256
8/21/79
UC-79 (Special)
(Local Project)

14. 3036
HEDL-TME 79-22
UC 79p

USERS' GUIDE TO CACECO CONTAINMENT ANALYSIS CODE

MASTER

Hanford Engineering Development Laboratory

APPLIED TECHNOLOGY

Any further distribution by any holder of this document or of the data therein to third parties representing foreign interests, foreign governments, foreign companies and foreign subsidiaries of foreign divisions of U.S. companies should be coordinated with the Director, Division of Reactor Research and Technology, U.S. Department of Energy.

Released for Announcement in Energy
Research Abstracts. Distribution Limited
to Participants in the LMFBR Program.
Others request from TIC

HANFORD ENGINEERING DEVELOPMENT LABORATORY
Operated by Westinghouse Hanford Company
P.O. Box 1970 Richland, WA 99352
A Subsidiary of Westinghouse Electric Corporation
Prepared for the U.S. Department of Energy
under Contract No. DE-AC14-76FF02170.

DISCLAIMER

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

DISCLAIMER

Portions of this document may be illegible in electronic image products. Images are produced from the best available original document.

NOTICE

This report was prepared as an account of work sponsored by the United States Government. Neither the United States nor the U.S. DOE, nor any of their employees, nor any of their contractors, subcontractors, or their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness or usefulness of any information, apparatus, product or process disclosed, or represents that its use would not infringe privately owned rights.

Printed in the United States of America
Available from
DOE Technical Information Center
P. O. Box 62
Oak Ridge, Tenn. 37830
Price: Printed Copy \$9.25

USERS' GUIDE TO CACECO CONTAINMENT ANALYSIS CODE

Hanford Engineering Development Laboratory

R. D. Peak

June 1979

NOTICE

This report was prepared as an account of work sponsored by the United States Government. Neither the United States nor the United States Department of Energy nor any of their employees nor any of their contractors subcontractors or their employees makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy completeness or usefulness of any information apparatus product or process disclosed or represents that its use would not infringe privately owned rights.

Released for Announcement in Energy
Research Abstracts Distribution Limited
to Participants in the LMFBR Program.
Others request from TIC

HANFORD ENGINEERING DEVELOPMENT LABORATORY
Operated by Westinghouse Hanford Company
P.O. Box 1970 Richland, WA 99352
A Subsidiary of Westinghouse Electric Corporation
Prepared for the U.S. Department of Energy
under Contract No. DE-AC14-76FF02170.

.

.

.

.

.

USERS' GUIDE TO CACECO
CONTAINMENT ANALYSIS CODE

R. D. Peak

ABSTRACT

The CACECO containment analysis code was developed to predict the thermodynamic responses of LMFBR containment facilities to a variety of accidents. The code is included in the National Energy Software Center Library at Argonne National Laboratory as Program No. 762.

This users' guide describes the CACECO code and its data input requirements. The code description covers the many mathematical models used and the approximations used in their solution. The descriptions are detailed to the extent that the user can modify the code to suit his unique needs, and, indeed, the reader is urged to consider code modification acceptable.

.

.

.

.

.

CONTENTS

	<u>Page</u>
Abstract	iii
Figures & Tables	vii
1.0 Summary	1-1
2.0 CACECO Example	2-1
2.1 Case 731 Facility	2-1
2.2 Case 731 Scenario	2-3
2.3 Case 731 Input	2-4
3.0 CACECO Summary	3-1
3.1 Analysis Cells	3-1
3.2 Heat Structures	3-2
3.3 Chemical Reactions	3-4
3.4 Differential Equations	3-7
4.0 CACECO Code Organization	4-1
5.0 CACECO Code Description	5-1
5.1 Main Program CACECO	5-3
5.2 Subroutine CINPUT	5-8
5.3 Subroutines HINPUT and HPSET	5-18
5.4 Subroutine CCALC	5-26
5.5 Subroutine EMVBAL	5-60
5.6 Subroutine HCALC	5-78
6.0 Helpful Hints	6-1
7.0 References	7-1

CONTENTS (Cont'd)

	<u>Page</u>
Appendix A Flow Diagram for the CACECO Code	A-1
Appendix B CACECO Input Format	B-1
Appendix C Data Format	C-1
Appendix D List of CACECO Common Block Variables	D-1
Appendix E Solution of the Heat Conduction Equation	E-1
Appendix F Case 731	F-1

FIGURES

	<u>Page</u>
1.1 CACECO Code Capabilities	1-2
2.1 Secondary Containment Model	2-2
4.1 Basic CACECO Flow Diagram	4-2
5.1 Node Point Layout	5-20

TABLES

2.1 Sodium-Concrete Reaction	2-8
2.2 Case 731 Heat Structures	2-11
2.3 Moving Boundary Simulation	2-13

USER'S GUIDE TO CALECO CONTAINMENT ANALYSIS (WXYAEC)

1.0 SUMMARY

The CACECO Code was developed at the Hanford Engineering Development Laboratory in order to predict the thermodynamic responses of a liquid metal fast breeder reactor (LMFBR) facility to a variety of facility accidents such as: (a) sodium leakage as spray and spill, (b) air injection, (c) decay heat release, and (d) space cooler operation. CACECO, the acronym for cavity, cells and containment, was developed from the CONTEMPT code⁽¹⁾ which had been written in 1967 to predict the containment responses of a light water reactor (LWR) facility. Both codes are included in the National Energy Software Center Library at Argonne National Laboratory, the CONTEMPT code as program No. 433 and the CACECO code as program No. 762.

The CACECO code analyzes the conditions in the reactor vessel, reactor cavity, vault, equipment cell, sodium pipeway, containment building, etc., according to input data supplied by the user. The codes treats each vessel, cavity, vault, cell, pipeway, or building as an analysis cell and can analyze the conditions in one cell, two cells, or in three or four cells, according to the needs of the accident scenario. These analysis cells are called Cell A, Cell B, Cell C, and Cell D. There is little specialization in the code; each cell has the same provisos as Cell A and can be used to obtain the same results. The code calculates cell conditions such as temperature and pressure from mass and energy balances that include chemical reactions, carbon dioxide and water vapor releases upon heating of concrete, decay power, atmosphere leakages, sodium transfers between cells, etc., accommodating the complex phenomena of accident scenarios for LMFBR facilities. The code capabilities are portrayed in Figure 1.1.

The CACECO code was written in the FORTRAN extended language for operation in the Control Data Corporation (CDC) computer systems, in particular the CDC-Cyber 74 computer system at Hanford Engineering Development Laboratory and the CDC-7600 systems at Brookhaven National Laboratory and Lawrence Berkeley Laboratory. With few exceptions (DATA statements mainly), the code

FIGURE 1.1. CACECO Code Capabilities.

HEDL 7801-137.1

statements comply with the American National Standards Institute (X3.9-1978) FORTRAN language and the code should be readily converted to other computers.

The code has been verified by 36 cases that set up problems that could be solved by (a) hand calculations or by reference to (b) textbook examples and (c) literature examples.⁽²⁾ These 36 cases were called Debug Cases because they were invented to debug the code; that is, to prove to the code writer (as well as to the user) that the code operates as described in this Users' Guide. The many problems treated in this verification suggest the large variety of situations that can be analyzed by the code.

Development and use of the CACECO code to predict the containment response to any particular accident situation entails three major steps which are sketched below:

1. The first step covers the invention and solution of the mathematical models that represent the containment situation. This step involves the simplification of real-life phenomena and uses chemistry, physics, thermodynamics, and other disciplines, in order to obtain the models that are incorporated into the code. These models are usually expressed as algebraic and differential equations. In the CACECO code there are five main differential equations to express the mass and energy balances of each cell and one differential equation to express the energy balance of each heat structure. This first step was mainly the responsibility of the code author.
2. The second step covers the selection and detailing of the descriptive models that provide parameter values for the mathematical models. This step involves the examination of objects in the facility (or on drawings) in order to obtain the areas, volumes, and material properties of the facility model and the development of accident scenarios, in order to estimate the initiators of the accident model. This second step is mainly the responsibility of the code user, who must be familiar with

the mathematical models provided by the code author, so that these descriptive models fit.

3. The third step covers the numerical solution of the mathematical models for the parameter values representing the descriptive models. This third step is mainly the task of the digital computer used to effect the solution. Naturally, there is interaction between the method of numerical solution of the mathematical models, which was the responsibility of the code author, and the parameter values of the descriptive models, which are the responsibility of the code user. This interaction between method and parameter values should give a wholesome solution in which the user has confidence. However, many parameter values may give an unreasonable solution for which the user has concern and, of course, there are parameter values which result in no solution because the interaction was destructive.

The purpose of the foregoing is to emphasize that the user has the important responsibility to make his descriptive models fit the mathematical models of the code.

Common sense must be exercised in interpreting the results of the CACECO code. These results, perhaps more than the results of other codes, depend upon the input data supplied by the code user because there is no standard (or default) cell, atmosphere, heat structure or concrete. The input data are so important that, to aid the user, the code input subroutines (CINPUT, HINPUT AND TINPUT) check the input data for being the proper quantity and kind (real, integer or alphanumeric). Unfortunately, the subroutines do not check for proper magnitude or for internal consistency. Nevertheless, these input subroutines discover a surprising number of data errors and the typical user can be elated upon passing their checks. Further, the user can and should monitor the validity of code results by two checks: (1) the magnitude of the code convergence parameter CONV(1) and its effect on the size STEP of the time step and (2) the presence of oscillatory pressures and temperatures. These two checks are facilitated because cell atmosphere

temperatures and pressures and both CONV(1) and STEP are listed in the data summary that is printed periodically.

An accident scenario that can be analyzed by the CACECO code is presented in the next section. Then the CACECO main program and important subroutines are described. The final section has hints and experiences that may help the user understand operating situations.

2.0 CACECO EXAMPLE

The CACECO code has been used at HEDL to analyze 700 cases which cover various situations ranging from a small-scale concrete dehydration test to a hypothetical reactor vessel melt-through incident in a 2500 Mwt LMFBR facility. The latter situation provides an excellent illustration of the code capabilities to follow a complex accident scenario.

2.1 Case 731 Facility

The situation for Case 731 was based on a 2500 Mwt reactor in a containment typical of pipe loop LMFBRs. For these, the primary sodium system has several circulating loops; each loop has an intermediate heat exchanger through which reactor heat is transferred to a secondary sodium loop and, thence, to steam generators. The reactor and primary sodium loops are located in inerted atmosphere cells with thick shielding roofs, walls, and floors below the operating floor of a reactor containment building (RCB). The inerted atmosphere cells constitute one barrier and the containment building constitutes the second barrier to the escape of radioactivity that may leak from the primary sodium loops. The cell that holds the primary pump and intermediate heat exchanger of one primary sodium loop is called a heat transport system (HTS) cell. The containment building, reactor cavity, and HTS cells are portrayed in Figure 2.1. In this case, the atmosphere of the cavity is vented to the atmospheres of the cells by ducts that prevent circulation. All concrete surfaces in the cavity and cells are lined with steel plates to conserve the inerting gas atmosphere and to prevent contact between sodium and concrete in the event of a sodium leak. These liner plates are connected with water vapor vent piping to avoid liner failure from the pressure of water vapor released from concrete in the event that the concrete is heated, as by a sodium spill. The water vapor vent piping leads to air-filled equipment rooms which are adjacent but not otherwise connected to the reactor cavity and HTS cells. The roof, wall, and floor surfaces of the equipment rooms are bare concrete. The atmosphere of these equipment rooms is vented to the atmosphere of the containment building by stairways which

HEDL 7905-048.1

FIGURE 2.1. Secondary Containment Model.

prevent circulation. The containment building roof and walls are steel which is covered on the outside with a plastic foam insulation which was assumed to melt when the steel is heated.

2.2 Case 731 Scenario

In the accident scenario for Case 731, a hypothetical core-disruptive accident was assumed to result in a whole-core meltdown which melted through the reactor vessel and guard vessel in 1,000 seconds. Sodium was released from the reactor vessel into the reactor cavity, along with molten fuel-core debris. The molten fuel was assumed to fragment into a bed of particulates and be cooled by the overlying sodium. Although there are reasons to believe that the cavity floor liner would remain intact, the assumption was made that the liner failed. The failed liner permitted a limited reaction between sodium and concrete and a complete reaction between sodium and water vapor released from the concrete. The latter reaction resulted in the formation of sodium oxide and hydrogen. The sodium oxide remained in the sodium pool. The hydrogen formed sodium hydride or leaked to the containment building, based on a design leak rate for the cavity and HTS cells of 100 volume percent per day at 10 psig. Under appropriate conditions in the building, the hydrogen recombined with oxygen to form water vapor. Decay energy from the core debris rapidly heated the sodium in the cavity to boiling, and the sodium vapor vented to the HTS cells through passages provided for that purpose. Sodium vapor condensed on the structural surfaces in the HTS cells, and accumulated there.

