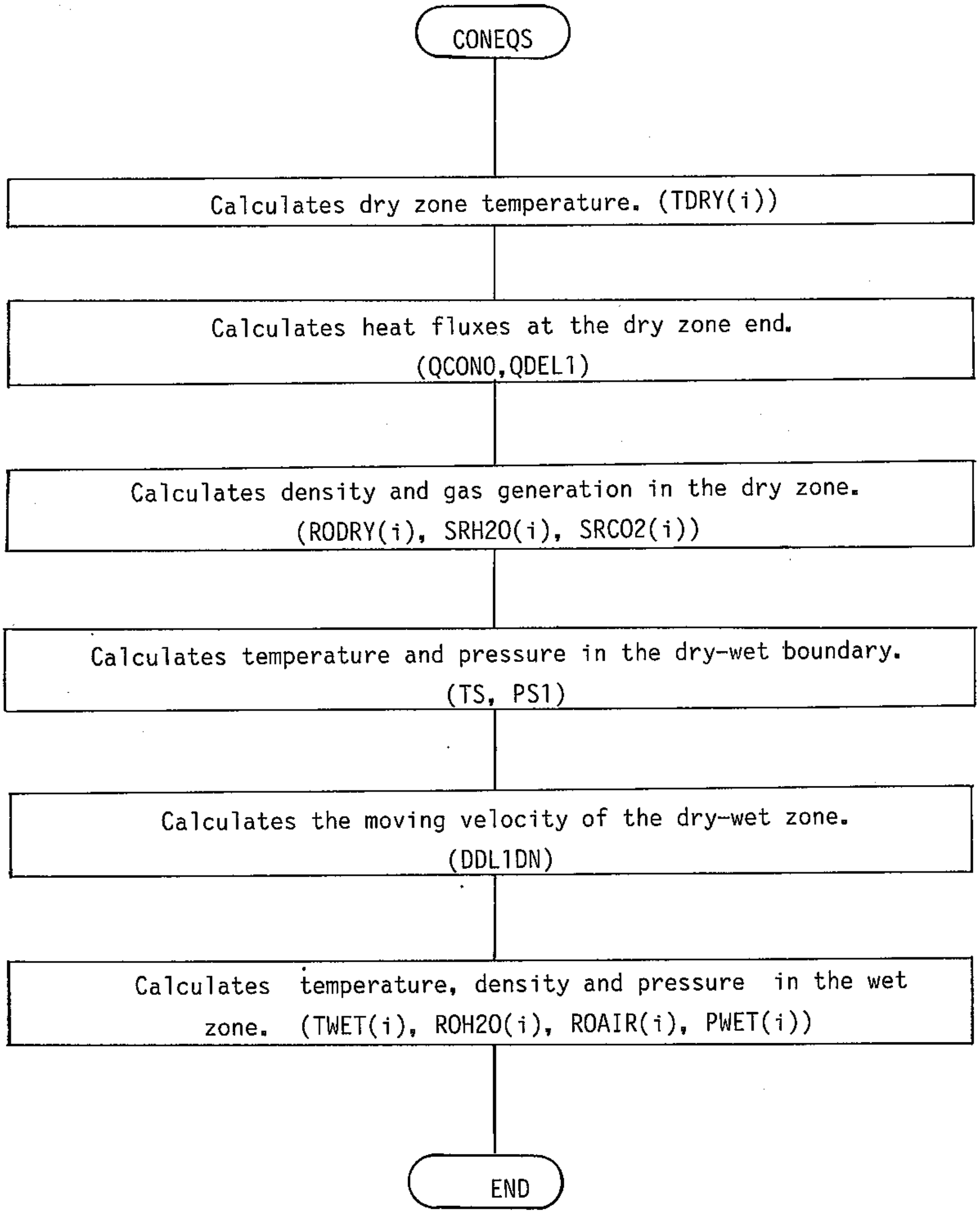


Figure 5.4 Flow Diagram of CONEQS

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/m3
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/m2/s
PINFXT	R	I	atmospheric pressure	Pa
TINFXT	R	I	atmospheric temperature	K

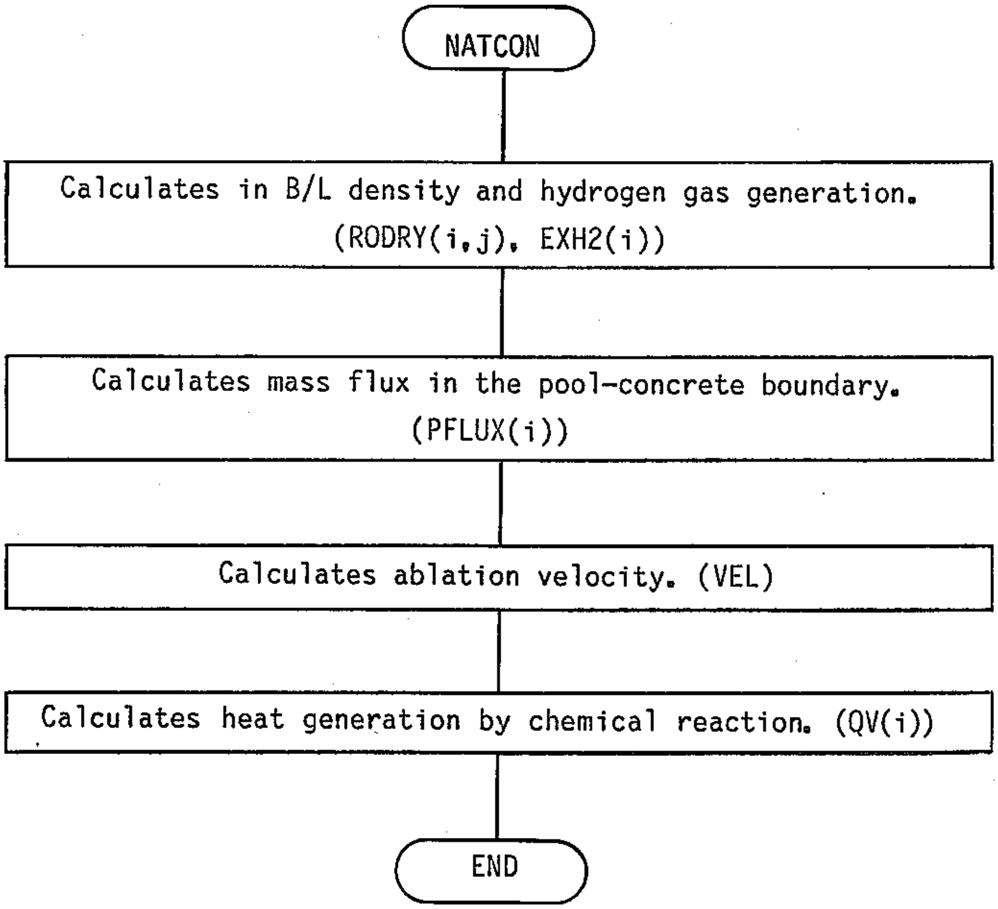


Figure 5.5 Flow Diagram of NATCON

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(*)	O	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, TOTH2G,
                  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	m2
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 tempera-ture	K
CTA(*)	R	I	concrete average layer temperature	K
CRM(*)	R	I	node density in the concrete layer	kg/m3
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)	m2/s2/K	gas constant/molecular weight
RHO(17,3)	kg/m3	material density
RHOL(17)	kg/m3	liquid phase micro density
ROMH2O	kg/m3	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
SRH2O(51)	kg/m3/s	H_2O gas generation rate per volume at the dry zone node point
EXH2(51)	m3/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/m3/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/m2	integrated value of heat generation by chemical reaction
QCONO	W/m2	heat flux from the pool
QCON	W/m2	heat flux from the wet zone
TKWET	W/m/K	thermal conductivity of the wet zone
QVSUM	W/m2	integral of heat source and heat sink in the dry zone

(continued)

QDEL1	W/m2	heat flux in the dry-wet boundary
H2OFLX	kg/m2/s	mass flux of evaporating water
H2OSRC	kg/m2/s	H_2O generation rate in the dry zone
WFLXDT	kg/m2	integral of H2OFLX and H2OSRC
CO2SRC	kg/m2/s	CO_2 generation rate in the dry zone
XNTCO2	kg/m2	amount of CO_2 generated in the dry zone
VGAS	m/s	converted rate of H_2 generation
XMH2XT	kg/m2/s	amount of hydrogen gas generation
H2FLXT	kg/m2	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/m3	H_2O 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/m3	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/m3	concrete gaseous phase density
ALPH(51)	-	air volume ratio
RODDYI (51,17)	kg/m3	density in the dry zone
RODRYO (51,17)	kg/m3	density in the dry zone
QV(52)	W/m3	heat generation by chemical reaction
QVP(51)	W/m3	heat generation by chemical reaction in the pool
SRCO2(51)	kg/m3/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          +--I1MAGH------>(ABORT )
        I          I          +--->(VALUE )
        I          +--->(TIMDAT)
        I          +--GLOSET ------>(SETCOM)
        I          I          +--BANNER
        I          +--->(RESTAR)
        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          +--->(SECOND)
        I          +--ISPACE
        I          +--IGLOBL------>(NEXT  )
        I          I          +--->(ABORT )
        I          I          +--->(VALUE )
        I          I          +--->(IPROPS)
        I          I          +--PTRGLO-----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          +--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          +--IFLOW ------>(NEXT  )
        I          I          I          +--->(VALUE )
        I          I          I          +--->(ITABLE)
        I          I          +--->(IENGSY)
        I          I          +--->(DECAY )
        I          +--CELSET------>(SETCOM)
        I          +--ICELL ------>(VALUE )
        I          I          +--->(NEXT  )
        I          I          +--PTRCEL------>(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          +--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                     +--->(IITEMP)
        I          I          I                     +--->(VALUE )
        I          I          I                     +--->(ITABLE)
        I          I          I                     +--->(ABORT )
        I          I          I                     +--IZERO
        I          I          +--IREBEL------>(NEXT  )
        I          I          I          +--->(SETCOM)
        I          I          I          +--->(ABORT )
        I          I          I          +--IRCON------>(NEXT  )
        I          I          I          I          +--->(VALUE )
```

```
MAIN  -----INPUT -----ICELL -----IREBEL○----IRCON○-----IRCTYP○------>(NEXT  )
        I         I          I          I          I          +--->(ENTHX )
        I         I          I          I          I          +--->(RHOX  )
        I         I          I          I          I          +--->(VALUE )
        I         I          I          I          I        ○ +--->(ABORT )
        I         I          I          I          +--INACON------>(NEXT  )
        I         I          I          I          I          +--SLPROP*
        I         I          I          I          I          +--->(VALUE )
        I         I          I          I          +--SLPROP*
        I         I          I          I          +--->(ISOURC)
        I         I          I          I          +--->(ITABLE)
        I         I          I          I          +--->(ABORT )
        I         I          I          I          +--->(RHOX  )
        I         I          I          +--->(UNLOAD)
        I         I          I          +--IRINT○------>(NEXT  )
        I         I          I          I          +--->(VALUE )
        I         I          I          I          +--->(ISOURC)
        I         I          I          I          +--->(ITABLE)
        I         I          I          I          +--->(ABORT )
        I         I          I          I          +--->(RHOX  )
        I         I          I          +--IRPOO○------>(NEXT  )
        I         I          I          I          +--->(VALUE )
        I         I          I          I          +--->(ISOURC)
        I         I          I          I          +--IPFIRE------>(NEXT  )
        I         I          I          I          I          +--->(VALUE )
        I         I          I          I          +--->(ITABLE)
        I         I          I          I          +--->(ABORT )
        I         I          I          I          +--->(RHOX  )
        I         I          I          +--->(VALUE )
        I         I          I          +--IRATM ------>(RHOX  )
        I         I          I          I          +--->(NEXT  )
        I         I          I          I          +--->(ITABLE)
        I         I          I          I          +--->(ISOURC)
        I         I          I          I          +--->(ABORT )
        I         I          I          +--IANSQ ------>(VALUE )
        I         I          I          I          +--->(ABORT )
        I         I          I          I          +--->(NEXT  )
        I         I          I          +--->(LOAD  )
        I         I          I          +--SLINPT*---SLCHEM*
        I         I          I          I          +--CONCPT*----IMOVE
        I         I          I          I          +--SLCOOR*
        I         I          I          I          +--->(RHOX  )
        I         I          I          I          +--->(TRAND*)
        I         I          I          +--->(RBOUT )
        I         I          +--->(LOAD  )
        I         I          +--REBPLT
        I         I          +--->(IENGSY)
        I         I          +--->(SAVCEL)
        I         I          +--->(ABORT )
        I         +--INITL
        I         +--RMOVE
        I         +--->(ABORT )
        +--CONTRL-----OUTPUT-----EDITIN------>(TIMDAT)
                  I          I          +--->(NXTCEL)
                  I          I          +--->(LOAD  )
                  I          +--->(TIMDAT)
                  I          +--->(SECOND)
                  I          +--->(NXTCEL)
```

```
MAIN -----CONTRL-----OUTPUT-----PRCELL------>(SETGAS)
                I          I          +--CPLOT ------>(LOAD  )
                I          I          I          +--REBPLT
                I          I          +--->(PRATM )
                I          I          +--->(RBOUT )
                I          I          +--->(IRTEMP)
                I          I          +--PRAERO
                I          I          +--HTMOUT
                I          I          +--PRSPRY
                I          I          +--PRENGS------>(LOAD  )
                I          I                     +--->(UNLOAD)
                I          +--PRGLBL-----GPLOT ------>(SECOND)
                I          I          +--USERO ------>(NXTCEL)
                I          I          +--PRFLOW
                I          I          +--->(PRDF  )
                I          +--RESOUT------>(UNLOAD)
                I                     +--->(ABORT )
                I                     +--->(SECOND)
                I                     +--->(NXTCEL)
                +--->(ABORT )
                +--BTIME
                +--CHOZDT------>(SECOND)
                +--GLREST-----ZERO
                I          +--RMOVE
                +--CLCNTR------>(NXTCEL)
                I          +--->(SETGAS)
                I          +--PRCEDT------>(TSATU )
                I          I          +--->(PSATX )
                I          I          +--->(SECOND)
                I          +--CELLDT-----CELLCK
                I          +--CLREST-----ZERO
                I          +--RBCNTR-----PHYDT
                I          I          +--LAYMOD------>(LOAD  )
                I          I          I          +--->(UNLOAD)
                I          I          +--->(IRTEMP)
                I          I          +--->(IITEMP)
                I          I          +--FPHEAT-----ANSQ  -----FXPFIT
                I          I          I          I          +--INTERP
                I          I          I          +--->(IRTEMP)
                I          I          I          +--->(LOAD  )
                I          I          +--ZERO
                I          I          +--ATTOCV-----ATMPOL------>(PSATX )
                I          I          I          I          +--->(SHX   )
                I          I          I          I          +--CNVNOS
                I          I          I          I          +--->(EVACON)
                I          I          I          I          +--->(ENTHX )
                I          I          I          I          +--->(CONDX )
                I          I          I          +--->(RHOX  )
                I          I          I          +--RADIAT
                I          I          +--BCTSET------>(LOAD  )
                I          I          +--HTSET ------>(LOAD  )
                I          I          I          +--->(CONVX )
                I          I          I          +--->(SETCOM)
                I          I          I          +--ZERO
                I          I          I          +--->(ABORT )
                I          I          I          +--->(UNLOAD)
                I          I          +--->(LOAD  )
                I          I          +--CONCRE------>(LAYPRO)
```

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

```
MAIN  -----CONTRL-----CLCNTR-----CCNTRL-----SORATM------>(ENTHX )
                I          I          +--ENGCTL------>(LOAD  )
                I          I          I          +--ZERO
                I          I          I          +--->(ENTHX )
                I          I          I          +--->(VISCX )
                I          I          I          +--->(RHOX  )
                I          I          I          +--ENGORF-----EST
                I          I          I          +--->(IRTEMP)
                I          I          I          +--SORENG------>(FLUX  )
                I          I          I          I          +--->(ABORT )
                I          I          I          I          +--->(ENTHX )
                I          I          I          +--->(TEMEN )
                I          I          I          +--ENGSPR------>(ENTHX )
                I          I          I          I          +--->(RHOX  )
                I          I          I          I          +--->(PSATX )
                I          I          I          I          +--->(TSATU )
                I          I          I          I          +--VELT
                I          I          I          I          +--ENGSPW------>(ENTHX )
                I          I          I          I          +--->(TEMEN )
                I          I          I          I          +--SPREFF-----RHODD
                I          I          I          I          I          +--CHI
                I          I          I          I          I          +--FGAMMA
                I          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          +--CNVNOS
                I          I          I          I          +--->(EVACON)
                I          I          I          I          +--->(ENTHX )
                I          I          I          I          +--->(IRTEMP)
                I          I          I          I          +--DCONTM
                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          +--CNVNOS
                I          I          I          +--->(CONDX )
                I          I          I          +--->(EVACON)
                I          I          I          +--->(ENTHX )
                I          I          +--CHEMRE-----AERREA------>(IRTEMP)
                I          I          +--HBURN -----VFLAME
                I          I          I          +--UNBRNF
                I          I          I          +--RBURN
                I          I          I          +--FORCE ------>(PRATM )
                I          I          +--QXLOS ------>(IRTEMP)
                I          I          I          +--HSETUP------>(CONDX )
                I          I          I          I          +--->(SHX   )
                I          I          I          I          +--->(DEBOUT)
```

```
MAIN  -----CONTRL-----CLCNTR-----CCNTRL-----QXLOS -----TRIDAG------>(DEBOUT)
                   I          I          +--SPRAY ------>(SHX   )
                   I          I                     +--->(CONDX )
                   I          I                     +--->(VISCX )
                   I          I                     +--SORSPR------>(FLUX  )
                   I          I                     +--->(RHOX  )
                   I          I                     +--VELT
                   I          I                     +--->(ENTHX )
                   I          +--SOR   -----SORAER------>(FLUX  )
                   I          I                     +--->(ABORT )
                   I          +--->(SAVCEL)
                   +--GLCNTR-----ACNTRL------>(IRTEMP)
                              I          +--AERSL -----RHODD
                              I                     +--COEF  -----GAUSBT----#FUN
                              I                     I          I          +-@F
                              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
                              +--FPMOVE-----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-----RDLCM
        +--SHFIN
        +--->(LOAD  )

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

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

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

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

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

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

--------------------------------------------------------------------------------
```

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

--------------------------------------------------------------------------------------------------
TIMDAT-----TDCOC
        +--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
----------------------------------------------------------------------------------------------------
IPROPS------>(NEXT  )
        +--->(VALUE )
        +--->(ENTHX )
        +--->(ABORT )
----------------------------------------------------------------------------------------------------
ISOURC------>(ABORT )
        +--->(VALUE )
        +--->(NEXT  )
        +--->(CONVER)
----------------------------------------------------------------------------------------------------
ITABLE------>(ABORT )
        +--->(NEXT  )
        +--->(VALUE )
        +--->(ARRAY )
----------------------------------------------------------------------------------------------------
RESTARO------>(LOAD  )
        +--->(NXTCEL)
        +--->(TRANA*)
        +--SLPROP*
        +--SLCHEM*
        +--->(SAVCEL)
----------------------------------------------------------------------------------------------------
PRATM ------>(TSATU )
        +--->(PSATX )
----------------------------------------------------------------------------------------------------
PROF  ------>(IITEMP)
----------------------------------------------------------------------------------------------------
RBOUT O------>(IRTEMP)
        +--->(LOAD  )
        +--SLOUT
----------------------------------------------------------------------------------------------------
CONDX O------>(ABORT )
----------------------------------------------------------------------------------------------------
CONVX ------>(ABORT )
        +--->(RHOX  )
```

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

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

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

--------------------------------------------------------------------------------
PSATX ------>(ABORT )

--------------------------------------------------------------------------------
RHOX  ------>(ABORT )

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

--------------------------------------------------------------------------------
SHX   ------>(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

--------------------------------------------------------------------------------------------
LAYMS ------>(SHX   )
        +--->(RHOX  )

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

--------------------------------------------------------------------------------------------
RBNLS ----@RBEQNS
        +--->(DEBOUT)
        +--->(ABORT )
        +--SGEFS -----SGECO -----SASUM
                   I          +--SGEFA -----ISAMAX
                   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 'CONCRM' 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 'CONCRM' * 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/XZI(51),WTZ(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            XNH2XT,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 /CPROP/ 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-SETMA.166
*/ ***********************************************************************
*/ * SLOB  :  MODIFICATION OF ORIGINAL SLAM SUBROUTINES                  *
*/ *                                                                     *
*/ *          1.  SLINPT  2.  SLCHEM   3.  SLCOOR   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 SIO3          - 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
      CALL  SLCHEM
C
      KREG = JCONC
      IMPL = 0
      IMPL2 = 0
```

```
      ISRCP1 = 0
      ISRC = IH(LRNNP+KREG)/4
      NDRY = ISRC*2.3
      NWET = IH(LRNNP+KREG)-NDRY+2
      NTERM = 12
      NPOOL = 1
      NSLM  = 15
C
      IF(NFCONC) ISRC=0
C
C
C   DETERMINES POINTERS OF CONCRETE STORAGE ARRAY ICH( )
C
      CALL  CONCPT
C
C   INITIAL SET  OF CONCRETE  COMMON  VARIABLE
C
      SDT  = 1.0E -6
      DEL1 = 0.003
      DIST = 0.0
      DDL1DT = 0.0
      DDL1DN = 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
      EPSO = RHOCON*FH2OE/ROMH2O
      TAV = 400.
      QCV = 0.0
      QCONO = 0.0
      QCON = 0.0
      QVSUM = 0.0
      QDEL1 = 0.0
      H2OFLX = 0.0
      H2OSRC = 0.0
      WFLXDT = DEL1*RHOCON*FH2OE
      CO2SRC = 0.0
      XNTCO2 = 0.0
      XMH2XT = 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
  350 DO  400  I = IDRY,NDRY
      RODRY(I,10) = RHOCON * FCACO3
      RODRY(I,13) = RHOCON * FMGCO3
      RODRY(I,2)  = 1.E-06
      IF (I.GE.NDRY-3) RODRY(I,2)=RHOCON*FH2OB
      RODRY(I,5) = 0.0
      RODRY(I,1) = RHOCON * FSIO2
      RODRY(I,15) = RHOCON * (1.-FCACO3-FMGCO3-FSIO2)
  400 CONTINUE
```

```
C
C     INITIAL  SET  OF  SLAM  COODINATE  SYSTEM
C
      CALL    SLCOOR
C
      KZ = IH(LRKZ0+KREG)
      J = 3*(KZ-1) + 2
      TC = H(LRCT + J)
      IF( NFNACO )  THEN
          KPOOL = JCONC + JINT + 1
          KP = IH(LRKZ0+KPOOL)
          K  = 3*(KP-1)+1
          TPOOL = H(LRPT+K)
          WTEMP  = TPOOL
      ELSE
          KINT = JCONC + 1
          KI = IH(LRKZ0+KINT)
          K  = 3*(KI-1)+1
          TINT  = H(LRIT+K)
          WTEMP  = TINT
      ENDIF
      DO  500  I = 1,NDRY
      TDRY(I) = WTEMP  + (TS-WTEMP)*XZI(I)
      TKDRY(I) = TKA*TDRY(I) + TKB
  500 CONTINUE
      TKWET =  TKA * TC + TKB
C
      DO  700  I = 1,NDRY
      ALPHAL(I)=1.0
      QV(I) =0.0
  700 CONTINUE
C
C     THIS  ROUTINE  DETERMINES  A  CONCRETE  DEPTH
C
      DELM=0.0
      KZ0=IH(LRKZ0+KREG)
      KZ1=KZ0+IH(LRNNP+KREG)-1
      DO 710 KZ =KZ0,KZ1
      DO 720 J=2,NM
      CMA = H(LRCMA+(KZ-1)*NM+J)
      CMB  = RHOX(J,H(LRCTA+KZ),H(LPGAS+NCELL))*H(LRCA+KZ)
      DELM = DELM + CMA/CMB
  720 CONTINUE
  710 CONTINUE
C
C     INITIAL  SET OF  TEMPERATURE ,PRESSURE ,MASS DENSITY IN WET ZONE
C
C
      DO 750  I = 1,NWET
      TWET(I) = TC
      PWET(I)=PBOT
      ROAIR(I)=(PWET(I)/RGAS(2)*TC)*(EPSM-EPSO)
      ROH2OL(I)=ROMH2O*EPSO
      XETA(I)=XETA(I)*(DELM-DEL1)
      ALPH(I)=EPSM - EPSO
  750 CONTINUE
C
C     INITIAL  SET OF  COODINATE  VARIABLE
C
      AM(1) = 0.0
      SIGN(1) = -1.0
      DO 800 M = 2,NTERM
      AM(M) = 0.0
      SIGN(M) = -SIGN(M-1)
  800 CONTINUE
      DO  900  N = 1,NTERM
      DO  900  M = 1,NTERM
      SUM=0.0
      DO  880  I = 1,NDRY
      SUM=SUM+XZI(I)*COSX(M,I)*SINX(N,I)*WTZ(I)
  880 CONTINUE
      ZCOSI(N,M) = SUM
  900 CONTINUE
C
C   STORE CONCRETE COMMON BLOCK WITH THE CONCRETE STORAGE ARRAY CH(*)
C
      CALL  TRANB
      RETURN
```

```
      END
*/ ***********************************************************************
*/ *        SUBROUTINE   SLCHEM                                          *
*/ ***********************************************************************
      SUBROUTINE  SLCHEM
C
C     THIS  ROUTINES  SETS  CHEMICAL  REACTION  DATA
C
*CA,/IMPLIC/
*CA,/SLCOM1/
*CA,/SLCOM2/
C
      DATA  NREACT/6/
      DATA  (SAK(I),I=1,8)/99.0E+05,1.0E+05,8.85E+12,8.85E+12,
     1                     40.0E+05,40.0E+05,0.0,0.0/
      DATA  (EK(I),I=1,8)/1.8E+03,1.8E+03,6.0E+04,6.0E+04,
     1                     3.4E+04,3.4E+04,0.0,0.0/
      DATA  (AKF(I,1),I=1,8)/0.3,0.3,4*0.1,0.0,0.0/
      DATA  (AKF(I,2),I=1,8)/1.,1.,2*0.1,1.,1.,0.,0./
      DATA  (AKF(I,3),I=1,8)/1.0,1.0,2*0.6,1.0,1.0,0.,0./
      DATA  (COFR(1,I),I=1,17)/0.,1.,1.,14*0./
      DATA  (COFR(2,I),I=1,17)/0.,0.,4.,7*0.,1.,6*0./
      DATA  (COFR(3,I),I=1,17)/0.,0.,4.,6*0.,3.,7*0./
      DATA  (COFR(4,I),I=1,17)/0.,0.,4.,9*0.,3.,4*0./
      DATA  (COFR(5,I),I=1,17)/4*0.,2.,4*0.,1.,7*0./
      DATA  (COFR(6,I),I=1,17)/1.,3*0.,2.,12*0./
C
      DATA  (COFP(1,I),I=1,17)/3*0,0.5,1.,12*0./
      DATA  (COFP(2,I),I=1,17)/7*0,2.0,3*0,1.,5*0./
      DATA  (COFP(3,I),I=1,17)/6*0,2.0,0.,3.,2*0.,1.,5*0./
      DATA  (COFP(4,I),I=1,17)/6*0,2.0,4*0.,1.,0.,3.,3*0./
      DATA  (COFP(5,I),I=1,17)/0.,1.,4*0.,1.,0.,1.,8*0./
      DATA  (COFP(6,I),I=1,17)/0.,1.,3*0.,1.,11*0./
      DATA  (IPHASR(1,I),I=1,17)/0,3,2,14*0/
      DATA  (IPHASR(2,I),I=1,17)/0,0,2,7*0,3,6*0/
      DATA  (IPHASR(3,I),I=1,17)/0,0,2,6*0,1,7*0/
      DATA  (IPHASR(4,I),I=1,17)/0,0,2,9*0,1,4*0/
      DATA  (IPHASR(5,I),I=1,17)/4*0,2,4*0,1,7*0/
      DATA  (IPHASR(6,I),I=1,17)/1,3*0,2,12*0/
      DATA  (IPHASP(1,I),I=1,17)/3*0,3,2,12*0/
      DATA  (IPHASP(2,I),I=1,17)/7*0,1,3*0,1,5*0/
      DATA  (IPHASP(3,I),I=1,17)/6*0,2,0,1,2*0,1,5*0/
      DATA  (IPHASP(4,I),I=1,17)/6*0,2,4*0,1,0,1,3*0/
      DATA  (IPHASP(5,I),I=1,17)/0,3,4*0,2,0,1,8*0/
      DATA  (IPHASP(6,I),I=1,17)/0,3,3*0,1,11*0/
      DATA  (NRCT(I),I=1,8)/ 6*2,0,0/
      DATA  (NPROD(I),I=1,8)/ 2,2,3*3,2,0,0/
      DATA  (MRR(I,1),I=1,8)/ 2,11,10,13,10,5,0,0/
      DATA  (MRR(I,2),I=1,8)/ 4*3,5,1,0,0/
      DATA  (MRR(I,3),I=1,8)/ 8*0/
      DATA  (MPP(I,1),I=1,8)/ 5,8,7,7,9,6,0,0/
      DATA  (MPP(I,2),I=1,8)/ 4,12,9,14,2,2,0,0/
      DATA  (MPP(I,3),I=1,8)/ 0,0,12,12,7,3*0/
      RETURN
      END        -
C     **************************************
*/ ***********************************************************************
*/ *        SUBROUTINE   SLCOOR                                          *
*/ ***********************************************************************
      SUBROUTINE SLCOOR
C
C     INITIAL  SET   OF  MATERIAL  PROPERTIES OF  CONCRETE
C
*CALL,/IMPLIC/
*CALL,/SLCOM1/
*CALL,/SLCOM2/
*CALL,/CONST/
C
      PI=3.1415926535
      NTERM=12
      XNTM1=NDRY-1
      ETA(1)=0.
      SUM=0.0
      FR=NDRY*0.2
      TRMX=FR**6
      DO 15 I=2,NDRY
      TERM=((NDRY-I)+1.)**6
      IF(TERM.LT.TRMX) TERM=TRMX
```

```
      DETA=1./TERM
      ETA(I)=ETA(I-1)+DETA
      SUM=SUM+DETA
15    CONTINUE
      DO 20 I=1,NDRY
      ETA(I)=ETA(I)/SUM
20    CONTINUE
C
C      WATER MIGRATION COORDINATES
C
      XETA(1)=0.
      SUMX=0.0
      DO 28 I=2,NWET
      TERM=(NWET-I+1.)
      DXETA=1./TERM
      XETA(I)=XETA(I-1)+DXETA
      SUMX=SUMX+DXETA
28    CONTINUE
      DO 29 I=1,NWET
      XETA(I)=XETA(I)/SUMX
29    CONTINUE
      DO 30 I=2,NDRY
      WT(I)=ETA(I)-ETA(I-1)
30    CONTINUE
      WT(1)=ETA(1)/2.
      WT(NDRY)=WT(NDRY)/2.
C
C      LINEAR FUNCTIONS AND WEIGHTING
C
      DXZI=1./XNTM1
      WTZ(1)=0.333333*DXZI
      XZI(1)=0.0
      DO 40 I=2,NDRY
      XZI(I)=XZI(I-1)+DXZI
      WTZ(I)=1.333333*DXZI
      IF((I/2)*2.NE.I) WTZ(I)=0.666666*DXZI
      DO 50 M=1,NTERM
      TERM=M*PI*XZI(I)
      SINX(M,I)=SIN(TERM)
      COSX(M,I)=COS(TERM)
50    CONTINUE
      DO 52 M=1,NTERM
      SINX(M,1)=0.0
      COSX(M,1)=1.0
52    CONTINUE
40    CONTINUE
      WTZ(NDRY)=WTZ(1)
      RETURN
      END
C     *********************************************
*/ ***********************************************************************
*/ *         SUBROUTINE   SLPROP
*/ ***********************************************************************
      SUBROUTINE SLPROP
C
*CALL,/IMPLIC/
*CALL,/SLCOM1/
*CALL,/SLCOM2/
*CALL,/REBLAY/
C
C
C      0.0 ALL DATA
C
      DO 100 I=1,17
      W(I)=0.0
      GAMMA(I)=0.0
      ZETA(I)=0.0
      TCRIT(I)=0.0
      PSTAR(I)=0.0
      TSTAR(I)=0.0
      HSTAR(I)=0.0
      XMUL(I)=0.0
      RHOL(I)=1200.
      RGAS(I)=8314.
      TMLT(I)=298.
      TSAT(I)=298.
      HF(I)=0.0
      HFUS(I)=0.0
```