The heating of the cavity and HTS cell walls released water from the structural concrete. The water vapor was vented to the equipment rooms which were below the building floor and external to the cavity and HTS cells. The water initially condensed, but the rooms eventually heated to the point where water vapor passed into the building through the connecting stairways. The building heated and pressurized to 9 psig. At that pressure a relief system operated to vent gas, as required to limit the pressure to 10 psig.

The accident scenario ended with the cavity boiled dry of sodium. The CACECO code analysis of this scenario would provide the information needed to compute offsite radiological consequences, principally the time of containment venting, the rate of sodium transport along with fission products to the containment building, and the rate of building leakage to the environs.

Following sodium boil-dry, the core debris was no longer cooled by sodium evaporation, and melting of the cavity concrete floor was postulated. Concrete would continue to dehydrate, and some of the water vapor would be expected to react with molten steel in the core debris, forming hydrogen. Thus, additional off-gassing from the RCB would occur. This phase of the accident was not included in the CACECO analysis.

2.3 Case 731 Input

The input data for Case 731 are listed in Appendix F. These data follow the input data format of Appendix B to describe the containment building, reactor cavity, HTS cells and equipment rooms, along with the 15 heat structures which model the equipment, roofs, walls, and floors of these spaces. The task of assembling the input data for such complicated cases entails efforts such as the following:

1. Reading drawings, sketches, reports and letters for arrangements, dimensions, materials, thermal-physical properties, sodium-concrete reaction parameters and accident scenario scenes,
2. Interrogating designers and other sources for further information,
3. Copying and adapting applicable data of earlier CACECO cases having similar facility and scenario pieces,
4. Calculating thicknesses, areas, volumes, leakage coefficients, heat transfer coefficients, decay power allotments, and other values as needed, and

5. Guessing at unavailable values needed by the code.

In Case 731, the building, cavity, cells, and rooms filled the four analysis cells available. The author preferred to assign the cavity to Cell A and the building to Cell C. Then arbitrary choice assigned the cells to Cell B and the rooms to Cell D. These containment volumes and their initial conditions are entered on Input Cards 2 through 9 as described in Appendix B.

The CACECO code has restart capability and almost all of the input data can be modified at a restart. Case 731 used 12 runs which entailed 11 restarts. The usual operating procedure beyond the first run consists of: (a) restart for a run, (b) check the run output, and then either (c) restart for a rerun with modified input data, or (d) restart with a new run to continue the case. A complicated scenario such as Case 731 can take 500 or more cards to enter the input data. Of these, many are comment cards which describe the features of the case in plain language. The author urges free use of comment cards and their continual updating in any serial of cases in order to keep straight the significance of each case.

Integration time step and printout interval control data are entered on Input Card Series 9001. To speed the analysis when the predictor-corrector convergence meets internal criteria, the code will automatically lengthen the analysis time step by a factor of 1.58..., and again, and again, to a limit of 10 times the entered time step. Conversely, when the convergence fails criteria, the code will automatically shorten the analysis time step by a factor of $1/1.58...$, and again, and again, to a limit of 0.01 times the entered time step. The code user will adjust the time step in order to minimize run cost by using short steps for periods of rapid change and long steps for periods of slow change. Case 731 experience shown below was typical of this adjustment.

<u>Time Period</u>	<u>Time Step Entered</u>
0-1 hr	0.00025 hr
1-12 hr	0.0005 hr
12-36 hr	0.001 hr
35-end hr	0.002 hr

There may be design or scenario importance to the leakrate specifications, as in the example below, or to the nozzle size. Otherwise, the leakage coefficients and nozzle areas are best set by trial, in order to obtain a smooth running case. A common difficulty is to input coefficients or areas that are so large that 10% and more of the cell atmosphere is leaked in a single time step. A correlative difficulty with large coefficients or areas is to obtain leakage flow that oscillates direction in successive time steps. In Case 731 the containment building leakage was specified as being 0.1 percent per day. Local usage has taken this specification to mean 0.1 volume percent per day at the leak test conditions of air at 80°F and at the design pressure of 10 psig. At these test conditions the air density is 0.1237 lb/ft³. The leakage coefficient was calculated by the defining equation:

$$\begin{aligned}
 L_c &= \frac{(\text{leakrate, lb/hr})}{\sqrt{(\text{differential pressure, psid})(\text{density, lb/ft}^3)}} & (2.1) \\
 &= \frac{(2.3 \times 10^6 \text{ ft}^3)(0.001 \text{ fraction})(0.1237 \text{ lb/ft}^3)/(24 \text{ hr})}{\sqrt{(10 \text{ psid})(0.1237 \text{ lb/ft}^3)}} \\
 &= 10.7 \text{ lb/hr} - \sqrt{\text{psid-lb/ft}^3}
 \end{aligned}$$

Nozzle areas and leakage coefficients are entered in Input Cards 10 through 91.

The decay power history for Case 731 included only fission product decay power which was independent of the accident scenario. The factors for apportioning decay power among the cell atmospheres and pools did depend

upon the accident scenario because these factors specify the movement of noble gas fission products (Kr and Xe) and of volatile solid fission products (As, Br, Cd, Cs, I, Rb and Se). In particular, the Case 731 scenario assumed that the whole core meltdown released all noble gas fission products and that these blew out of the reactor vessel into the building at the outset of the analysis (1,000 seconds after the HCDA). Also, the scenario assumed that the volatile solid fission products boiled off with the sodium and followed the sodium vapor from the reactor cavity into the HTS cells. For example, at 24 hr after the HCDA and 23.72 hr after the start of Case 731 analysis, the fission product decay power was 4.438×10^7 Btu/hr and the noble gas fission products contribute 3.11% of this power and the volatile solid fission products contribute 19.89% of this power. At 23.72 hr the factor of 0.0311 was input to apportion the noble gas fission products to the building atmosphere. With the guess that boiloff had moved 10.9% of the sodium inventory into the HTS cells, the factor of $(0.1985 \times 0.1088 =)$ 0.0216 was input to apportion the volatile solid fission products to the HTS cell pools and the factor of $(1.000 - 0.0311 - 0.0216 =)$ 0.9473 was input to apportion the rest of the fission products to the cavity pool. Generally, when fission products follow sodium, as in Case 731, the sodium movements have to be guessed for the preparation of input data for the first time the case is run. Then, after the case has run, the CACECO calculated sodium movements can be used to modify the input data for the second time the case is run. This sort of iteration is common with CACECO code cases and illustrates why the code user should know the results of a case in order to prepare the input data to obtain these same results. Decay power and its apportioning factors are entered on Input Card Series 101, 201, and 301.

The accident scenario for Case 731 assumed local failures of the steel liner on the floor of the reactor cavity such that sodium passed through the failures and attacked the underlying concrete across the whole area of the floor. Generally, the extent of the sodium attack will depend upon the aggregate and sand materials used in the concrete and, because basalt aggregate and sand were used in the concrete of the Fast Flux Test Facility, basalt concrete has been the subject of a series of sodium-concrete

experiments. The particular mixes of these experiments had Portland cement (15.6 to 18.1 lb/ft³ concrete), water (8.4 to 9.1 lb/ft³ concrete) and basalt aggregate and sand to obtain concrete densities of about 150 lb/ft³. The concrete chemical composition was SiO₂ (51 to 56 weight percent), Al₂O₃ (10 to 11 percent), CaO (10 to 13 percent), FeO-Fe₂O₃ (6 to 7 percent), etc. Differential thermal analysis of this sodium-concrete reaction indicated that the reaction occurs above 930°F and is exothermic with a heat of reaction of about 52,200 Btu/ft³ concrete. The usual reaction products are the silicates, Na₂SiO₃ and Na₂O·CaO·SiO₂. The sodium-concrete reaction rate decreased with time, perhaps due to an accumulation of reaction products, when the sodium covered a flat, horizontal concrete surface. The reaction data could be correlated by a model having the rate, dX/dt, reduced in proportion to the reaction depth, X, viz: dX/dt = a - bX. For basalt concrete, a = 1.0 in./hr and b = 1/3 hr, so that:

$$dX/dt = 1.0 - X/3 \quad (2.2)$$

$$X = 3.0 - 3.0 \text{ Exp}(-t/3) \quad (2.3)$$

Case 731 applied this reaction equation to the sodium attack of the cavity floor. The case approximated the equation by using four time periods, each with a fixed reaction rate as listed below:

TABLE 2.1
SODIUM-CONCRETE REACTION

<u>Time Period</u> <u>hours</u>	<u>Reaction Rate</u> <u>inches/hour</u>	<u>Reaction Depth</u> <u>inches</u>	<u>Comments</u>
0.0 to 1.216	0.8824	0.25 to 1.25	Floor has steel liner to the 0.25-inch depth.
1.216 to 3.296	0.4808	1.25 to 2.25	
3.296 to 10.204	0.1448	2.25 to 3.25	
10.204 to end	0.0000	3.25	No further reaction.

The sodium-concrete reaction data are entered on Input Card Series 2401.

Concrete is the universal material of construction of LMFBR facilities because, by common experience, it is a stable, durable material for floors, walls, and, where shielding is important, for roofs as well. These roof, wall, and floor structures hold most of the overall heat capacity of the facility and are important for their heat absorption in accident analyses. Also, these concrete structures will release their water content, 9 lb $\text{H}_2\text{O}/\text{ft}^3$ concrete and more, when the concrete is heated above 200°F . Limestone concrete structures will release their carbon dioxide content, 50 lb CO_2/ft^3 concrete and more, when the concrete is heated above 1200°F . These water and carbon dioxide releases occur the first time the concrete is heated and cause significant dehydration and calcination energy absorption inside the concrete structures. These concrete effects are largely independent of the accident scenario and the data entered on Input Card Series 2201 through 2801 cover the following topics:

- a. Effective volumetric heat capacity vs. temperature to represent the dehydration and calcination energy effects of water and carbon dioxide releases,
- b. Thermal conductivity vs. temperature,
- c. Water release vs. temperature and carbon dioxide releases vs. temperature,
- d. Cell atmosphere or cell pool destination for water releases and for carbon dioxide releases from each concrete heat structure.

Case 731 assumed a facility having basalt concrete structures and these dehydrated when heated above 200°F by accident effects.

Concrete roofs, walls, and floors, for their heat absorption capabilities, and hot equipment as heat sources are important parts of any accident analysis. The CACECO code calculates these energy effects by using heat conduction structures to model the roofs, walls, floors, and equipment, and by

solving the one-dimensional, multi-region transient heat conduction equation for each structure. The 15 heat structures for Case 731 are listed in Table 2.2. These heat structures can be set up with several regions having different materials and different temperature node spacings in each region. Further, these structures may have planar, cylindrical, or spherical geometries. The planar geometry is used most often because flat walls and floors are common. For example, Case 731 used planar geometry for the cavity floor (Structure 5), the HTS cell floors (Structure 9), the building floor (Structure 12), and the equipment room floors (Structure 15). The cylindrical geometry is used as appropriate and Case 731 used this geometry for the cavity walls (Structures 3 and 4) to match their true shape. The spherical geometry is available for special purposes and Case 731 used it for the reactor vessel head (Structure 1). The actual head has a complex geometry with a larger area on its top surface than on its bottom because of the control rod and fuel handling equipment and because of the closure flange. This head was modeled as a sector of a thick spherical shell by setting up the area and thickness equations as follows:

$$\text{Top area} = 560 \text{ ft}^2 = 4\pi F R_2^2 \quad (2.4)$$

$$\text{Bottom area} = 326 \text{ ft}^2 = 4\pi F R_1^2 \quad (2.5)$$

$$\text{Head thickness} = 8 \text{ ft} = R_2 - R_1 \quad (2.6)$$

The solution of these three equations was an inside (bottom) radius of $R_1 = 25.8 \text{ ft}$ and outside (top) radius of $R_2 = 33.8 \text{ ft}$, and a fraction of the spherical surface of $F = 0.0390$ (dimensionless) which was used for the heat transfer calculations at the inside (bottom) and outside (top) surfaces of this structure. Case 731 used 15 heat structures to describe the features of Figure 2 and the preparation of the input data for these was a significant part of the case effort. Input data for these heat structures are entered following Card 9999 and there are six or more cards for each heat structure.