```
      HFG(I)=0.0
      TKMAT(I)=.5
      IGAS(I)=0
      DO 200 J=1,3
      SP(I,J)=0.0
      IF(I.NE.2) RHO(I,J)=0.0
200   CONTINUE
100   CONTINUE
C
C      CONCRETE THERMAL CONDUCTIVITY FUNCTION  TK=TKA*TEMP(KELVIN)+TKB
C
C
C     SPECIES 1 IS SI-O2 PHASE1 ONLY = SOLID
C
      W(1)=60.
      SP(1,1)=1100.
      RHO(1,1)=RHOCON
      HF(1)=-205.
      TMLT(1)=1743.
300   CONTINUE
C
C      SPECIES 2 IS WATER PHASE 1 IS LIQUID PHASE 2 IS GAS
C
      SP(2,2)=4184.
      HF(2)=-68.317
      SP(2,3)=1860.
      RHO(2,3)=1.013E5/(8314.*800./18)
      W(2)=18.
      GAMMA(2)=1.33
      ZETA(2)=0.33
      TCRIT(2)=647.
      PSTAR(2)=1.758E10
      TSTAR(2)=4500.
      HSTAR(2)=3.1E6
      TSAT(2)=373.
      HFG(2)=HSTAR(2)*(1.-373./TCRIT(2))**ZETA(2)
      XMUL(2)=2.3E-4
      RHOL(2)=600.
      ROMH2O=1000.
      TKMAT(2)=0.11
      RGAS(2)=8314./18.
      IGAS(2)=2
C
C      SPECIES 3 IS SODIUM PROPERTIES: RATIO OF SPEC HEATS, DIETERICI
C      EQN OF STATE, PRESS FUNCTION=RT/(V-B),TEMP FUNCTION=A/(VR),CRITIC
C      TEMP, SAT TEMP, HEAT OF VAPORIZATION, CONDUCTIVITY
C
      SP(3,2)=1296.
      SP(3,1)=SP(3,2)
      HFUS(3)=1.14E05
      RHO(3,2)=800.
      RHOL(3)=800.
      TMLT(3)=370.9
      HF(3)=-.575
      SP(3,3)=4481.*4184/(23*902)
      RHO(3,3)=1.013E5/(8314*1000./23.)
      W(3)=23.
      GAMMA(3)=1.666
      ZETA(3)=0.341
      HSTAR(3)=4.81E6
      PSTAR(3)=3.27E9
      TSTAR(3)=1.202E4
      TCRIT(3)=2509.
C     TSAT(3)=-TSTAR(3)/ALOG(PINFXT/(PSTAR(3)*XMOLNA))
C     HFG(3)=HSTAR(3)*(1.-TSAT(3)/TCRIT(3))**ZETA(3)
      TKMAT(3)=50.
C
C     SPECIES 4 IS HYDROGEN GAS  1 PHASE ONLY: SPECIFIC HEAT,
C     DENSITY, MOLECULAR WEIGHT, CONDUCTIVITY
C
      SP(4,3)=(6404.*4184.)/(902.*2.)
      RHO(4,3)=1.013E5/(8314.*1000/2.)
      W(4)=2.
      TKMAT(4)=0.05
      RHOL(4)=3000.
      IGAS(4)=4
C
C     SPECIES 5 IS NAOH
```

```
C
      W(5)=40.
      RHOL(5)=850.
      SP(5,2)=3614.
      HF(5)=-100.0
      SP(5,1)=3827.
      HFUS(5)=2.76E5
      TMLT(5)=595.
      PSTAR(5)=4.002E9
      TSTAR(5)=1.748E4
C
C
C      SPECIES 6 IS NA2 SIO3 SOLID PHASE ONLY
C
C
      W(6)=122.
      SP(6,1)=1485.
      HF(6)=-372.
      HFUS(6)=3.53E5
      TMLT(6)=1360.
C
C      SPECIES 7 NA2CO3
C
      W(7)=106.
      HF(7)=-270.
      SP(7,2)=1776.
      SP(7,1)=1578.
      HFUS(7)=2.79E5
      TMLT(7)=1123.
C
C      SPECIES 8 NA2O
C
      W(8)=62.
      HF(8)=-99.9
      SP(8,2)=1635.
      SP(8,1)=1649.
      HFUS(8)=7.57E5
      TMLT(8)=1405.
C
C      SPECIES 9 CAO
C
      W(9)= 56.
      HF(9)=-151.6
      SP(9,1)=877.
      TMLT(9)=2888.
      RHO(9,1)=1540.
C
C      SPECIES 10 CACO3
C
      W(10)=100.
      HF(10)=-288.4
      SP(10,1)=1173.
      TMLT(10)=3000.
      RHO(10,1)=2200.
      HFUS(11)=2.E05
C
C      SPECIES 11 CO2
C
      W(11)=44.
      HF(11)=-94.
      RHOL(11)=3000.
      TKMAT(11)=0.075
      SP(11,3)=1378.
      RHO(11,1)=RHO(10,1)-RHO(9,1)
      IGAS(11)=11
C
C      SPECIES 12 C - GRAPHITE
C
      W(12)=12.
      HF(12)=0.
      SP(12,1)=1243.
C
C      MGCO3
C
      W(13)=84.3
      HF(13)=-267.6
      SP(13,1)=1173.
      RHOL(13)=1400.
```

```
C
C      MGO
C
      W(14)=40.3
      HF(14)=-143.8
      SP(14,1)=877.
      RHOL(14)=1400.
C
C      INERTS
C
      W(15)=100.
C
C     SPECIES  16  STEEL
C
      SP(16,2)=750.
      SP(16,1)=639.
      HFUS(16)=2.7E5
      RHO(16,2)=6979.
      RHOL(16)=6979.
      TMLT(16)=1700.
      HF(16)=-.575
      SP(16,3)=492.
      RHO(16,3)=1.0116E5/(81614*3000./56.)
      W(16)=56.
      GAMMA(16)=1.26
      ZETA(16)=0.36
      HSTAR(16)=8.17E6
      PSTAR(16)=1.33E11
      TSTAR(16)=4.337E4
      TCRIT(16)=1E4
C     TSAT(16)=-TSTAR(16)/ALOG(H(LPGAS+NCELL)/PSTAR(16))
      HFG(16)=HSTAR(16)*(1.-TSAT(16)/TCRIT(16))**ZETA(16)
      TKMAT(16)=20.
      XMUL(16)=5.36E-3
C
C     SPECIES 17  UO2
C
      SP(17,2)=504.
      SP(17,1)=638.
      HFUS(17)=2.08E5
      RHO(17,2)=8699.
      RHOL(17)=8699.
      TMLT(17)=3100.
      HF(17)=-272.
      SP(17,3)=511.
      RHO(17,3)=1.0116E5/(81614*3000./270.)
      W(17)=270.
      GAMMA(17)=1.05
      ZETA(17)=.597
      HSTAR(17)=2.62E6
      PSTAR(17)=1.44E11
      TSTAR(17)=5.17E4
      TCRIT(17)=8.4E3
C     TSAT(17)=-TSTAR(17)/ALOG(H(LPGAS+NCELL)/PSTAR(17))
      HFG(17)=HSTAR(17)*(1.-TSAT(17)/TCRIT(17))**ZETA(17)
      TKMAT(17)=5.0
      XMUL(17)=4.3E-3
C
C
C      CONVERT HEAT OF FORMATION TO JOULES/SI-MOLE
C
C
      DO 270 I=1,17
      HF(I)=HF(I)*4184000.
      IF(W(I).NE.0.0) RGAS(I)=8314./W(I)
270   CONTINUE
      RETURN
      END
C  ***********************************************************
*I,O-RESOUT.147
*DECK,O-SLAM
*/ ***********************************************************************
*/ *        SUBROUTINE   SLOUT                                           *
*/ ***********************************************************************
      SUBROUTINE SLOUT
C
C  ***   PRINTS THE CALCULATIONAL SUMMARY
C
```

```
C
*CALL,/IMPLIC/
*CALL,/FLAGS/
*CALL,/SLCOM1/
*CALL,/SLCOM2/
*CALL,/REBLAY/
*CALL,/TIME/
      CHARACTER*51 WORDA(18)
C
      DATA WORDA/'DUMMY',
     *'AVERAGE DRY ZONE TEMPERATURE,  TDRY   (K)          = ',
     *'WET-DRY INTERFACE TEMPERATURE, TS     (K)          = ',
     *'CONCRETE REACTION HEAT,        QCV    (W/M**2)     = ',
     *'CONCRETE SURFACE HEAT FLUX,    QCONO  (W/M**2)     = ',
     *'HEAT FLUX INTO WET ZONE,       QCON   (W/M**2)     = ',
     *'PENETRATION DEPTH,             DIST   (M)          = ',
     *'ABLATION VELOCITY,             VEL    (M/S)        = ',
     *'DRY ZONE DEPTH,                DEL1   (M)          = ',
     *'DRY ZONE GROWTH RATE,          DDL1DT (M/S)        = ',
     *'DRY ZONE HEAT SOURCES,         QVSUM  (W/M**2)     = ',
     *'WET-DRY INTERFACE HEAT FLUX,   QDEL1  (W/M**2)     = ',
     *'WET-DRY INTERFACE H2O PRESS.,  PS1    (PA)         = ',
     *'INTERFACE H2O EVAP. FLUX,      H2OFLX (KG/M**2-S)= ',
     *'DRY ZONE BOUND H2O FLUX,       H2OSRC (KG/M**2-S)= ',
     *'INTEGRAL OF BOUND + EVAP H2O,  WFLXDT (KG/M**2)   = ',
     *'DRY ZONE BOUND CO2 FLUX,       CO2SRC (KG/M**2-S)= ',
     *'INTEGRAL OF BOUND CO2,         XNTCO2 (KG/M**2)   = '/
C
      TDRI=TDRY(NDRY/2+1)
      IF (SLMOUT) WRITE (6,80) PTPDT,PDT
      IF (WATOUT) WRITE (6,85) PTPDT,PDT
80    FORMAT(//' SLAM   SODIUM-CONCRETE INTERACTION:',20X,'PTPDT = ',
     *G13.6,16X,'TIMESTEP = ',G13.6//)
85    FORMAT(//' WATER  MIGRATION  CALCULATION      ',20X,'PTPDT = ',
     *G13.6,16X,'TIMESTEP = ',G13.6//)
      WRITE (6,90) WORDA(2), TDRI, WORDA(3),TS
      WRITE (6,90) WORDA(4), QCV, WORDA(5), QCONO
      IF(SLMOUT) WRITE (6,90) WORDA(6), QCON, WORDA(7), DIST
      IF(SLMOUT) WRITE (6,90) WORDA(8), VEL , WORDA(9), DEL1
      IF(SLMOUT) WRITE (6,90) WORDA(10), DDL1DT,WORDA(11), QVSUM
      WRITE (6,90) WORDA(12), QDEL1,WORDA(13), PS1
      IF(WATOUT) WRITE (6,90) WORDA(9), DEL1,WORDA(10), DDL1DT
      WRITE (6,90) WORDA(14), H2OFLX,WORDA(15),H2OSRC
      WRITE (6,90) WORDA(16), WFLXDT,WORDA(17),CO2SRC
      WRITE (6,90) WORDA(18),XNTCO2
90    FORMAT(1X,A51,G12.5,5X,A51,G12.5)
      RETURN
      END
C ***************************************************************
*I,U-TRIDAG.62
*DECK,RB-SLAM
*/ ********************************************************************
*/ *         SUBROUTINE   STIME                                        *
*/ ********************************************************************
      SUBROUTINE STIME (SLTIME,WDT,ISLAF,TR,TSN,WDTDT)
C
C **  THIS  ROUTINE DETERMINES TIME STEP FOR SODIUM CONCRETE INTERACTION
C
*CALL,/IMPLIC/
*CALL,/TIME/
*CALL,/NACON/
*CALL,/SLCOM1/
*CALL,/SLCOM2/
*CALL,/REBLAY/
C
C     *-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*
C     TIME STEP POINT
C     I SAVE SYSTEM TIME STEP DT13 BEFORE ENTERING LOOP
      DETEX=PDT
C     I GET 2ND FROM LAST VALUE OF SDT TO CALC DT1 IN NEXT LOOP
100   CONTINUE
      DT1=SDT*1.2
      IF(SLTIME.LT.TNATO+1.) DT1=SDT*2.
      DT2=1.E3
      DT3=1.E3
      DT4=1.E3
      DT5=1.E3
      DT6=1.E3
```

```
      DT7=1.E3
      DT8=1.E3
      DT9=1.E3
      DT10=1.E3
      DT11=1.E03
      DT12=1.E03
      IF(DDL1DT.NE.0.0) DT2=(ABS(1.E-2*DEL1/DDL1DT))
      IF(DDL1DT.NE.0.0) DT3=(0.5*TDIF/DDL1DT**2)
      IF(VEL.NE.0.0) DT4=(ABS(0.15*DEL1/(VEL*NDRY)))
      DTOA=AMAX1(ABS(TR-TS),1.,0.01*ABS(TS-TSN))
      IF(DTSDT.NE.0.0) DT5=(ABS(DTOA/DTSDT))
      IF(WDTDT.NE.0.0) DT6=(ABS(DTOA/WDTDT))
      IF(COURNT.GT.0.5) DT7=(SDT/COURNT*0.5)
      DTIME=2.+ABS(SLTIME-TNATO)
      DTMAX=3.*ALOG(DTIME)
      DT8=(DTMAX)
      IF(IMPL2.NE.0) DT9=SDT*(20./FLOAT(IMPL2))
      IF(IMPL.NE.0) DT10=SDT*(20./FLOAT(IMPL))
      TERM=QVSUM+QCONO-QDEL1+VEL*ROCP*(TR-TS)
      DTDT=TERM/(RHOCON*DEL1*CPCON)
      IF(DTDT.GT.0.0) DT12=100./DTDT
      DTINF=SDT
C     ! SAVE 2ND FROM LAST VALUE OF DT TO CALC DT1 IN NEXT LOOP
C     THE LAST VALUE OF DT IN THE LOOP WILL USUALLY BE SMALLER
C     ! DT CANNOT BE GEATER THAN REMAINDER OF SYSTEM TIME STEP
      SDT=AMIN1(DT1,DT2,DT3,DT4,DT5,DT6,DT7,DT8,DT9,DT10,DT11,DT12
     * ,DETEX)
C     ! REMAINDER OF ORIG SYSTEM TIME STEP
      DETEX=DETEX-SDT
      IF(SDT.EQ.DT1) ISTEP=1
      IF(SDT.EQ.DT2) ISTEP=2
      IF(SDT.EQ.DT3) ISTEP=3
      IF(SDT.EQ.DT4) ISTEP=4
      IF(SDT.EQ.DT5) ISTEP=5
      IF(SDT.EQ.DT6) ISTEP=6
      IF(SDT.EQ.DT7) ISTEP=7
      IF(SDT.EQ.DT8) ISTEP=8
      IF(SDT.EQ.DT9) ISTEP=9
      IF(SDT.EQ.DT10) ISTEP=10
      IF(SDT.EQ.DT11) ISTEP=11
      IF(SDT.EQ.DT12) ISTEP=12
      IF(SDT.EQ.DETEX) ISTEP=13
      IF(SDT.LT.1.E-6)  PRINT 9823, SDT,ISTEP
C
9823  FORMAT(' ******'/' SMALL TIME STEP IN SLAM ',E10.3,
     *',   ISTEP=  ',I4,/' ******')
C
C    COMPARISON  SLAM  TIME  STEP  WITH  CONTAIN TIME  STEP
C
      PDTX = PDT - 0.0000001
      IF (SDT .GT. PDTX) THEN
                                ISLAF = 1
                                SDT = PDT
                                RETURN
      ELSE
      WDTS = WDT - SDT
      ENDIF
      IF (WDTS.LE.-0.000001) THEN
         SDT = WDT
         ISLAF = 1
      ELSE
      IF (WDTS.LE.SDT*0.25) THEN
          SDT =  WDT
          ISLAF = 1
      ELSE
          WDT = WDTS
      ENDIF
      ENDIF
      SLTIME = SLTIME + SDT
C
C
C     NSTEPO=NSTEP
C     NSTEP=NSTEP+1
      RETURN
      END
C
C
*/ *********************************************************************
```

```
*/ *        SUBROUTINE   CONEQS
*/ ******************************************************************
      SUBROUTINE CONEQS(TIME,DT,WTEMP,PINFXT,TINFXT,WDTDT,DEL10,TRO,
     1 H2OFLO,EPP,ERR1,COMP1,TOK,SUMTRM,FX,AMAT,CCOL,XVAR,AMO,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
C     ITERATION POINT ON THE END OF TIME STEP VALUES
C
      MIG = 0
      TR = WTEMP
      IMPL=IMPL+1
C     ! DRY REG H2O HEAT OF EVAP
      HFG(2)=HSTAR(2)*(1.-TS/TCRIT(2))**ZETA(2)
C
C     AVE THERMAL COND IN DRY CONC
C
      TAVE=0.5*(TDRY(ISRCP5)+TDRY(NDRY))
C     ! CONDUCTIVITY IN DRY REGION
      TK=TKA*TAVE+TKB
      IF(TK.LT.0.1) TK=0.05
      DO 545 I=ISRCP1,ISRCP5
      TKDRY(I)=AMAX1(0.05,TKA*TDRY(I)+TKB)
545   CONTINUE
      TDIF=TK/(ROCP)
C
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*N*PI/DEL1
      IF(M.NE.N .AND. ((M+N)/2)*2 .NE. (M+N) )
     *      SUM2=(2.*N/((N**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=XZI(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(M,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))*XZI(2)*0.5
      IF(I.LE.ISRCP5)
     * SUM9=SUM9+ (TKDRY(I)-TKDRY(I-1))*SINX(N,I)*TRM*WTT
```

```
112   CONTINUE
      CCOL(N)=-(SUM3+TDDL+SUM5+SUM7-SUM9)*DT+AMO(N)*0.5
110   CONTINUE
C
C      INVERT MATRIX AND CALC SOLUTION
C
      CALL GAUSS (AMAT,AM,CCOL,NTERM,12)
C
C      CALC HEAT FLUX AND ENERGY BALANCE
C
      SUM=0.0
      SUMDEL=0.0
      DO 160 M=1,NTERM
      XM=M
      SUM=SUM+XM*AM(M)
      SUMDEL=SUMDEL+SIGN(M)*XM*AM(M)
160   CONTINUE
C     ! HEAT FLUX AT CONC SURFACE
      QCONO=-TKDRY(1)*(SUM*PI+(TS-TR))/DEL1
C     QDEL1=-TK*((TS-TR)+PI*SUMDEL)/DEL1
      QDEL1=-TK*(TDRY(NDRY)-TDRY(NDRY-1))/(XZI(2)*DEL1)
      SUMDT1=0.0
      SUMDT2=0.0
      DO 161 M=1,NTERM
      SUMDT1=SUMDT1+(AM(M)-AMO(M))/DT*(1./XM-SIGN(M)/XM)
      SUMDT2=SUMDT2-AM(M)*(SIGN(M)-1.)/XM
161   CONTINUE
      TERM1=ROCP*DEL1*SUMDT1/PI
      TERM2=ROCP*DDL1DT*SUMDT2/PI
      TERM3=ROCP*(TR-TS)*0.5*DDL1DT
      TERM4=ROCP*WDTDT*DEL1
      TERM5=ROCP*(DTSDT-WDTDT)*DEL1*0.5
      ENBAL1=TERM1+TERM2+TERM3+TERM4+TERM5
      ENBAL2=ROCP*VEL*(TS-TR)
      ENBAL3=QCONO-QDEL1
      ENBAL4=QVSUM
      XVAR(9)=ENBAL1
      XVAR(10)=ENBAL2
      XVAR(11)=ENBAL3
      XVAR(12)=ENBAL4
      IF(ENBAL3.EQ.0.0) GO TO 440
      IF(ABS(ENBAL4/ENBAL3).LT.1) ENBAL=(ENBAL1+ENBAL2-ENBAL4)/ENBAL3
      IF(ABS(ENBAL4/ENBAL3).GT.1.) ENBAL=(ENBAL1+ENBAL2-ENBAL3)/ENBAL4
C     ! HEAT FLUX INTO WET REGION
440   QCON=-TKWET*2.*(TWET(1)-TS)/XETA(2)
C
C
C      DARCY EQN NUMERICAL INTEGRALS
C
      AP=0.3949E-18
      BP=0.01416
      TAVE=(TR+TS)/2.
      CALL SLDIFC(TAVE,PINFXT,DIFF,2,4,XMU)
      COMP1(1)=0.0
      DO 85 I=1,NDRY
      SUM1=0.0
      DO 86 M=1,NTERM
      SUM1=SUM1+AM(M)*SINX(M,I)
86    CONTINUE
      TDRY(I)=TR+(TS-TR)*XZI(I)+SUM1
      IF(TDRY(I).LE.0.0) WRITE(6,9445) I,VEL,TDRY(I),QV(I)
9445  FORMAT('   I,VEL,TDRY(I),QV(I)=  ',I6,4E12.5)
      PRMDRY=AP*EXP(BP*TDRY(I))
      TOK(I)=TDRY(I)/PRMDRY
      COMP1(I+1)=COMP1(I)+TOK(I)*WTZ(I)
85    CONTINUE
C
C      GAS SOURCE INTEGRALS
C
      CALL SORGAS(TIME,DT)
C
      SINT(1)=0.0
      DO 285 I=2,NDRY
      SRCGAS=0.5*(SRH2O(I)+SRCO2(I)+SRH2O(I-1)+SRCO2(I-1))
      SINT(I)=SINT(I-1)+SRCGAS*(XZI(I)-XZI(I-1))
285   CONTINUE
      SUMTRM(1)=0.0
      DO 280 I=2,NDRY
```

```
      SUMTRM(I)=SUMTRM(I-1)+TOK(I)*(SINT(NDRY)-SINT(I))*WTZ(I)
280   CONTINUE
C
C      SOLVE FOR TS AND PS1 SIMULTANEOUSLY BY NEWT RAPH
C
      TRMO=2.*XMU*RGAS(2)*DEL1
      TRM1=TRMO*COMP1(NDRY)
      TRM2=TRMO*DEL1*SUMTRM(NDRY)
C     ! USES H2O HEAT OF VAPORIZATION
      TRM3=2.*TKWET/(XETA(2)*HFG(2))
      ITNR=0
      TSN=TS
      ERM=1.E-05
      EPNR=0.25
54    CONTINUE
      ITNR=ITNR+1
      IF(ITNR.GT.15) EPNR=0.5
      PS1=PSTAR(2)*EXP(-TSTAR(2)/TSN)
C     ! H2O SAT PRESS IN WET REGION
      HH = QDEL1/HFG(2)+TRM3*(TWET(1)-TSN)
      IF (HH .LT. 0.0) HH = 0.0
      F=PS1*PS1-TRM1*HH-TRM2
     * -PINFXT*PINFXT
      DFDTS=2.*PS1*PS1*(TSTAR(2)/(TSN*TSN))+TRM1*TRM3
      TSNR=TSN-F/DFDTS
      ERR=ABS((TSN-TSNR)/TSN)
      IF(ITNR.GT.16) WRITE(6,99) ITNR,TSN,TSNR,ERR,PS1,PINFXT
99    FORMAT(1X,I3,5E12.4)
      TSNR=EPNR*TSN+(1.-EPNR)*TSNR
      IF(ITNR.GT.15) ERM=1.E-03
      IF(TSNR.LT.TWET(1)) TSNR=0.5*(TWET(1)+TSN)
      TSN=TSNR
      IF(ITNR.LT.20.AND.ERR.GT.ERM) GO TO 54
      ERR1=ABS((TS-TSN)/TSN)
      TS=TSN
C     ! USES H2O HEAT OF VAP
      H2OFLX=QDEL1/HFG(2)+TRM3*(TWET(1)-TS)
      IF (H2OFLX.LE.0.0) H2OFLX = 0.0
C
C      CALC PRESSURES AND FLUXES BY THE RATE AT WHICH AIR IS PUSHED
C      FROM THE SATURATION REGION
C
      CALL SLDIFC(TC,PINFXT,DIFF,2,4,XMU)
      CALL MGRATE(TIME,DT,WTEMP,EPSMI,LAST,PINFXT,TINFXT,XMU)
C
C
C      ERROR CHECK AND DEBUG PRINTS
C
      IF(IMPL.GT.20) WRITE(6,92) IMPL,NSTEP,TIME,DT,ERR1
92    FORMAT(6H IMPL=,I4,7H NSTEP=,I4,6H TIME=,F10.3,4H DT=,
     * E10.3,5H ERR=,E10.3)
      IF(IMPL.GT.20) WRITE(6,901) PS1,H2OFLX,DDL1DT,DEL1,TS,PS1
     *,TR,VEL
901   FORMAT(5H PS1=,E10.3,8H H2OFLX=,E10.3,8H DDL1DT=,E10.3
     * ,6H DEL1=,E10.3,/,4H TS=,E10.3,5H PS1=,E10.3
     * ,4H TR=,E10.3,5H VEL=,E10.3)
      IF(IMPL.GT.21) ERR1=0.0
C
      RETURN
      END
*/ ***********************************************************************
*/ *        SUBROUTINE   SORGAS                                           *
*/ ***********************************************************************
      SUBROUTINE SORGAS (TIME,DT)
C
*CALL,/IMPLIC/
*CALL,/SLCOM1/
*CALL,/SLCOM2/
*CALL,/NACON/
*CALL,/REBLAY/
C
C
C
C     RELEASE GAS CALCULATIONS FROM DRY REGION
C
      NM1=NDRY-1
      DO 102 I=1,NM1
      DO 102 J=1,NSLM
```

```
      RODRYO(I,J)=RODRY(I,J)
102   CONTINUE
103   CONTINUE
C
C      THE SOURCES TERMS FOR GAS RELEASE
C
      ROMIN=1.E-5
      DO 300 I=ISRCP1,NM1
      SRH20(I)=0.0
      SRCO2(I)=0.0
C
C      BOUND H20
C
      SH20=0.0
      RODRY(I,2)=RODRYO(I,2)
      IF(RODRY(I,2).GT.ROMIN)SH20=-3.3E10*EXP(-20560./TDRY(I))
      IF(SH20.NE.0.0) RODRY(I,2)=RODRYO(I,2)*EXP(SH20*DT)
C
C      BOUND CO2
C
      SCO2=0.0
      SCO2D=0.0
      RODRY(I,10)=RODRYO(I,10)
      RODRY(I,13)=RODRYO(I,13)
      TDAVE=0.5*(TDRY(I)+TDRY(I+1))
      IF(RODRY(I,10).GT.ROMIN)SCO2=-3.6E5*EXP(-19362./TDAVE)
      IF(RODRY(I,13).GT.ROMIN)SCO2D=-4.2E8*EXP(-19362./TDAVE)
      IF(SCO2 .LT.-0.15) SCO2=-0.15
      IF(SCO2D .LT.-0.15) SCO2D=-0.15
      IF(SCO2.NE.0.0) RODRY(I,10)=RODRYO(I,10)*EXP(SCO2*DT)
      IF(SCO2D.NE.0.0) RODRY(I,13)=RODRYO(I,13)*EXP(SCO2D*DT)
      RODRY(I,9)=(FCACO3*RHOCON-RODRY(I,10))*W(9)/W(10)
      IF(RODRY(I,9).LT.0.0) RODRY(I,9)=0.0
      RODRY(I,14)=(FMGCO3*RHOCON-RODRY(I,13))*(W(14)/W(13))
      IF(RODRY(I,14).LT.0.0) RODRY(I,14)=0.0
C
C      THE GAS SOURCE TERMS
C
      SRH20(I)=-(RODRY(I,2)-RODRYO(I,2))/DT
      SRCO2(I)=-(RODRY(I,10)-RODRYO(I,10))*W(11)/(W(10)*DT)
     *         -(RODRY(I,13)-RODRYO(I,13))*W(11)/(W(13)*DT)
300   CONTINUE
      H20SRC=0.0
      CO2SRC=0.0
      DO 115 I=ISRCP1,NM1
      H20SRC=H20SRC+SRH20(I)*WTZ(I)*DEL1
      CO2SRC=CO2SRC+SRCO2(I)*WTZ(I)*DEL1
      DO 115 J=1,NSLM
      RODRYI(I,J)=RODRY(I,J)
115   CONTINUE
      RETURN
      END
*/ *********************************************************************
*/ *        SUBROUTINE   MGRATE                                        *
*/ *********************************************************************
      SUBROUTINE MGRATE(TIME,DT,WTEMP,EPSMI,LAST,PINFXT,TINF,XMU)
C
*CALL,/IMPLIC/
*CALL,/SLCOM1/
*CALL,/SLCOM2/
*CALL,/NACON/
*CALL,/REBLAY/
*CALL,/REBCOM/
C
      DIMENSION  PP(51),RH2OLO(51),ROAIRO(51),TWETO(51)
      DIMENSION  DRG(51),DRL(51),FPRS(51,2),POLD(51,2)
      DIMENSION  PERML(51),PERMG(51)
      DATA PERMG/51*0.0/
C
C      BEFORE FIRST TIME STEP PRESET ALL THE VARIABLES FOR
C      THE WATER MIGRATION MODEL
C
      LAST=NWET
      SEAL=0.
      ALINC=0.006
      ISATO=0
      ISATL=1
C
```

```
C      UPDATE BEGINNING OF TIME STEP VALUES
C
       DO 102 I=1,LAST
       ROAIRO(I)=ROAIR(I)
       RH2OLO(I)=ROH2OL(I)
       TWETO(I)=TWET(I)
       PWET(I)=PBOT
       APW=2.94E-28
       BPW=0.069
       IF(TWET(I).GT.383.) APW=2.27E-22
       IF(TWET(I).GT.383.) BPW=0.0336
       PERML(I)=APW*EXP(BPW*TWET(I))
102    PERMG(I)=PERML(I)*2.
1011   CONTINUE
C
C      FIND LAST NODE
C
       LASTN=LAST
       XLEN=DELM-DEL1-DIST
       DO 330 I=1,LAST
       IF(XETA(I).LT.XLEN) GO TO 330
       LASTN=I
       XETA(I)=XLEN
       IF(XETA(I)-XETA(I-1) .LT. XETA(2) ) THEN
       LASTN = I - 1
       NWET = LASTN
       ENDIF
       GO TO 340
330    CONTINUE
340    LAST=LASTN
       IF(LAST.EQ.4) TWFIN=TIME
       NM1=LAST-1
       IMPL2=0
       EPSMI=0.5
C
C       IMPLICIT ITERATION POINT ON THE END OF TIME STEP TEMP AND PRESS
C
50     CONTINUE
       IMPL2=IMPL2+1
C
C       THE COORD SYSTEM VELOCITY DDL1DN
C
       IF(MIG.EQ.0) GO TO 220
       IF(ALPH(1).GT.ALINC) GO TO 200
       ISATO=1
       DO 210 I=2,LAST
       IF(ALPH(I).LT.ALINC) GO TO 210
       ISATL=I
       GO TO 220
210    CONTINUE
C
C       INCOMPRESIBLE LIQUID WATER FLOW
C
C
220    CONTINUE
       IF (MIG.EQ.0) ISATL=LAST
       DXSAT=XETA(ISATL)
       DPDXO=(PS1-PWET(ISATL))/DXSAT
       SUM=0.0
       LST=ISATL-1
       DO 295 I=1,LST
       SUM=SUM+(XETA(I+1)-XETA(I))/PERML(I)
295    CONTINUE
       PERMT=XETA(ISATL)/SUM
       DRL(1)=PERMT*DPDXO/XMUL(2)
       GO TO 230
200    CONTINUE
       DPDXO=2.*(PS1-PWET(1))/(XETA(2)-XETA(1))
       XKRL=(1.-ALPH(1)/EPSM)**3
       DRL(1)=PERML(1)*XKRL*DPDXO/XMUL(2)
230    CONTINUE
       XKRM=ALPH(1)/EPSM
       IF(XKRM.LT.1.E-10) XKRM=1.E-10
       DRG(1)=PERMG(1)*XKRM*DPDXO/XMU
       DDL1DO=DDL1DN
       DDL1DN=H2OFLX/ROH2OL(1)+DRL(1)
642    CONTINUE
C
```

```
C       END OF TIME STEP TEMP
C
        TWAVE=0.5*(TWET(1)+TWET(LAST))
C       ! AVG CONDUCTIVITY IN WET REGION
        TKWET=TKA*TWAVE+TKB
        TDIFW=TKWET/(ROCP)
        RFDX2=1./(2.*XETA(2)*XETA(2))
        RFDX1=1./((XETA(3)*XETA(2)))
        TSP=2.*TS-TWET(1)
        DXID=XETA(2)*0.5+0.5*(XETA(3)-XETA(2))
        TID=TS
        IF(DDL1DN.GT.0.0) TID=TWET(2)
        VV=-DDL1DN
        DENOM=(1./DT-VV/DXID)+2.*TDIFW*(RFDX1+RFDX2)
        XNUM=TWETO(1)/DT-VV*TID/DXID+2.*TDIFW*(TWET(2)*RFDX1
     *      +TSP*RFDX2)
        TWET(1)=XNUM/DENOM
        PWET(1)=PS1
        RFDX2=1./((XETA(LAST)-XETA(LAST-2))*(XETA(LAST)-XETA(NM1)))
        DXID=XETA(LAST)-XETA(NM1)
        TID=TWET(NM1-1)
        IF(DDL1DN.GT.0.0) TID=TWET(NM1)
        DENOM=(1./DT-VV/DXID+4.*TDIFW*RFDX2)
        XNUM=TWETO(NM1)/DT-VV*TID/DXID+2.*TDIFW*TWET(NM1-1)*RFDX2
        XNUM=XNUM+2.*TDIFW*TXL*RFDX2
        TWET(NM1)=XNUM/DENOM
        NM2=NM1-1
        DO 199 I=2,NM2
        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))
        IF(DDL1DN.GT.0.0) DXID=0.5*(XETA(I+2)-XETA(I))
        TID=TWET(I-1)
        IF(DDL1DN.GT.0.0) TID=TWET(I+1)
        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 )
        TWET(I)=XNUM/DENOM
        IF(IMPL2.GT.1) GO TO 199
C       ! H2O SAT PRESS IN WET REGION
        PWVAP=PSTAR(2)*EXP(-TSTAR(2)/TWET(I))
        PWET(I)=ROAIR(I)*RGAS(2)*TWET(I)/(ALPH(I))+PWVAP
199     CONTINUE
        IF(MIG.EQ.0) RETURN
C       IF((NSTEP/MIG)*MIG.NE.NSTEP .AND. NSTEP.NE.1) RETURN
C
C       END OF TIME STEP DENSITIES AND PRESSURE
C
        PWET(LAST)=PINFXT
        ROH2OL(LAST)=ROH2OL(NM1)
        ROAIR(LAST)=ROAIR(NM1)
        IF(ISATO.NE.1) GO TO 260
        DO 270 I=ISATO,ISATL
        DRL(I)=DRL(1)
        DRG(I)=0.0
        IF(I.EQ.ISATL) GO TO 270
        DXP=XETA(I+1)-0.5*(XETA(I+1)-XETA(I))
        PWET(I)=PS1-(PS1-PWET(ISATL))*DXP/XETA(ISATL)
270     CONTINUE
260     CONTINUE
        DO 31 I=ISATL,NM1
        ALPHL=EPSM-ALPH(I)
        DPDX=(PWET(I)-PWET(I+1))/(XETA(I+1)-XETA(I))
        XKRL=(1.-ALPH(I)/EPSM)**3
        DRL(I+1)=PERML(I)*DPDX*XKRL/(XMUL(2))
        IF(ALPH(I).GT.ALINC) GO TO 280
280     CONTINUE
        IF(I.EQ.NM1) DRL(I+1)=AMAX1(0.0,DRL(I+1))
        XKRM=ALPH(I)/EPSM
        IF(XKRM.LT.1.E-10) XKRM=1.E-10
        DRG(I+1)=PERMG(I)*DPDX *XKRM /(XMU)
C
C       LIQUID
C
        VELI=DRL(I)-DDL1DN
        VELIP1=DRL(I+1)-DDL1DN
        IF(I.EQ.NM1 .AND. SEAL.EQ.1) VELIP1=-DDL1DN
        DLI=0.0
```