TABLE 2.2
CASE 731 HEAT STRUCTURES

<u>Structure No.</u>	<u>Structure Titles</u>	<u>Boundary Temperatures</u>	<u>Area</u>	<u>Comprised of</u>
10	RCB* Equipment with 6.0 x 10 ⁶ lb steel	Building atmosphere	120,000 ft ²	1.2 inch steel
11	RCB Roof and Walls	Building atmosphere /outside air	70,000 ft ²	1.5 inch steel and 1 inch insulation
12	RCB Floor	Building atmosphere	11,600 ft ²	6.5 feet concrete
13	RCB Floor over HTS Cells	Building atmosphere /HTS Cell roof	6,600 ft ²	3.25 feet concrete
1	Reactor Vessel Head	Cavity atmosphere /RCB atmosphere	326 ft ² , bottom 560 ft ² , top	2.5 feet shielding steel, 2.5 steel 3.0 feet insulation
2	Reactor Vessel and Internals with 2.4 x 10 ⁶ lb steel	Cavity atmosphere	24,700 ft ²	0.2 feet steel
3	Cavity Roof and Upper Wall	Cavity atmosphere	3,300 ft ²	Steel liner and 6 feet concrete
4	Cavity Submerged Wall	Cavity pool	3,930 ft ²	Steel liner and 6 feet concrete
5	Cavity Floor	Cavity pool	1,260 ft ²	Failed steel liner and 6 feet concrete

* RCB denotes the Ractor Containment Building.

TABLE 2.2 (Cont'd)
CASE 731 HEAT STRUCTURES

<u>Structure No.</u>	<u>Structure Titles</u>	<u>Boundary Temperatures</u>	<u>Area</u>	<u>Comprised of</u>
6	HTS Cell** Equip- ment with 7.5×10^6 lb sodium and 2.0×10^6 lb steel	Cell atmosphere	15,000 ft ²	Sodium, steel and 9 inches of insulation
7	HTS Cell Roof	Cell atmosphere/ 6,600 ft ² RCB Floor	6,600 ft ²	Steel liner and 3.25 feet concrete
8	HTS Cell Walls above pool	Cell atmosphere	42,000 ft ²	Steel liner and 4 feet concrete
9	HTS Cell Submerged Wall and Floor	Cell Pool	5,400 ft ²	Steel liner and 4 feet concrete
14	Equipment Room Roof and Walls	Room atmosphere	12,700 ft ²	3 feet concrete
15	Equipment Room Floors	Room pool	7,100 ft ²	3 feet concrete

** HTS Cell denotes the Heat Transport System Cell.

The materials generally used in the CACECO heat structures have constant values of volumetric heat capacity and thermal conductivity. These material properties are entered on Input Card Series 300001. Concrete is an exceptional material because its properties may be temperature-dependent, as already noted. The accident scenario for Case 731 assumed that the sodium-concrete reaction products of the cavity floor would mix with the overlying sodium pool so that the attack depth of Equation (2.3) was really the effective, moving boundary between the sodium pool and unreacted concrete. This moving boundary was simulated in the analysis by changing the material index of successive one-inch layers of the floor (Structure 5) from the index for concrete (with thermal conductivity of about 1.0 Btu/hr-ft-⁰F) to the index for reaction products (with assumed thermal conductivity of 200 Btu/hr-ft-⁰F). This index change was made when the reaction had penetrated two thirds of the way through each layer described in Table 2.1, as listed below.

TABLE 2.3
MOVING BOUNDARY SIMULATION

<u>Time for Change</u>	<u>Reaction Depth</u>	<u>Comment</u>
0.76 hr	0.92 in.	Floor has steel liner to 0.25 inch depth; change index for top layer.
2.44 hr	1.92 in.	Change index for second layer.
6.62 hr	2.92 in.	Change index for third layer.

Heat transfers that occur between equipment, cell roofs, walls, and floors, cell atmospheres and pools, form an important part of the analysis. The equipment roofs, walls and floors are modeled as heat structures and each structure has a left-hand (inside) surface and a right-hand (outside) surface which can be connected by convection and radiation to any cell atmosphere or pool, or to another heat structure. For example, Case 731 had the reactor vessel head (Structure 1) connected by convection heat transfer to the cavity atmosphere on its bottom (inside) surface and by convection heat

transfer to the building atmosphere on its top (outside) surface. Thick concrete structures, such as the cavity walls (Structures 3 and 4) and the building floor (Structure 12), can be connected to an atmosphere or pool on one surface and to nothing on the other surface (as an adiabatic or insulated surface) without effect to the results because the heat wave from the atmosphere or pool will not penetrate the structure during the analysis. Each connection for convection (or radiation) heat transfer uses: (a) an index to the coefficient for convection (or total interchange area for radiation) heat transfer and (b) another index to the atmosphere, pool or structure temperature. The convection coefficients and interchange areas can be constant or dependent upon either time or temperature. Sometimes several heat transfer items can be combined to simplify the modeling. For example, Case 731 assumed that the cavity walls were covered with 0.25-inch-thick steel liner plate as a standard LMFBR design feature which conserves inert atmosphere during normal operation and protects the concrete in event of a sodium spill. This case could have modeled the steel liner as one heat structure and the concrete wall as another and let the code calculate the conduction and radiation heat transfer across the assumed 0.125-inch-thick gas gap that separated the liner from the concrete. Instead, Case 731 combined the liner, gap and concrete into one structure with the liner and gap homogenized into one region and the concrete divided into three regions having different temperature node spacings. Further, the convection coefficient, h_c , on the atmosphere or pool side of the liner was combined with the conduction, h_g , and radiation, h_r , coefficients of the gap into one overall coefficient, U , by using the equation below:

$$\frac{1}{U} = \frac{1}{h_c} + \frac{1}{h_g + h_r} \quad (2.7)$$

With the liner at 800°F,

$$h_g = K/x = 2.68 \text{ Btu/hr-ft}^2\text{-}^\circ\text{F}, \text{ the conduction coefficient for the gap having gas thermal conductivity } K \text{ and thickness } x.$$

$h_r = \sigma \bar{e} 4 T^3 = 8.17 \text{ Btu/hr-ft}^2\text{-}^\circ\text{F}$, the radiation coefficient calculated using the Stefan-Boltzmann constant σ and emissivities e of 0.7 for the steel liner and 0.8 for the concrete wall.

The overall coefficient depends upon the convection coefficient as illustrated below:

<u>Convection Coefficient</u> <u>Btu/hr-ft²-°F</u>	<u>Overall Coefficient</u> <u>Btu/hr-ft²-°F</u>	<u>Index to Coefficient</u>	<u>Application</u>
1.07	0.97	105	Natural convection in atmosphere of nitrogen saturated with sodium vapor
30.0	7.99	106	Natural convection in sodium pool

These overall heat transfer coefficients and the total interchange areas for radiation heat transfer are entered on Input Card Series 400001 through 4900001.

The CACECO code calculates steady-state temperatures in all heat structures before beginning the transient analysis. When structure boundary conditions index the surface temperatures of other structures (instead of the temperatures of cell atmospheres or pools), then this steady-state calculation can be helped by the input of appropriate surface temperatures for both structures involved. In Case 731 this situation arose with a 6.5-ft-thick slab of concrete which was simultaneously (above, looking down) a portion of the building floor and (below, looking up) the roofs of the HTS cells. This thick slab was modeled as two 3.25-ft-thick slabs, one called the building floor over the HTS cells (Structure 13) and the other called the HTS cell roofs (Structure 7). With the building at 80°F and the HTS cells at 150°F, initially, the interface temperatures of 119.9°F for Structure 7 and 119.3°F for Structure 13 were entered on temperature Input Cards 7 and 13 which were

located between Input Card 500000 and 600000, near the end of the card deck for run 1 of this case.

Sometimes the CACECO code calculates incorrect steady-state temperatures in certain heat structures. This happened in Case 731 for the reactor vessel head (Structure 1), the reactor vessel and internals (Structure 2), and for the HTS cell equipment (Structure 6). The easiest remedy is to enter correct temperatures on input cards. This was done on temperature Input Cards 1, 2, and 6, which were located between Input Card 600000 and Card 10000000, the latter card being the last card of the input data for run 1 of Case 731.

3.0 CACECO SUMMARY

The CACECO code was developed to predict the thermodynamic response of LMFBR secondary containment to facility accidents such as: (a) loss of space cooling in a pipeway or cell, (b) sodium leakage as a spray or spill, and (c) postulated HCDA releases. The code analyzes the conditions in the reactor vessel, in the reactor cavity, in adjacent pipeways and cells, and in the containment building, according to input data supplied by the user.

The code uses a predictor-corrector scheme to advance the mass and energy differential, conservation equations over each time step. There is restart capability, and almost all input data can be changed at each restart. The restart capability permits the user to make an analysis in a sequence of computer runs, with opportunity at each restart to modify input data, and thereby modify the scenario; this permits user interaction throughout the analysis.

3.1 Analysis Cells

The code treats each vessel, cavity, pipeway, cell, or building as an analysis cell. The code can analyze the conditions in one cell, two cells, or in three or four cells, according to the needs of the accident scenario. Each analysis cell has an atmosphere and may have a pool. The atmosphere may include carbon dioxide, hydrogen, nitrogen, oxygen, water vapor, and sodium vapor, but not all of these simultaneously, as explained later. This atmosphere is well mixed and is represented by a single temperature and pressure, and by a uniform composition which, of course, will change during the analysis. The pool, if it exists, may have liquid sodium or water, but not both simultaneously, as explained later. The pool is well mixed and is represented by a single temperature. Further, as a result of chemical reactions, the cell may contain carbon, sodium hydride, sodium hydroxide, and sodium oxide.

The code user defines accident effects in terms of the following: (a) sodium spray, (b) sodium spill, (c) nitrogen-oxygen injection, (d) decay-heat release, and (e) cell cooler operation. With the sodium spray, the code neglects spray droplet dynamics and burning phenomena, and mixes the spray with the cell atmosphere. The spray is consumed when there is an excess of carbon dioxide, hydrogen, oxygen, or water vapor, but when there is an excess of sodium, the amount above saturation condenses and is added to the pool.

Thermodynamic conditions, which include the cooling of a saturated atmosphere, will cause condensation, and the liquid condensate will be accumulated. Also, the condensate, sodium, or water can boil off into the atmosphere. Generally, any cell can leak into any other cell or to the atmosphere outside of the containment building. The leakage rates are calculated from flow equations, according to input data that specify leakage coefficients or nozzle areas.

The cell temperature may range from 50⁰F to 2500⁰F. The code has subroutines for calculating the thermodynamic properties of sodium liquid and vapor over this range, of water liquid and water vapor to 300⁰F, of water vapor to 2500⁰F, and of sodium hydride. The code has functions for calculating energy and enthalpy values for carbon dioxide, hydrogen, nitrogen, and oxygen, and for energy values for carbon, sodium hydroxide, and sodium oxide.

3.2 Heat Structures

The code user defines the heat structures which represent the heat absorption and source of the cells. For example, one structure can model the cell roof and walls above the sodium pool, another structure can model the walls and floor submerged by the pool, and still another can model the equipment. Each structure is analyzed by a one-dimensional, multilayer, transient heat conduction equation in the user's choice of planar, cylindrical, or spherical geometry. Each structure can have up to 25 temperature nodes, with as many as 20 regions of different node spacing and materials. The boundary

conditions include convection and radiation to any cell atmosphere or pool, to any other structure, to outside air, or constant temperature heat sink.

The code user can direct special treatment for concrete structures in the following respects:

- (a) water vapor and carbon dioxide release as concrete is heated,
- (b) concrete heat capacity effect of dehydration and calcination as a function of temperature,
- (c) concrete thermal conductivity as a function of temperature, and
- (d) sodium-concrete chemical reaction.

The code analysis of concrete water vapor and carbon dioxide release uses the assumption that the only resistance to their release is the conduction heating of the concrete. As noted before, the code treats the conduction heating (and cooling) of a heat structure, such as the concrete cell wall, by solving a one-dimensional, transient heat-conduction equation. In each equation for concrete structures, the code uses temperature-dependent thermal conductivity. Also, during the first heating, the code adds temperature-dependent heat capacity effects of dehydration and calcination to the concrete heat capacity. The code keeps track of the maximum temperature attained at each temperature node, and uses the maximum temperature to determine the water and carbon dioxide release. The code user can direct the disposition of the released water and carbon dioxide to any cell atmosphere or pool. For example, concrete surfaces in a sodium equipment cell are normally covered by steel liner plate to protect the concrete in event of a sodium leak. Since a massive spill can heat the concrete and release enough water to burst the liner, the lower wall and floor liners of these cells are normally connected to vent piping systems which will vent the water and carbon dioxide releases to another containment space. The code user directs the releases of a cell floor into an adjacent equipment room,

in order to model proper operation of liner and vents, or directs the releases into the sodium pool on the floor, in order to model liner failure.