```
      DRI=1.0
      IF(VELI.GT.0.0) GO TO 5
      DLI=-1.0
      DRI=0.0
5     DLIP1=0.0
      DRIP1=1.0
      IF(VELIP1.GT.0.0) GO TO 6
      DLIP1=-1.0
      DRIP1=0.0
6     CONTINUE
      XNUM=ROH2OL(I-1)*VELI*DRI+ROH2OL(I+1)*VELIP1*DLIP1
      DXETA=XETA(I+1)-XETA(I)
      XNUM=XNUM/DXETA+RH2OLO(I)/DT
      DENOM=1./DT+(VELI*DLI+VELIP1*DRIP1)/DXETA
      ROH2OL(I)=XNUM/DENOM
C
C      GAS
C
      VELI=DRG(I)-DDL1DN
      VELIP1=DRG(I+1)-DDL1DN
      IF(I.EQ.NM1 .AND. SEAL.EQ.1) VELIP1=-DDL1DN
      IF(I.EQ.1 .AND. VELI.GT.0.0) VELI=0.0
      DLI=0.0
      DRI=1.0
      IF(VELI.GT.0.0) GO TO 7
      DLI=-1.0
      DRI=0.0
7     DLIP1=0.0
      DRIP1=1.0
      IF(VELIP1.GT.0.0) GO TO 8
      DLIP1=-1.0
      DRIP1=0.0
8     CONTINUE
      XNUM=ROAIR(I-1)*VELI*DRI+ROAIR(I+1)*VELIP1*DLIP1
      DXETA=XETA(I+1)-XETA(I)
      XNUM=XNUM/DXETA+ROAIRO(I)/DT
      DENOM=1./DT+(VELI*DLI+VELIP1*DRIP1)/DXETA
      ROAIR(I)=XNUM/DENOM
      ALPH(I)=EPSM-ROH2OL(I)/ROMH2O
      DXDT=ABS(DDL1DN)
C      I H2O SAT PRESS IN WET REG
      PWVAP=PSTAR(2)*EXP(-TSTAR(2)/TWET(I))
      PP(I)=ROAIR(I)*RGAS(2)*TWET(I)/(ALPH(I))+PWVAP
31    CONTINUE
C
C      UPDATE AND CHECK FOR CONVERGENCE
C
      ERRMX=0.0
      DO 40 I=ISATL,NM1
      ERR=ABS((PP(I)-PWET(I))/PWET(I))
      IF(ERR.GT.ERRMX) IMX=I
      ERRMX=AMAX1(ERRMX,ERR)
40    CONTINUE
C
C      NEWTON RAPHSON OR WIEGHTED ITERATION UPDATE ON PRESSURE
C
      DO 145 I=ISATL,NM1
116   IF(IMPL2.GT.20 .AND. I.EQ.IMX) WRITE(6,909)IMPL2,I,PWET(I)
     * ,PP(I),ERRMX
909   FORMAT(2I5,5E12.4)
      FPRS(I,2)=FPRS(I,1)
      FPRS(I,1)=PP(I)-PWET(I)
      POLD(I,2)=POLD(I,1)
      POLD(I,1)=PWET(I)
      IF(ABS(FPRS(I,1)/PP(I)).LT.1.E-08) GO TO 115
      IF(IMPL2.EQ.1) GO TO 115
      IF(FPRS(I,2)/FPRS(I,1).GT.0.0) GO TO 115
      IF(PWET(I)-POLD(I,2).EQ.0.0) GO TO 115
      DFDP=(FPRS(I,1)-FPRS(I,2))/(PWET(I)-POLD(I,2))
      IF(DFDP.EQ.0.0) GO TO 115
      PWET(I)=PWET(I)-FPRS(I,1)/DFDP
      GO TO 145
115   PWET(I)=EPSMI*PP(I)+(1.-EPSMI)*PWET(I)
145   CONTINUE
      IF((IMPL2/10)*10.EQ.IMPL2) EPSMI=EPSMI*0.9
      IF(IMPL2.EQ.50) STOP
      EPSMI=AMAX1(EPSMI,0.2)
      IF(ERRMX.GT.5.E-04) GO TO 50
```

```
C
C     CALCULATE EXPLICIT PARAMETERS AFTER CONVERGENCE
C
      PPWM=DRL(1)/DDL1DN
      COURNT=0.0
      DO 80 I=1,LAST
      CRNT=((DRG(I)-DDL1DN)*DT)/(XETA(I+1)-XETA(I))
      IF(ROAIR(I).LT.1.E-6) CRNT=0.0
      COURNT=AMAX1(COURNT,ABS(CRNT))
80    CONTINUE
      RMIG=AMIN1(DDL1DN*ROH2OL(1)/H2OFLX,5.0)
      DRL1O=DRL(1)
      RETURN
90    FORMAT(6E11.4)
      END
*/ ********************************************************************
*/ *         SUBROUTINE   SLENTH                                       *
*/ ********************************************************************
      SUBROUTINE SLENTH(TEMP,ENTH,MAT,IPHS)
*CALL,/IMPLIC/
*CALL,/SLCOM1/
*CALL,/SLCOM2/
C
      TREF=298.
      IF(IPHS.EQ.2) GO TO 10
      IF(IPHS.EQ.3) GO TO 20
C     | ENTHALPY OF SOLID- PROP
      ENTH=HF(MAT)+SP(MAT,1)*(TEMP-TREF)*W(MAT)
      RETURN
10    CONTINUE
      SENS=SP(MAT,1)*(TMLT(MAT)-TREF)+SP(MAT,2)*(TEMP-TMLT(MAT))
C     | ENTHALPY OF LIQUID- PROP
      ENTH=HF(MAT)+(SENS+HFUS(MAT))*W(MAT)
      RETURN
20    CONTINUE
      SENS=SP(MAT,1)*(TMLT(MAT)-TREF)+SP(MAT,2)*(TSAT(MAT)-TMLT(MAT))
     * +SP(MAT,3)*(TEMP-TSAT(MAT))
C     | ENTHALPY OF VAPOR- PROP
      ENTH=HF(MAT)+(SENS+HFUS(MAT)+HFG(MAT))*W(MAT)
      RETURN
      END
C**********************************************
*/ ********************************************************************
*/ *         SUBROUTINE   SLDIFC                                       *
*/ ********************************************************************
      SUBROUTINE SLDIFC(TAVE,PRESS,DIFF,I,J,XMU)
C
C      THIS SUBROUTINE CALCULATES THE AVERAGE DIFFUSION COF
C      FOR SPECIES I AND J AT THE AVERAGE TEMP TAVE
C
*CALL,/IMPLIC/
*CALL,/SLCOM1/
*CALL,/SLCOM2/
C
      DIMENSION EPSOK(17),SIGMA(17)
C
      PATM=PRESS/1.013E05
      EPSCON=0.20
      EPSOK(2)=32.
      EPSOK(3)=1375.
      EPSOK(4)=60.
      EPSOK(5)=1962.
      SIGMA(2)=3.737
      SIGMA(3)=3.567
      SIGMA(4)=2.827
      SIGMA(5)=3.804
      TERM1=(TAVE)**3*(1./W(I)+1./W(J))
      SIG=(SIGMA(I)+SIGMA(J))/2.
      EPS=(EPSOK(I)*EPSOK(J))**0.5
      TEPS=TAVE/EPS
      XMD=0.82515+1.9066E-3*TEPS+0.66448*TEPS**(-1.0676)-0.050246*
     * SQRT(TEPS)
      DIFF=0.018768*(TERM1)**0.5/(SIG*SIG*XMD*PRESS)
      TERM1=2.67E-6*SQRT(W(I)*TAVE)/SIGMA(I)**2
      TEPS=TAVE/EPSOK(I)
      XMD=0.92495+2.07368E-3*TEPS+0.719288*TEPS**(-1.151049)-5.46452E-2
     * *SQRT(TEPS)
      XMU=TERM1/XMD
```

```
      RETURN
      END
*/ ************************************************************************
*/ *        SUBROUTINE   SLTRIS                                           *
*/ ************************************************************************
      SUBROUTINE SLTRIS(A,B,C,D,N)
C
*CA /IMPLIC/
      DIMENSION A(1),B(1),C(1),D(1)
C
C     A=LOWER DIAGONAL
C     B=DIAGONAL
C     C=UPPER DIAGONAL
C     D=RESULTANT
C     N=NUMBER OF UNKNOWNS
C
C
      DO 10 I=2,N
      FCTR=A(I)/B(I-1)
      B(I)=B(I)-FCTR*C(I-1)
      D(I)=D(I)-FCTR*D(I-1)
10    CONTINUE
C
      D(N)=D(N)/B(N)
      K=N
      DO 20 I=2,N
      K=K-1
      D(K)=(D(K)-C(K)*D(K+1))/B(K)
20    CONTINUE
C
      RETURN
      END
*/ ************************************************************************
*/ *        SUBROUTINE   GAUSS                                            *
*/ ************************************************************************
      SUBROUTINE GAUSS(A,X,B,M,NDIM)
*CA /IMPLIC/
C
      DIMENSION A(NDIM,1),X(1),B(1),IS(30),JS(30),LOC(30)
      N=0
      DO 13 K=1,M
      IF(A(K,K).EQ.0.0)GO TO 13
      N=N+1
      LOC(N)=K
 13   CONTINUE
      IF(N.EQ.0) RETURN
      IF (N .GT. 1) GO TO 14
      K=LOC(1)
      X(K)=B(K)/A(K,K)
      RETURN
 14   N1=N-1
      DO 4 KK=1,N1
      K=LOC(KK)
      CMAX=0.0
      DO 1 II=KK,N
      I=LOC(II)
      DO 2 JJ=KK,N
      J=LOC(JJ)
      IF(ABS(A(I,J)).LE.CMAX) GO TO 2
      IM=I
      JM=J
      IS(KK)=II
      JS(KK)=JJ
      CMAX=ABS(A(I,J))
 2    CONTINUE
 1    CONTINUE
      DO 11 JJ=KK,N
      J=LOC(JJ)
      TEMP=A(K,J)
      A(K,J)=A(IM,J)
      A(IM,J)=TEMP
 11   CONTINUE
      DO 12 II=1,N
      I=LOC(II)
      TEMP=A(I,K)
      A(I,K)=A(I,JM)
      A(I,JM)=TEMP
 12   CONTINUE
```

```
      K1=KK+1
      DO 3 II=K1,N
      I=LOC(II)
      A(I,K)=A(I,K)/A(K,K)
 3    CONTINUE
      DO 4 II=K1,N
      I=LOC(II)
      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 II=K1,N
      I=LOC(II)
      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                                            *
*/ ************************************************************************
      SUBROUTINE NATCON (TIME,DT,TPOOL,ROPOOL,RLIQ,DIFO,PFLUX,PINFXT,
     1                   TINFXT)
C
*CALL,/IMPLIC/
*CALL,/NACON/
*CALL,/SLCOM1/
*CALL,/SLCOM2/
*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
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
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)
      DO 510 IR=1,8
      RLIQ(I,IR)=0.0
510   CONTINUE
      SUM = 0.0
      DO 520 I=1,ISRC
      CALL REACT (TIME,DT,TPOOL,I,1,RLIQ)
      SUM = SUM + EXH2(I)*DEL1*XZI(2)
520   CONTINUE
      XMH2XT = SUM
      H2FLXT = XMH2XT*DT
C
C     THE DIFFUSION
C     ! DENSITY OF H2O(V) IN DRY REGION
      RHO2=PINFXT/(RGAS(2)*TDRY(ISRC))
C     ! DENSITY OF CO2 IN DRY REGION
      RHO11=PINFXT/(RGAS(11)*TDRY(ISRC))
      VGAS=(H2OFLX+H2OSRC)/RHO2
      DXXI=DEL1*XZI(2)
      GAMV=GAMO*VGAS*DXXI
      DO 500 I=1,ISRCP1
      AAM=GAM2*DEL1*(XZI(ISRCP1)-XZI(I))
      BB=0.0
      IF(AAM.LT.14.) BB=EXP(-AAM)
      GAMBL(I)= (GAMV*(1.+GAM1*(1.-BB))+GAMINF*DXXI)
500   CONTINUE
      DO 560 J=1,NSLM
      DO 600 I=1,ISRC
      A(I)=-GAMBL(I)
      B(I)=DXXI**2/DT+GAMBL(I)+GAMBL(I+1)
      C(I)=-GAMBL(I+1)
      D(I)=RODRYI(I,J)*DXXI**2/DT
600   CONTINUE
      D(1)=D(1)+ROPOOL(J)*GAMBL(1)
      B(ISRC)=B(ISRC)-GAMBL(ISRC+1)
      A(1)=0.0
      C(ISRC)=0.0
      CALL SLTRIS(A,B,C,D,ISRC)
      DO 550 I=1,ISRC
      RODRYI(I,J)=D(I)
550   CONTINUE
560   CONTINUE
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
      VIPM=AMIN1(0.0,VELIP1-VB)
      VIPP=AMAX1(0.0,VELIP1-VB)
      VIP=AMAX1(0.0,VELI-VB)
      VIM=AMIN1(0.0,VELI-VB)
      DENOMT=DENOM+(VIPP-VIM)/DXXI
      ROIM1=RODRY(I-1,J)
      ROIP1=RODRY(I+1,J)
```

```
      IF(I.EQ.ISRCP1) ROIM1=0.0
      IF(I.EQ.1) ROIM1=AMAX1(0.,ROPOOL(J))
      IF(I.EQ.ISRC) DENOMT=DENOMT-VIPP/DXXI
      ROVE=(ROIM1*VIP-VIPM*ROIP1)/DXXI
C
C     SOURCES IN THE INTERFACE NODE DUE TO ABLATION
C
      SRC=0.0
      IF(I.NE.ISRC) GO TO 838
      SRC=-VEL*RODRY(I+1,J)
      SRC=AMAX1(0.0,SRC/DXXI)
838    CONTINUE
      XNUM=ROVE+RODRYI(I,J)/DT + SRC
      RODRY(I,J)=XNUM/(DENOMT)
      IF (RODRY(I,J) .LE. 0.0) RODRY(I,J)=0.0
      IF(I.EQ.1) PFLUX(J)=-VIM*RODRY(I,J)-VIP*ROIM1
     *             +  GAMBL(1)*(RODRY(I,J)-ROIM1)/DXXI
      IF(I.GT.ISRC) GO TO 31
      VOLFR=RODRY(I,J)/RHOL(J)
      COR1=COR1+VOLFR
      CPBL=SP(J,2)
      IF(CPBL.EQ.0.0) CPBL=CPCON
      IF(J.EQ.3 .AND. VOLFR.LT.0.25) VOLFR=0.0
      TKBL=TKBL+VOLFR*(TKMAT(J)+GAMBL(I+1)*RHOL(J)*CPBL)
      IF(I.EQ.ISRC) DIFO=GAMBL(I+1)
31    CONTINUE
C
C      END OF MATRL DO LOOP
C
      IF(I.GT.ISRC) GO TO 32
      TKDRY(I)=(ALPHAL(I)*TKBL+(1.-ALPHAL(I))*0.03)*ALPHAL(I)
      F=1.-COR1
      IF(ABS(F).LT.0.00002) GO TO 32
      IF(ITNR.EQ.1) GO TO 360
      DFDV=(F-FOLD)/(V(I)-VOLD)
      FOLD=F
      VOLD=V(I)
      VNEW=V(I)-F/DFDV
      CHG=1.
      IF(VNEW.NE.0.0) CHG=ABS((VNEW-V(I))/VNEW)
      IF(CHG.GT.0.5) V(I)=(V(I)+VNEW)*0.5
      IF(CHG.LE.0.5) V(I)=VNEW
      IF(ITNR.GT.10) V(I)=(VOLD+VNEW)*0.5
      GO TO 205
360   FOLD=F
      VOLD=V(I)
      V(I)=V(I)*1.05-1.E-05
      GO TO 205
32    CONTINUE
33    CONTINUE
C
C      CALCULATE THE LOCAL BUBBLE VELOCITY BASED UPON STOKES THEOREM
C
      DO 710 I=1,ISRC
      RHOBL=0.0
      DO 711 J=1,NSLM
      RHOBL=RHOBL+RODRY(I,J)
711   CONTINUE
      ANAL=RODRY(I,3)/RHOL(3)
      ANAOH=RODRY(I,5)/RHOL(5)
      ALRE=1.-ANAL-ANAOH
      AMUL=ANAL*1.86E-4+ANAOH*1.E-3
      XMUBL=(AMUL*XVAR1 + ALRE*2.)/(AMUL*XVAR1+ALRE)
      VBUB(I)=AMIN1(0.5,2.*RBUB**2*RHOBL/XMUBL)
      ALPHAL(I)=1.-VGAS/(VBUB(I)+VGAS)
710   CONTINUE
C
C      CALCULATE ABLATION VELOCITY BY CHEMICAL REACTION
C
      IR=3
      AKK=AMIN1(AKF(IR,1),SAK(IR)*EXP(-EK(IR)/(1.98*TDRY(ISRCP1))))
      IF(RODRY(I,3)/RHOL(3).GT.AKF(IR,3)) AKK=AKK*AKF(IR,2)
      DRIVE=RODRY(ISRC,3) /(COFR(IR,3)*W(3))
      VELN=0.0
      IF(TIME.GT.TNATO) VELN=-AKK*DRIVE*FRAC*DEL1*XZI(2)
      VELN=AMIN1(0.0,VELN)
89    CONTINUE
C
```

```
C
C        HEAT SOURCES DUE TO CHEMISTRY
C
      CALL HEAT (1,RLIQ)
      HBH2O=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)-(SRH2O(I)*HBH2O+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
*/ ***********************************************************************
*/ *         SUBROUTINE   WTRRLS                                          *
*/ ***********************************************************************
      SUBROUTINE WTRRLS (TIME,DT,WTEMP,WRO,RLIQ,DIFO,PFLUX,PINFXT,
     1                  TINFXT)
C
*CALL,/IMPLIC/
*CALL,/NACON/
*CALL,/SLCOM1/
*CALL,/SLCOM2/
*CALL,/REBLAY/
C
C
      DIMENSION GAMBL(51),WRO(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
      VIPM=AMIN1(0.0,VELIP1-VB)
      VIPP=AMAX1(0.0,VELIP1-VB)
```

```
      VIP=AMAX1(0.0,VEL1-VB)
      VIN=AMIN1(0.0,VEL1-VB)
      DENOMT=DENOM+(VIPP-VIM)/DXXI
      ROIM1=RODRY(I-1,J)
      ROIP1=RODRY(I+1,J)
      IF(I.EQ.ISRCP1) ROIM1=0.0
C
      IF(I.EQ.1) THEN
         IF((J.EQ.2).OR.(J.EQ.4).OR.(J.EQ.11)) THEN
                 ROIM1 = AMAX1(0.,WRO(J))
         ELSE
                 ROIM1 = 0.
         ENDIF
      ENDIF
C
      IF(I.EQ.ISRC) DENOMT=DENOMT-VIPP/DXXI
      ROVE=(ROIM1*VIP-VIPM*ROIP1)/DXXI
C
C     SOURCES IN THE INTERFACE NODE DUE TO ABLATION
C
      SRC=0.0
      IF(I.NE.ISRC) GO TO 838
      SRC=-VEL*RODRY(I+1,J)
      SRC=AMAX1(0.0,SRC/DXXI)
838    CONTINUE
      XNUM=ROVE+RODRYI(I,J)/DT + SRC
      RODRY(I,J)=XNUM/(DENOMT)
      IF (RODRY(I,J) .LE. 0.0) RODRY(I,J)=0.0
      IF(I.EQ.1) PFLUX(J)=-VIM*RODRY(I,J)-VIP*ROIM1
C    *           +  GAMBL(1)*(RODRY(I,J)-ROIM1)/DXXI
      IF(I.GT.ISRC) GO TO 31
      VOLFR=RODRY(I,J)/RHOL(J)
      COR1=COR1+VOLFR
      CPBL=SP(J,2)
      IF(CPBL.EQ.0.0) CPBL=CPCON
      IF(J.EQ.3 .AND. VOLFR.LT.0.25) VOLFR=0.0
      TKBL=TKBL+VOLFR*(TKMAT(J)+GAMBL(I+1)*RHOL(J)*CPBL)
      IF(I.EQ.ISRC) DIFO=GAMBL(I+1)
31    CONTINUE
C
C      END OF MATRL DO LOOP
C
      IF(I.GT.ISRC) GO TO 32
      TKDRY(I)=(ALPHAL(I)*TKBL+(1.-ALPHAL(I))*0.03)*ALPHAL(I)
      F=1.-COR1
      IF(ABS(F).LT.0.00002) GO TO 32
      IF(ITNR.EQ.1) GO TO 360
      DFDV=(F-FOLD)/(V(I)-VOLD)
      FOLD=F
      VOLD=V(I)
      VNEW=V(I)-F/DFDV
      CHG=1.
      IF(VNEW.NE.0.0) CHG=ABS((VNEW-V(I))/VNEW)
      IF(CHG.GT.0.5) V(I)=(V(I)+VNEW)*0.5
      IF(CHG.LE.0.5) V(I)=VNEW
      IF(ITNR.GT.10) V(I)=(VOLD+VNEW)*0.5
      GO TO 205
360   FOLD=F
      VOLD=V(I)
      V(I)=V(I)*1.05-1.E-05
      GO TO 205
32    CONTINUE
33    CONTINUE
      RETURN
90    FORMAT(1H ,6E11.4)
      END
*/ **********************************************************************
*/ *        SUBROUTINE   REACT                                           *
*/ **********************************************************************
      SUBROUTINE REACT(TIME,DT,WTEMP,I,ICALC,RLIQ)
C
*CALL,/IMPLIC/
*CALL,/SLCOM1/
*CALL,/SLCOM2/
*CALL,/NACON/
C
       DIMENSION RLIQ(51,8)
C
```

```
C      LIQUID NAOH - SLURRY REACTIONS (SLAM)
C
C      SOURCES DUE TO GAS RELEASE
C
      EXH2(I)=0.0
      IF(ICALC.EQ.0 .OR. I.NE.ISRC) GO TO 70
      DO 60 JJ=1,2
      IF(JJ.EQ.1) J=2
      IF(JJ.EQ.2) J=11
      IF(J.EQ.2) SRC=H2OFLX+H2OSRC
      IF(J.EQ.11) SRC=CO2SRC
      SRC=AMAX1(0.0,SRC/(DEL1*(XZI(I+1)-XZI(I))))
      XNUM=RODRYI(I,J)/DT+SRC
      DENOM=1./DT
      RODRYI(I,J)=XNUM/DENOM
60    CONTINUE
70    CONTINUE
      IF(ICALC.EQ.0) TREACT=WTEMP
      IF(ICALC.NE.0) TREACT=TDRY(I)
C
C      NEXT THE CHEMICAL REACTIONS
C
C
      DO 121 IR=1,NREACT
      IF(SAK(IR).EQ.0.0) GO TO 121
      ICNTP=NPROD(IR)
      IF(NRCT(IR).GT.2) GO TO 100
C
C      2 REACTANTS
C
      XX1=RODRYI(I,MRR(IR,1))/(W(MRR(IR,1))*COFR(IR,MRR(IR,1)))
      XX2=RODRYI(I,MRR(IR,2))/(W(MRR(IR,2))*COFR(IR,MRR(IR,2)))
      FMIN=AMIN1(XX2,XX1)
      IF(FMIN.LE.1.E-3) GO TO 121
      FMAX=AMAX1(XX2,XX1)
      AKK=AMIN1(AKF(IR,1),SAK(IR)*EXP(-EK(IR)/(1.98*TREACT)))
      IF(RODRYI(I,3)/RHOL(3).GT.AKF(IR,3)) AKK=AKK*AKF(IR,2)
      IF(AKK.LT.1.E-03) GO TO 121
      BB=-(XX1+XX2+1./(AKK*DT))
      CC=XX2*XX1
      TERM=BB*BB-4.*CC
      XX=0.5*(-BB+SQRT(TERM))
      XX2=0.5*(-BB-SQRT(TERM))
      IF(XX.GT.FMIN) XX=XX2
      IF(FMAX/FMIN.GT.100.) XX=FMIN*(1.-EXP(-AKK*DT*FMAX))
      RLIQ(I,IR)=XX/DT
      DO 8855 IK=1,2
      RODRYI(I,MRR(IR,IK))=RODRYI(I,MRR(IR,IK))-XX*W(MRR(IR,IK))
     *   *COFR(IR,MRR(IR,IK))
8855   CONTINUE
      DO 30 IP=1,ICNTP
      MM=MPP(IR,IP)
      RODRYI(I,MM)=RODRYI(I,MM)+XX*W(MM)*COFP(IR,MM)
30    CONTINUE
      EXH2(I)=EXH2(I)+XX*W(4)*COFP(IR,4)/DT
335   CONTINUE
      GO TO 121
C
C      3 REACTANTS
C
100   CONTINUE
      IF(NRCT(IR).GT.3) WRITE(6,966) NRCT(IR)
966   FORMAT('    CODE NOT SET UP FOR ',I5,'    REACTANTS')
      XA=RODRYI(I,MRR(IR,1))/(COFR(IR,MRR(IR,1))*W(MRR(IR,1)))
      XB=RODRYI(I,MRR(IR,2))/(COFR(IR,MRR(IR,2))*W(MRR(IR,2)))
      XC=RODRYI(I,MRR(IR,3))/(COFR(IR,MRR(IR,3))*W(MRR(IR,3)))
      FMIN=AMIN1(XA,XB,XC)
      FMAX=AMAX1(XA,XB,XC)
      IF(FMIN.LE.1.E-3) GO TO 121
      IF(XA.GT.FMIN .AND. XA.LT.FMAX) FMID=XA
      IF(XB.GT.FMIN .AND. XB.LT.FMAX) FMID=XB
      IF(XC.GT.FMIN .AND. XC.LT.FMAX) FMID=XC
      AKK=AMIN1(AKF(IR,1),SAK(IR)*EXP(-EK(IR)/(1.98*TREACT))*FMAX)
      IF(RODRYI(I,3)/RHOL(3).GT.AKF(IR,3)) AKK=AKK*AKF(IR,2)
      IF(AKK.LT.1.E-03) GO TO 121
      BB=-(FMIN+FMID+1./(AKK*DT))
      CC=FMID*FMIN
      TERM=BB*BB-4.*CC
```

```
      XX=0.5*(-BB+SQRT(TERM))
      XX2=0.5*(-BB-SQRT(TERM))
      IF(XX.GT.FMIN) XX=XX2
      IF(FMAX/FMIN.GT.100.) XX=FMIN*(1.-EXP(-AKK*DT*FMAX))
      DMODT=(XX)/DT
      RLIQ(I,IR)=DMODT
      DO 5599 IK=1,3
      RODRYI(I,MRR(IR,IK))=RODRYI(I,MRR(IR,IK))-XX*W(MRR(IR,IK))
     *   *COFR(IR,MRR(IR,IK))
5599  CONTINUE
      DO 50 IP=1,ICNTP
      MM=MPP(IR,IP)
      RODRYI(I,MM)=RODRYI(I,MM)+XX*W(MM)*COFP(IR,MM)
50    CONTINUE
      EXH2(I)=EXH2(I)+COFP(IR,4)*W(4)*DMODT
121   CONTINUE
      RETURN
      END
*/ ********************************************************************
*/ *        SUBROUTINE   HEAT                                          *
*/ ********************************************************************
      SUBROUTINE HEAT (ICALC,RLIQ)
C
*CALL,/IMPLIC/
*CALL,/SLCOM1/
*CALL,/SLCOM2/
*CALL,/NACON/
C
      DIMENSION RLIQ(51,8)
C
C      DEFINE COFS IN THE CHEMICAL REACTION EQN
C
      QV(1)=0.0
      IF(ICALC.NE.0) QCV=0.0
      IMAX=ISRC
      IF(ICALC.EQ.0) IMAX=NPOOL
      DO 100 I=1,IMAX
      SUMR=0.0
      SUMP=0.0
      DO 10 IR=1,NREACT
      IF(RLIQ(I,IR).LT.1.E-05) GO TO 10
      DO 30 M=1,14
      IF(COFR(IR,M).EQ.0.0) GO TO 20
C     I ENTHALPY OF DRY REGION- PROP
      CALL SLENTH(TDRY(I),ENTH,M,IPHASR(IR,M))
      SUMR=SUMR+ENTH*RLIQ(I,IR)*COFR(IR,M)
20    CONTINUE
      IF(COFP(IR,M).EQ.0.0) GO TO 30
      CALL SLENTH(TDRY(I),ENTH,M,IPHASP(IR,M))
      SUMP=SUMP+ENTH*RLIQ(I,IR)*COFP(IR,M)
30    CONTINUE
10    CONTINUE
      IF(ICALC.EQ.0) GO TO 99
      QV(I+1)=SUMR-SUMP
      QCV=QCV+(QV(I+1)+QV(I))*0.5*DEL1*XZI(2)
      GO TO 100
99    QVP(I)=SUMR-SUMP
100   CONTINUE
      IF(ICALC.NE.0) QCV=QCV+0.5*QV(ISRCP1)*DEL1*XZI(2)
      RETURN
      END
*I RB-CONCR.397
*/ ********************************************************************
*/ * SLOC  :  NEWLY  ESTABLISHED SUBROUTINES                           *
*/ *          1.  CONCRM    2.  SLAM   3.  CONCPT                      *
*/ *          4.  TRANA     5.  TRANB  6.  TRANC                       *
*/ *          7.  LAYMOD                                               *
*/ ********************************************************************
*DECK,RB-CONCM
*/ ********************************************************************
*/ *        SUBROUTINE   CONCRM                                        *
*/ ********************************************************************
      SUBROUTINE CONCRM
     1 (ATA,ATF,DIAG,FJAC,FVEC,XSOL,WRK,IWRK, CKM,CCP,HQ,KFLPL,KSPPL,
     2  ONAMES,OMMTPL,OTYP,CQFP,CQ3P,CA,CDZ,CDZO,CMA,CMAO,CRM,CRMO,
     3  TBCL,TBCU,CT,CTA,CTAO,VCON, PMA,CTDX,CTDY,OCTBNX,OCTBNY,OCTBNM,
     4  ICTBFL,ICTBNP, KZO,NNP, ASRE,ASRM,PSRE,PSRM,RISRE,RISRM,RIMA,
     5  RIQ3P,CSRE,CSRM,MW,HL,CMAS,NIND,NMAT,N,PT,PRM,PTA,PTAO,PDZ,
```

```
     6  PQ3P,AQ3P,RIT,RITA,RITAO,RIDZ,RIRM)
C ... INCLUDE 'IMPLIC.CMN'
*CA /IMPLIC/
C
      REAL ATA(N,*),ATF(*),DIAG(*),FJAC(N,*),FVEC(*),XSOL(*),WRK(*)
      REAL CKM(*),CCP(*),HQ(*),CQFP(*),CQ3P(*),CA(*),CDZ(*),CDZO(*)
      REAL CMA(NMAT,*),CMAO(NMAT,*),CRM(*),CRMO(*),TBCL(*),TBCU(*)
      REAL CT(*),CTA(*),CTAO(*),VCON(*),PMA(NMAT,*),PRM(*),PT(*)
      REAL CTBX(*),CTBY(*),PTA(*),PTAO(*),PDZ(*),PQ3P(*),AQ3P(*)
      REAL ASRE(NIND,NMAT,*),ASRM(NIND,NMAT,*),PSRE(NIND,NMAT,*)
      REAL PSRM(NIND,NMAT,*),RISRE(NIND,NMAT,*),RISRM(NIND,NMAT,*)
      REAL CSRE(NIND,NMAT,*),CSRM(NIND,NMAT,*)
      REAL RIMA(NMAT,*),RIQ3P(*),HL(*),CMAS(*)
      REAL NW(*),RIT(*),RITA(*),RITAO(*),RIDZ(*),RIRM(*)
      INTEGER IWRK(*),KFLPL(*),KSPPL(*),ICTBFL(*),ICTBNP(*)
      INTEGER KZO(*),NNP(*),NIND,NMAT,N,ISLCN(17),ICNSL(54)
      REAL ROPOOL(17),PFLUX(17),TOTMS(17),ROINT(17),WRO(17)
      REAL V(51),RLIQ(51,8)
      CHARACTER*8 ONAMES(*),OMMTPL(*),OTYP(*)
      CHARACTER*8 OCTBNX(*),OCTBNY(*),OCTBNM(*)
      EXTERNAL CONEQN
C ... INCLUDE 'ATMOS.CMN'
C ... INCLUDE 'CNCRT.CMN'
C ... INCLUDE 'CONST.CMN'
C ... INCLUDE 'FLAGS.CMN'
C ... INCLUDE 'HCOEF.CMN'
C ... INCLUDE 'MATER.CMN'
C ... INCLUDE 'NACON.CMN'
C ... INCLUDE 'PTRS.CMN'
C ... INCLUDE 'REBCOM.CMN'
C ... INCLUDE 'REBLAY.CMN'
C ... INCLUDE 'SOURC.CMN'
C ... INCLUDE 'SPACE.CMN'
C ... INCLUDE 'TAPE.CMN'
C ... INCLUDE 'TIME.CMN'
C
C *** ATMOSPHERIC GAS PROPERTIES/PARAMETERS ***  RI=KKM
C
*CA /ATMOS/
*CA /CONCRT/
*CA /CONST/
*CA /FLAGS/
*CA /OUTFLG/
*CA /HCOEF/
*CA /MATER/
*CA /NACON/
*CA /PTRS/
*CA /REBCOM/
*CA /REBLAY/
*CA /SOURCE/
*CA /SPACE/
*CA /TAPE/
*CA /TIME/
*CA /SLCOM1/
*CA /SLCOM2/
C
      LOGICAL ACTIV,ACTIVH,ABLAFL
      DATA OROUT/'CONCRM'       /
C     CONVERSION ARRAY FROM SLAM MATERIAL NUMBER TO CONTAIN MATERIAL
C     NUMBER
      DATA ISLCN/25,8,11,4,21,22,47,19,41,52,6,26,53,40,54,50,15/
C     CONVERSION ARRAY FROM CONTAIN MATERIAL NUMBER TO SLAM MATERIAL
C     NUMBER
      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
     1          ,14,9,5*0,7,4*0,10,13,15/
C ... INCLUDE 'DEBUGO.CMN'
*CA /DEBUGO/
C
C
C  *********************************************************************
C  *                                                                   *
C  *      THE PHYSICS CONTROLING ROUTINE FOR A CONCRETE LAYER          *
C  *                                                                   *
C  *********************************************************************
C
C
      OBLOC = 'CONCRT'
      CALL  LOAD ( OBLOC )
```

```
C
C     RESET  OUTPUT FLAGS
C
      SLMOUT = .FALSE.
      WATOUT = .FALSE.
C
      IF (JINT .EQ. 0) LNFAIL =.TRUE.
C
C
C     SWITCH  SODIUM CONCRETE INTERACTION OR WATER  RELEASE
C
      IF  (LNFAIL.AND.NFCONC)   THEN
                     NFCONC  =  .FALSE.
                     NFNACO  =  .TRUE.
                     ISRC = IH(LRNNP+KREG)/4
                     DO  10  I = 1,ISRC
                     DO  10  J = 1,NSLM
                     RODRY(I,J)=0.0
                     IF (J .EQ. 3) RODRY(I,J)=RHOL(3)
10                   CONTINUE
                     CALL  TRANB
      ENDIF
C
C                          *********
C
C        TABLE MODIFICATIONS TO THE HEAT TRANSFER COEFFICIENT
C
C                          *********
C
      IF( HTTBFL ) THEN
C
         IF( OCTBNX(IHTTBL).EQ.'TIME' ) THEN
         TIME= PTIME + PDT/2.0
         CALL TABLE
     1       ( IHTTBL,CTBX,CTBY,ICTBFL,ICTBNP, ACTIVH,TIME,HTVAL )
C
         ELSE IF( OCTBNX(IHTTBL).EQ.'TEMP' ) THEN
         TEMP= ( TBCU(KREG) + CT( (KZO(KREG)+NNP(KREG)-1)*3 ) ) *0.5
         CALL TABLE
     1              ( IHTTBL,CTBX,CTBY,ICTBFL,ICTBNP, ACTIVH,TEMP,HTVAL )
C
C  TERMINATE :       IF( OCTBNX(IHTTBL).EQ.'XXXXXXX' ) THEN
C
         ENDIF
C
C   TERMINATE :      IF( HTTBFL ) THEN
C
      ENDIF
C
C   IF THE OVERLYING STRUCTURE MATCHES THE CHARACTER NAME OF THE TABEL
C         THEN RESET THE DEFAULT HEAT TRANSFER COEFFICIENT
C
      IF( ACTIVH .AND. OCTBNM(IHTTBL).EQ.OTYP(KRUP) ) THEN
C
      HQ(KREG+1)= HTVAL
C
      ELSE
C
        IF( KREG.EQ.(NREG-1) ) THEN
        LFCOEF= HTUPLY
        ELSE
        LFCOEF= HTLAY
        ENDIF
      IF( LFCOEF ) THEN
      HQ(KREG+1)= HL(2*KREG)*HL(2*KREG+1) / (HL(2*KREG)+HL(2*KREG+1))
      ELSE
      HQ(KREG+1)= 0.0
      ENDIF
C
C  TERMINATE :  IF( ACTIVH .AND. OCTBNM(IHTTBL).EQ.OTYP(KREG+1) ) THEN
C
      ENDIF
C
      ACTIV = .TRUE.
      KPOOL = JCONC + JINT + 1
      KINT  = JCONC + 1
C *********************************************************************
C *      SODIUM   CONCRETE  INTERACTIONS                              *
```

```
C  *********************************************************************
         IF( LNFAIL .AND. NFNACO .AND. NRFAST) THEN
C
C  CHECK ON INTERACTION CONSTRAINTS : SODIUM M > 0.0
C                                     PTIME > TNATO
C                                     PTIME < TNATN
C
           KPOOL= JCONC + JINT + 1
C
           IF( PMA(LNAL,KZO(KPOOL)).GT.0.0 ) THEN
           ACTIV= .TRUE.
           ELSE
           ACTIV= .FALSE.
           ENDIF
C
C  TERMINATE :      IF( LNFAIL .AND. NFNACO .AND NRFAST ) THEN
C
C
         ENDIF
C
C
C
C
      IF (.NOT.ACTIV) RETURN
C
C
      KZ=KZO(KREG)
      AREA = H(LRCA+KZ)
      PINFXT=H(LPGAS+NCELL)
      TINFXT=H(LTGAS+NCELL)
      TOTMAS=0.0
C  *
C  *  DETERMINATION  OF  DENSITY IN LAYER
C  *
      IF ( NFNACO )  THEN
           KZ = KZO(KPOOL)
           J = 3*(KZ-1) + 1
           TPOOL = PT(J)
           DTRDT=(PTA(KZO(KPOOL))-PTAO(KZO(KPOOL)))/PDT
           WDTDT=DTRDT
           DO 55 J = 1, NMAT
           TOTMAS = TOTMAS + PMA(J,KZ)
 55        CONTINUE
           PRM(KZ) = TOTMAS/(AREA*PDZ(KZ))
      ELSE
           KZ = KZO(KINT)
           J = 3*(KZ-1) + 1
           TINT = RIT(J)
           DTIDT=(RITA(KZO(KINT))-RITAO(KZO(KINT)))/PDT
           WDTDT=DTIDT
           DO 56 J = 1, NMAT
           TOTMAS = TOTMAS + RIMA(J,KZ)
 56        CONTINUE
           RIRM(KZ) = TOTMAS/(AREA*RIDZ(KZ))
      ENDIF
      SUM=0.0
      DO 50 I=1,NSLM
      IF(I.NE.15)  THEN
      II = IMATER(ISLCN(I)+1)
      IF( NFNACO ) THEN
      ROPOOL(I)=PMA(II,KZO(KPOOL))/(AREA*PDZ(KZO(KPOOL)))
      WRO (I) = ROPOOL(I)
      SUM = SUM + ROPOOL(I)
      ELSE
      ROINT(I)=RIMA(II,KZO(KINT))/(AREA*RIDZ(KZO(KINT)))
      WRO (I) = ROINT(I)
      SUM = SUM + ROINT(I)
      ENDIF
      ENDIF
   50 CONTINUE
      IF( NFNACO ) THEN
          II = KZO(KPOOL)
          ROPOOL(15)=PRM(II)-SUM
          WRO(15)=ROPOOL(15)
          WATOUT=.FALSE.
          SLMOUT=.TRUE.
          WTEMP = TPOOL
      ELSE
```

```
          II = KZO(KINT)
          ROINT(15)=RIRM(II)-SUM
          WRO(15)=ROINT(15)
          WATOUT=.TRUE.
          SLMOUT=.FALSE.
          WTEMP = TINT
      ENDIF
C
C   ***********************************************************************
C   *  LOAD STORAGE ARRAY CH,ICH WITH COMMON BLOCK                        *
C   ***********************************************************************
      CALL  TRANA
C
      TSO    = TS
      TOTH2O = 0.0
      TOTCO2 = 0.0
      TOTH2G = 0.0
      TOTQNP = 0.0
      TOTQNI = 0.0
      TOTQNA = 0.0
      DO  60  I = 1,NSLM
   60 TOTMS(I) = 0.0
C
      SLTIME = PTIME
      ISLAF  = 0
      WDT    = PDT
  100 CONTINUE
C
C     DETERMINATION  OF  TIME  STEP
C
      CALL  STIME (SLTIME,WDT,ISLAF,WTEMP,TSO,WDTDT)
C   ***********************************************************************
C   *                                                                     *
C   *     SOLVE THE CONSERVATION EQUATIONS FOR THE CONCRETE LAYER         *
C   *                                                                     *
C   ***********************************************************************
C
      TOTM= 0.0
      DO 300 KZ=KZO(KREG),(KZO(KREG)+NNP(KREG)-1)
         DO 455 J=2,NMAT
         TOTM= TOTM + CMA(J,KZ)
  455    CONTINUE
  300 CONTINUE
C
      KZ = KZO(KPOOL)
      J = 3*(KZ-1) +1
      TPOOL = PT(J)
C
      IF( TOTM.GT.0.0 ) CALL SLAM
     1  ( SLTIME,SDT,WTEMP,WRO,PINFXT,TINFXT,WDTDT,RLIQ,PFLUX)
C
      DTSDT = (TS-TSO)/SDT
      TSO   =   TS
C
C CALCULATION OF MASS AND ENERGY WHICH IS RELEASED TO POOL OR INTERM
C
C
      IF (NFNACO) THEN
      TOTH2G = TOTH2G + PFLUX(4) * SDT * AREA
      TOTCO2 = TOTCO2 + PFLUX(11) * SDT * AREA
      TOTQNP = TOTQNP - QCONO * SDT * AREA
      TOTQNA = TOTQNA + SP(4,3)*(WTEMP-TINFXT)*PFLUX(4)*SDT*AREA
     1                + SP(11,3)*(WTEMP-TINFXT)*PFLUX(11)*SDT*AREA
      DO  320 I = 1,NSLM
      IF (I.EQ.4 .OR. I.EQ.11) GO TO  320
      TOTMS(I) = TOTMS (I) + PFLUX(I)*AREA*SDT
  320 CONTINUE
      ELSE
      TOTQNI = TOTQNI - QCONO * SDT * AREA
      ENDIF
C
      IF (ISLAF.EQ.0) GO TO 100
      CALL  TRANB
C   ***********************************************************************
C   *                                                                     *
C   * AT THIS POINT THE MASS SOURCES HAVE BEEN DETERMINED, NOW DEFINE     *
C   *           THE NEW MASS TOTALS IN THE  LAYER                         *
C   *                                                                     *
```

```
C  ***********************************************************************
C
      CALL TRANC(CA,CDZ,CMA,CT,CTA,CRM,KZO,NNP,ISLCN,PSRM,PSRE,ASRM,
     1  ASRE,TOTH2G,TOTH2O,TOTCO2,TOTQNP,TOTQNI,TOTQNA,TOTMS,PQ3P,AQ3P,
     2  RISRM,RIQ3P,NIND,NMAT)
C
  350 CONTINUE
C
C                          *********
C
      IM(LRNNP+KREG)=NWET+NDRY-2
C
C
C  RESET THE NEXT LAYERS LOWER BOUNDRY CONDITION
C
      TBCL(KREG+1)= CT( 3*(KZO(KREG)+NNP(KREG)-1) )
C
C   TERMINATE :  IF( ACTIV .AND. ONACO.EQ.'SLAM' ) THEN
C
      CALL UNLOAD ( OBLOC )
      RETURN
      END
*/ ***********************************************************************
*/ *         SUBROUTINE   SLAM                                            *
*/ ***********************************************************************
      SUBROUTINE SLAM(TIME,DT,WTEMP,WRO,PINFXT,TINFXT,WDTDT,
     1                 RLIQ,PFLUX)
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),WRO(*)
     2         ,PFLUX(*),RLIQ(51,8)
C ***********************************************************
C *                                                         *
C *    PHYSICS OF THE CONCRETE                              *
C *                                                         *
C ***********************************************************
C
C       UPDATE BEGINNING OF TIME STEP VALUES
C
      DO 55 N=1,NTERM
55    AMO(N)=AM(N)
      DEL10=DEL1
      TRO=WTEMP
      H2OFLO=H2OFLX
      VELOLD=VEL
      VEL=0.5*(VELOLD+VELN)
      IF(ABS(VEL).LT.1.E-10) VEL=0.0
      DIST=DIST-VEL*DT
      IMPL=0
      EPP=0.5
66    CONTINUE
C
C ***********************************************************************
C *   WATER   RELEASE  AND  HEAT  TRANSFER  CALCULATION                  *
C ***********************************************************************
C
      CALL   CONEQS(TIME,DT,WTEMP,PINFXT,TINFXT,WDTDT,DEL10,TRO,H2OFLO,
     1    EPP,ERR1,COMP1,TOK,SUMTRM,FX,AMAT,CCOL,XVAR,AMO,SINT,VELOLD)
C
      DDL1DT=(DDL1DN+VEL+DDL1DT)*0.5
      DEL1=DEL10+(DDL1DT)*DT
      IF(ERR1.GT.1.E-03) GO TO 66
C
C  **********************************************************************
C  *   WE MUST NOW DECIDE WHICH  CONCRETE MODEL TO USE                  *
C  *          ACCORDING TO ACCIDENT SCENARIO                            *
C  **********************************************************************
      IF ( NFNACO ) THEN
C                          *********
C
C                SODIUM CONCRETE INTERACTIONS
```

```
C
C                           *********
          CALL NATCON (TIME,DT,WTEMP,WRO,RLIQ,DIFO,
     1                     PFLUX,PINFXT,TINFXT)
C
C                           *********
C
C               WATER RELEASE ROUTINE
C
C                           *********
      ELSE
          CALL WTRRLS (TIME,DT,WTEMP,WRO,RLIQ,DIFO,
     1                     PFLUX,PINFXT,TINFXT)
      ENDIF
C
C
C      EXPLICIT CALCULATIONS AND INTEGRALS FOR THE NEXT TIME STEP
C
C
      WFLXDT=WFLXDT+(H2OFLX+H2OSRC)*DT -VEL*RODRY(ISRCP1,2)*DT
      XNTCO2=XNTCO2+CO2SRC*DT
C
C
C      EVALUATE PRESSURE DISTRIBUTION IN DRY ZONE
C
      DO 850 I=1,NDRY
      PDRY(I)=((H2OFLX*COMP1(I)+DEL1*SUMTRM(I))*TRO+PINFXT*PINFXT)**0.5
850   CONTINUE
      RETURN
90    FORMAT(6E11.3)
      END
*/ *********************************************************************
*/ *         SUBROUTINE   CONCPT                                        *
*/ *********************************************************************
      SUBROUTINE  CONCPT
C
C     THIS ROUTINE DETERMINES POINTERS OF CONCRETE
C       STORAGE ARRAY CH(*) AND ALSO STORES PARAMETERS.
C
*CA /IMPLIC/
*CA /PTRS/
*CA /SPACE/
*CA /SLCOM1/
*CA /SLCOM2/
      NTM = 31
      LICH = (NCELL-1)*NTM
      ICH(LICH + 1) = IMPL
      ICH(LICH + 2) = IMPL2
      ICH(LICH + 3) = ISRC
      ICH(LICH + 4) = NDRY
      ICH(LICH + 5) = NWET
      ICH(LICH + 6) = NTERM
      ICH(LICH + 7) = NPOOL
      ICH(LICH + 8) = NSLM
C
C     SET  POINTERS
C
      LC1 = 0
      IF (NCELL .GT. 1) THEN
      LICHM = LICH-NTM
      LC1 = ICH(LICHM+NTM)
      ENDIF
      JSLM = 33
      LC2 = LC1 + JSLM
      LC3 = LC2 + NDRY*NSLM
      LC4 = LC3 + NDRY
      LC5 = LC4 + NDRY
      LC6 = LC5 + NDRY
      LC7 = LC6 + NWET
      LC8 = LC7 + NWET
      LC9 = LC8 + NWET
      LC10 = LC9 + NWET
      LC11 = LC10 + NWET
      LC12 = LC11 + NWET
      LC13 = LC12 + 52
      LC14 = LC13 + NTERM
      LC15 = LC14 + NTERM
      LC16 = LC15 + NTERM*NDRY
```

```
      LC17 = LC16 + NTERM*NDRY
      LC18 = LC17 + NTERM*NTERM
      LC19 = LC18 + 51
      LC20 = LC19 + 51
      LC21 = LC20 + 51
      LC22 = LC21 + 51
      LC23 = LC22 + 51
      CALL  IMOVE (LPOINT(1),ICH(LICH+9),23)
      RETURN
      END
C
*/ ***********************************************************************
*/ *        SUBROUTINE   TRANA                                           *
*/ ***********************************************************************
      SUBROUTINE  TRANA
C
C     THIS ROUTINE LOADS STORAGE ARRAY CH,ICH
C          WITH   CONCRETE  COMMON BLOCK
C
*CA /IMPLIC/
*CA /SPACE/
*CA /PTRS/
*CA /SLCOM1/
C
C    READ POINTERS
C
      NTM =  31
      LICH = (NCELL-1) * NTM
      IMPL =  ICH(LICH + 1)
      IMPL2 =  ICH(LICH + 2)
      ISRC =  ICH(LICH + 3)
      NDRY =  ICH(LICH + 4)
      NWET =  ICH(LICH + 5)
      NTERM =  ICH(LICH + 6)
      NPOOL=  ICH(LICH + 7)
      NSLM =  ICH(LICH + 8)
      CALL IMOVE(ICH(LICH+9),LPOINT(1),23)
C
      ISRCP1 = ISRC + 1
      ISRCP3 = ISRC + 3
      ISRCP5 = ISRC + 5
C
C     LOAD VALUES ESTIMATED FROM CONCRETE STORAGE ARRAY  CH( )
C
      CALL  RMOVE (CH(LC1+1),RSLCEL,JSLAMR)
C
      DO  200  J=1,NSLM
      DO  200  I=1,NDRY
      LLL  = LC2+I+NDRY*(J-1)
      RODRY(I,J)=CH(LLL)
  200 CONTINUE
C
      DO  400  I=1,NDRY
      TDRY(I)=CH(LC3+I)
      TKDRY(I)=CH(LC4+I)
      ALPHAL(I)=CH(LC5+I)
  400 CONTINUE
C
      DO  600  I=1,NWET
      XETA(I)=CH(LC6+I)
      TWET(I)=CH(LC7+I)
      PWET(I)=CH(LC8+I)
      ROAIR(I)=CH(LC9+I)
      ROH2OL(I)=CH(LC10+I)
      ALPH(I)=CH(LC11+I)
  600 CONTINUE
C
      DO  800  I=1,52
      QV(I)=CH(LC12+I)
  800 CONTINUE
C
      DO  1000 I=1,NTERM
      AM(I)=CH(LC13+I)
      SIGN(I)=CH(LC14+I)
 1000 CONTINUE
      DO 1200  I = 1,NTERM
      DO 1200  J = 1,NDRY
      LLL  = LC15+I+NTERM*(J-1)
```

```
      KKK   = LC16+I+NTERM*(J-1)
      SINX(I,J) = CH(LLL)
      COSX(I,J) = CH(KKK)
 1200 CONTINUE
      DO 1300  I = 1,NTERM
      DO 1300  J = 1,NTERM
      MMM  = LC17+I+NTERM*(J-1)
      ZCOSI(I,J)=CH(MMM)
 1300 CONTINUE
C
      DO  1400 I = 1,51
      XZI(I) = CH(LC18+I)
      WTZ(I) = CH(LC19+I)
      XETA(I) = CH(LC20+I)
      ETA(I) = CH(LC21+I)
      WT(I) = CH(LC22+I)
 1400 CONTINUE
      RETURN
      END
*/ ************************************************************************
*/ *        SUBROUTINE   TRANB                                              *
*/ ************************************************************************
      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)
 1200 CONTINUE
      DO 1300 I = 1,NTERM
      DO 1300 J = 1,NTERM
      MMM  = LC17+I+NTERM*(J-1)
      CH(MMN) = ZCOSI(I,J)
 1300 CONTINUE
C
      DO  1400 I=1,51
      CH(LC18+I) = XZI(I)
      CH(LC19+I) = WTZ(I)
      CH(LC20+I) = XETA(I)
      CH(LC21+I) = ETA(I)
      CH(LC22+I) = WT(I)
 1400 CONTINUE
C
      RETURN
      END
*/ ********************************************************************
*/ *        SUBROUTINE   TRANC                                         *
*/ ********************************************************************
      SUBROUTINE TRANC (CA,CDZ,CMA,CT,CTA,CRM,KZO,NNP,ISLCN,PSRM,PSRE,
     1   ASRM,ASRE,TOTH2G,TOTH2O,TOTCO2,TOTQNP,TOTQNI,TOTQNA,TOTMS,PQ3P,
     2   AQ3P,RISRM,RIQ3P,NIND,NMAT)
C
*CALL,/IMPLIC/
C
      REAL  CA(*),CDZ(*),CMA(NMAT,*),CT(*),CTA(*),CRM(*)
      REAL PSRM(NIND,NMAT,*),PSRE(NIND,NMAT,*),TOTMS(*),PQ3P(*),AQ3P(*)
      REAL ASRM(NIND,NMAT,*),ASRE(NIND,NMAT,*)
      REAL RISRM(NIND,NMAT,*),RIQ3P(*)
      INTEGER ISLCN(17),KZO(*),NNP(*)
*CALL,/SLCOM1/
*CALL,/SLCOM2/
*CALL,/PTRS/
*CALL,/MATER/
*CALL,/TIME/
*CALL,/REBCOM/
*CALL,/REBLAY/
*CALL,/SPACE/
C  *********************************************************************
C  *  TRANSPORT  SLAM  CALCULATION RESULTS TO  CONTAIN VARIABLE        *
C  *********************************************************************
      AREA = H(LRCA+KZO(KREG))
C     WET  REGION   ********************
      NM1 = NWET - 1
      DO  200  I = 1,NM1
      J = I + 1
      KZ = KZO(KREG)+NWET-I-1
C
C     CALCULATE  NEW  LAYER  THICKNESS
C
      CDZ(KZ)=XETA(J)-XETA(J-1)
C
C     NEW  MASS  IN  THE  LAYER
C
      DO  100  L = 1,NMAT
  100 CMA(L,KZ) = 0.0
      VOL = AREA*CDZ(KZ)
C
      CMA(IMATER(ISLCN(1)+1),KZ)=RHOCON*FSIO2*VOL
      CMA(IMATER(ISLCN(2)+1),KZ)=RHOCON*(FH2OE+FH2OB)*VOL
      CMA(IMATER(ISLCN(10)+1),KZ)=RHOCON*FCACO3*VOL
      CMA(IMATER(ISLCN(13)+1),KZ)=RHOCON*FMGCO3*VOL
      II  = IMATER(ISLCN(15)+1)
      CMA(II,KZ)=RHOCON*(1.-FCACO3-FMGCO3-FSIO2-FH2OE-FH2OB)*VOL
C
C     REDEFINE  MIXTURE  DENSITY
C
      CRM(KZ)=RHOCON
C
C     DEFINE  TEMPERATURE
C
      II = KZ*3-1
      IF(I .EQ. 1) THEN
        CT(II+1)=TDRY(NDRY)
        CT(II)=(TDRY(NDRY)+TWET(I))/2.0
      ELSE
```

```
          CT(II+1)=TWET(I-1)
          CT(II)=(TWET(I-1)+TWET(I))/2.0
        ENDIF
        CT(II-1)=TWET(I)
        CTA(KZ)=CT(II)
  200 CONTINUE
C       DRY   REGION
        NM1 = NDRY - 1
        DZZ = DEL1*XZI(2)
        DO  500  I = 1,NM1
        J = I + 1
        KZ = KZO(KREG)+NWET-2+NDRY-I
C
C       CALCULATE  NEW  LAYER  THICKNESS
C
        CDZ(KZ)=DZZ
C
C       NEW  MASS  IN  THE  LAYER
C
        DO  300  L = 1,NMAT
  300   CMA(L,KZ)=0.0
        VOL = AREA*DZZ
        CRM(KZ)=0.0
C
        DO  400 L = 1,NSLM
        CMA(IMATER(ISLCN(L)+1),KZ)=RODRY(I,L)*VOL
  400   CRM(KZ)=CRM(KZ)+RODRY(I,L)
C
C       DEFINE  TEMPERATURE
C
        II = KZ*3-1
        CT(II+1)=TDRY(J-1)
        CT(II)=(TDRY(J-1)+TDRY(J))/2.0
        CT(II-1)=TDRY(J)
        CTA(KZ)=CT(II)
  500 CONTINUE
C ********************************************************************
C *            TRANSPORT  MASS  AND  ENERGY                          *
C ********************************************************************
C
        IF ( NFNACO ) THEN
        KPOOL = JCONC + JINT + 1
        DO  700  KZ =KZO(KPOOL),KZO(KPOOL)+NNP(KPOOL)-1
        DO  600  I = 1,NSLM
        II = IMATER(ISLCN(I)+1)
  600 PSRM(1,II,KZ)=PSRM(1,II,KZ) + TOTMS (I)/PDT/NNP(KPOOL)
  700 PQ3P(KZ)=PQ3P(KZ) + TOTQNP /PDT/NNP(KPOOL)
C
        ELSE
        KINT = JCONC + 1
        KPOOL = JCONC + JINT + 1
        DO  900  KZ =KZO(KINT),KZO(KINT)+NNP(KINT)-1
  900 RIQ3P(KZ)=RIQ3P(KZ) + TOTQNI/PDT/NNP(KINT)
        PQ3P(KZO(KPOOL))=PQ3P(KZO(KPOOL)) + TOTQNP/PDT/NNP(KINT)
        ENDIF
        IF (NFCONC) RETURN
C
        KZATM=KZO(NREG)
        II = IMATER(ISLCN(4)+1)
        ASRM(1,II,KZATM) = ASRM(1,II,KZATM) + TOTH2G/PDT
        II = IMATER(ISLCN(11)+1)
        ASRM(1,II,KZATM) = ASRM(1,II,KZATM) + TOTCO2/PDT
        AQ3P(KZATM)=AQ3P(KZATM) + TOTQNA /PDT
        RETURN
        END
*I I-IREBEL.325
*/ ********************************************************************
*/ *         SUBROUTINE   LAYMOD                                     *
*/ ********************************************************************
        SUBROUTINE LAYMOD (OTYP,KZO,NNP)
*CA,/IMPLIC/
        CHARACTER*8 OTYP(*)
        INTEGER  KZO(*),NNP(*)
*CA /NACON/
*CA /PTRS/
*CA /REBCOM/
*CA /REBLAY/
*CA /SPACE/
```

```
      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         *
*/ *         MODIFICATIONS   OF   CONTAIN 1.03   ***                       *
*/ ************************************************************************
*/ ************************************************************************
*/ * SLOD01 : CREATED SUBROUTINE 'LAYMOD' IS CALLED WHEN LINER FAILURE *
*/ *           OCCURS                                                     *
*/ ************************************************************************
*I RB-RBCNT.53
C
C     RECONSTRUCT THE LAYER STRUCTURE WHEN LINER FAILURE OCCURS
C
      IF ((JINT .GE. 2) .AND. LNFAIL .AND. LDUMNC) THEN
      CALL LAYMOD (OH(LROTYP+1),IH(LRKZO+1),IH(LRNNP+1))
      ENDIF
*/ ************************************************************************
*/ * SLOD02 : CREATED SUBROUTINE 'CONCRM' IS CALLED IN CASE OF SLAM     *
*/ ************************************************************************
*I RB-RBCNT.160
         IF ( (.NOT.NFNACO) .AND. (.NOT.NFCONC) )  THEN
*I RB-RBCNT.185
         ELSE IF ( NFNACO .OR. NFCONC ) THEN
      CALL  CONCRM
     1   ( H(LRBTMP+LTATA),H(LRBTMP+LTATF),H(LRBTMP+LTDIAG),
     2     H(LRBTMP+LTFJAC),H(LRBTMP+LTFVEC),H(LRBTMP+LTXSOL),
     3     H(LRBTMP+LTWRK),IH(LIBTMP+LTIWRK),
     4     H(LRBTMP+LTCKM),H(LRBTMP+LTCCP),H(LRHQ+1),
     5     IH(MFLPL+1),IH(MSPPL+1),OH(INAMES+1),OH(MMTPL+1),
     6     OH(LROTYP+1),H(LRBTMP+LTCQFP),H(LRBTMP+LTCQ3P),
     7     H(LRCA+1),H(LRCDZ+1),H(LRCDZO+1),H(LRCMA+1),
     8     H(LRCMAO+1),H(LRBTMP+LTCRM),H(LRBTMP+LTCRMO),
     9     H(LRTBCL+1),H(LRTBCU+1),H(LRCT+1),H(LRCTA+1),
     1     H(LRCTAO+1),H(LRVCON+1),H(LRPMA+1),H(LCTBX+1),
     1     H(LCTBY+1),OH(LCTBNX+1),OH(LCTBNY+1),OH(LCTBNM+1),
     2     IH(LCTBFL+1),IH(LCTBNP+1),IH(LRKZO+1),IH(LRNNP+1),
     3     H(LRBTMP+LTASRE),H(LRBTMP+LTASRM),H(LRBTMP+LTPSRE),
     4     H(LRBTMP+LTPSRM),H(LRBTMP+LTISRE),H(LRBTMP+LTISRM),
     5     H(LRIMA+1),H(LRBTMP+LTIQ3P),H(LRBTMP+LTCSRE),
     6     H(LRBTMP+LTCSRM),H(IMW+1),H(LRBTMP+LTHL),H(LRCMAS+1),NIND,
     7     NM,NMAX,H(LRPT+1),H(LRBTMP+LTPRM),H(LRPTA+1),H(LRPTAO+1),
     8     H(LRPDZ+1),H(LRBTMP+LTPQ3P),H(LRBTMP+LTAQ3P),H(LRIT+1),
     9     H(LRITA+1),H(LRITAO+1),H(LRIDZ+1),H(LRBTMP+LTIRM))
      ENDIF
*/ ************************************************************************
*/ * SLOD03  :  REDUCE COMMENT STATEMENT AND SLAM DUMMY SUBROUTINE       *
```

```
*/ *           AND RELOCATE SLAM  SUBROUTINE                                 *
*/ ***********************************************************************
*D,I-INACON.55
      ELSE IF (OTEST.EQ.'SLAM')  THEN
*D I-INACON.61,62
*D I-INACON.65,80
*D C100K.104,115
*/ ***********************************************************************
*/ * SLOD04  :  RELOCATION  OF  SLAM  SUBROUTINE                             *
*/ ***********************************************************************
*I I-IREBEL.295
      KREG = JCONC
      CALL  LOAD( OREBLA )
      IF (LDUMNC)  THEN
          CALL  SLINPT
      ENDIF
*/ ***********************************************************************
*/ * SLOE01  :  ALLOCATES  STORAGE  ARRAY  FOR  SLAM CALCULATION         *
*/ ***********************************************************************
*I /SPACE/.27
C
C    ***    SLAM   POINTERS    ****
C
      REAL  CH
      INTEGER  ICH
      COMMON /CSLAM1/  CH(9000)
      COMMON /CSLAM2/  ICH(200)
*/ ***********************************************************************
*/ * SLOE02 : EXPAND DIMENSION OF REAL STORAGE ARRAY                         *
*/ *           FOR    SLAM  CALCULATION                                      *
*/ ***********************************************************************
*D /SPACE/.13
      PARAMETER (JRSPAC=10000)
*/ ***********************************************************************
*/ * SLOE03 : REDUCE REGION WHICH IS NOT USED BY SLAM CALCULATION         *
*/ ***********************************************************************
*D PTRCEL.457
*D PTRCEL.500,515
*I RB-RBCNT.60
      IF (LDUMNC)   THEN
          NMAX  =  4
      ELSE
          NMAX  =  3*NDEFLT(1) + 1
      ENDIF
C
      LTATA  =  LTMTOT + NREG
      LTATF  =  LTATA  + NMAX*NMAX
      LTDIAG =  LTATF  + NMAX
      LTFJAC =  LTDIAG + NMAX
      LTFVEC =  LTFJAC + NMAX*NMAX
      LTXSOL =  LTFVEC + NMAX
      LTWRK  =  LTXSOL + NMAX
      NEND   =  LTWRK  + NMAX * 5
C
      LTHL   =  NEND
      NEND   =  LTHL   + 2*NREG
C
      NTMPR  =  NEND
      LTIWRK =  1
      NEND   =  LTIWRK + NMAX + NMAX
      NTMPI  =  NEND
*/ ***********************************************************************
*/ * SLOF01 :  ADD NEW COMMON VARIABLES TO COMMON BLOCK                    *
*/ ***********************************************************************
*D /REBLAY/.24,26
     2  FAL2O3,FCR2O3,FCO2,FH2OE,FH2OB,FCACO3,FMGCO3,TABLAT,EFUSN,
     3  RHOCON,CPCON,TKA,TKB,AP,BP,RBUB,GAM0,GAM1,GAM2,GAMINF,
     4  DAVDIA,DMNDEV,TFAIL,XVAR1
      PARAMETER (JRREBL=32)
*D /REBCOM/.22,23
     1  ,HTXCON,DMOX,DMNA,DMPER,DMMON,QFM,BOILMS,BOILEN,PBOT
      PARAMETER ( JRCPOL=20 )
*/ ***********************************************************************
*/ * SLOF02 :  INITIAL  SET  OF  SLAM  COMMON  VARIABLES                  *
*/ ***********************************************************************
*I I-IRCTYP.49
C
C
```