3.3 Chemical Reactions

The chemical reactions covered by the code are analyzed in the following sequence:

- (a) Sodium pool reactions with concrete are first. These reactions occur when a hot sodium pool submerges concrete surfaces that are bare or covered by a steel liner that is postulated to fail. The code analysis of these reactions is nonmechanistic. The code user supplies the concrete attack rate, the heat of reaction, and the sodium consumption.
- (b) Sodium pool reactions with carbon dioxide and water vapor are second. The carbon dioxide would be released upon heating concrete made from limestone, while water vapor is released upon heating of all kinds of concrete. The analysis situation has the concrete submerged by the sodium pool, and the carbon dioxide and water vapor bubbling up through the pool. The code has the following reactions:

<u>Pool Reactions</u>	<u>Energy Release</u>
$4 \text{ Na} + \text{CO}_2 \rightarrow 2\text{Na}_2\text{O} + \text{C}$	1,988 Btu/lb Na
$2 \text{ Na} + \text{H}_2\text{O} \rightarrow \text{Na}_2\text{O} + \text{H}_2$	1,600 Btu/lb Na

- (c) Water vapor reacts with sodium oxide next. The analysis situation has the vapor being released from the concrete floor, which is covered by hot sodium oxide. The sodium oxide comes from atmospheric reactions, described below, of sodium that enters a cell containing oxygen. The water vapor must pass through the oxide deposit in order to reach the atmosphere. The code has the following reaction:

Deposit Reaction Energy Release

1,500 Btu/lb Na

- (d) Sodium vapor reactions occur next. The sodium vapor reacts with carbon dioxide, oxygen, and water vapor. The carbon dioxide would come from heating limestone concrete. Oxygen is present at a concentration of one or two percent in the initially inerted atmosphere cells, and air is initially present in the containment building. Also, the user can inject air into any cell. Water vapor is present from heating concrete. Finally, the carbon dioxide, oxygen, and water vapor can leak from one cell to another and to the outside. The code has the following reactions:

Atmosphere Reactions Energy Release

3,800 Btu/lb Na

3,400 Btu/lb Na

5,700 Btu/lb Na

Sodium reacts with carbon dioxide, oxygen, and water according to their molar ratios when the concentration of oxygen is less than the concentration of water. When concentration of oxygen is greater than the concentration of water, sodium reacts with carbon dioxide and oxygen according to their molar ratios, as before, but the sodium and water reaction is omitted and no hydrogen is formed. The oxygen/water concentration effects on the sodium vapor reaction are based on experiments conducted at Hanford Engineering Development Laboratory (HEDL).

- (e) Hydrogen may react with oxygen next. The hydrogen would be generated by the sodium-water reaction, either in an inerted cell, with the hydrogen leaking into the air-filled containment building or in the building itself. The code neglects flammability limits and near-limit flame phenomena. The code covers the naturally

occurring hydrogen and oxygen reaction when sodium vapor accompanies the hydrogen leakage. Results of recent tests at HEDL show that this reaction occurs, and that hydrogen is completely consumed when the following conditions prevail:

- (1) The hydrogen and sodium vapor leak from a cell that is at least 500°F and has a sodium aerosol concentration of at least 5.5 g/m³.
- (2) The leakage goes into an air space that has an oxygen concentration of at least 11 percent.

In the code analysis, the hydrogen reaction is as follows:

<u>Hydrogen Recombination Reaction</u>	<u>Energy Release</u>
$2 \text{ H}_2 + \text{O}_2 \rightarrow 2 \text{ H}_2\text{O}$	52,425 Btu 1b H ₂

- (f) Last, sodium reacts with hydrogen to form sodium hydride. This situation occurs in a cell following a sodium spill that contacts concrete, and hydrogen is generated by the sodium-water reaction. The hydride reaction delays the escape of hydrogen from the cell. In the code analysis, this reaction is as follows:

<u>Sodium Hydride Formation</u>	<u>Energy Release</u>
$2 \text{ Na} + \text{H}_2 \rightarrow 2 \text{ NaH}$	915 to 1,010 Btu/lb Na

The sodium hydride, as a separate phase, or as dissolved in the sodium pool, is in equilibrium with the hydrogen in the atmosphere over the pool.

3.4 Differential Equations

The principal mathematical equations of the CACECO code are the five differential equations that express the mass and energy balances of each cell analyzed by the code. Since there are four cells analyzed by the code, there are four sets of these five equations. Each set of these differential equations includes three equations which describe the mass balances of the atmosphere, sodium pool, and water pool, and two equations which describe the energy balances of the atmosphere and pool. One set of these five equations is listed in the following pages. The nomenclature of these equations is informal. Usually the letter denoting each variable is the first alphabetic letter of the phenomenological name of that variable. For example, M represents Mass and I represents Inter data.

The simple Euler (predictor) method⁽³⁾ is used to solve the five differential equations for the conservation of cell mass and energy. This Euler method is a marching process which uses the cell conditions at the end of one step to calculate the various rates of change over the next step and thereby the cell conditions at the end of the next step.

The modified Euler (predictor-corrector) method⁽³⁾ is used to solve the two differential equations for the conservation of energy with respect to their terms for heat transfer from heat structures, ΣQ_{HS} . Unfortunately, these differential equations and their Euler methods of solution are lost from view in the following sections which cover the code operations in detail.

ATMOSPHERE MASS DIFFERENTIAL EQUATION

$$\frac{d}{dt} M_a = \text{atmosphere mass change in Cell J} \quad (3.1)$$

$$= I_{Na} + I_{N_2} + I_{O_2}$$

Input tables for sodium spray and gas injection

$$+ \sum_I L_{IJ} - \sum_K L_{JK}$$

Leakage from Cell I and to Cell K

$$- R_{Na + CO_2} - R_{Na + O_2} - R_{Na + H_2O}$$

Chemical reactions of sodium with carbon dioxide, oxygen, and water vapor

$$+ R_C + R_{Na_2O} + R_{H_2}$$

Chemical reaction products of carbon, sodium oxide, and hydrogen

$$- R_{Na + H_2} - R_{Na_2O + H_2O}$$

Chemical reaction of sodium with hydrogen and of sodium oxide with water vapor

$$- R_{H_2 + O_2} + R_{H_2O}$$

Chemical reaction of hydrogen and oxygen to make water vapor

$$+ R_{NaH} + R_{NaOH}$$

Chemical reaction product of sodium hydride and sodium hydroxide

$$+ \sum_N R_{CO_2} + \sum_N R_{H_2O}$$

Concrete releases of carbon dioxide and water vapor from N structures

$$- C + B$$

Vapor condensation and pool boiloff

SODIUM POOL MASS DIFFERENTIAL EQUATION

$$\frac{d}{dt} M_p = \text{sodium pool mass change in Cell J} \quad (3.2)$$

$$= I_{Na} \quad \text{Input table for sodium spill}$$

$$+ L_{IJ} - L_{JK} \quad \text{Sodium recycle from Cell I or to Cell K}$$

$$- R_{Na + CO_2} - R_{Na + H_2O} \quad \text{Chemical reactions of sodium with carbon dioxide and water vapor}$$

$$+ R_C + R_{Na_2O} \quad \text{Chemical reaction products of carbon and sodium oxide (actually carried in the atmosphere mass balance)}$$

$$- R_{Na + Conc} \quad \text{Chemical reaction with concrete}$$

$$+ \sum_N R_{CO_2} + \sum_N R_{H_2O} \quad \text{Concrete releases of carbon dioxide and water vapor from N structures}$$

$$+ C - B \quad \text{Vapor condensation and pool boiloff}$$

WATER POOL MASS DIFFERENTIAL EQUATION

$$\frac{d}{dt} M_w = \text{water pool mass change in Cell J} \quad (3.3)$$

$$= I_{H_2O} \quad \text{Input mass for water pool}$$

$$+ C - B \quad \text{Water condensation and pool boiloff}$$

ATMOSPHERE ENERGY DIFFERENTIAL EQUATION

$$\frac{d}{dt} U_a = \text{atmosphere energy change in Cell J} \quad (3.4)$$

$$= H_{Na} I_{Na} + H_{N_2} + O_2 I_{N_2} + O_2$$

Input tables for sodium spray and gas injection have enthalpy effects

$$+ \sum_I H_{IJ} L_{IJ} - \sum_K H_{JK} L_{JK}$$

Enthalpy leakage from Cell I to Cell K

$$+ Q_{Na + CO_2} + Q_{Na + O_2} + Q_{Na + H_2O}$$

Chemical reaction heat of sodium with carbon dioxide, oxygen, and water vapor

$$+ Q_{H_2 + O_2}$$

Chemical reaction heat of hydrogen and oxygen

$$+ \sum_N U_{CO_2} R_{CO_2} + \sum_N U_{H_2O} R_{H_2O}$$

Concrete releases of carbon dioxide and water vapor from N structures have energy effects

$$- U_f C + U_g B$$

Vapor condensation and pool boiloff have energy effects

$$+ Q_{FP}$$

Fission product decay power

$$+ \sum_N Q_{HS}$$

Heat transfer from N structures

$$- Q_{SC}$$

Space cooler

POOL ENERGY DIFFERENTIAL EQUATION

$$\frac{d}{dt} U_p = (\text{sodium or water}) \text{ pool energy change in Cell J} \quad (3.5)$$

$$+ H_{Na}^I Na + U_{H_2O}^I H_2O$$

Input table for sodium spill has enthalpy effects; input mass for water pool implies energy effect

$$+ H_{IJ}^L - H_{JK}^L$$

Sodium recycle from Cell I or to Cell K has enthalpy effect

$$+ Q_{Na + CO_2} + Q_{Na + H_2O}$$

Chemical reaction heat of sodium with carbon dioxide and water vapor

$$+ Q_{Na + H_2}$$

Chemical reaction heat of sodium with hydrogen

$$+ Q_{Na + Conc}$$

Chemical reaction heat of sodium with concrete

$$+ \sum_N U_{CO_2}^R CO_2 + \sum_N U_{H_2O}^R H_2O$$

Concrete releases of carbon dioxide and water vapor from N structures into sodium pool have energy effects

$$+ U_f C - U_g B$$

Vapor condensation and pool boiloff have energy effects

$$+ Q_{FP}$$

Fission product decay power

$$+ \sum_N Q_{HS}$$

Heat transfer from N structures

4.0 CACECO CODE ORGANIZATION

The main program unit is called CACECO. Each computer run begins in CACECO, either at 0.0 hour time for a new case or at some later time for a restarted case. The basic flow diagram is summarized in Figure 4.1 and the detailed flow diagram is given in Appendix A. The basic diagram shows three phases:

1. Data input and parameter initialization are done by calls to subroutines CINPUT, CCALC, and TINPUT. In turn, subroutine CINPUT calls subroutines CLZRO, HINPUT, and HPSET.
2. Initial steady-state temperatures are calculated for all heat structures of a new case by calls to subroutines CCALC and HCALC. Input of temperatures for any heat structure is done by call to subroutine TINPUT.
3. The transient is calculated by a call to subroutine CCALC. Note that subroutine CCALC used to be the main program unit of the code and is still involved in all three phases of code operation.

A case restart will automatically skip the Phase 2 calculation of initial steady-state temperatures for all heat structures, but new temperatures can be entered by card input which is read by subroutine TINPUT.

The input phase for a new case exercises CINPUT, HINPUT, and TINPUT, while the input for a restarted case usually skips HINPUT because heat structures are retained. The description and format for all input data are given in Appendix B. Generally the input data are repeated as labeled output before being modified for computational convenience. The sequence of data input is as follows:

1. Cards 1 to 100000 are read under the supervision of CINPUT. These cards enter (1) cell volume and initial conditions, accident effects, and concrete properties.

Main program unit

Data input and
parameter initialization

Steady state temperatures
in all heat structures

Transient calculations

Simple (predictor)
Euler method

Modified (corrector)
Euler method for
heat transfer terms

CACECO

CINPUT
HINPUT
TINPUT

HCALC
TINPUT

CCALC
Calculate parameter rates and their time increments: decay heat, sodium spray, sodium spill, sodium transfer, air injection, sodium-concrete reaction, space cooler operation, concrete water and carbon dioxide releases, atmosphere leakages and heat transfers to cells.

EMVBAL
Chemical reactions and first solution of energy, mass and volume conservation (balance) equations to predict cell conditions at end of time step.

HCALC
Solve transient heat conduction equations for new structure temperatures and new heat transfer rates to cells.

EMVBAL
Final solution of energy, mass and volume conservation (balance) equations to determine cell conditions at end of time step.

Output Data
Stop?