```
      RHOCON = 2250.0
      CPCON  = 1000.0
      XVAR1  = 1.000
      TKA    = -1.375E-03
      TKB    = 1.6128
      AP     = 0.3949E-18
      BP     = 0.01416
      RBUB   = 0.004
      GAMO   = 0.07700
      GAM1   = 200.0
      GAM2   = 1.0E-10
      GAMINF = 1.0E-08
*/ **************************************************************************
*/ * SLOG01 :  ADD  MATERIAL PROPERTIES OF SLAM TO CONTAIN                  *
*/ *           FOR  SLAM      CALCULATION                                   *
*/ **************************************************************************
*D C100K.1
     6                LAR,LNA2CO,LZR,LZRO2,LSS,LSSOX,LCACO3,LMGCO3,
     7                LINERT,LCOOL
*D /MATER/.16
      PARAMETER (JIMATR=56)
*D /INPUTW/.8
      PARAMETER (JPWORD=54)
*I I-SETUP.241
      DATA (OPWORD(I,52),I=1,2)/'CACO3',' '/
      DATA (DFALT(I,52),I=1,3),(IDFALT(I2,52),I2=1,3)
     1 /1.0E+4,1.E20,100.,3,0,1/
      DATA (OPWORD(I,53),I=1,2)/'MGCO3',' '/
      DATA (DFALT(I,53),I=1,3),(IDFALT(I2,53),I2=1,3)
     1 /1.0E+4,1.E20,84.,3,0,1/
      DATA (OPWORD(I,54),I=1,2)/'INERT',' '/
      DATA (DFALT(I,54),I=1,3),(IDFALT(I2,54),I2=1,3)
     1 /1.0E+4,1.E20,100.,3,0,1/
*D PR-RHOX.35
     4      41,42,43,44,45, 46,47,48,49,50,51,52,53,54),I
*I PR-RHOX.260
C
C     CACO3
   52 RHOX = 2200.
      RETURN
C
C     MGCO3
   53 RHOX = 1400.
      RETURN
C
C     INERT
   54 RHOX = 2885.
*D PR-SHX.41
     4      41,42,43,44,45, 46,47,48,49,50,51,52,53,54),I
*I PR-SHX.281
C
C     CACO3
   52 SHX  = 1173.
      RETURN
C
C     MGCO3
   53 SHX = 1173.
      RETURN
C
C     INERT
   54 SHX = 600.
      RETURN
*D PR-ENTHX.23
     4      41,42,43,44,45, 46,47,48,49,50,51,52,53,54),I
*I PR-ENTHX.234
C
C     CACO3
   52 GO TO  100
C
C     MGCO3
   53 GO TO 100
C
C     INERT
   54 GO TO 100
*D PR-CONDX.32
     4      41,42,43,44,45, 46,47,48,49,50,51,52,53,54),I
*I PR-CONDX.273
C
```

```
C     CACO3
   52 CONDX = 2.0
      RETURN
C
C     MGCO3
   53 CONDX = 2.
      RETURN
C
C     INERT
   54 CONDX = 2.
      RETURN
*/ **********************************************************************
*/ * SLOG02 :  MODIFICATION OF MASS FRACTION OF CONCRETE COMPONENTS    *
*/ *           EXTENSION OF ARRAY DIMENSION FOR CONCRETE               *
*/ **********************************************************************
*D C100K.117
      REAL BASALT(17),LIME(17),GENRIC(17)
*D I-IRCTYP.37
     1 0.062,0.0832,0.00,0.0150,0.0386,0.020,0.0,0.0,1575.0,5.225E+09/
*D I-IRCTYP.38,39
      DATA LIME/0.1105,7*0.0,
     1 0.0,0.0,0.,0.049,0.0305,0.567,0.243,1575.0,3.46E+09/
*D I-IRCTYP.41
     1 0.012,0.016,0.00004,0.35698,0.0394,0.02,0.0,0.0,1575.0,3.46E+09/
*I I-IRCTYP.68
      FCACO3=BASALT(14)
      FMGCO3=BASALT(15)
*D I-IRCTYP.69
      IF(TABLAT.LT.0.0001) TABLAT=BASALT(16)
*D I-IRCTYP.72
      EL(LCON)=BASALT(17)/RHOX(LCON,TABLAT,PRES)
*I I-IRCTYP.91
      FCACO3=LIME(14)
      FMGCO3=LIME(15)
*D I-IRCTYP.92
      IF(TABLAT.LT.0.0001) TABLAT=LINE(16)
*D I-IRCTYP.95
      EL(LCON)=LIME(17)/RHOX(LCON,TABLAT,PRES)
*I C100K.131
      FCACO3=GENRIC(14)
      FMGCO3=GENRIC(15)
*D C100K.132
      IF(TABLAT.LT.0.0001) TABLAT=GENRIC(16)
*D C100K.133
      EL(LCON)=GENRIC(17)/RHOX(LCON,TABLAT,PRES)
*I I-IRCTYP.137
      FCACO3=VALUE(DMY)
      FMGCO3=VALUE(DMY)
*/ **********************************************************************
*/ * SLOH01 :  ADD  NEW  FLAG  'SLMOUT','WATOUT' TO  COMMON BLOCK       *
*/ **********************************************************************
*D /FLAGS/.49,50
     5      NFZROX,SLMOUT,WATOUT
      PARAMETER (JLFLAC=22)
*D /FLAGS/.53
     3      HTLOLY,HTLAY,HTRAD,NFCHEM,NFENGS,NFSENG,NFSPWO,SLMOUT,WATOUT
*/ **********************************************************************
*/ *  SLOH02 :  CELL FLAG 'LDUMNC' IS 'TRUE' WHEN SLAM MODEL OR WATER   *
*/ *           MIGRATION MODEL IS SELECTED                             *
*/ **********************************************************************
*I I-IRCON.164
      LDUMNC = .TRUE.
*I I-IRCON.172
      CALL  SLPROP
      LDUMNC = .TRUE.
*/ **********************************************************************
*/ * SLOH03 : CELL FLAG 'SLMOUT' IS 'TRUE' WHEN SODIUM CONCRETE         *
*/ *           INTERACTION OCCURS                                      *
*/ **********************************************************************
*I I-IRCON.164
              SLMOUT = .TRUE.
*/ **********************************************************************
*/ * SLOH04 :  CELL FLAG 'WATOUT' IS 'TRUE' WHEN WATER  RELEASE        *
*/ *            OCCURS                                                 *
*/ **********************************************************************
*I I-IRCON.171
              WATOUT = .TRUE.
*/ **********************************************************************
```

```
*/ * SLOI01 :  LINER FAILURE TEMPERATURE 'TFAIL' IS  SET TO DEFAULT    *
*/ ***********************************************************************
*D I-IRINT.59
      TFAIL = 1.0E+10
*/ ***********************************************************************
*/ * SLOI02 :  LINER FAILURE FLAG 'LNFAIL' IS 'TRUE' WHEN LINER        *
*/ *             FAILURE OCCURS                                        *
*/ ***********************************************************************
*D I-IREBEL.46
      LNFAIL = .FALSE.
*D RB-INTER.240,241
      IF ( RITA( KZO( KREG )) .LT. TFAIL ) THEN
         LNFAIL = .FALSE.
      ELSE
         LNFAIL = .TRUE.
*/ ***********************************************************************
*/ * SLOJ01 :  ADD  A  ARGUMENT  TO  SUBROUTINE                         *
*/ ***********************************************************************
*D I-IREBEL.144
     4        OH(INAMES+1),OTYP,IH(LRKZO+1),IH(LRNNP+1),NM,H(LRITAO+1))
*D I-IREBEL.181
     4 OH(INAMES+1),IH(LRKZO+1),IH(LRNNP+1),NM,H(LTRANT+1),H(LRPTAO+1))
*D I-IRPOO.4
     2      OTBNY,OTBNM,ICTBFL,ICTBNP,ONAMES,KZO,NNP,NMAT,TRNT,PTAO)
*D I-IRPOO.7
      REAL PA(*),PDZ(*),PMA(NMAT,*),PT(*),PTA(*),PTAO(*),EL(*),EPH(*)
*D I-IRINT.5
     3                      ONAMES,OTYP,KZO,NNP,NMAT,RITAO)
*D I-IRINT.9
      REAL  TBCL(*),TBCU(*),CTBX(*),CTBY(*),RITAO(*)
*/ ***********************************************************************
*/ * SLOJ02 :  SAVE  POOL  AVERAGE TEMP FOR CURRENT TIME STEP          *
*/ ***********************************************************************
*I I-IRPOO.63
      PTAO(KZ) = PTA(KZ)
*I I-IRPOO.107
      PTAO(KZ) = PTA(KZ)
*/ ***********************************************************************
*/ * SLOJ03 :  SAVE  INTERM AVERAGE TEMP FOR CURRENT TIME STEP         *
*/ ***********************************************************************
*I I-IRINT.64
      RITAO(KZ)=RITA(KZ)
*I I-IRINT.107
      RITAO(KZ)=RITA(KZ)
*/ ***********************************************************************
*/ * SLOK01 :  ADD  SLAM  OUTPUT PRINT                                  *
*/ ***********************************************************************
*I,O-RBOUT.341
C          ***************************
C                SLAM OUTPUT PRINTS
C          ***************************
C
      IF( SLMOUT .OR. WATOUT)  THEN
         CALL SLOUT
      ELSE
C
C     *** END OF "IF( SLMOUT .OR. WATOUT)  THEN" ***
      ENDIF
*/ ***********************************************************************
*/ * SLOK02 :  WRITE SLAM CALCULATIONAL RESULTS TO PLOT FILE           *
*/ ***********************************************************************
*D PL-CPLOT.155
*I PL-CPLOT.222
  400 CONTINUE
*I PL-CPLOT.9
*CALL,/SLCOM1/
*CALL,/SLCOM2/
*I PL-CPLOT.129
      OFLAG= '400'
      WRITE(N12) OFLAG,OBLANL,NCELL,SS
      WRITE(N12) OBLANK,OBLANK,JCONC,JINT,JPOOL,JATM,NREG
*I PL-CPLOT.180
      IF (NFNACO.OR.NFCONC)  THEN
      TDRI = TDRY(NDRY/2+1)
      OFLAG = '335'
      WRITE(N11) OFLAG,OBLANK,NCELL,SS
      WRITE(N11) OBLANK,OBLANK,TDRI,TS,QCV,QCONO,QCON,DIST,VEL,DEL1,
     1    DDL1DT,QVSUM,QDEL1,PS1,H2OFLX,H2OSRC,WFLXDT,CO2SRC,XNTCO2
```

```
      ENDIF
C
*I PL-CPLOT.220
      IF (NFNACO.OR.NFCONC)  THEN
      OFLAG = '335'
      WRITE(NTM) OFLAG,OBLANK,NCELL,SS
      WRITE(NTM) OBLANK,OBLANK,TDRI,TS,QCV,QCONO,QCON,DIST,VEL,DEL1,
     1    DDL1DT,QVSUM,QDEL1,PS1,H2OFLX,H2OSRC,WFLXDT,CO2SRC,XNTCO2
      ENDIF
C
*/ **********************************************************************
*/ * SLOK03 :  READ  CONCRETE STORAGE ARRAY FROM RESTART FILE            *
*/ **********************************************************************
*I I-RESTAR.17
*CALL,/NACON/
*CALL,/SLCOM1/
*CALL,/SLCOM2/
*I C100H.210
      DO  80  NCELL = 1,NCELLS
      LCELP = LECS + 12*(NCELL-1)
      KREG = 1
      CALL  NXTCEL ( LCELP )
      IF ( .NOT. LDUMNC ) GO TO 80
      READ (NDSK,END=85) OBLANK,OBLANK,
     1     ( CH(II),II=1,9000),(ICH(II),II=1,200)
      CALL  LOAD ('REBLAY')
      CALL  TRANA
      CALL  SLPROP
      CALL  SLCHEM
      GO TO 85
80    CONTINUE
85    CONTINUE
*/ **********************************************************************
*/ * SLOK04 :  WRITE RESTART FILE FROM CONCRETE STORAGE ARRAY            *
*/ **********************************************************************
*I O-RESOUT.19
*CALL,/NACON/
      DATA OROUT / 'RESOUT' /
*I O-RESOUT.111
      DO  81  NCELL = 1,NCELLS
      LCELP = LECS + 12*(NCELL-1)
      CALL  NXTCEL ( LCELP )
      IF ( .NOT. LDUMNC ) GO TO 81
      WRITE (NDSK) OBLANK,OBLANK,
     1     ( CH(II),II=1,9000),(ICH(II),II=1,200)
      GO TO 82
81    CONTINUE
82    CONTINUE
```

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
SiO2	0.1105
TiO2	0.0
MnO	0.0
MgO	0.0
CaO	0.0
Na2O	0.0
K2O	0.0
Fe2O3	0.0
Al2O3	0.0
Cr2O3	0.0
CO2	0.0
H2O(E)	0.049
H2O(B)	0.0305
CaCO3	0.567
MgCO3	0.243

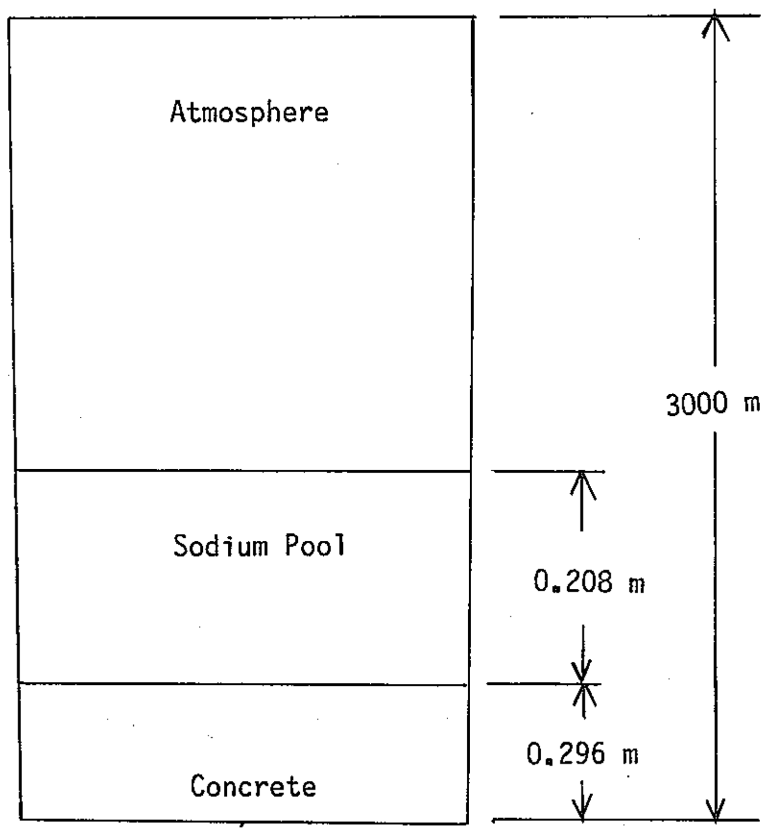


Figure 9.1 System for Sodium-Concrete Interaction Test Calculation

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
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=SIO2  2=H2O  3=NA  4=H2  5=NAOH  6=NA2SIO3 7=NA2CO3 8=NA2O
     9=CAO  10=CACO3 11=CO2 12=C  13=MGCO3 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 1.80000E 03 0.3
 H2O # COEFR IPHASR COEFP IPHASP
     2 1.00000E 00      3
 NA
     3 1.00000E 00      2
 NAOH # COEFR IPHASR COEFP IPHASP
     5 0.00000E 00      0 1.00000E 00      2
 H2
     4 0.00000E 00      0 0.50000E 00      3
 REACTION # 2 #SPECIES AND AK  (CO2+4NA--2NA2O+C)
     4 1.00000E+05 1.80000E 03 0.3
 CO2
    11 1.00000E 00      3
 NA
     3 4.00000E 00      2
 NA2O
     8 0.00000E 00      0 2.00000E 00      1
 C
    12 0.00000E 00      0 1.00000E 00      1
 REACTION # 3   4NA + 3CACO3 ----  2NACO3 + 3CAO + C
     5 8.85000E+12 60000.0E 00 0.10000E 00 0.10000E 00 0.60000E 00
 CACO3
    10 3.00000E 00      1
 NA
     3 4.00000E 00      2
 NACO3
     7 0.00000E 00      0 2.00000E 00      2
 CAO
     9 0.00000E 00      0 3.00000E 00      1
 C
    12 0.00000E 00      0 1.00000E 00      1
 REACTION # 4 4NA + 3MGCO3 ---- 2NACO3 + 3MGO + C
     5 8.85000E+12 60000.0E 00 0.10000E 00 0.10000E 00 0.60000E 00
 MGCO3
    13 3.00000E 00      1
 NA
     3 4.00000E 00      2
 NACO3
     7 0.00000E 00      0 2.00000E 00      2
 MGO
    14 0.00000E 00      0 3.00000E 00      1
 C