CCALC
OUTPUT
OUTSUM
Write heat structure temperatures and heat transfer rates, cell conditions and, periodically, two pages of summary data.

New
Time
Step

FIGURE 4.1. Basic CACECO Flow Diagram.

2. Cards with heat structure titles and Cards 101000 to 120406 are read under the supervision of HINPUT. These cards enter the titles, dimensions, volumetric heat generation parameters, and the indices to the materials and boundary conditions for all heat structures.
3. Cards 300001 to 500000 are read under the supervision of CINPUT. These cards enter the thermal properties for the materials which have been indexed in the heat structures. Also, these cards enter the heat transfer coefficients (for convective heat transfer) and the total interchange areas (for radiation heat transfer) which have been indexed in the boundary conditions of the heat structures.
4. Structure Cards 1 to 20 with temperatures needed to ease the calculation of initial steady-state temperatures are read by TINPUT in between Cards 500000 and 600000.
5. Structure Cards 1 to 20 with correct steady-state or transient temperatures are read by TINPUT in between Cards 600000 and 10000000.

Restart entails reading all of the data of labeled and blank common which were output in binary format at the conclusion of the preceding, successful run. This reading for restart is initiated by Card 1 of the sequence (1), above, and then new or modified input data can be entered to change the course of the transient analysis.

The transient phase exercises CCALC. This subroutine used to be the main program and much of its length and seeming complexity accumulated during the code development to accommodate more and more features of LMFBR accident phenomena. CCALC also controls the writing of output during the run and, at the end of a successful run, the binary output of all of the data of labeled and blank common for subsequent restart, CCALC evaluates and numerically integrates the mass and energy differential equations of Section 3.4. This numerical integration is a marching process which uses the simple (predictor) Euler method⁽³⁾ for most of the terms of Equations (3.1) through

(3.5). In this method, the cell conditions of concentrations, densities, pressures, temperatures, etc., at the end of the previous time step are used to calculate the mass and energy differential changes for the next time step. For example, the cell density and cell pressures at the end of the previous time step are used to calculate the leakrates between cells and the associated mass and energy differential changes in Equations (3.1) and (3.4) for the next time step. Figure 4.1 shows that this simple Euler method is completed by the first call to subroutine EMBVAL to predict the cell conditions at the end of the time step. The modified (predictor-corrector) Euler method⁽³⁾ is used with the terms for heat transfer from the structures, ΣQ_{HS} , of Equations (3.4) and (3.5). Figure 4.1 shows that this modified Euler method is completed by the second call to subroutine EMBVAL to correct the cell conditions at the end of the time step.

CCALC calls subroutine HCALC to solve the transient heat conduction equations over the time step, one equation for each heat structure. The boundary conditions for these heat structures use the cell atmosphere and pool temperatures from:

- a. the cell conditions at the end of the time step just preceding and
- b. the cell conditions at the end of the present time step as predicted by the first call to subroutine EMBVAL.

The boundary conditions are usually convection heat transfer, but can include radiation heat transfer to one to six boundary temperatures. The heat transfer coefficients for convective heat transfer and the total interchange areas for radiative heat transfer are both entered as input data and may be constant or dependent upon time, surface temperature, boundary temperature, average film (average of surface and boundary temperatures) temperature, or temperature difference. The typical total interchange area is entered as a constant and is based on assumptions of grey surfaces having constant areas, emissivities and geometrical view factor. The usual materials for heat structures have constant thermal conductivity and

volumetric heat capacity but concrete material, as noted previously, has these thermal properties dependent upon temperature. The solution method for the heat conduction equations was originally that of the CONTEMP code⁽¹⁾ and a summary of this method is presented in Appendix E. This solution method is implicit and is stable for any size of time step.

CCALC checks the convergence of the atmosphere energy, Equation (3.4), between the first (predicted) call to EMBVAL and the second (corrected) call to EMBVAL. Of course, as shown in Figure 4.1, the HCALC solution of the heat conduction equations and the associated heat transfer rates to the cell atmospheres and pools are the only changes to the atmosphere energy and the only reason that the convergence parameter CONV(1) can go out of the bounds of 5.0×10^{-5} to 5.0×10^{-3} . CCLAC checks the size of this convergence parameter and, if out of bounds, will call subroutine CHANGE to modify the size of the time step.

CCALC calls subroutine OUTPUT at intervals entered in Input Card Series 9001 to write out cell conditions at the end of the step. CCALC calls subroutine OUTSUM at intervals to write out two pages of summary data. This marching process continues with the next time step until:

- a. the final time of the case run is reached according to parameter TFNL which was entered on Input Card 1,
- b. the allowed computer time is reached as sensed by subroutine TCHEK (and local system routines) according to local usage, or else
- c. the CACECO code or the computer system sensed an error and halts the processing.

The code stops by (a) or (b), above, permit CCALC to write (binary) output of the data of labeled and blank common for subsequent restart.

5.0 CACECO CODE DESCRIPTION

The CACECO containment analysis code has the main program unit, also named CACECO, and 68 functions and subroutines. The flow diagram of the code is given in Appendix A along with a one-line description of each function and subroutine. This section describes the CACECO main program unit and the major subroutines.

Note that 11 of the 68 functions and subroutines are dependent upon the computer and computer system; these will be the main concern of the code user who must convert CACECO to another computer and system. Only 2 of these 11 functions and subroutines are so vital to the operation of the code that they must be converted by one means or another. Both of these two vital subroutines are written in FORTRAN, but they use CDC FORTRAN capabilities and functions which may not be immediately convertible to another computer. These two particular subroutines are the following.

1. CVIRDP reads mixed integer and real number data words which are arranged in free field format on input cards according to Appendix C. This subroutine reads nearly all of the input cards which are submitted to CACECO. These input data are described in Appendix B and the rules for entering numerical data are given in Appendix C. Note that comment cards (identified by the letter C or an asterisk [*] in Column 1) are simply copied to output and thereby provide an easy means to describe each case. The code user is urged to make free use of comment cards to provide permanent description of salient features of each case, particularly for cases done in sequence during a parametric study. CVIRDP reads the 80 characters of an input card and then examines each character in sequence, from left (Column 1 of the card) to right (Column 80), in order to determine the location, length, and kind of numeric data words on the card. The subroutine then prepares a FORTRAN format specification and re-reads the same card as mixed integer and real number data. The present version of CVIRDP uses the CDC FORTRAN

functions of DECODE and ERRSET and the CACECO supplied subroutine MOVE which, in turn, uses the CACECO supplied function FLDL.

- a. DECODE provides a memory core-to-core transfer of data similar to a formatted READ statement; DECODE reads from a string of code characters in a variable in-core, whereas the READ statement reads from an input device.
 - b. ERRSET enables the user to input data without the risk of termination when improper data are encountered.
 - c. MOVE moves characters from one computer word to another by using the FLDL function.
 - d. FLDL moves certain bits in one computer word to another by using the CDC FORTRAN functions of MASK and SHIFT.
2. FPTCK verifies that a number is "real" (a floating point number) and writes an error message when it is not. This simple check discovers many mistakes in the input data. The present version of FPTCK uses the CACECO supplied function FLD.
- a. FLD obtains certain bits in a computer word by using the CDC FORTRAN functions of MASK and SHIFT.

The 9 of the 11 functions and subroutines which are dependent upon the computer and system, but which may be skipped or bypassed in conversion to another computer, are listed below.

1. DOY obtains the run date from the computer. The present version uses the CDC FORTRAN subroutine DATE and re-formats the date to a form that is easier to read. DOY is called only by subroutine CINPUT.

2. ETIMEF obtains the computer processing time since the last call to the entry ETIME of this subroutine. The present version of ETIMEF uses the CDC FORTRAN subroutine SECOND.
3. FLD obtains certain bits in a computer word by using the CDC FORTRAN functions of MASK and SHIFT.
4. FLDL moves certain bits in one computer word to another by using the CDC FORTRAN functions of MASK and SHIFT.
5. ITLIMIT returns 1000 seconds as the maximum computer processing time permitted for each CACECO job. This function was peculiar to the CDC Cyber-74 system at HEDL. It obtained the job time limit from the "job card" which was submitted as the first card of any input card deck.
6. MOVE moves characters from one computer word to another by using the FLDL function.
7. RTCHEK is called by subroutine CCALC to check the remaining amount of computer processing time available. It calls subroutine TCHEK and uses the normal and non-normal return permitted in CDC FORTRAN.
8. TCHECK calls the ITLIMIT function, above, and the CDC FORTRAN subroutine SECOND.
9. TOD obtains the run time of day from the computer. The present version uses the CDC FORTRAN subroutine TIME and re-formats the time to a form that is easier to read. TOD is called only by subroutine CINPUT.

5.1 Main Program CACECO

The main program CACECO includes almost all of the FORTRAN variables and arrays which are used by the code. These variables and arrays comprise the labeled and blank common blocks which are clustered in the CDC computer

core for easy writing and reading of all data necessary for restart. The present arrangement makes two clusters of these common blocks: one for the labeled common blocks and the other for the blank common. The present code version has the following sizes for these clusters.

Labeled common, /CELL1/ to /LABLET/, 13,751 octal = 6,121 decimal words.

Blank common 14,526 octal = 6,486 decimal words.

The amount of data which are saved and returned for restart is 12,607 decimal words. The actual writing of these common block clusters is done in subroutine CCALC and the actual reading of these at restart is done in subroutine CINPUT. The only labeled common data blocks which are not saved for restart are as follows:

/RUNDA/, which provides temporary communication between CCALC and CINPUT.

/HYDRID/, which provides temporary communication between EMVBAL and NAHPRO.

/SAVED/, which provides temporary storage between FETCH and STORE.

The FORTRAN arrays for heat structure data occupy the majority of the data storage used by the present version of the code. These particular arrays fill the blank common block and the following labeled common blocks: /ZCALC/, /CHEAT/, and /RHEAT/. The two parameters which set the size of these arrays are II and III.

Parameter II denotes the maximum number of heat structures permitted and must include the four structures used to represent the sodium pool in each of the four cells. At present, $II = 24$, which means that 20 structures is the limit for the heat structures to represent the equipment, roofs, walls, and floors of the facility under analysis.

Parameter III denotes the maximum number of temperature node locations in any heat structure. At present, $III = 25$, which means that each structure can have two surface nodes and up to 23 interior nodes. Note that array AS has the dimensions of $AS (III + 1)$.

The following equation gives the amount of core storage established by these two parameters:

$$\text{Storage} = 147 \times II + 5 \times III + 7 \times II \times III \quad (5.1)$$

The present version with $II = 24$ and $III = 25$ has 7,853 decimal words of storage for the heat structures. Changing one or the other of these parameters to change the number and size of heat structures for an analysis is a common experience. Besides the arrays in the blank common block and the three labeled common blocks, which are obvious, there are the following statements which must be changed to keep the code error-free.

Parameter II: in CACECO, the data statement for II.

Parameter III: in CACECO, the data statement for III,

in CO2F, the dimension statement for ICS, and

in CWRF, the dimension statement for ICS.

The heat structures which represent the sodium pool can have up to five boundary temperatures in convection contact with each surface. For example, the top surface (temperature node 1) will be in convection contact with the cell atmosphere and can be in contact with one, two, three, or four more temperatures. The bottom surface (temperature node 7) can be in contact with one to five temperatures. This dimension can be changed to fewer or more by the following statements.

Pool heat structures: in CACECO, the labeled common block /PHEAT/, all arrays except NP1 and NPN,

 in HCALC, the DO 140 statement, and

 in HPSET, the DO 120 statement.

All heat structures can have up to six boundary temperatures in radiation contact with each surface. This dimension can be changed to fewer or more by the following statements.

Radiation contact: in CACECO, the labeled common block /RHEAT/, all arrays except NR1 and NRN,

 in HCALC, the DO 110 statement, and

 in HINPUT, the IF (IPOL6...) GO TO 920 statement.

CACECO calls the subroutines CINPUT, CCALC and TINPUT in the following sequence.

1. CINPUT, in order to read all input data from the case title card (denoted by either a dollar sign [\$] or an asterisk [*] in column 1) to CARD 500000.
2. CCALC with variable IN = 1, in order to initialize various parameters at the beginning of a new case and at the restart of the same case. More parameters are initialized at the beginning than at a restart.
3. TINPUT, in order to read temperature data which will aid the calculation of the steady-state temperatures in the heat structures. Each of these temperature input data cards has at least three numerical data words:

- (1st Word) structure number, an integer number denoted by I,
- (2nd Word) temperature node number, an integer number denoted by N, and
- (3rd Word) temperature, a real number denoted by T.

TINPUT stores the temperature in the following arrays:

US(N,I), the temperature at Node N of heat Structure I,

USM(N,I), the maximum temperature thus far attained at Node N of Structure I.

TINPUT also stores the temperature in one of the following two arrays, either

US1(I), the "left-hand" surface temperature at Node 1 if N = 1, or

USN(I), the "right-hand" surface temperature if N = 2 or more.

The important temperatures to enter at this time are the surface temperatures which may be used as boundary temperatures for other structures. The last card entered is Card 600000.

4. CCALC with variable IN = 2, in order to call HCALC to solve the heat conduction equations for the steady-state temperatures in the heat structures. These steady-state temperatures are stored in the arrays, US(N,I), USM(N,I), and US1(I) and USN(I), identified above.
5. TINPUT, in order to read temperature data which replace those stored in the arrays US(N,I), USM(N,I), US1(I) and USN(I). The temperatures entered at this time are the correct temperatures, either steady-state for a new case or transient for a restart, which the analysis is to use henceforth.

6. CCALC with variable IN = 3, in order to do the transient calculations. The code operation returns to CACECO at the end of these calculations, even when they terminate in error as detected by a CACECO subroutine.

A restarted case, denoted by the logical variable RESTAR = "TRUE," will skip the TINPUT call of Step (3) and the CCALC call of Step (4), above. When one case is finished, the processing returns to the CINPUT call of Step (1), above, to read in the next case which may be (a) a restart of the same case but with different input data which will modify the course of the accident scenario, (b) a new case, and (c) nonexistent.

5.2 Subroutine CINPUT

The first call to subroutine CINPUT (of the two or more calls of every CACECO run) sets up the restart parameters L1 through L4. Initially these parameters are the computer memory core locations of the first and last words of the clusters of labeled and blank common blocks which are saved and returned for case restart. CINPUT calls the CDC FORTRAN function LOCF to find these core locations. The parameters are as follows:

L1 is the core location of APT(1), the first word of the cluster of labeled common blocks.

L2 is the location of LABEL(6), the last word of this cluster.

L3 is the location of NON(1), the first word of blank common, and

L4 is the location of XS(III,II), the last word of blank common.

CINPUT writes these locations out, as a check for the user, and then redefines L2 to be the number of words in the cluster of labeled common and L4 to be the number of words in blank common. These parameters are stored in the labeled common block /RUNDA/ which is used only by CCALC and CINPUT. On a case restart, the first Card 1 of the restart card deck should command

CINPUT to read the data stored on disk file or magnetic tape. The actual FORTRAN read statements are as follows:

```
REWIND 7  
READ(7) (APT(J),J=1,L2)  
READ(7) (NON(J),J=1,L4)
```

The write statements, which are quite similar, are in CCALC. Note that the restart parameters L2 and L4 are so vital to this operation that they are not used for any other purpose in the CACECO code.

CINPUT initializes various parameters and arrays. As an example, the parameter NCRD3 is set to 40 as the limit of the number of calls to CVIRDP between the case title card and Card 9999 (or Card 10000 for a restart). This limit stops the convulsion when the subroutines get confused and attempt to re-read over and over again some card with some peculiar error. CINPUT calls CLZRO to initialize the arrays and parameters of cell conditions.

CINPUT then reads input cards in sequence for the first card having a dollar sign (\$) or an asterisk (*) in Column 1. CINPUT grasps this first card, no matter what it says, as the case title card and stores its information in the array HEAD(I) for use as the heading or top line of each page of output. This case title card is counted as the first card read by CINPUT.

The second card read by CINPUT, if it exists in the input card deck, enters new six character output labels for the cells. This second card is read under the format specification: `FORMAT(4X,A6,3(4X,A6))`. This second card was read by CINPUT upon command by Card 0 which, in turn, had been read by CVIRDP.

CINPUT calls CVIRDP to read all input data cards from Card 0 onward until either Card 9999 or Card 10000 is encountered. These are all numeric data cards in free field format. In this format, the numeric data words are separated by a comma and/or by one or more blanks. The numeric words can

be real (floating point) numbers or integer (fixed) numbers and part of the CINPUT supervisory process is the check by call to FPTCK that each real number word is indeed a real number. The first numeric word on each card is the card number and, in a card series, these card numbers are in monotonic sequence: 101, 102, 103, etc., for the Card Series 101 with decay power. With the following two restrictions, these input data cards can be in any sequence according to the needs or whims of the user.

- (a) Cards 2, 4, 6, and 8 are associated with Cards 3, 5, 7, 9, respectively. The code results are different when the odd card precedes, as compared to when the odd card follows its even card.
- (b) The cards of a card series must be together and in sequence.

The CINPUT supervisory process also verifies that each data card has the proper number of numeric data words. For example, each Card 1 must have four numeric words and each Card 2 must have six numeric words. CVIRDP counts the number of numeric words on each card and CINPUT checks that the count is proper. CVIRDP also processes the information on comment cards by immediately copying the information to output as soon as each card is encountered. These comment cards have an alphabet C or an asterisk (*) in Column 1. The user is urged to make good use of these comment cards for descriptions of each case and its unique features.

The CACECO code uses an input card series to enter each table of data. CINPUT calls TABLD to supervise the reading of each data table as soon as the existence of the table is noted by the card number on the first card of the series. Remember that in a card series, the card numbers are in monotonic sequence. Comment cards that describe a data table should follow the card series in order to be printed before or above the table in output. This unusual arrangement--and it may be unique--comes about because CINPUT, TABLD, and CVIRDP do not know how many cards are in each card series that enters a data table. The author believes that modern computers can count more quickly and more accurately than any user, and so uses the computer,

to count input data. TABLD reads (by calling CVIRDP) the card series, one card at a time, until a card is encountered with its card number out of sequence. This out-of-sequence card signals to TABLD, and in turn to CINPUT, that the card series was completed and the data table was stored and counted. Then CINPUT, or one of its ancillary subroutines, prints the data table. Thus, comment cards that follow the card series will be copied to output before the encounter of the out-of-sequence card that causes the output of the table itself. Of course, the out-of-sequence card may be the first card of another input card series and CINPUT will supervise its reading next.

Input Cards 2, 4, 6, and 8 are the even-number cards that enter the cell volume and initial atmosphere conditions for analysis Cells A, B, C, and D, respectively. Each of these even-number cards causes CINPUT to call INISHL to complete the calculation of the initial conditions of the respective cell. Cards 3, 5, 7, and 9 are the odd-number cards and each is associated with the preceding even-number card; e.g., Card 3 with Card 2,..., Card 9 with Card 8. In the input card data deck submitted to CACECO, the odd-number card may precede or follow its associated even-number card and this order makes a difference as outlined below.

- (a) Odd-number card precedes its associated even-number card; sodium or water may be spilled into the cell pool at the outset of the run and the cell atmosphere may be humid.
- (b) Odd-number card follows its associated even-number card; sodium may be added or removed from the cell atmosphere at the outset of the run and the sodium pool of the cell has the surface area entered on the card and may have a non-zero surface emissivity for radiation heat transfer to the cell atmosphere.

The effects of odd- and even-number card order were illustrated and checked with various examples in Debug Cases 1 through 8 which were presented in the Analytical Validation...⁽²⁾ In any case, CINPUT calls WRITO to print out the non-zero data which were entered on the odd-number card.

CINPUT calls INISHL to calculate the cell conditions from the data entered on Card 2, 4, 6, or 8 and the data already stored in MST(Z), UAT(Z) and FPT(ZP). Of course, the preferred method of entering data into these FORTRAN variables is to precede the even-numbered card with the associated odd-numbered card having the desired values for these variables. In the following, Z is the integer argument of INISHL, with allowable values of 1, 2, 3, or 4 to index the FORTRAN arrays for Cell A, B, C, or D, respectively. Also, ZP is an associated integer index with values of $ZP = Z + 1$. The outline of subroutine INISHL is as follows:

1. If MST(Z) is non-zero, then when:
 - (a) MST(Z) is positive, its value is stored in MPT(Z) as the mass of the sodium pool and when
 - (b) MST(Z) is negative, its absolute value is stored in MPWT(Z) as the mass of the water pool.

Also, UAT(Z) is stored in UPT(Z) as the energy of the pool (either sodium or water) and then MST(Z) and UAT(Z) are set to zero.

2. Note that the initial conditions of the cell were entered on the even numbered card in terms of its volume, VOLT(Z) or V, and the pressure, PTT(ZP) or P, temperature TAT(ZP) or T_a , and oxygen content, MOT(Z) or F_{O_2} , of the atmosphere that fills this volume.
3. If MOT(Z) is negative or zero, then the cell atmosphere is assumed to consist of nitrogen and sodium vapor. The call to SRCHPN provides sodium saturation properties at the cell pressure, in particular the energy at which a sodium pool would boil, ULBOL(Z), and the boiling temperature, TBOLT(Z). The call to NAPROP provides sodium saturation properties at the cell temperature; vapor pressure PSIA or P_{Na} , specific volume, VGN or V_{Na} , and internal energy, UCN or U_{Na} . The mass of sodium vapor, MGT(Z) or M_{Na} , is then:

$$M_{Na} = V/V_{Na} \quad (5.2.1)$$

The partial pressure of nitrogen and oxygen together (with the partial pressure of oxygen understood to be zero) is then the cell pressure less the sodium vapor pressure:

$$P_{N_2 + O_2} = P - P_{Na} \quad (5.2.2)$$

4. If MOT(Z) is positive, then the cell atmosphere is assumed to consist of nitrogen, oxygen, and water vapor. The call to SRCHPW provides water saturation properties at the cell pressure, in particular the energy at which a water pool would boil, ULBOL(Z), and the boiling temperature, TBOLT(Z). The call to WAPROP provides the water properties at the cell temperature: vapor pressure, PSIW or P_{H_2O} , specific volume, VGW or V_{H_2O} , and internal energy, UGW or U_{H_2O} . The relative humidity of the water vapor, FPT(ZP) or F_{H_2O} , is used in the calculation of water vapor mass:

$$M_{H_2O} = F_{H_2O} V/V_{H_2O} \quad (5.2.3)$$

and the water vapor partial pressure,

$$P_{H_2O} = F_{H_2O} P_{H_2O} \quad (5.2.4)$$

The partial pressure of nitrogen and oxygen together is then the cell pressure less the water vapor pressure:

$$P_{N_2 + O_2} = P - P_{H_2O} \quad (5.2.5)$$

5. The subroutine uses the Ideal Gas Law with the Ideal Gas Constant of $R = 10.7314 \text{ psia-ft}^3/\text{lb-mole-}^\circ\text{R}$. First, the number of moles of nitrogen and oxygen together is:

$$n_{N_2 + O_2} = P_{N_2 + O_2} V/R(T_a + 459.67) \quad (5.2.6)$$

The number of moles of oxygen is then

$$n_{O_2} = n_{N_2} + 0.2 F_{O_2} \quad (5.2.7)$$

and the number of moles of nitrogen is

$$n_{N_2} = n_{N_2} + 0.2 - n_{O_2} \quad (5.2.8)$$

The mass of nitrogen is $M_{N_2} = 28.016 n_{N_2}$ the mass of oxygen is $M_{O_2} = 32.000 n_{O_2}$ and these are stored in MNT(Z) and MOT(Z), respectively.

6. The subroutine uses statement functions to evaluate the nitrogen energy, U_{N_2} , and oxygen energy U_{O_2} , at the cell temperature. The cell atmosphere energy, UAT(Z) or U, is the sum of the energies of the constituents (with some understood to be zero):

$$U = M_{Na} U_{Na} + M_{N_2} U_{N_2} + M_{O_2} U_{O_2} + M_{H_2O} U_{H_2O} \quad (5.2.9)$$

7. The mass and energy of the pool, stored in (1), above, for either the sodium pool or water pool, are corrected by subcontracting the mass of vapor and its energy.
8. INISHL calculates and stores other data for all atmosphere constituents. For example, with sodium vapor, these data include the number of moles, MLST(Z), and the mass fraction, RATIO(3,ZP). Also, the subroutine calculates the atmosphere density, RHOT(ZP).

CINPUT uses the FORTRAN-computed GO TO statements, with the variable based on the input card number, in order to transfer to the proper subroutine paragraph for processing. There, usually, CINPUT will call one or more ancillary subroutines to read, store, print, check, and convert (for computational convenience) the data on an input card or card series. These ancillary subroutines are summarized in alphabetic order.

HINPUT reads and processes the input data for all heat structures. It is described in Section 5.3, below.

HPSET sets up the heat structure parameters for the structure representing the sodium pool in each cell having a sodium pool area which was entered on Card 3, 5, 7, or 9. It is described in Section 5.3, below.

MONOPR checks the pressure (or time) arguments of a data table to verify that these arguments are monotonic increasing values.