    12 0.00000E 00      0 1.00000E 00      1
 REACTION #5    CACO3+2NAOH----CAO+H2O+NA2CO3
     5 40.0000E+05 34000.0E 00 0.1
 CACO3
    10 1.00000E 00      1
 NAOH
     5 2.00000E 00      2
 CAO
     9 0.00000E 00      0 1.00000E 00      1
 H2O
     2 0.00000E 00      0 1.00000E 00      3
 NA2CO3
     7 0.00000E 00      0 1.00000E 00      2
 REACTION #6 2NAOH + SIO2 ----- NA2SIO3 + H2O
     4 40.0000E+05 34000.0E 00 0.1
 NAOH
     5 2.00000E 00      2
 SIO2
     1 1.00000E 00      1
 NA2SIO3
     6 0.00000E 00      0 1.00000E 00      1
 H2O
     2 0.00000E 00      0 1.00000E 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          XMOLNA       TPOOL
3.00000E 00 1.00000E 05 298.000E 00 298.000E 00 0.80000E 00 873.000E 00
  FMGCO       FCACO        FSIO2        RHOCON       CPCON       EPSRAD
0.24300E 00 0.56700E 00 0.11050E 00 2250.00E 00 1000.00E 00 0.50000E 00
   DELM       POWER        DEPTH         TSET        TWFIN        DTMAX
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
 DTPLOT        TC(POOL)    TC1           TC2          TC3          TC4
30.0000E 00-1.00000E 00 0.01000E 00 0.02540E 00 0.07600E 00 0.15000E 00
  MULTI        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
&& -------------------------- SCI004----------------------------
&& CONTAIN TEST SCI004 -  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   MGO  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 SCI004 ( 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.

SLAM-1 < SLAM-2 < CONTAIN-2 < 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.

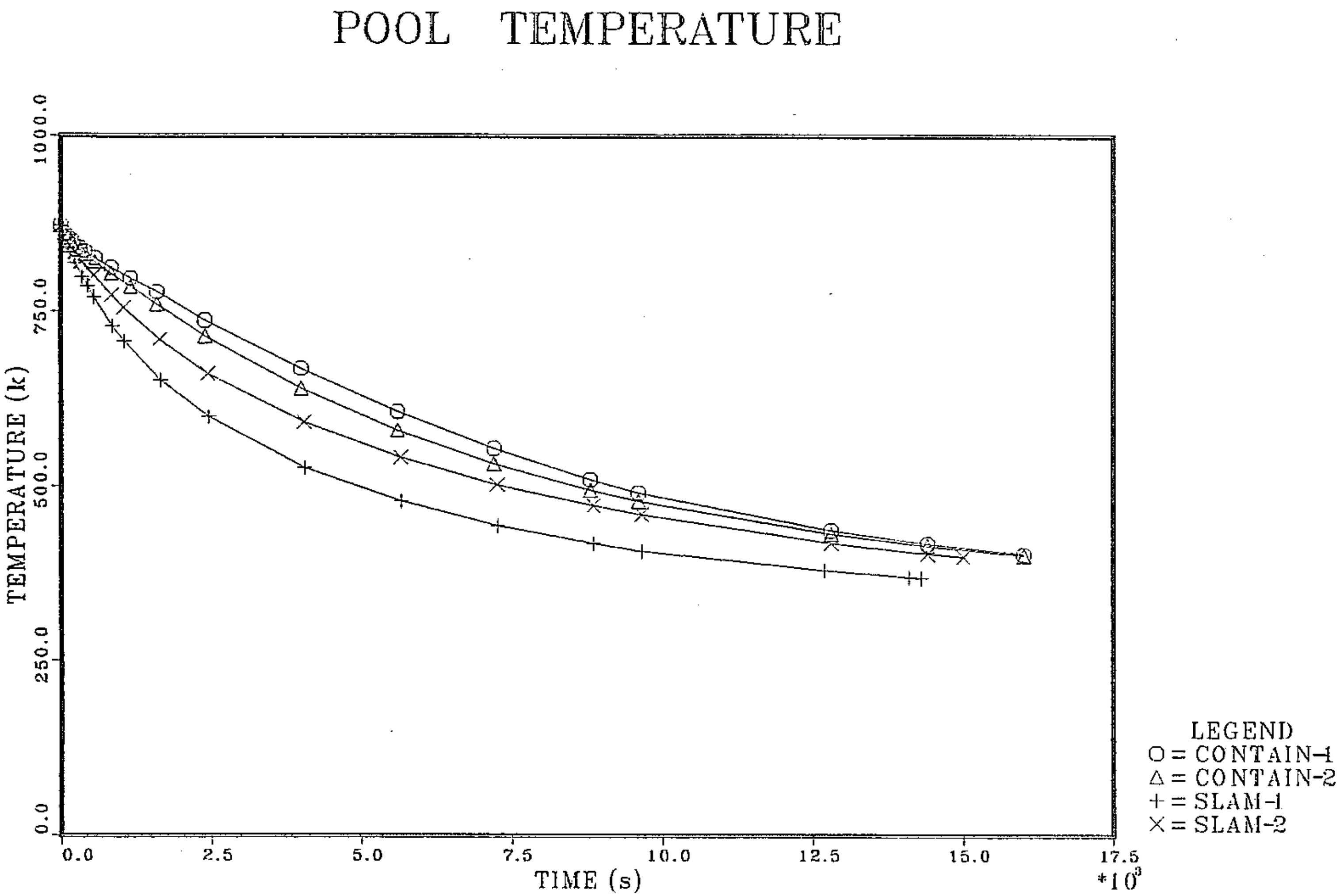


Figure 9.2 Comparison of Original SLAM and CONTAIN Calculations of Pool Temperature during Sodium-Concrete Reaction

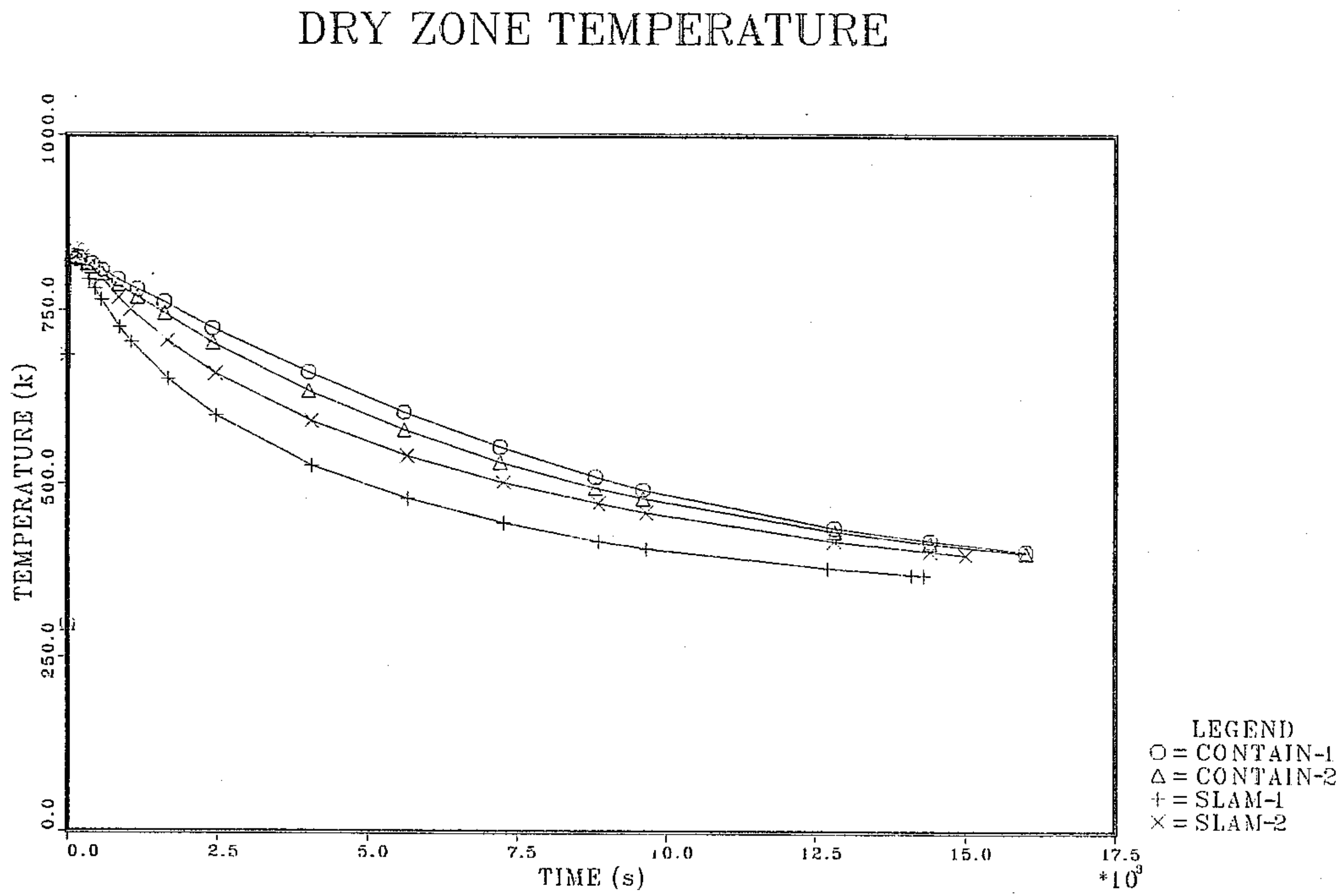


Figure 9.3 Comparison of Original SLAM and CONTAIN Calculations of Concrete DRY Zone Temperature during Sodium-Concrete Reaction

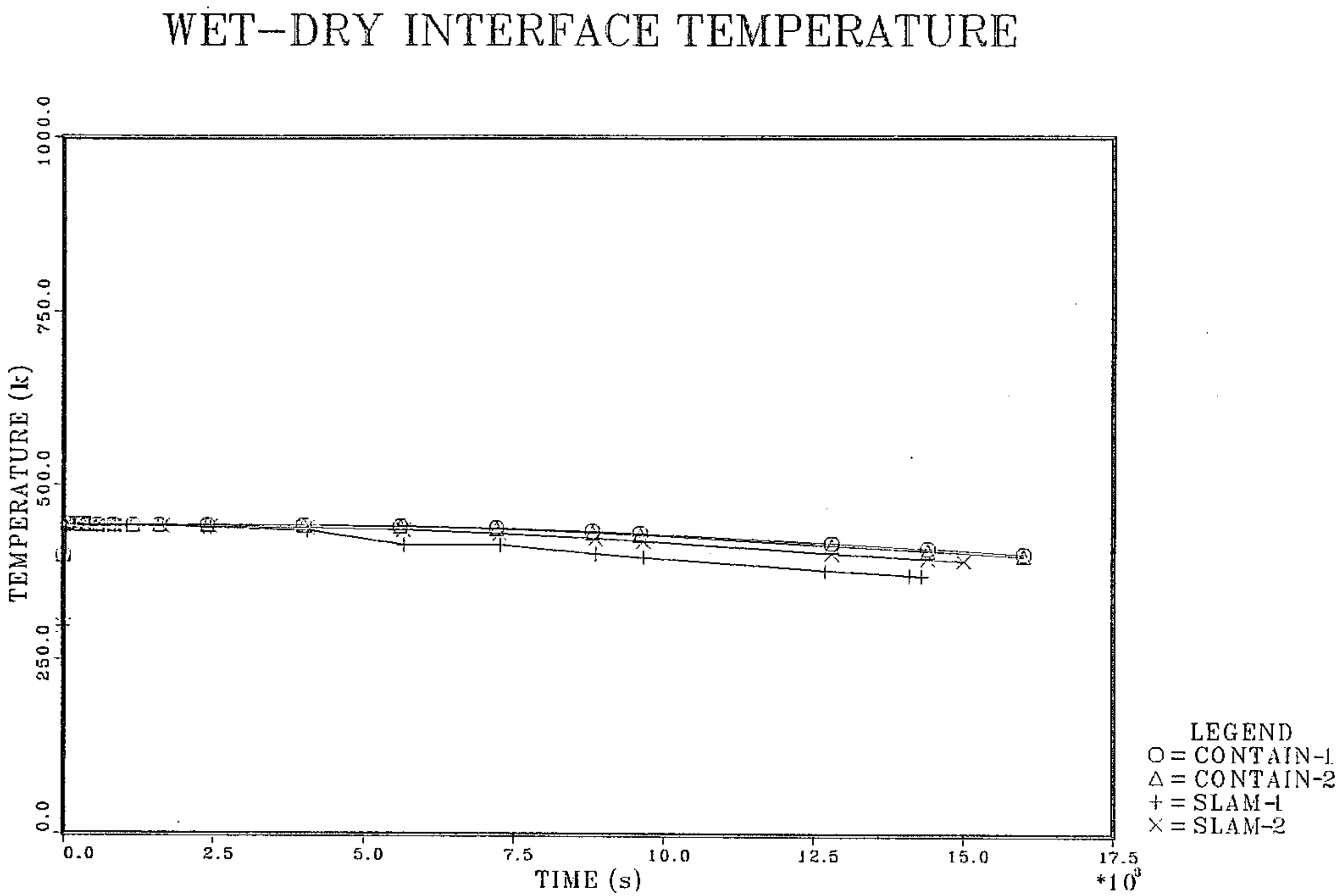


Figure 9.4 Comparison of Original SLAM and CONTAIN Calculations of Wet-Dry Interface Temperature during Sodium-Concrete Reaction

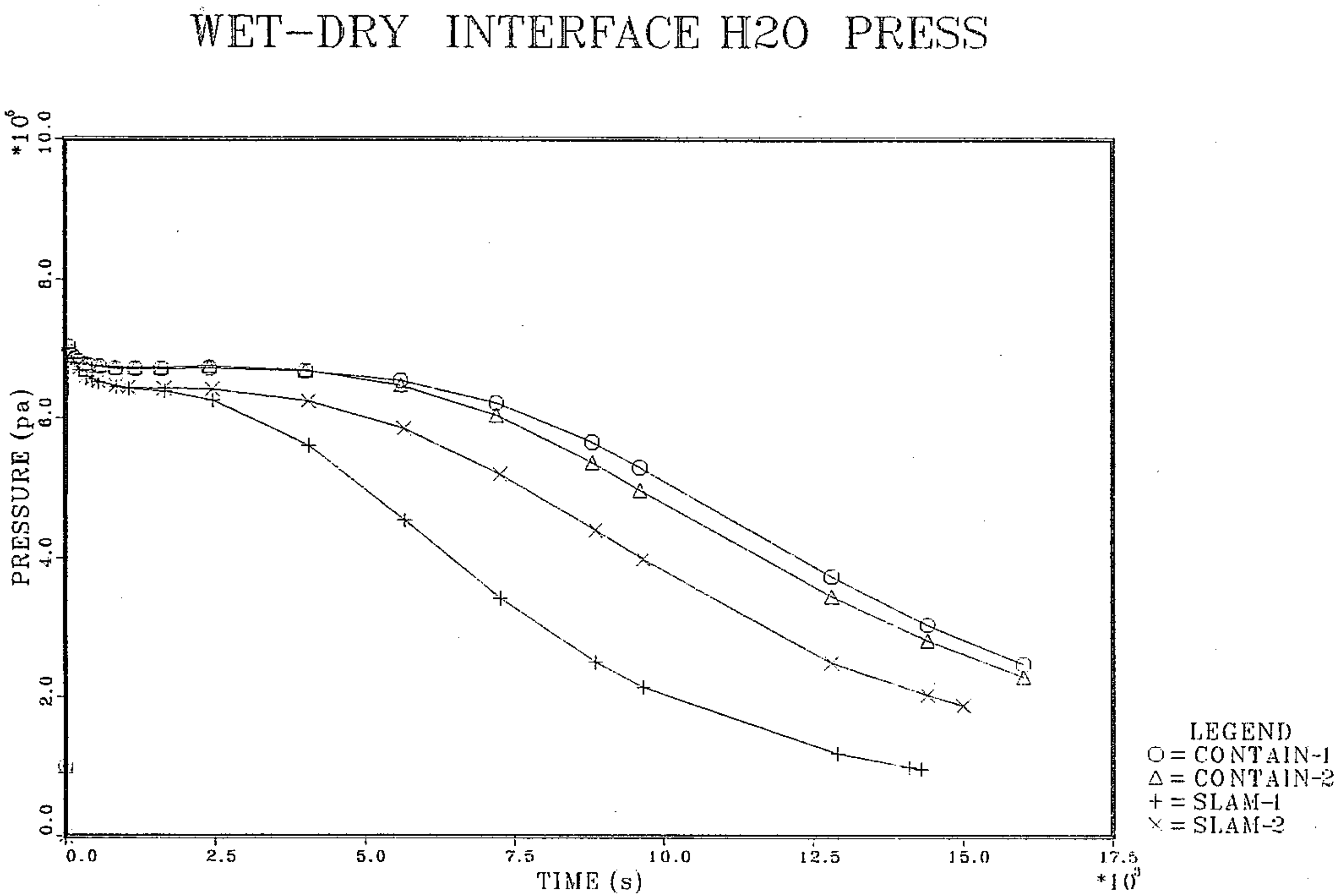


Figure 9.5 Comparison of Original SLAM and CONTAIN Calculations of Wet-Dry Interface H2O Pressure during Sodium-Concrete Reaction

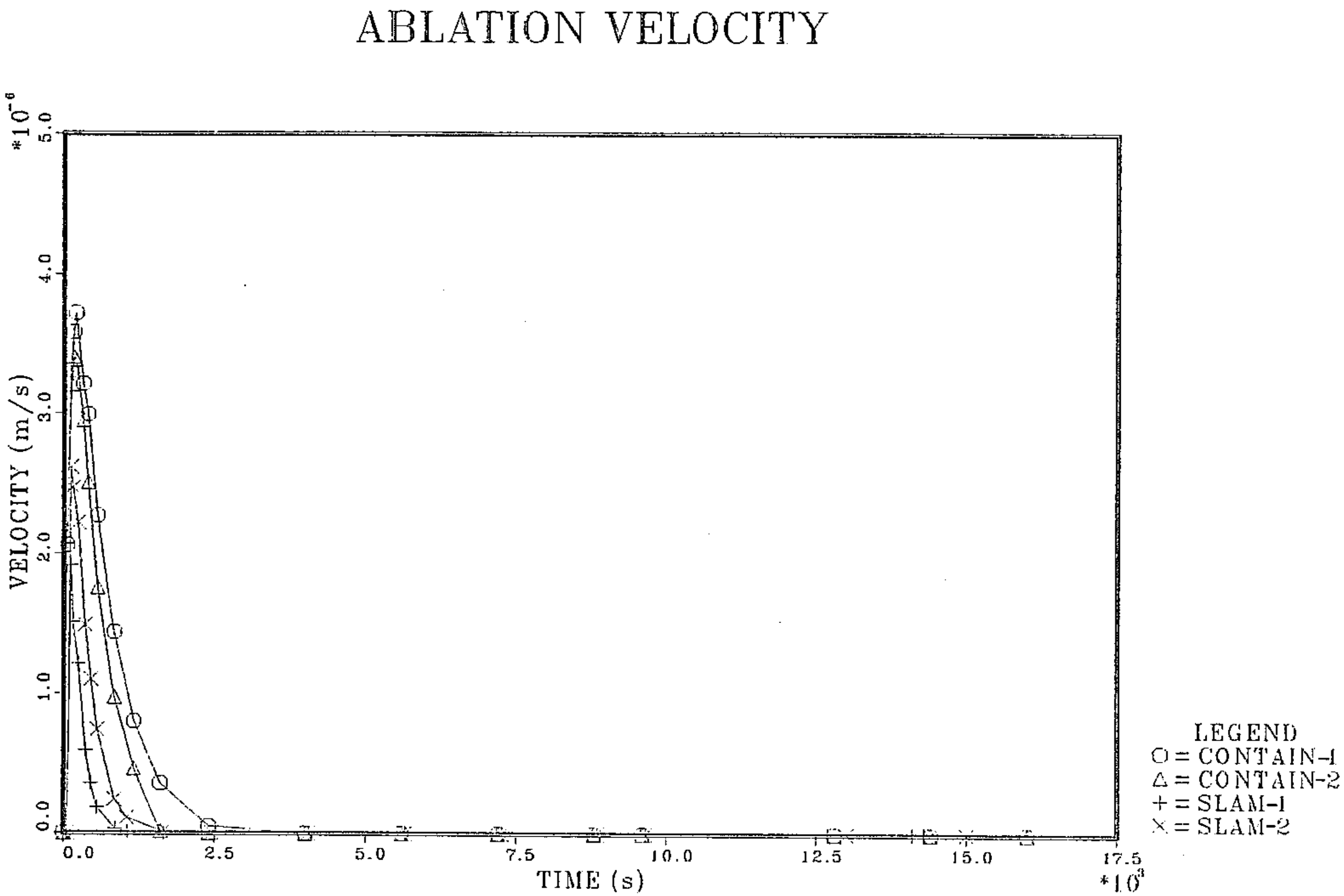


Figure 9.6 Comparison of Original SLAM and CONTAIN Calculations of Ablation Velocity during Sodium-Concrete Reaction

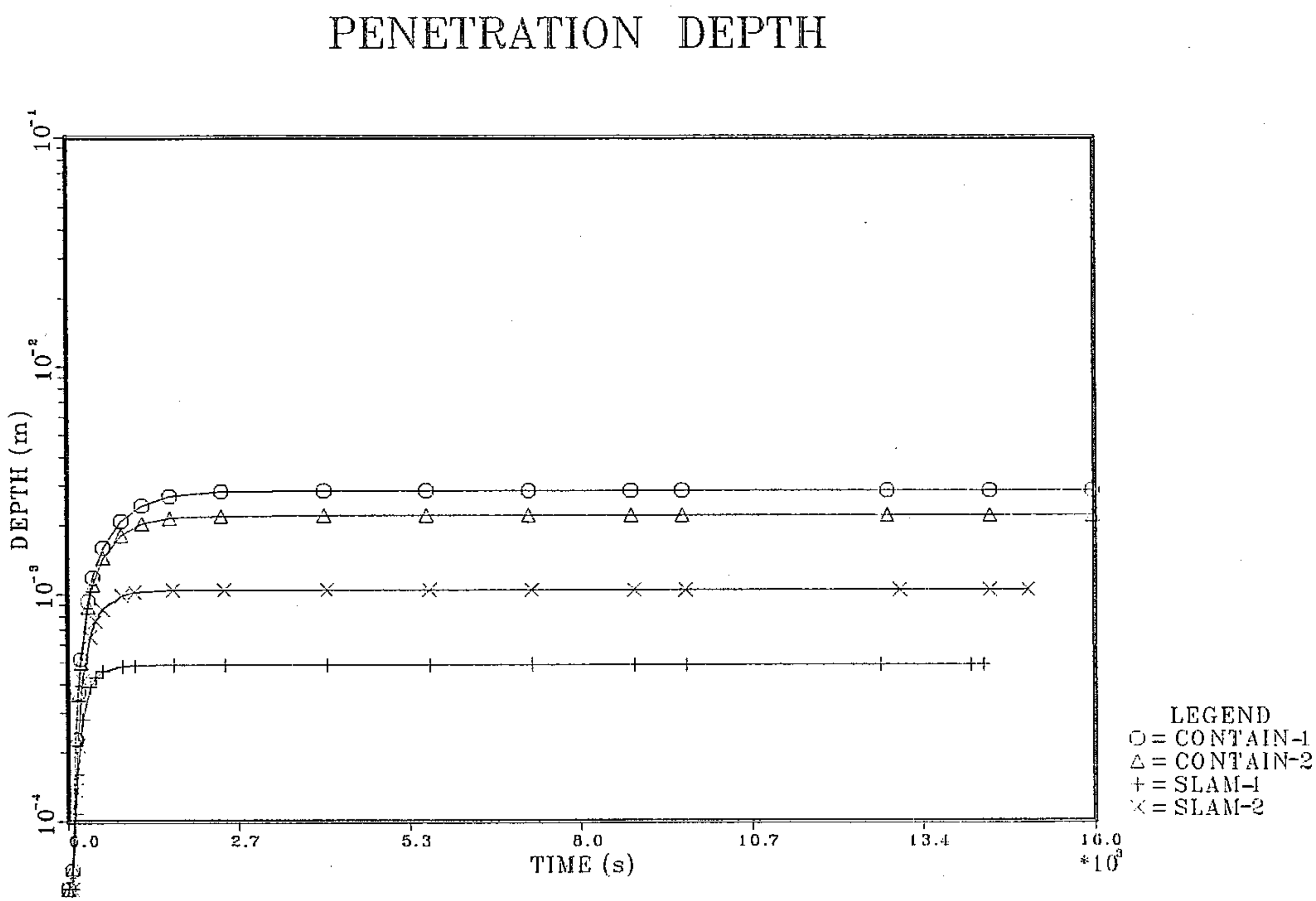


Figure 9.7 Comparison of Original SLAM and CONTAIN Calculations of Penetration Depth during Sodium-Concrete Reaction

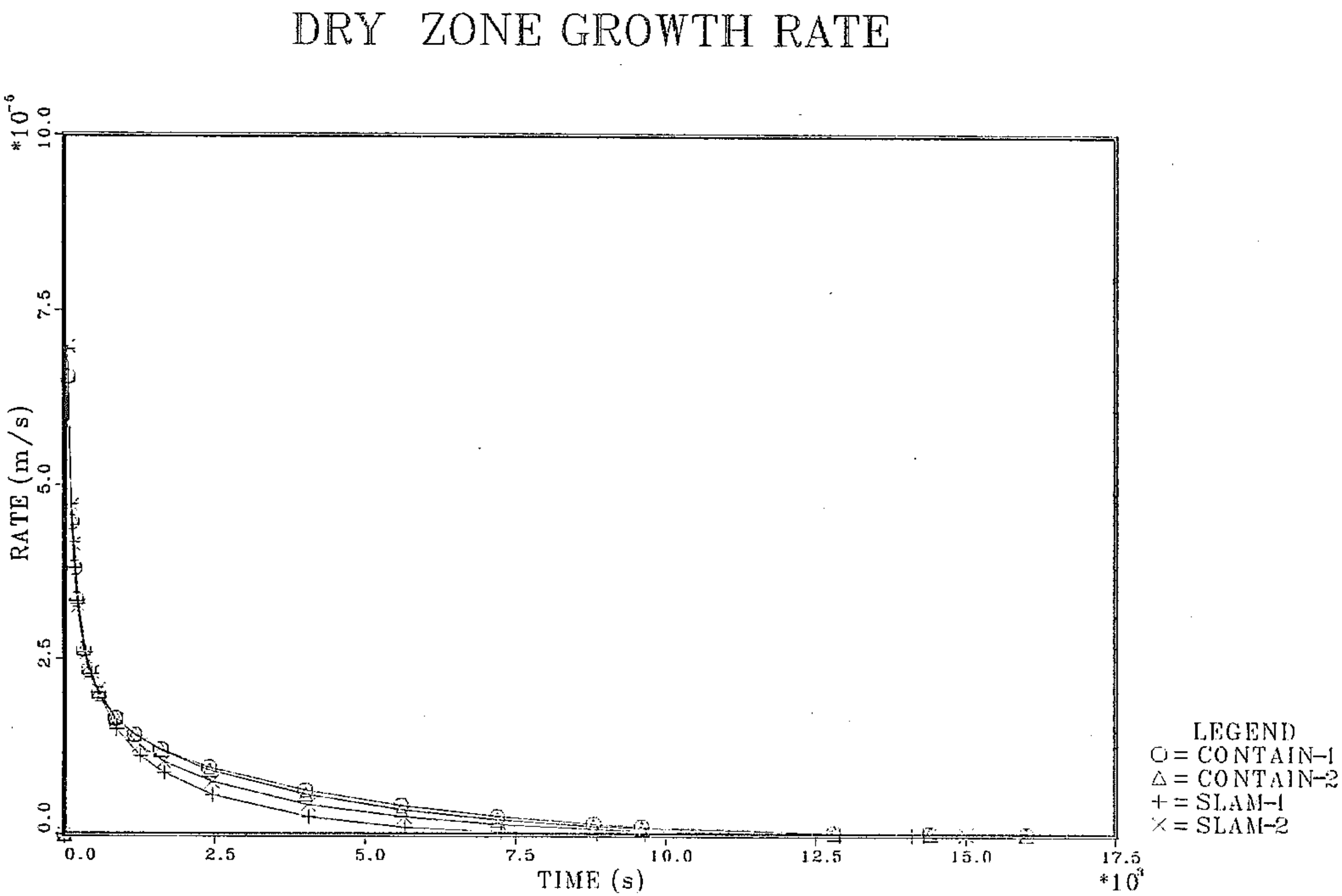


Figure 9.8 Comparison of Original SLAM and CONTAIN Calculations of Dry Zone Growth Rate during Sodium-Concrete Reaction

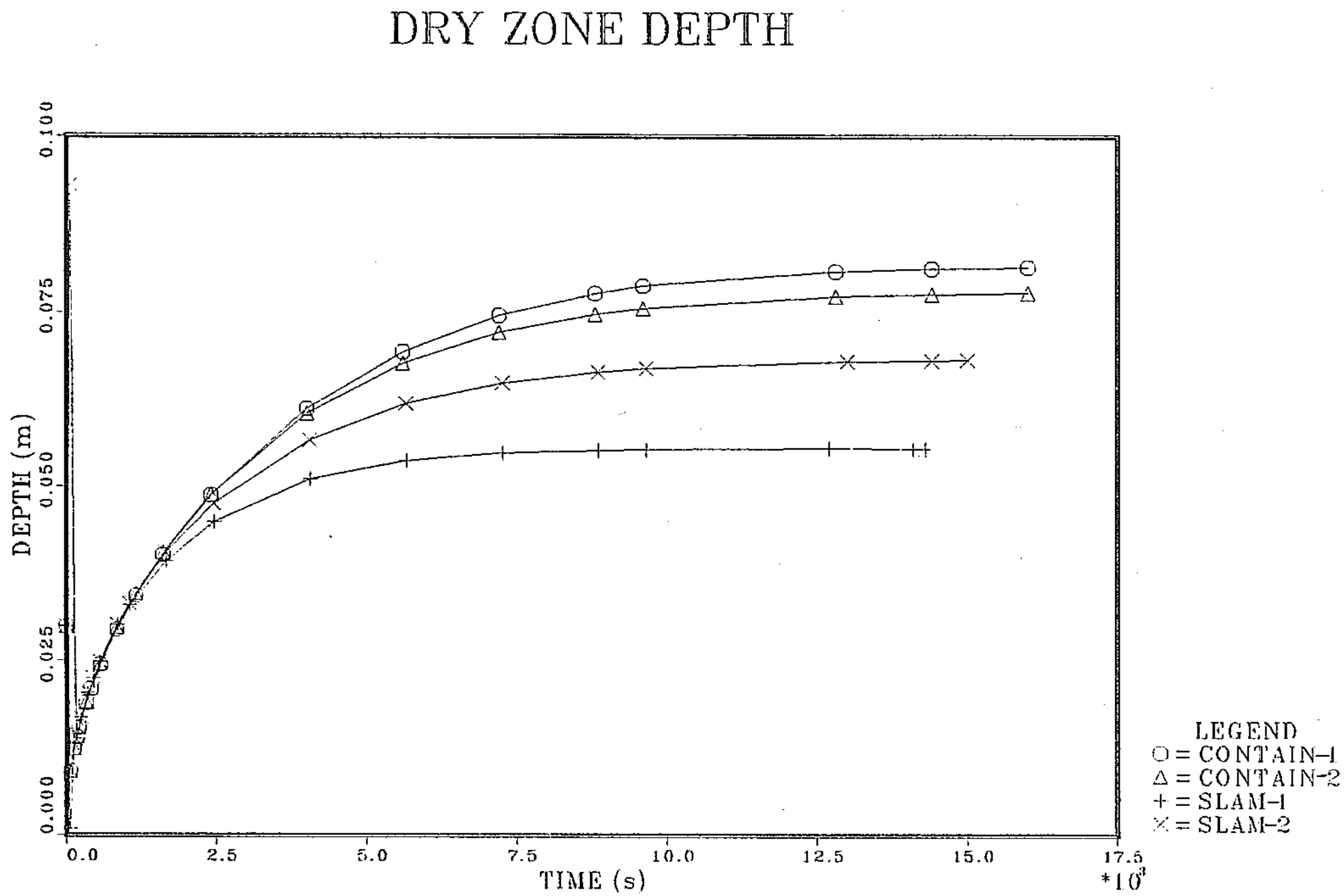


Figure 9.9 Comparison of Original SLAM and CONTAIN Calculations of Dry Zone Depth during Sodium-Concrete Reaction

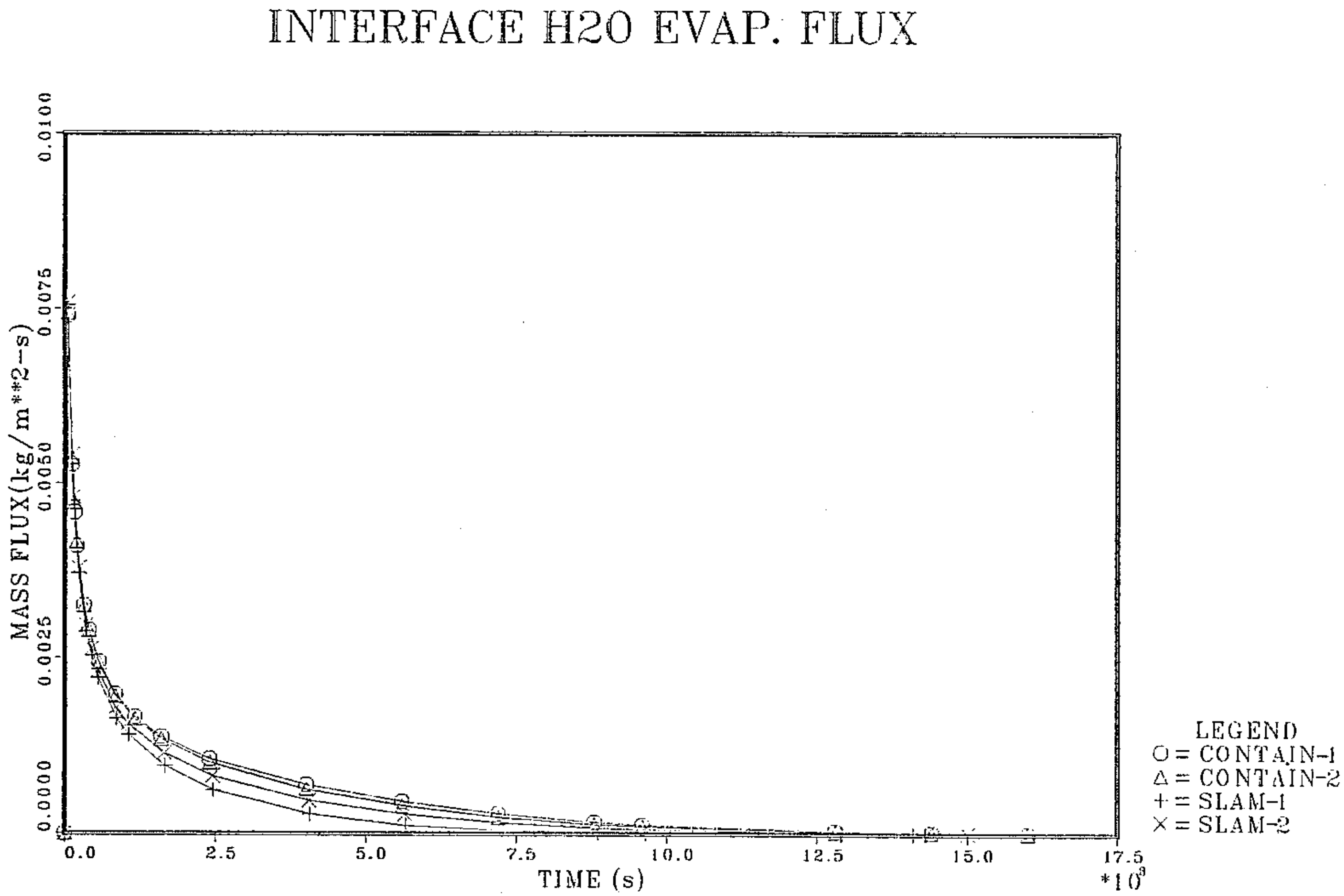


Figure 9.10 Comparison of Original SLAM and CONTAIN Calculations of Interface H2O Evaporated Flux during Sodium-Concrete Reaction

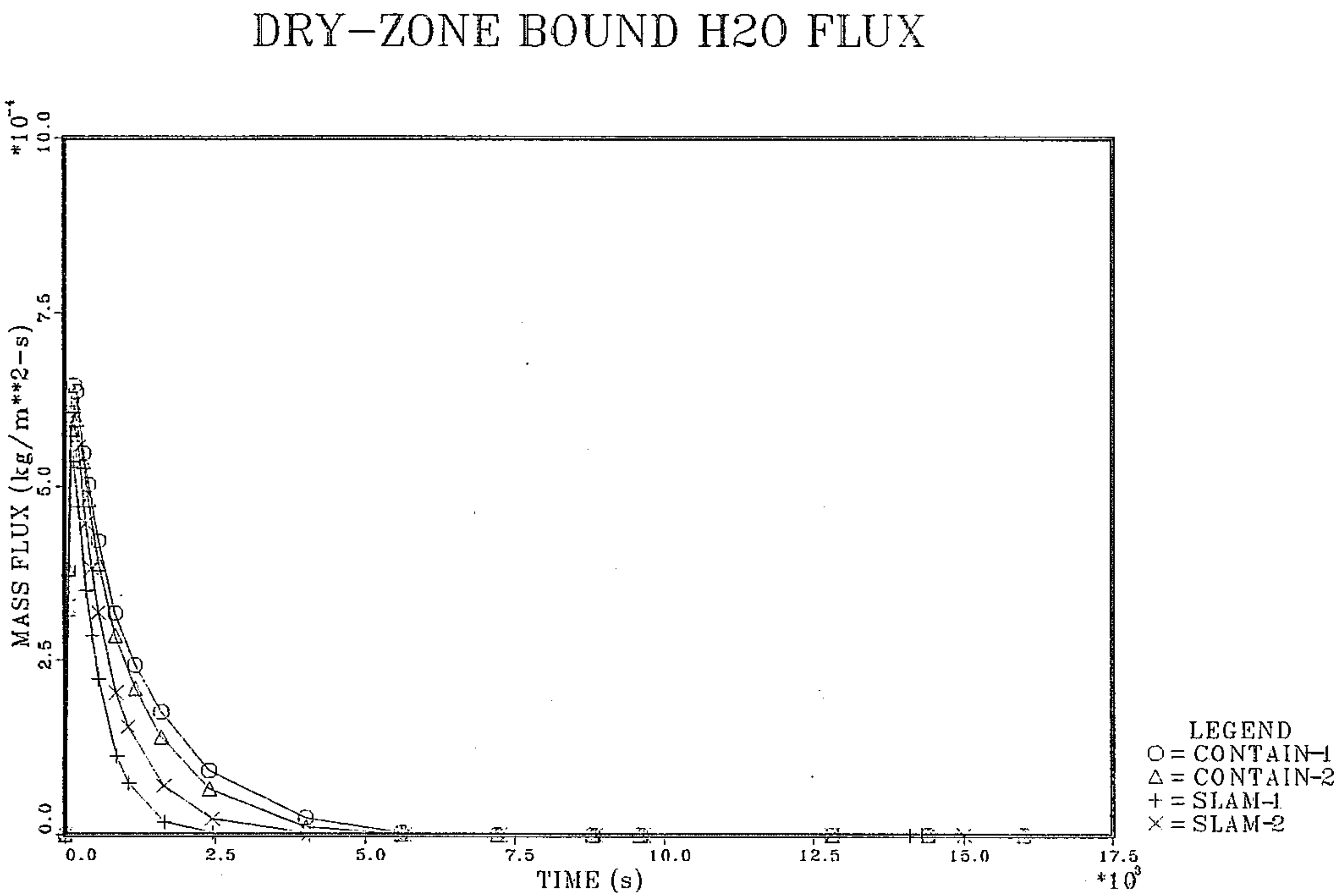


Figure 9.11 Comparison of Original SLAM and CONTAIN Calculations of Dry Zone Bound H2O Flux during Sodium-Concrete Reaction

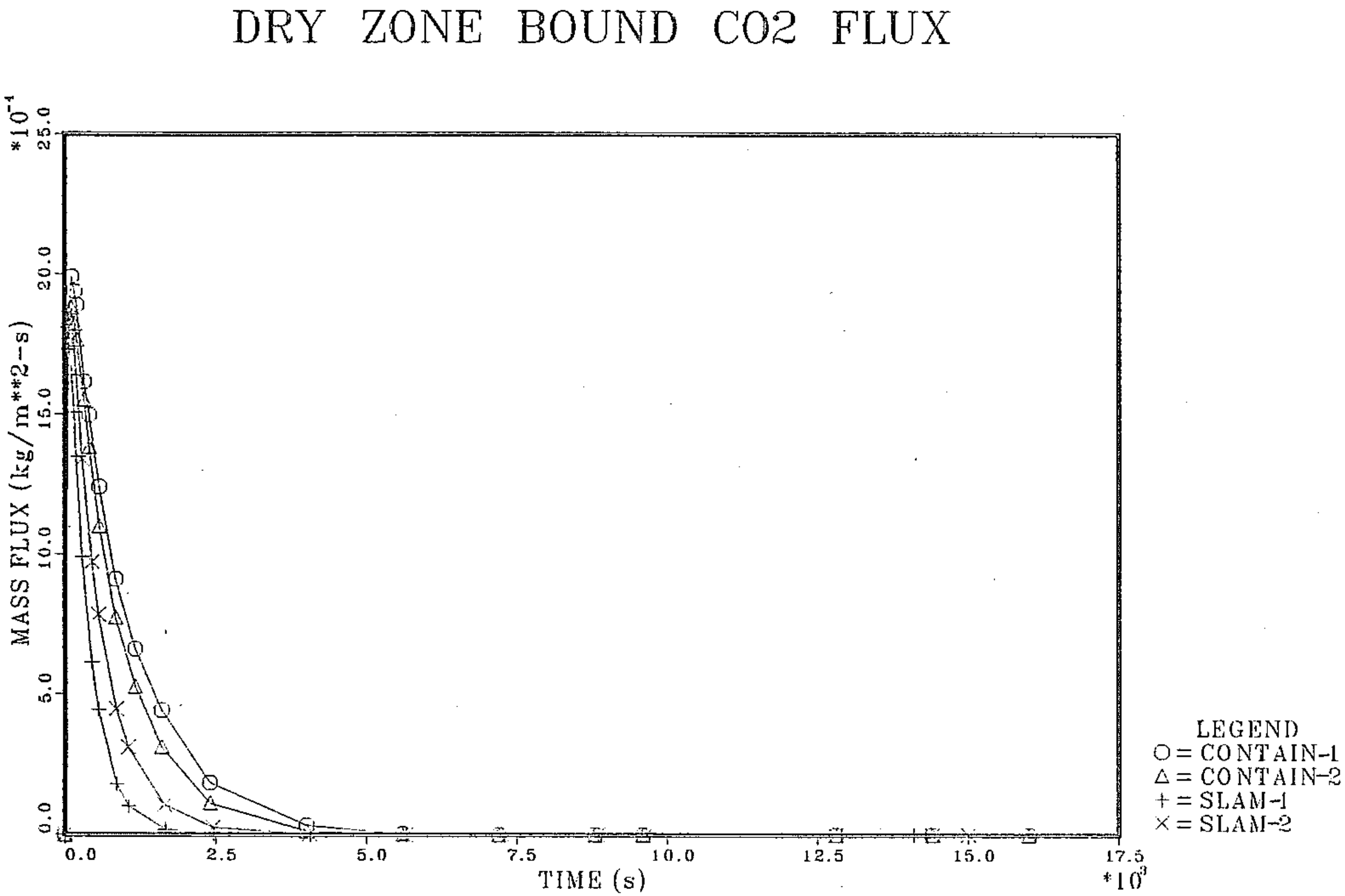


Figure 9.12 Comparison of Original SLAM and CONTAIN Calculation of Dry Zone Bound CO2 Flux during Sodium-Concrete Reaction

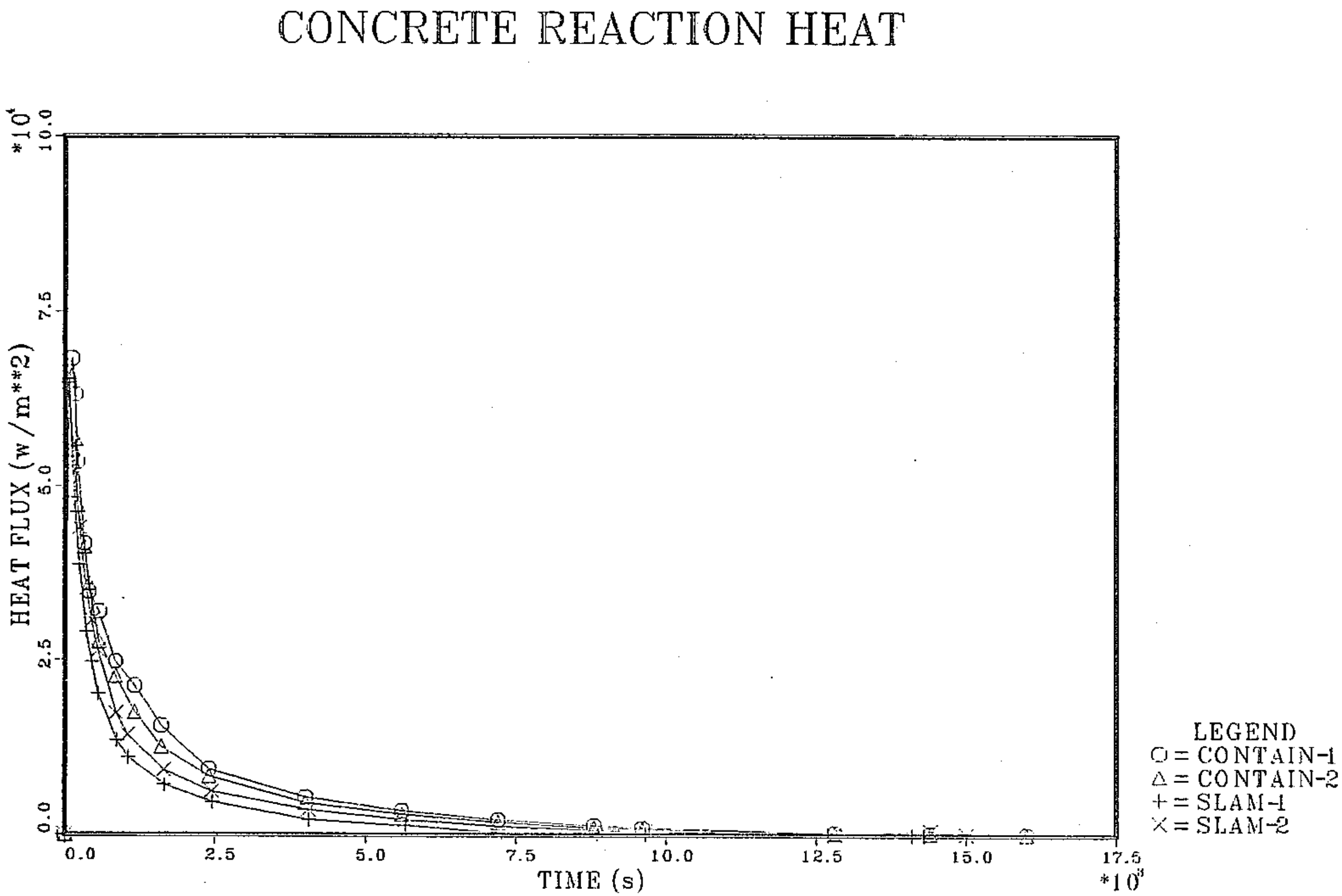


Figure 9.13 Comparison of Original SLAM and CONTAIN Calculations of Concrete Reaction Heat during Sodium-Concrete Reaction

CONCRETE SURFACE HEAT FLUX

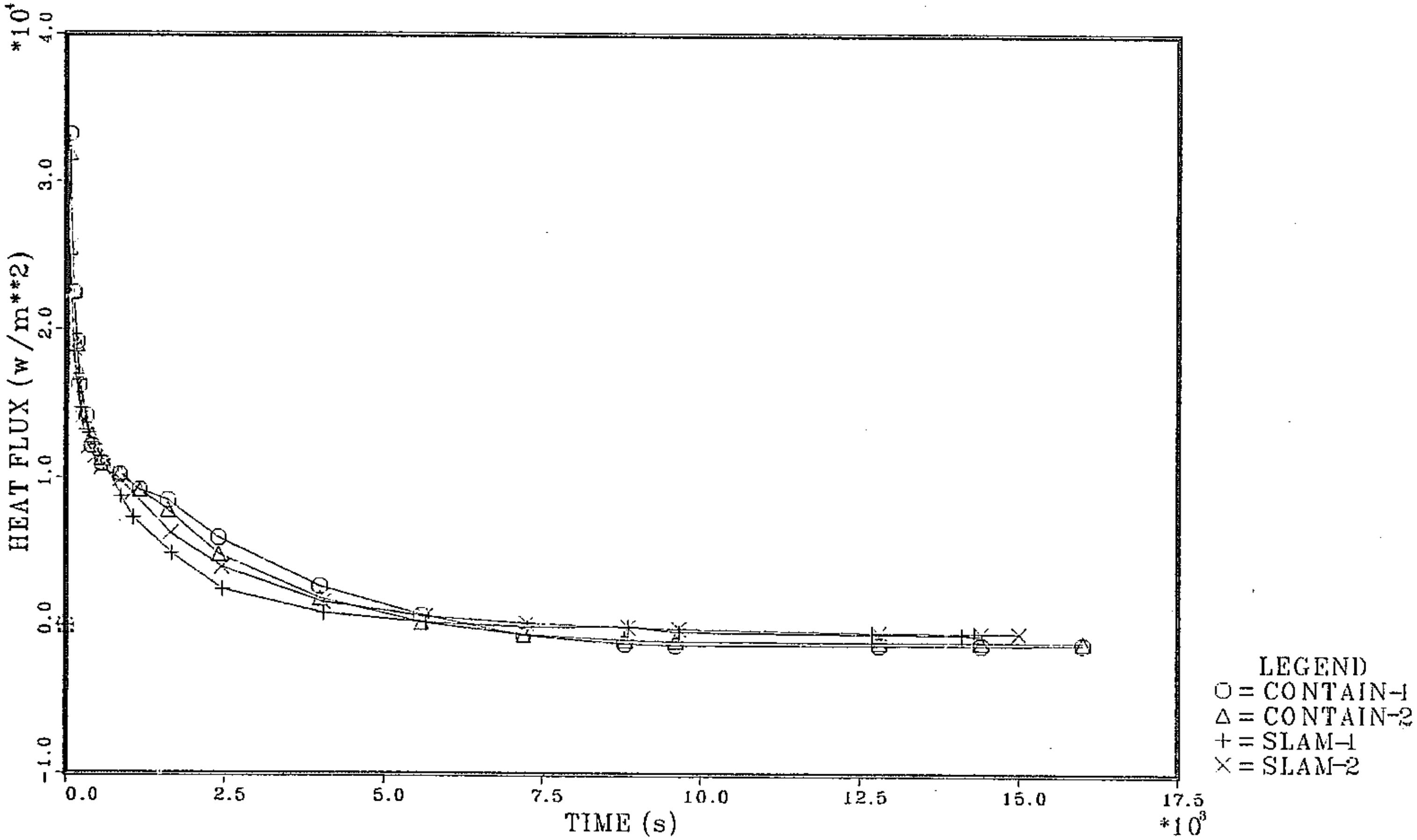


Figure 9.14 Comparison of Original SLAM and CONTAIN Calculation of Concrete Surface Heat Flux during Sodium-Concrete Reaction

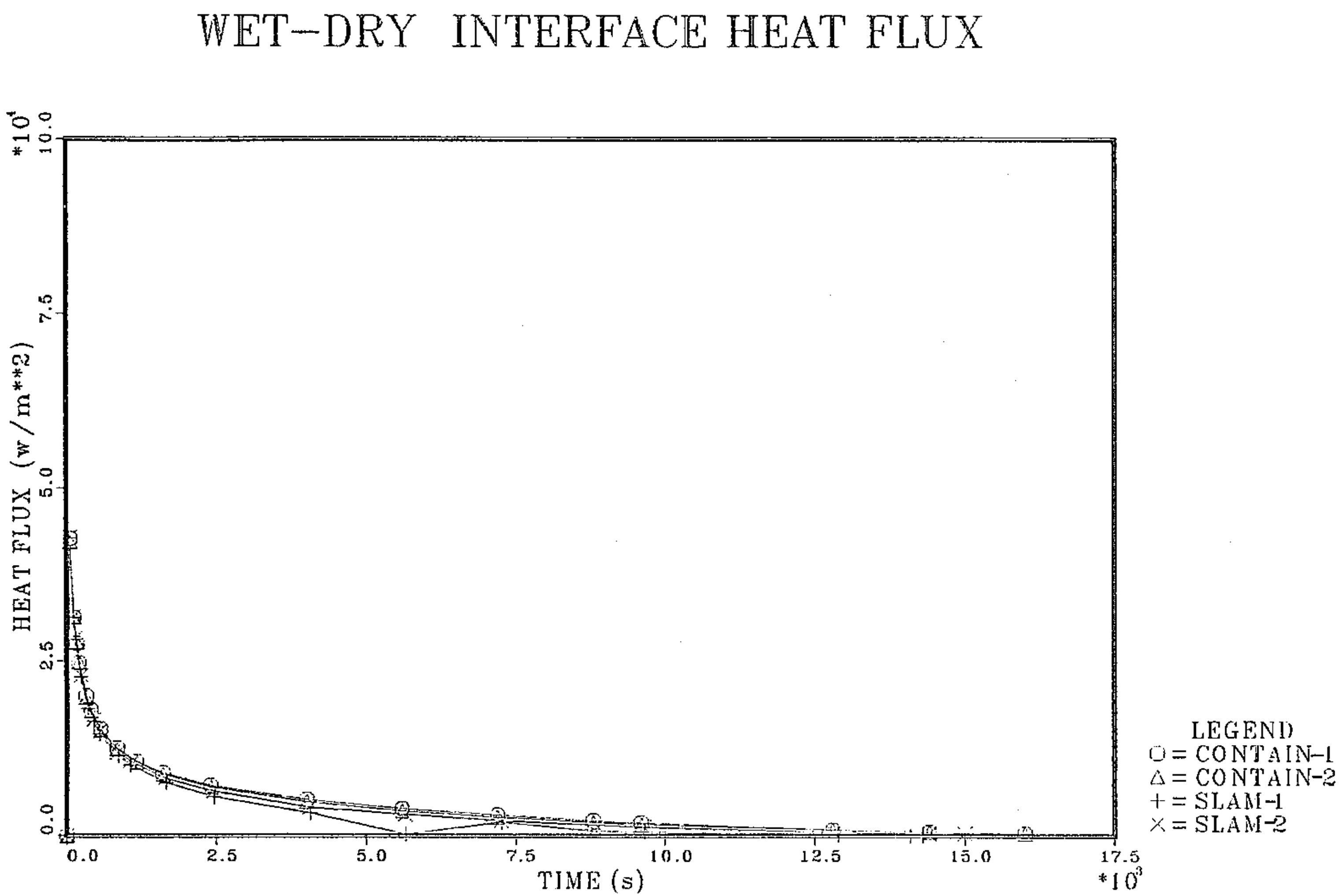


Figure 9.15 Comparison of Original SLAM and CONTAIN Calculations of Wet-Dry Interface Heat Flux during Sodium-Concrete Reaction

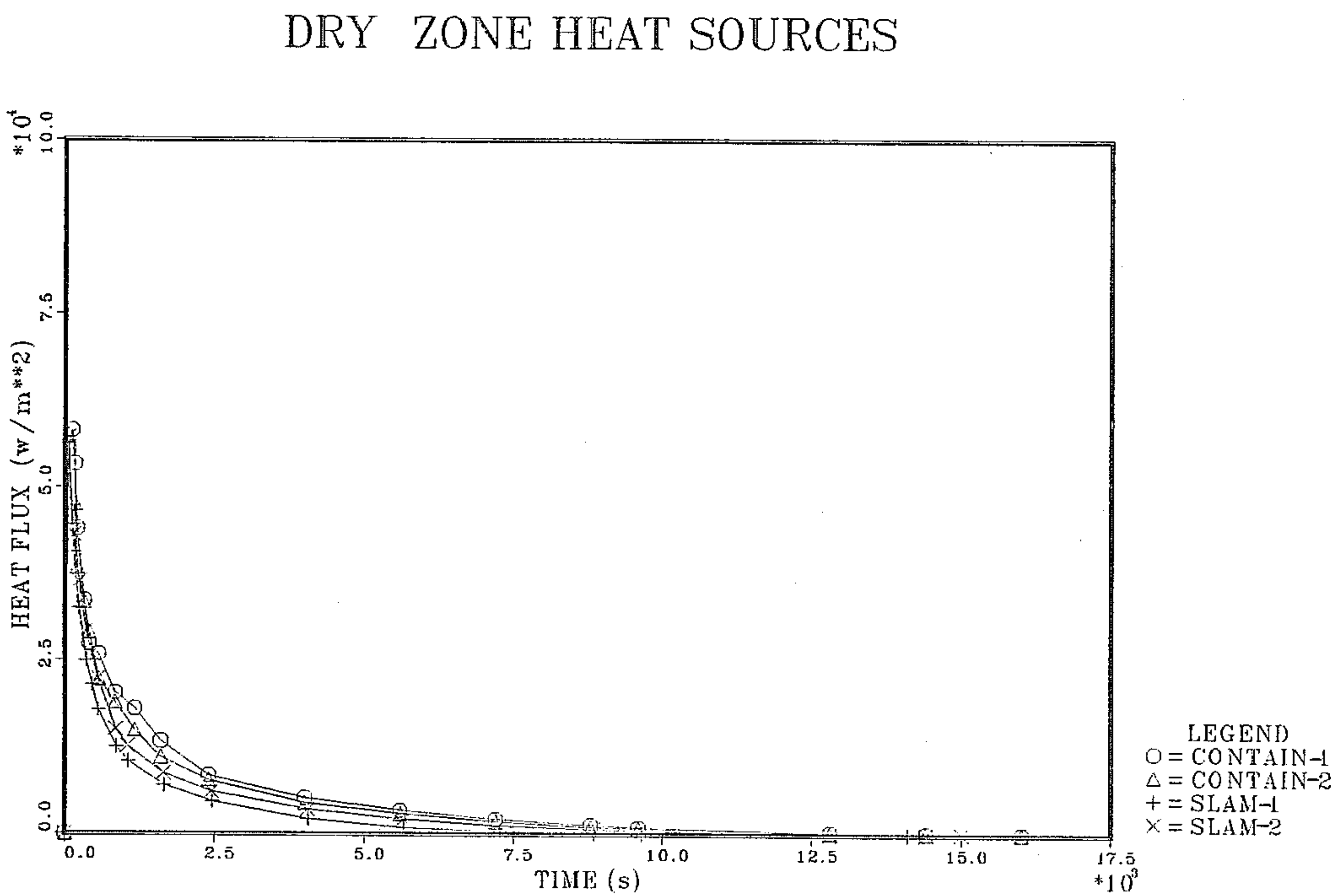


Figure 9.16 Comparison of Original SLAM and CONTAIN Calculations of Dry Zone Heat Sources during Sodium-Concrete Reaction

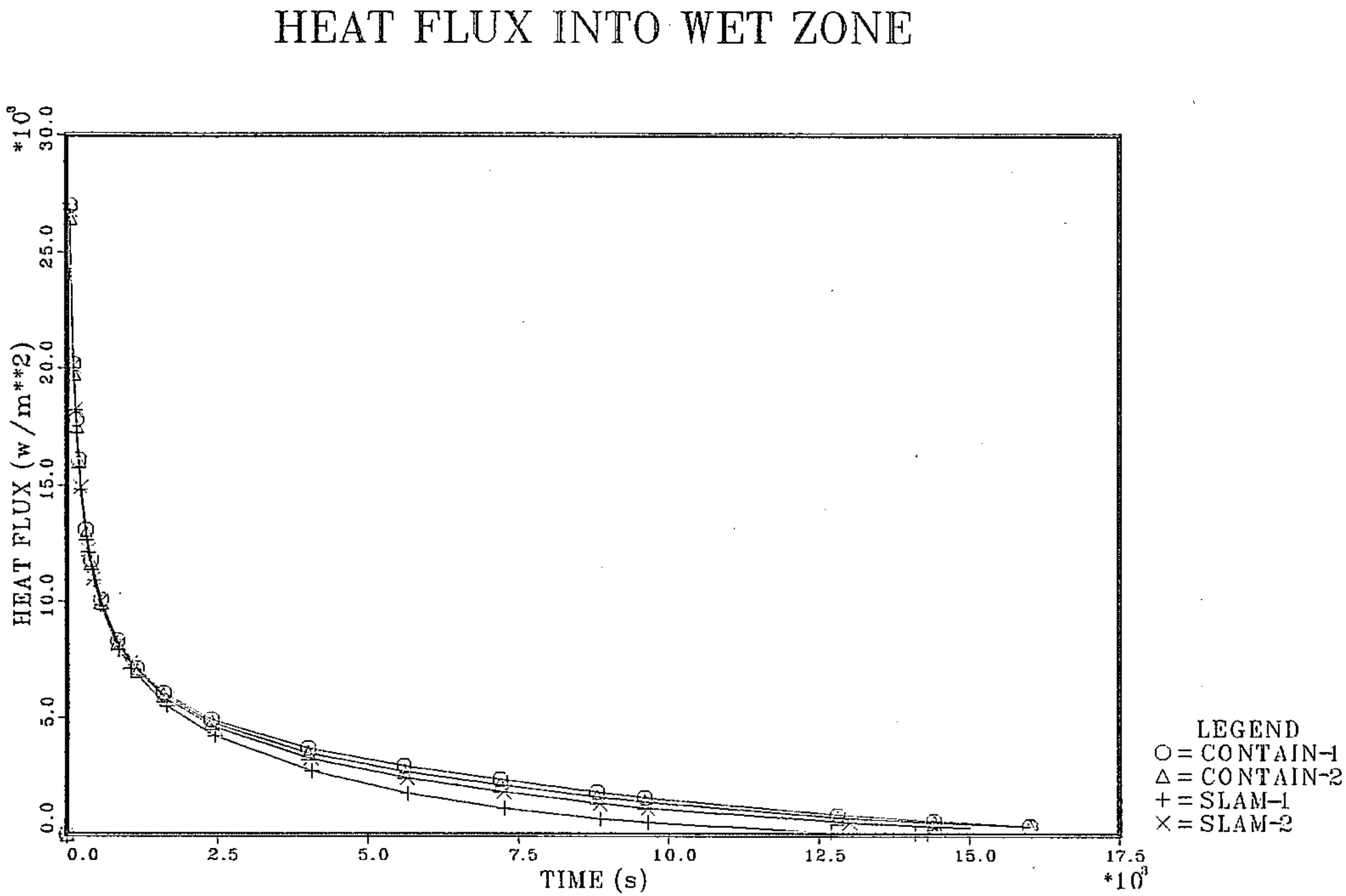


Figure 9.17 Comparison of Original SLAM and CONTAIN Calculations of Heat Flux into Wet Zone during Sodium-Concrete Reaction

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 .)

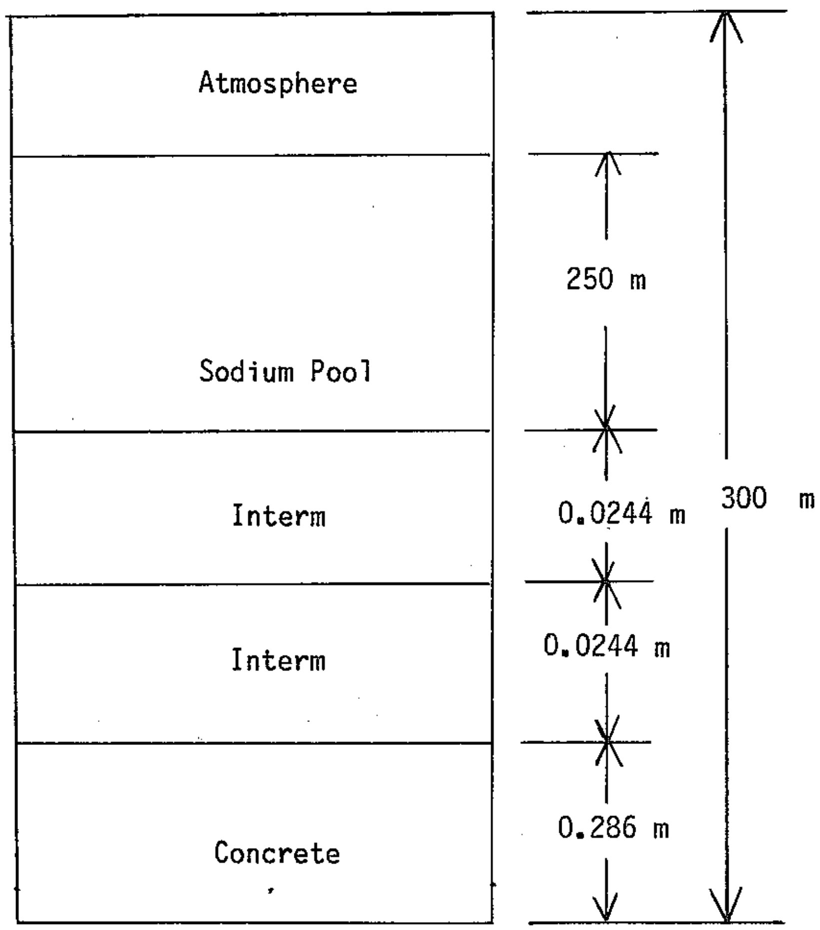


Figure 9.18 System for Water Release Test Calculation

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 RELESE 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 NA2SIO3 NA2CO3 NA2O CAO MGCO3 CACO3 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 perfomed, 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.

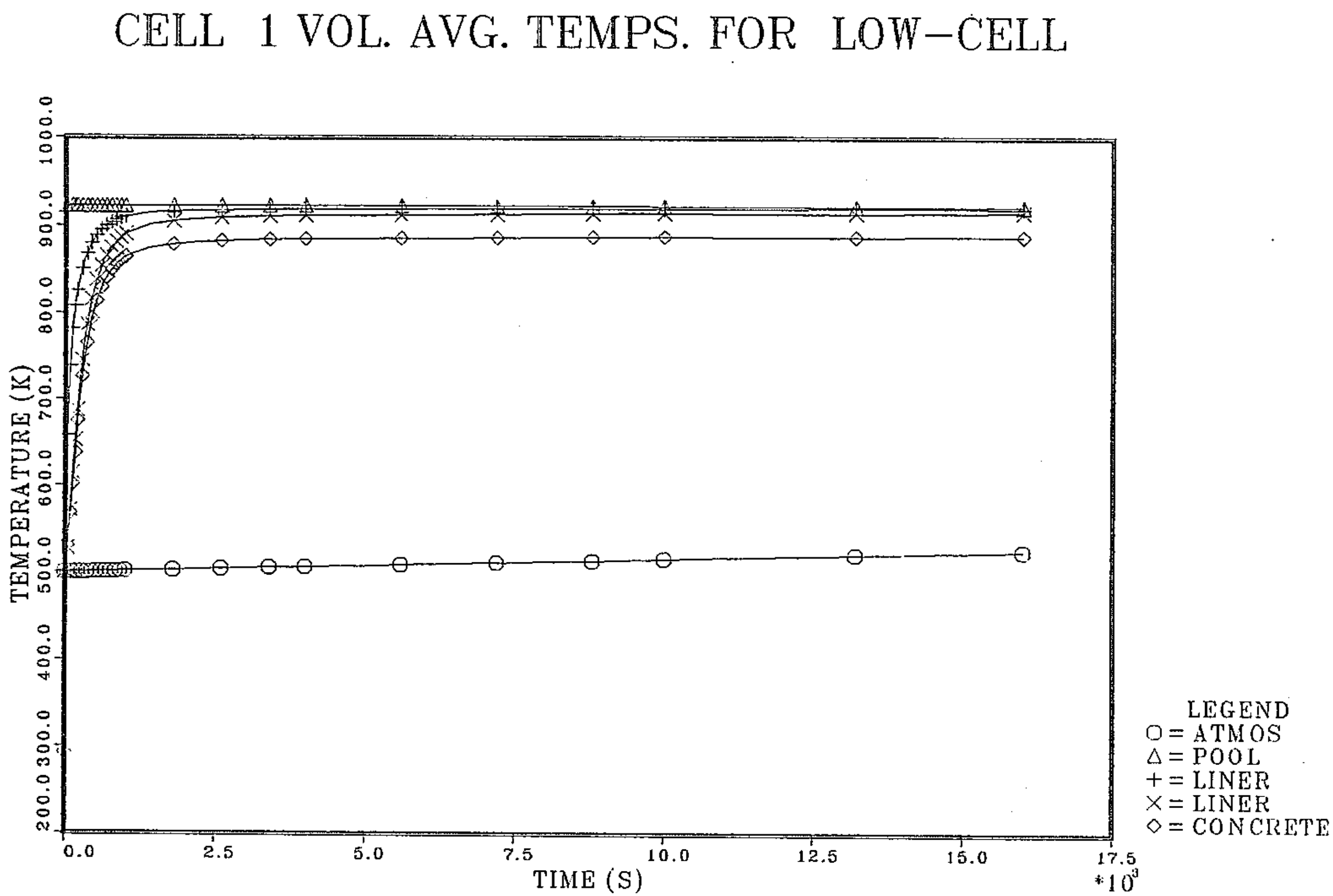


Figure 9.19 Results of Low-Cell Temperature of Water Release Calculation

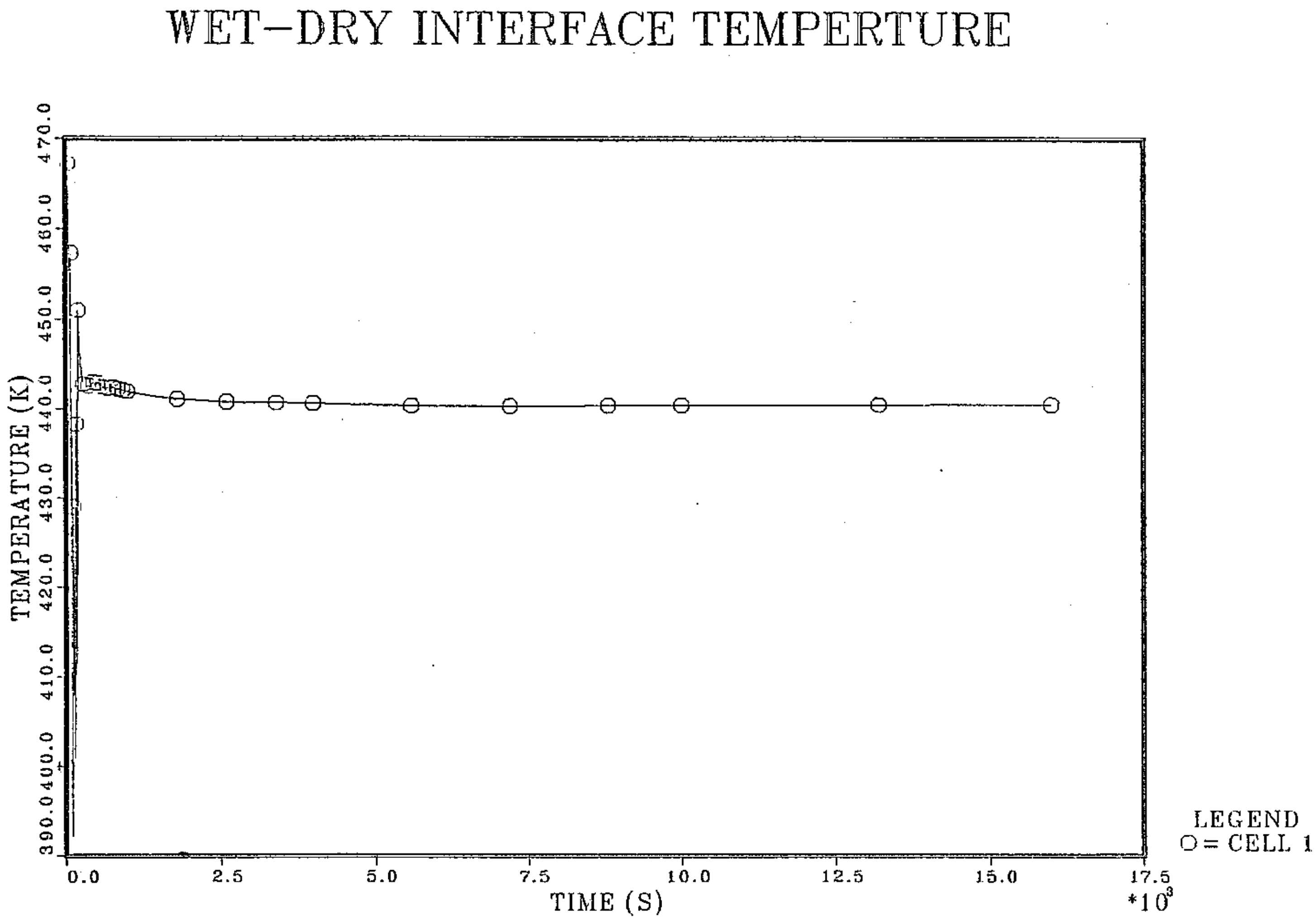


Figure 9.20 Result of Wet-Dry Interface Temperature of Water Release Calculation

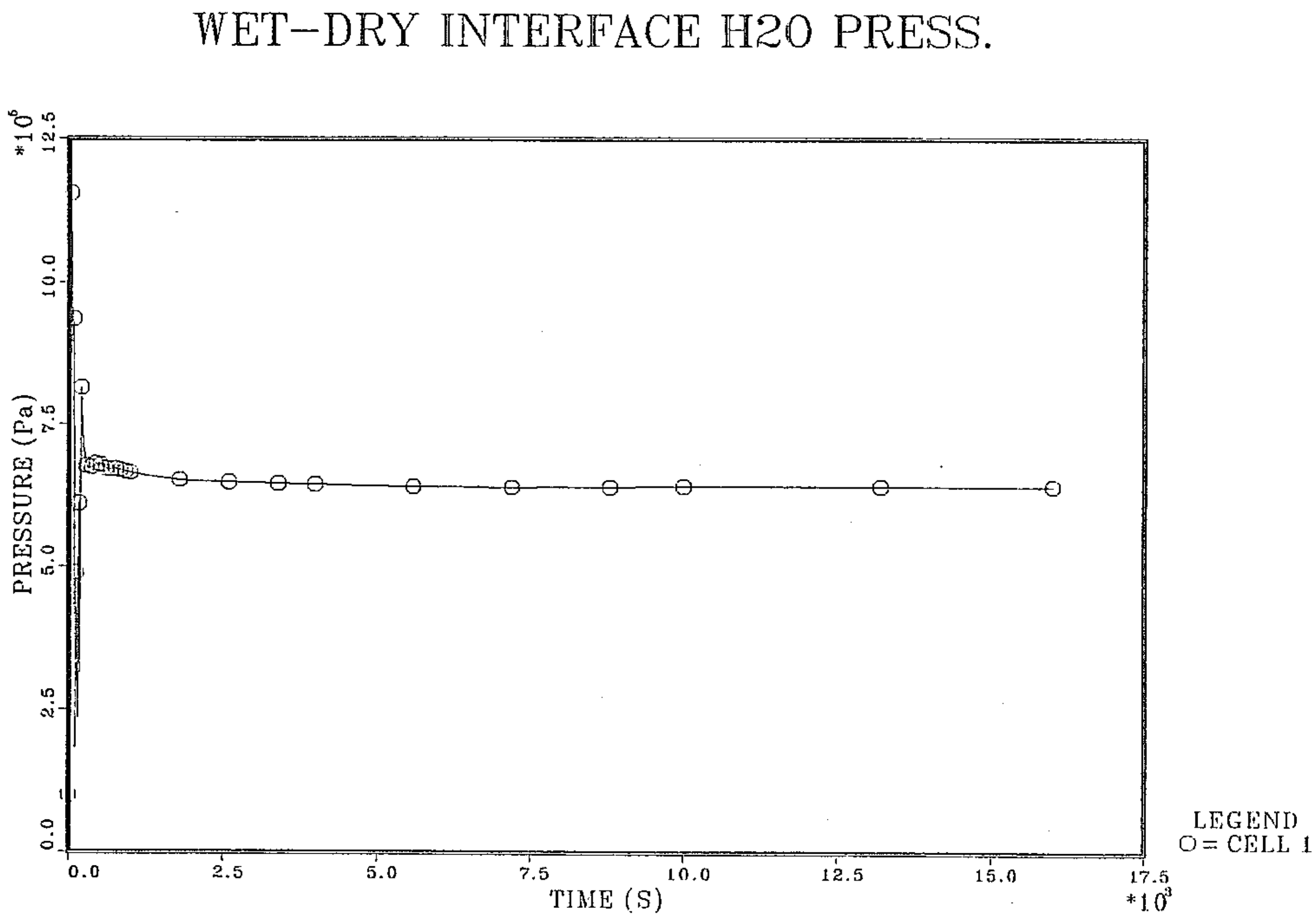


Figure 9.21 Result of Wet-Dry Interface H20 Pressure of Water Release Calculation

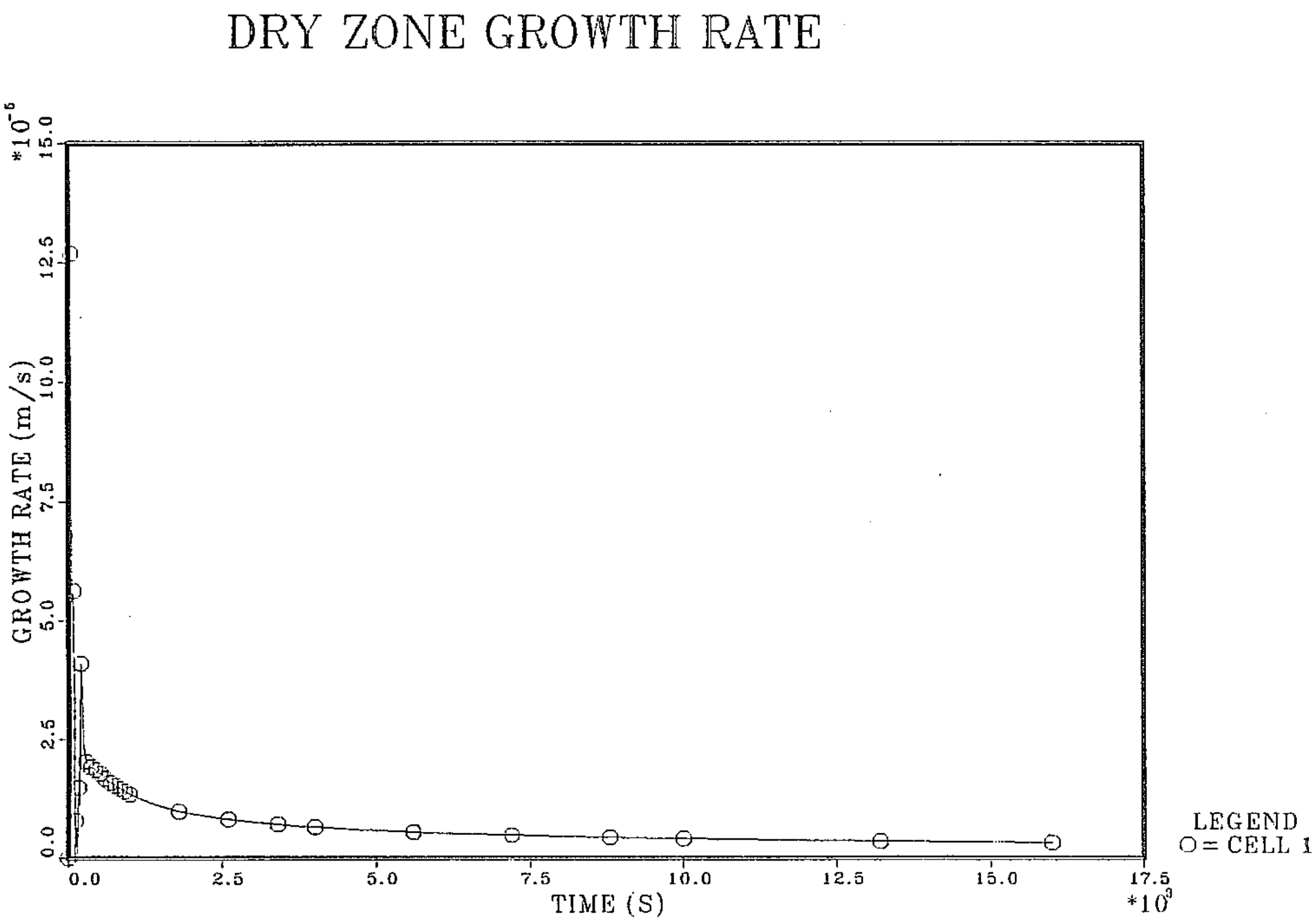


Figure 9.22 Result of Dry Zone Growth Rate of Water Release Calculation

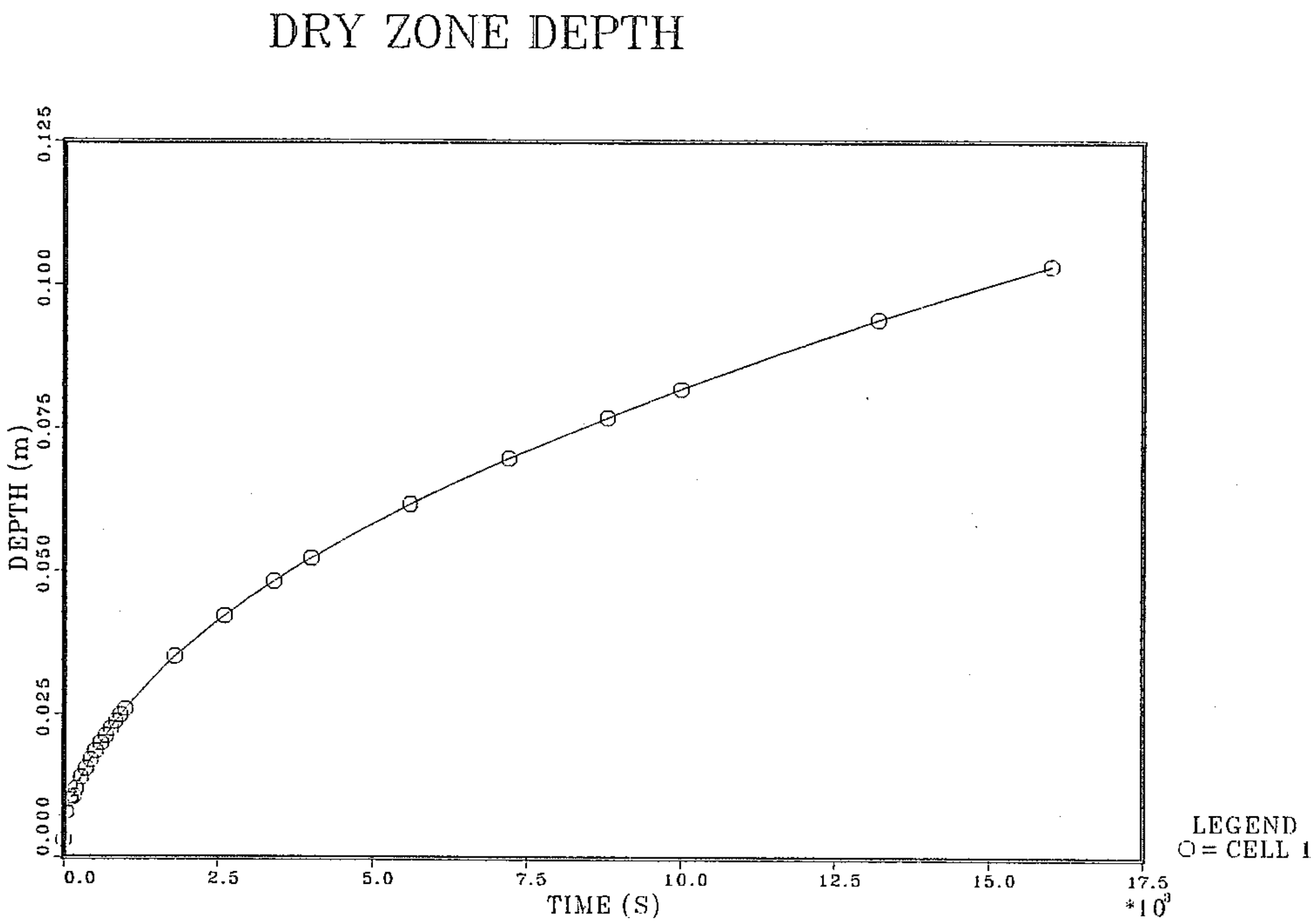


Figure 9.23 Result of Dry Zone Depth of Water Release Calculation

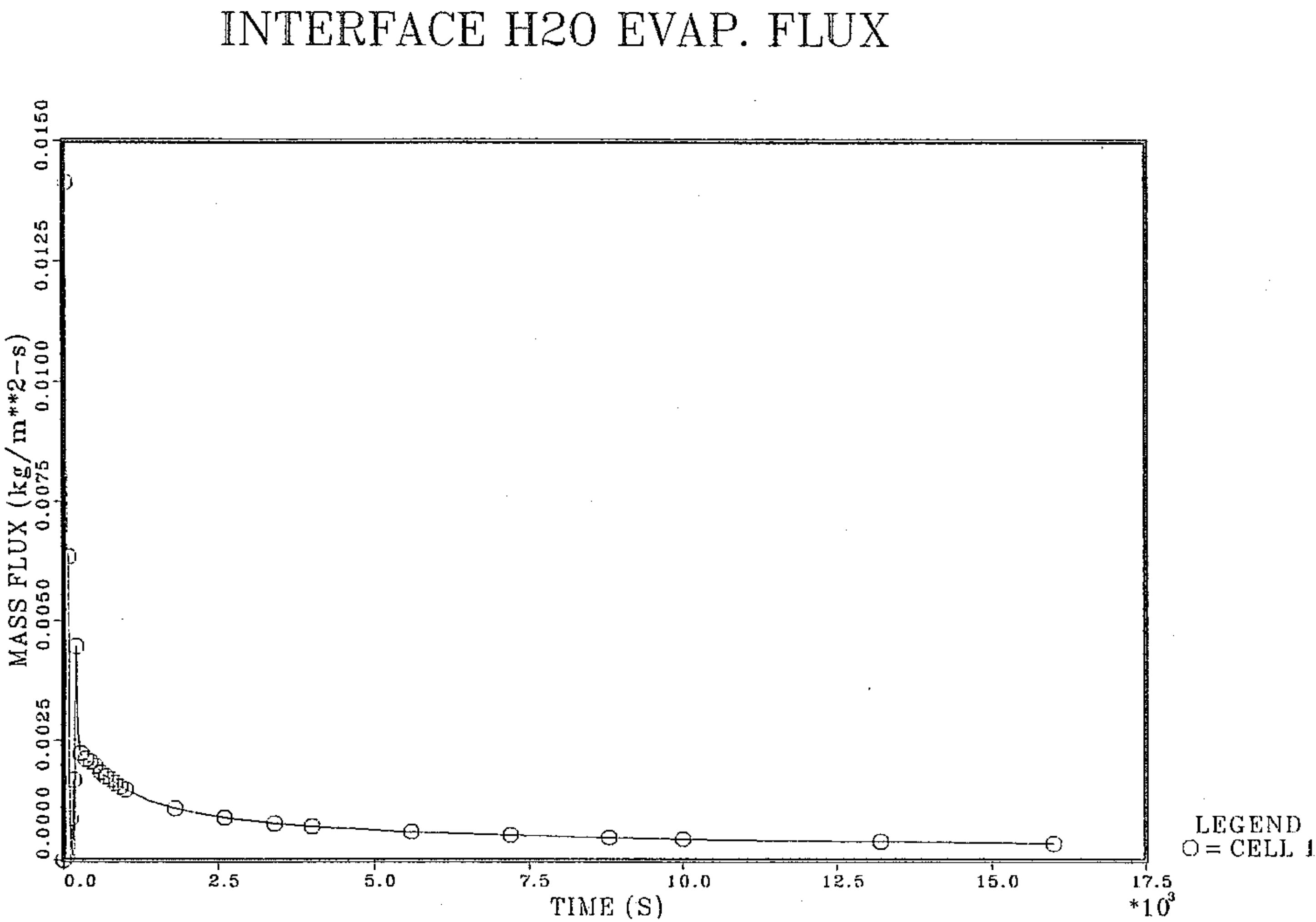


Figure 9.24 Result of Interface H20 Evaporated Flux of Water Release Calculation

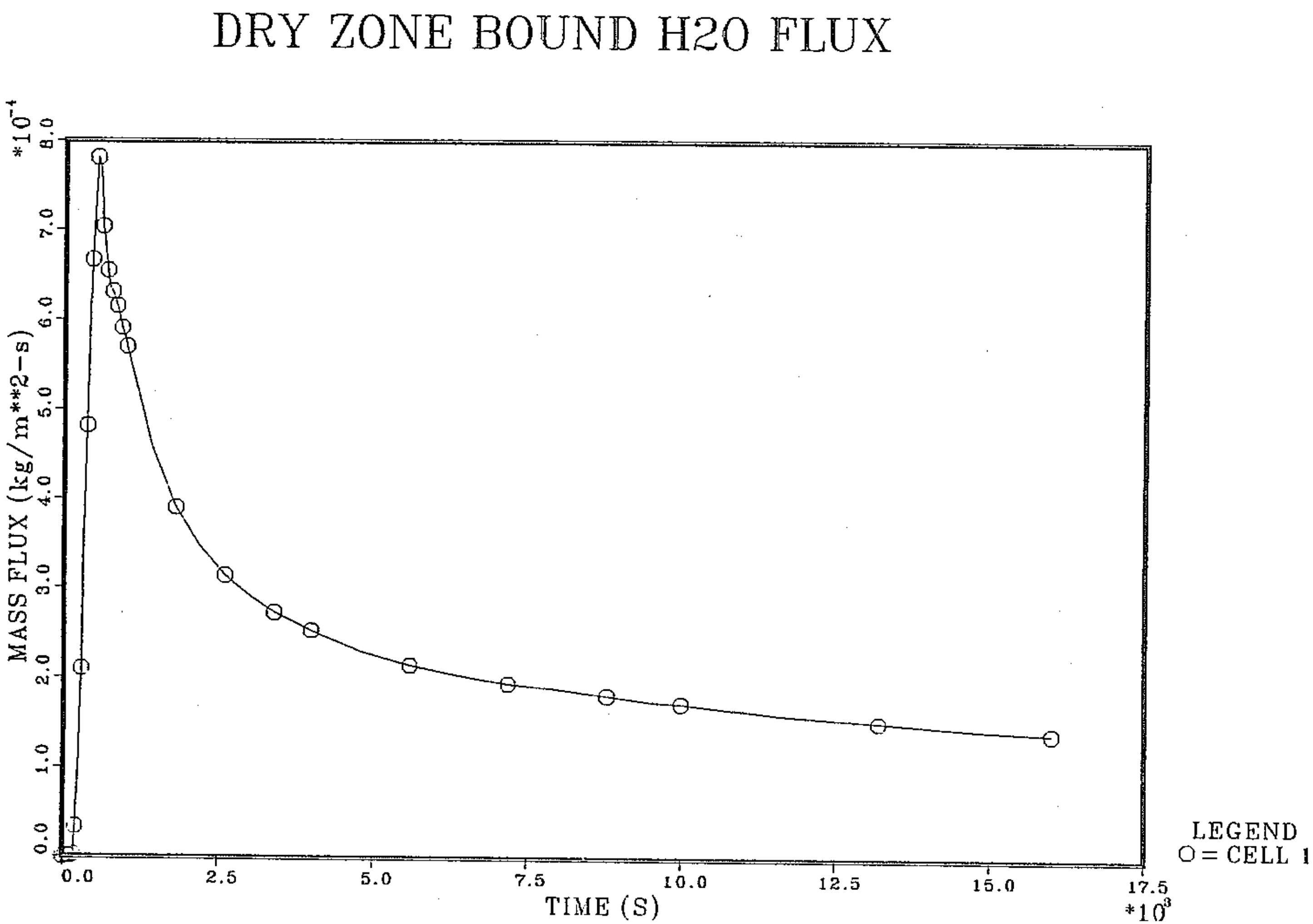


Figure 9.25 Result of Dry Zone Bound H2O Flux of Water Release Calculation

INTEGRAL OF BOUND + EVAP H2O

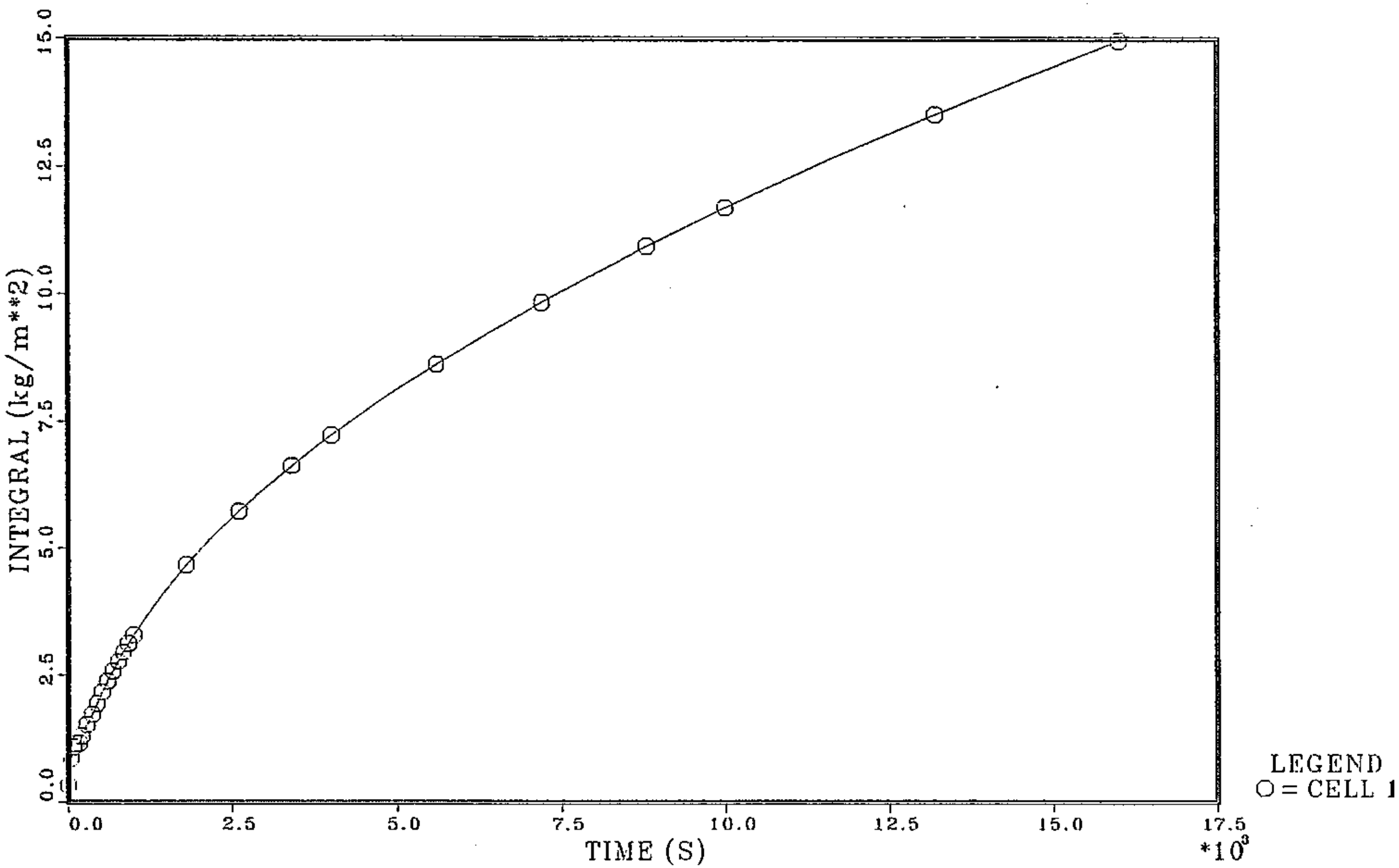


Figure 9.26 Result of Integral of Bound + Evaporated H2O of Water Release Calculation

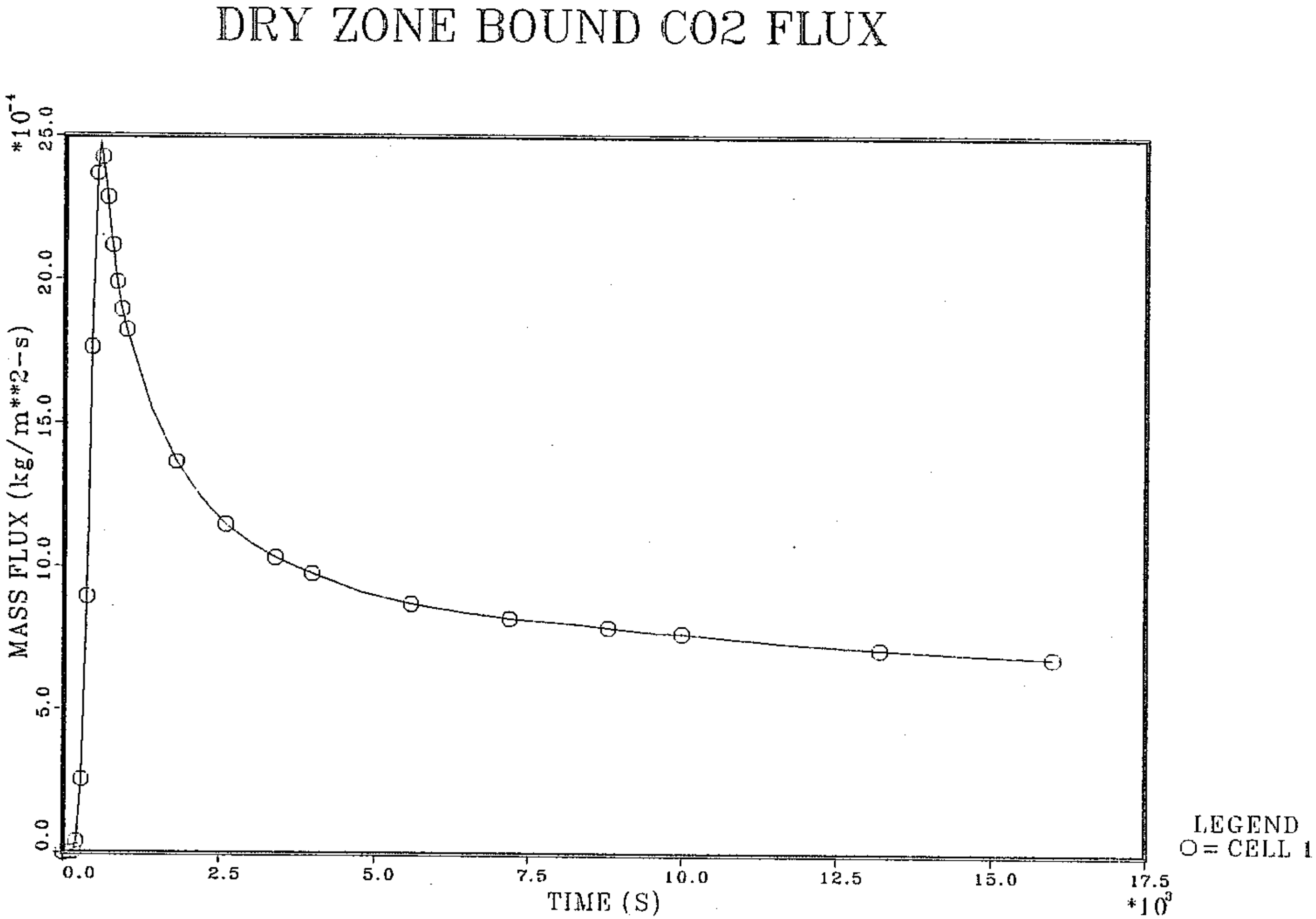


Figure 9.27 Result of Dry Zone Bound CO2 Flux of Water Release Calculation

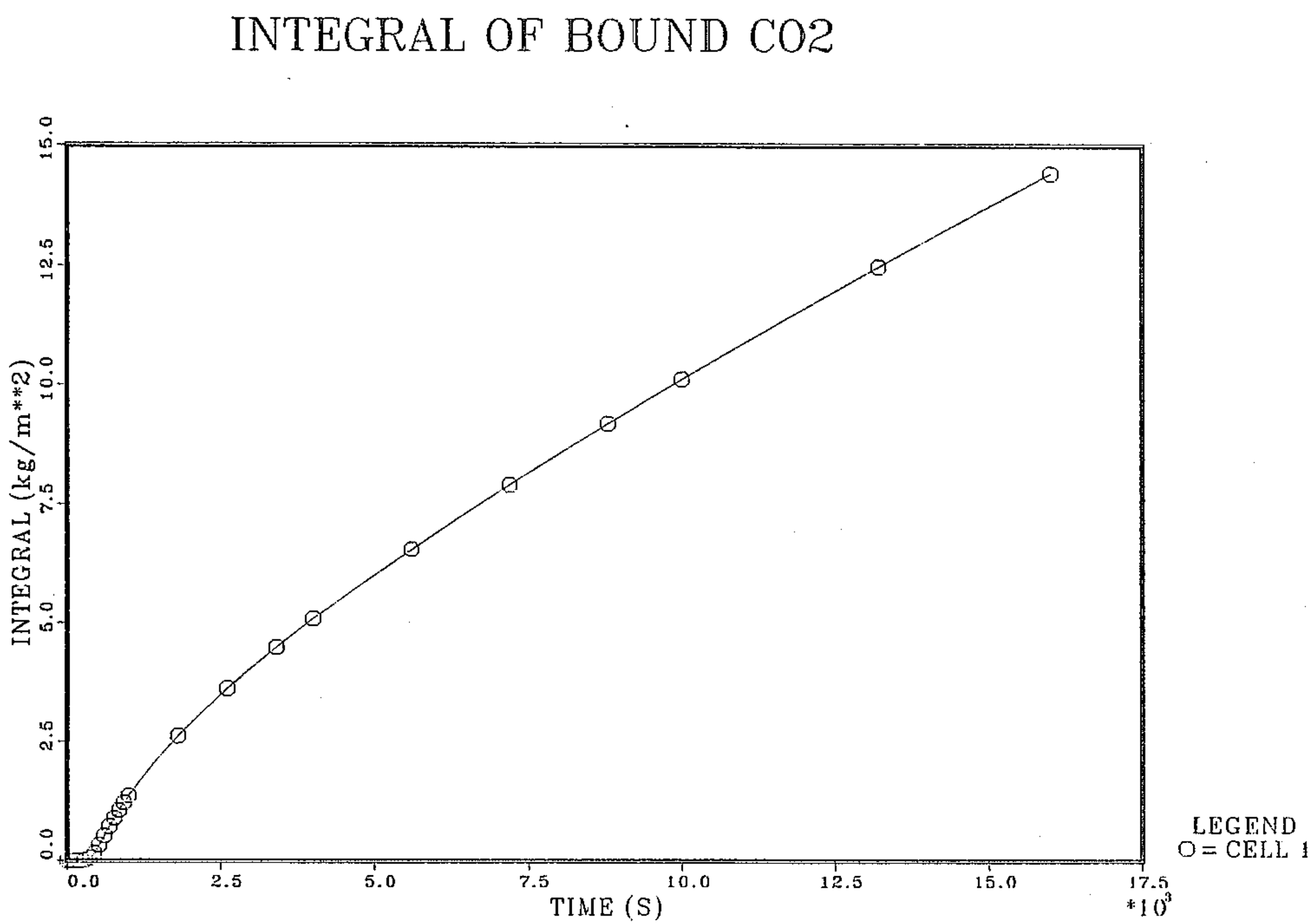


Figure 9.28 Result of Integral of Bound CO2 of Water Release Calculation

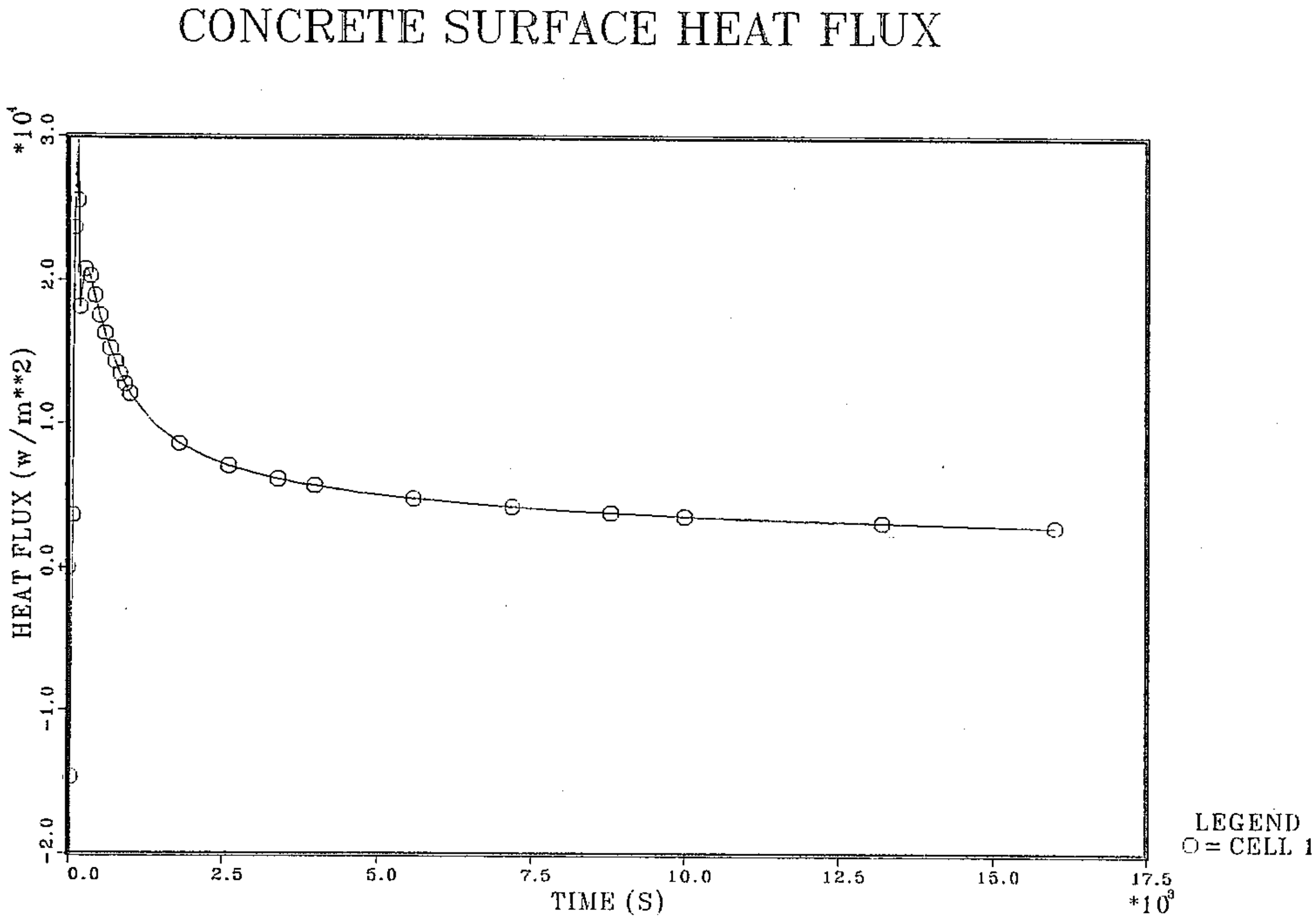


Figure 9.29 Result of Concrete Surface Heat Flux of Water Release Calculation

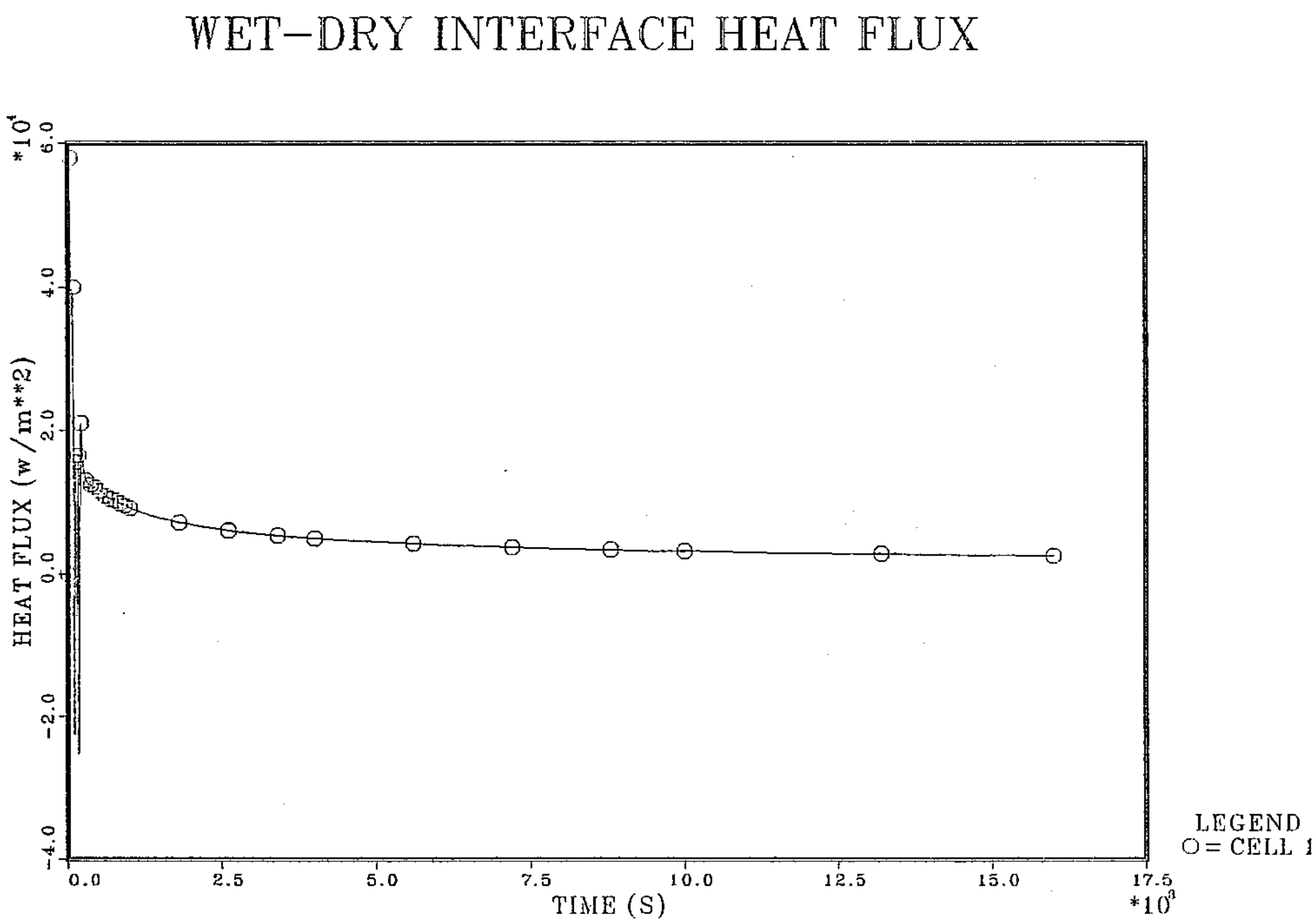


Figure 9.30 Result of Wet-Dry Interface Heat Flux of Water Release Calculation

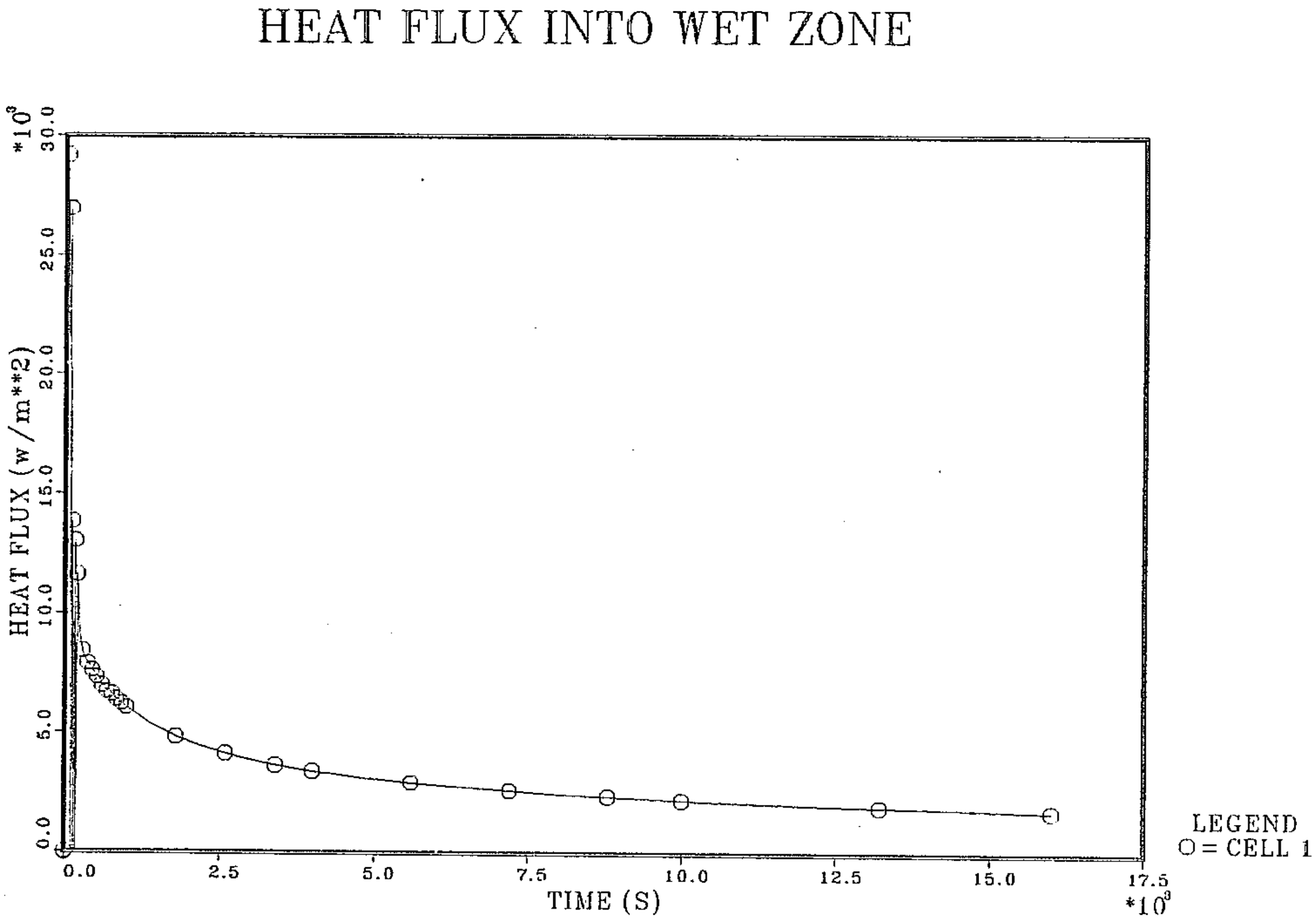


Figure 9.31 Result of Heat Flux into Wet Zone of Water Release Calculation

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.