MONOTM checks the time arguments of a data table to verify three things:

1. the first time argument is zero (i.e., 0.0 hours),
2. the remaining time values are monotonic increasing, and,
3. the last time argument is greater than or equal to the final time, TFNL, entered on Card 1 of the case. Incidentally, the time arguments for outside air conditions entered on Card Series 1601 need only to cover from 0.0 hours to 24.0 hours, one day.

PGCK checks the number of lines available, and skips to the top of the next output page, if necessary, to print the whole data table.

TABLC processes the space cooler data of input Cards 1700, 1800, 1900, and 2000. It is described later in Section 5.4 in connection with space cooler calculations.

TABLD supervises the reading and then stores the data table entered by each input card series. It checks that each card was read by CVIRDP without error and that the card has the proper number of numeric data words. Also, it checks each card of the card series for having its card number in sequence and that the values entered into the data table are real numbers.

TABLH is a modification of TABLD to supervise the reading and store the data tables entered by input Card Series 410001, 420001, ..., and 490001. These tables have heat transfer coefficients for convection heat transfer or total interchange areas for radiation heat transfer as a function of time or temperature. The second numeric data word of the first card of these card series is an integer that indicates the kind of data table that follows.

The allowable kinds are listed for these card series in Appendix B and in both CINPUT and TABLH. Note that Kind 4 is effectively the same as Kind 3 because the FORTRAN variable FACTOR is set to value 1.0 in subroutine CCALC: (a) just before the calculation of steady-state temperatures in the heat structures and again (b) just before beginning of the transient calculations.

TABLLC processes the leakage coefficient data tables entered on input Card Series 11, 21, ..., and 91.

TABLN processes the leakage nozzle data entered on input Cards 10, 20, ..., and 90. This subroutine is described later in Section 5.5 in connection with leakage calculations.

TABLNA processes the air injection data table entered on input Card Series 1201, 1301, 1401, and 1501.

TABLSA processes the sodium spray data table entered on input Card Series 401, 501, 601, and 701.

TABLSP processes the sodium spill data table entered on input Card Series 801, 901, 1001, and 1101.

TIMEDP processes the decay power data table entered on input Card Series 101. This subroutine is described later in Section 5.5 in connection with decay power calculations.

ZROPLS checks the heat transfer coefficients of a data table to verify that each has a positive value, or, at least, a zero value. This subroutine prints a notice at each zero value encountered.

CINPUT continues to read (via CVIRDP) cell data, accident effects, concrete properties, etc. until either Card 9999 or Card 10000 is encountered. These latter cards signal the end of the cell data and general tables and CINPUT checks the existence of certain tables and combinations thereof. For example, if I27 is zero, then Card Series 9001 was not entered, there are no time step and print controls stored in Array TSPC and this is a fatal error. On the other hand, if I14, I15, I16, and I17 are all zero, then Card Series 401, 501, 601, and 701 were not encountered, there are no sodium spray data stored in Arrays SATA, SATB, SATC, and SATD, and subroutine CCALC can skip the paragraph that evaluates sodium spray effects. CINPUT checks and sets many indices so that subroutine CCALC can skip paragraphs that have zero effect.

Card 9999 commands CINPUT to call subroutine HINPUT to read card data which describe NSL (entered on Card 1) number of heat structures. Then, CINPUT calls HPSET to set up the heat structures for the sodium pool in each cell. HINPUT and HPSET are described in Section 5.3. Card 10000 signals CINPUT to skip HINPUT and HPSET because (a) there are no heat structures used in the case or (b) the heat structure data read earlier in a previous run of this case are still suitable without modification.

CINPUT reads the tables of data common to all heat structures which are entered on Card Series 300001 through 490001. CINPUT had set parameter NCRD4 to 13 as the limit of the number of calls to CVIRDP for these tables. Again, this limit stops the convulsion when the subroutines get confused and repeatedly attempt to read a card showing some peculiar error. Card 500000 halts the card input supervised by CINPUT and then CINPUT checks that none of the cells described by Cards 2, 4, 6 and 8 have negative or zero volumes.

Most users, including the author many times, have expressed anger and frustration at the seemingly innocuous and inoffensive card punch and data entry errors that choke CINPUT and its ancillary subroutines. Actually, these subroutines perform only the most elementary checks on the input data and the user is advised, encouraged, and urged to scrutinize his data, even after it is acceptable to CINPUT, even after it has been used in a previous case, even after the user is sure that the data are perfect.

5.3 Subroutines HINPUT and HPSET

The user must count the number of heat structures whose descriptions are to be read following Card 9999. This count of heat structures, FORTRAN variable NSL, is entered on input Card 1, and can be of number 0 (zero) to 20 in the present version of the code. The count must be correct because the penalty for error may be that either HINPUT or CINPUT will choke as the input cards are read.

Subroutine HINPUT reads the input data for NSL number of heat structures, each structure being described in a set of six or more input data cards. Each card of the set has different data as described in Appendix B. The first card of each set is the title card which enters a line of 80 characters of alphanumeric information which is read by HINPUT and stored in array TLHS. This title line is printed to identify the data output for each structure. The remaining data cards of each set are read by subroutine CVIRDP under the supervision of HINPUT. This supervision includes check of data entered on each card, such as the following:

1. The six-digit card number includes the two correct digits for heat structure number and the three correct digits for data identification. For example, Card 1NN100; the NN denotes the two digits for heat structure number and 100 identifies the card with node locations.
2. The card entered the proper number of numeric data words.

3. Each numeric word was the proper kind of number, either integer or real, as specified for the identified data.
4. The numeric values are acceptable; for example, the following:
 - a. The number of temperature node points, variable NN, which is stored in array NON, is within the range of two to III; the upper limit being the FORTRAN dimension of the arrays that hold node point data.
 - b. The heat structure geometry, variable IGEOM, is either planar, cylindrical, or spherical.
 - c. The heat transfer surface area factor, stored in array ARA, is greater than zero. This factor is identically the area of planar structures and the height of cylindrical structures.
 - d. Etc.

Input data reading and processing continues for the remainder of the heat structure data in an attempt to discover all data errors in one pass. Unfortunately, however, the return from subroutine ERRWR may be inappropriate and the entry of the remaining heat structure data may result in numerous error messages of which only the first is valid.

Heat structure temperature node locations are illustrated in Figure 5.1. The input data locate x_1 , the left-hand boundary (the inside radius of cylindrical and spherical structures), x_N , the right-hand boundary (the outside radius of cylindrical and spherical structures), and the interfaces of regions in between these boundaries.

FIGURE 5.1. Node Point Layout.

Location x_1 of the figure is entered on structure Card 1NN000 and the locations of the successive interfaces and location x_N are entered on Card Series 1NN100, 1NN101, 1NN102, etc. HINPUT checks these entered locations for being in monotonic increasing order and then calculates for node N of structure NN, and stores in array RS(N,NN), and temporarily in array AS(N), the locations for all nodes of the structure. These locations are then used to calculate and store terms in the following FORTRAN arrays which are used in the steady-state and transient solutions of the heat conduction equation as outlined in Appendix E.

SR1(NN)	Surface geometry weight of left-hand boundary
SRN(NN)	Surface geometry weight of right-hand boundary
SS(N,NN)	Surface gradient weight
VS(N,NN)	Left volume gradient weight
WS(N,NN)	Right volume gradient weight
XS(N,NN)	True volume between node N and Node N + 1

The solution of the heat conduction equation includes volumetric heat generation and the code assumes that this heat generation has a space dependence and a time dependence which are separable. The space dependence

is described in Card Series 1NN300, 1NN301, 1NN302, etc. The description entered on this card series may be as follows:

1. Constant, perhaps zero throughout the structure,
2. Constant in each region, but different for successive regions, or else
3. Varying from node to node.

This space dependence is calculated and stored in the following FORTRAN array:

YS(N,NN) Volumetric heat distribution

The boundary conditions for each heat structure are entered on Card Series 1NN400, 1NN401, 1NN402, etc. Each card enters conditions for the left-hand surface and for the right-hand surface. The conditions for each surface will cover the 1 path of convection heat transfer to some boundary (or bulk) temperature and may include the 0, 1, 2, ..., 6 paths of radiation heat transfer to different boundary temperatures. The boundary conditions for each path of convection and radiation heat transfer are specified by two integer indices. These indices are described below along with the FORTRAN arrays used to store their values for heat structure number NN.

<u>Index</u>	<u>Description</u>	<u>Left-Hand Surface</u>	<u>Right-Hand Surface</u>
first	Index to coefficient for convection heat transfer	BH1(NN)	BHN(NN)
	Index to total interchange area for path I of the radiation heat transfer	RBH1(I,NN)	RBHN(I,NN)
second	Index to boundary temperature for convection heat transfer	BT1(NN)	BTN(NN)
	Index to boundary temperature for path I of the radiation heat transfer	RBT1(I,NN)	RBTN(I,NN)

The code user should note the following idiosyncrasies of the heat transfer coefficients and total interchange areas.

1. When there is convective heat transfer between two structures of different areas, say A_1 and A_2 , for example, then each structure should index a different heat transfer coefficient, say h_1 and h_2 , respectively, such that the area-coefficient products are the same. That is, $A_1 h_1 = A_2 h_2$. If these products are not the same, then the heat transfer from Structure 1 to Structure 2 will differ in magnitude and sign from the heat transfer from Structure 2 to Structure 1.
2. When there is radiation heat transfer between any two structures, then both structures should index the same total interchange area.

Subroutine HPSET stores data in FORTRAN variables and arrays as if the sodium pool in each cell had been described as a heat structure by card input data. The code has four analysis cells (e.g., Cell A, etc.) and the sodium pool, if it is to exist at some time during the CACECO analysis, must be described by having its pool surface area entered on an odd number card, e.g., the area on Card 3 for cell A, the area on Card 5 for Cell B, etc. HPSET checks these areas, stored in FORTRAN array APT, and sets up the sodium pool heat structure for each cell having a pool area. This pool heat structure will have planar geometry, seven uniformly spaced temperature nodes in one region, and appropriate boundary conditions. The material properties and structure thickness will depend upon pool temperature and volume and will be calculated later as needed during the transient by subroutine HPOOL.

These sodium pool heat structures have little use in the present version of the CACECO code. The pool structures were installed along with the presently inactive subroutine POOLR in order to provide capability to evaluate heat and mass transfer processes between the sodium pool and the cell atmosphere. For example, POOLR would calculate the following items of natural convection phenomena:

- a. the heat transfer coefficient between the pool and atmosphere which would be stored in the FORTRAN array HPA,
- b. the sodium evaporation rate from the pool to the atmosphere through a stagnant nitrogen film, and
- c. the oxygen transfer rate from the atmosphere to the pool surface through a stagnant nitrogen film.

The oxygen transfer rate was taken to be the sodium-oxygen chemical reaction rate which is also called the sodium burning rate. The models of the sodium burning rate and of the sodium pool heat structure were adapted from the models of the SOFIRE-II code.⁽⁴⁾ POOLR is inactive and these pool heat structures have little use because the present version of the code assumes that: (1) the pool and atmosphere are at the same temperature and (2) the atmosphere is saturated with the pool fluid, either sodium or water. These assumptions simplify the evolution of the heat and mass transfer processes between the pool and atmosphere and thereby speed the operation of the code. The hurried reader is advised that he can skip the rest of this section on HPSET.

There are two conventions used in the CACECO code for indexing the sodium pool heat structure data.

1. The first convention stores sodium pool heat structure data at the top end of the FORTRAN arrays used by all heat structures. In particular, the number of heat structures entered on Card 1 and whose descriptions were read in by HINPUT is the FORTRAN variable NSL. These heat structures of concrete walls, heat exchanger equipment, etc., are numbered: 0 (if none exist), 1, 2, ..., NSL with the limit of NSL being 20 in the present code. The sodium pool heat structures are then numbered as follows:

NSL + 1 for the sodium pool in Cell A,
NSL + 2 for the sodium pool in Cell B,
NSL + 3 for the sodium pool in Cell C, and
NSL + 4 for the sodium pool in Cell D.

These heat structure numbers (indices) are used in FORTRAN arrays already described: SR1(NN), SRN(NN), SS(N,NN), etc.

2. The second convention stores the boundary conditions for the sodium pool heat structures in special FORTRAN arrays wherein the cell pools are numbered as follows:

1 for the sodium pool in Cell A,
2 for the sodium pool in Cell B,
3 for the sodium pool in Cell C, and
4 for the sodium pool in Cell D.

The sodium pool heat structures differ from the other heat structures because the sodium pools can have 1, 2, ..., 5 paths of convection heat transfer from each surface, whereas the other heat structures can have only one path. Of course, the sodium pool and the other heat structures can each have 0, 1, 2, ..., 6 paths of radiation heat transfer to each surface. The boundary conditions for each path of convection and radiation heat transfer are specified by two integer indices. These indices are described below, along with the FORTRAN arrays used to store the conditions for the pool heat structures. In these arrays, NC denotes the cell pool with values of 1, 2, 3 and 4, while NH denotes the heat structure with values of NSL + NC.

<u>Index</u>	<u>Description</u>	<u>Top (left-hand) Surface</u>	<u>Bottom (right-hand) Surface</u>
first	Index to coefficient for path K of the convection heat transfer	PBH1(K,NC)	PBHN(K,NC)
	Index to total interchange area of path I of the radiation heat transfer	RBH1(I,NH)	RBHN(I,NH)
second	Index to boundary temperature for path K of the convection heat transfer	PBT1(K,NC)	PBTN(K,NC)
	Index to boundary temperature for path I of the radiation heat transfer	RBT1(I,NH)	RBTN(I,NH)

The code assumes that the top surface of the sodium pool enjoys convection heat transfer with the atmosphere of the cell. Therefore, the first path for convection heat transfer has the coefficient indexed as 110 in arrays PBH1(1,1), PBH1(1,2), PBH(1,3) and PBH(1,4) for the four pools and has the boundary temperature indexed as follows:

<u>Index</u>	<u>Array</u>	<u>Meaning</u>
101	PBT1(1,1)	atmosphere of Cell A
104	PBT1(1,2)	atmosphere of Cell B
107	PBT1(1,3)	atmosphere of Cell C
110	PBT1(1,4)	atmosphere of Cell D

The user may specify that the top surface of the sodium pool enjoys radiation heat transfer with the cell atmosphere by entering a positive value for pool emissivity (say e_1) on the same odd-number card (say Card 3) that entered the pool surface area (say A_1 for the pool of Cell A). HPSET calculates a total interchange area of $A_1 e_1$ and stores this in FORTRAN array HTC at index $95 + NC$, then stores this index ($95 + NC$) for the first radiation heat transfer path in RBH1(1,NH), and stores the boundary temperature index of 101, 104, ..., 110 in RBT1(1,NH) for the specified cell.

The real worth of subroutine HPSET is its search of the boundary conditions of the NSL number of other heat structures to set up the boundary conditions for the inverse heat transfer from the sodium pool heat structures. HPSET searches for all convection paths and for all radiation paths because each pool surface may be connected by none, one, or more than one path of heat transfer. It is automatic, and it works.

5.4 Subroutine CCALC

Subroutine CCALC used to be the main program of the code and much of its length and seeming complexity accumulated during the CACECO development to accommodate more and more features of LMFBR accident phenomena. CCALC evaluates and numerically integrates the mass and energy differential equations of Section 3.4. Indeed, the CCALC description that follows will evaluate each term of these differential equations. CCALC also controls the writing of output during the run, and at the end of a successful run, the binary output of all of the data of labeled and blank common for subsequent restart.

CCALC is called three times by the CACECO main program during the first run of each case and two times during each restarted run thereafter. CCALC has two FORTRAN arguments, IN and RESTAR. The values of these arguments and the CCALC operations are summarized below.

<u>FORTTRAN Arguments</u>			<u>CCALC Operations</u>
<u>Call</u>	<u>IN</u>	<u>RESTAR</u>	
(1)	1	(either)	Reset locators (J11, etc.) for data tables of accident effects, evaluate sodium pool parameters (used in the inactive subroutine POOLR), set up logical switches for skipping leakage calculations, and zero array SUMMRY, which is used to store 116 lines of cell summary data.
		TRUE	(means case restart, return to CACECO)

<u>Call</u>	<u>IN</u>	<u>RESTAR</u>	<u>CCALC Operations</u>
		FALSE	Initialize cell variables not already set by CINPUT calls to CLZRO and INISHL, zero arrays used to store leakage data, and set up variables in preparation for calculating steady-state temperatures in the heat structures.
(2)	2	FALSE	Call HCALC to solve the heat conduction equation for the steady-state temperatures in the heat structures, then print these temperatures and the initial cell conditions.
(3)	3	FALSE	Call CWRP to calculate the concrete water release and call CO2F to calculate the concrete carbon dioxide release from the heat structures listed in input Card Series 2601 and 2801, respectively. These releases are based on the heat structure temperatures which were either (a) calculated by the call to HCALC, above, or (b) read by TINPUT.
		(either)	Write heat structure descriptions, cell conditions and heat structure temperatures to the binary file PTAPE for subsequent use in report summaries (made by the CCLIST code), plots (made by the CCPLLOT code), and further accident analyses by the HAA-3C code ⁽⁵⁾ and the COMRADEX-H code. ⁽⁶⁾ Enter the FORTRAN code loop that evaluates and integrates the mass and energy differential equations of Section 3.4.

CCALC will continue the FORTRAN code loop until the loop is broken by one of the following events.

1. Normal stop due to conditions that turn the FORTRAN variable STOP to TRUE. The normal stop will write labeled and blank common data to the binary file DTAPE for subsequent restart. The normal stop conditions are:
 - a. The analysis time, FORTRAN variable T, equals or exceeds the run final time, FORTRAN variable TFNL, which was entered on input Card 1.
 - b. The computer run time, evaluated each time step by the call to subroutine RTCHEK, exceeds 990 seconds.
2. Error halt due to CCALC or one of its ancillary subroutines sensing an error that turns the FORTRAN variable ERROR to TRUE. An error diagnostic is printed by subroutine ERRWR but there is no binary write of data for subsequent restart. A typical error might be exceeding the differential pressures of some leakage coefficient data table.
3. Code crash when the computer system senses a job time overrun or an arithmetic error. Computer sensed errors are most unusual but "address out of range" and "infinite operand" have been encountered.

The operations of the FORTRAN code loop are summarized below. Hereafter, the FORTRAN variables and arrays will be denoted by capital letters, without further identification as such.

CCALC begins the FORTRAN code loop by incrementing NSTEP, the number of time steps taken since the beginning of the run, and T, the analysis time. Various arrays are shifted from present time to old time: TAT to TATO with the cell atmosphere temperatures, and TPT to TPTO with the cell pool temperatures. Print switches are set: SW4 to FALSE if cell conditions are to be printed and SW3 to FALSE if heat structure data are to be printed. The analysis is now at TOLD or t, taking a time step DT or dt to the new time T or t + dt.

Accident effect data tables are skipped if their indices (I1 through I38 and N1 through N20) show that the tables are empty. For example, if index I11 is zero, there are no decay power data in array DECAYP because there was no input of Card Series 101 and CCALC skips the evaluation of decay power and the apportionment of this power to the cell atmospheres and pools. Likewise, if index I14 is negative, there are no sodium spray data in arrays SATA, SATB, SATC and SATD because there was no input of Card Series 401, 501, 601, and 701 and CCALC skips the evaluation of these spray effects. However, if index I14 is zero, there are no sodium spray data in array SATA but indices I15, I16 and I17 have to be checked in turn for spray data in arrays SATB, SATC and SATD, respectively.

Accident effect data tables are evaluated at the midtime ST01, halfway between TOLD and T. ST01 is a labeled common variable used by subroutines DECPOW and TABLU as the argument for table lookup and interpolation. Subroutine DECPOW performs a log-log interpolation of the decay power data stored in array DECAYP. These data had been input on Card Series 101 and read by subroutine TIMEDP (an ancillary of CINPUT). TIMEDP had checked the time data by using subroutine MONOTM, had replaced the first time value (entered as 0.0 hours) by the value 1×10^{-6} , and then had stored the logarithms of all time and power values in array DECAYP for future use by DECPOW. The decay power at time ST01 is POWIN or Q'_{FP} .

Subroutine TABLU performs a linear interpolation of each accident effect data table for the first dependent variable and stores the interpolating factor ST02. Subroutine TABLG uses the factor ST02 to perform the interpolation for the second, third, and other dependent variables. For example, if index I12 is positive, there are data in array APABCD from input Card Series 201 and TABLU looks up this table and interpolates the apportioning factor TEMP or F_{FP} for the decay power going into the Cell A atmosphere. Then TABLG interpolates the respective factors for the power going into the Cell B atmosphere, Cell C atmosphere, and Cell D atmosphere. The product of decay power and apportioning factor is the CCALC evaluation of the following portion of the differential equation for atmosphere energy, Equation (3.4):

$$\frac{d}{dt} U_a = Q_{FP} \quad (5.4.1)$$

Note that the actual FORTRAN statements use temporary variables PVAL and ST03 to perform the following mathematical operations in this evaluation.

$$(PVAL) = Q'_{FP} \, dt \quad (5.4.2)$$

$$(ST03) = Q'_{FP} \, dt \, F_{FP} \quad (5.4.3)$$

$$U'_a = U_a + dU_a = U_a + Q_{FP} \, dt = U_a + Q'_{FP} \, F_{FP} \, dt \quad (5.4.4)$$

The new atmosphere energy, U'_a , replaces the former energy value in the atmosphere energy array UAT. The writer asserts the equivalence of Equations (5.4.1) and (5.4.4) and will continue this description of CCALC as if the mathematical operations were like Equations (5.4.1). If index I13 is positive, there are data in array PPABCD from input Card Series 301 and the result is the evaluation of the following portion of the differential equation for pool energy, Equation (3.5):

$$\frac{d}{dt} U_p = Q_{FP} \quad (5.4.5)$$

CCALC sums the increments of decay heat going into the atmosphere (array DHAT) and pool (array DHPT) of each cell and prints these sums along with other cell data.

CCALC calls subroutine SATIN to process the sodium spray tables in arrays SATA, SATB, SATC, and SATD which were entered on Card Series 401, 501, 601 and 701 for Cells A, B, C, and D, respectively. Each table includes the sodium spray rate, I_{Na} , and the sodium enthalpy, H_{Na} . The latter defines whether the spray is liquid, vapor, or a two-phase mixture. The code neglects the liquid spray phenomena of spray droplet dynamics, particle evaporation, and particle burning. The CACECO model of the spray simply

mixes the spray homogeneously with the cell atmosphere. The result is the evaluation of the following portions of the differential equations for the mass and energy of the cell atmosphere, Equations (3.1) and (3.4):

$$\frac{d}{dt} M_a = I_{Na} \quad (5.4.6)$$

$$\frac{d}{dt} U_a = H_{Na} I_{Na} \quad (5.4.7)$$

CCALC calls subroutine SPTIN to process the sodium spill tables in arrays SPTA, SPTB, SPTC, and SPTD which were entered on Card Series 801, 901, 1001, and 1101 for Cells A, B, C, and D, respectively. Each table includes the sodium spill rate, I_{Na} , and its temperature, T_{Na} , the latter of which can be either positive (+) or negative (-) as a control. The positive spill temperature causes SPTIN calculations that increase the cell volume, array VOLT, by the volume of the spill increment and the negative sign causes SPTIN to skip these calculations. The result is that the positive spill temperature increases the cell volume by adding the spill volume as if the spill came from a tank located inside the cell; the negative spill temperature pressures the cell because the atmosphere is squeezed into a smaller net volume. Hereafter, T_{Na} denotes the absolute value of the spill temperature. Except for the above volume control, the CACECO model of the spill simply mixes the liquid spill with the cell sodium pool. The code adjusts the node point temperatures of the sodium pool heat structure, on the basis that the products of sodium mass and temperature are additive. That is, the product of the spill mass and temperature, $I_{Na} dt T_{Na}$, is additive to the product of the pool mass and temperature, $M_p T_p$, to determine the new mass of the pool, $M_p + I_{Na} dt$, and its temperature, T_p . With the heat structure number NH and its nodes as number N, this adjustment modifies pool as the arrays US(N,NH), US1(NH) and USN(NH). SPTIN calls subroutine NAPROP to determine the liquid specific volume and enthalpy, H_{Na} , at the spill temperature. The result is the evaluation of the following portions of the differential equations for the mass and energy of the cell sodium pool, Equations (3.2) and (3.5):

$$\frac{d}{dt} M_p = I_{Na} \quad (5.4.8)$$

$$\frac{d}{dt} U_p + H_{Na} I_{Na} \quad (5.4.9)$$

CCALC calls subroutine SPRCL to process the sodium recycle instructions of the array RECYL which were entered on Card 2100. This subroutine is the preferred way to transfer sodium from the pool of one cell to the pool of another cell (the alternative way uses a negative spill in one cell simultaneous with a positive spill in another cell). The transfer is made as if the sodium from the first cell overflowed a dam or weir into the second cell, regardless of cell temperatures and temperatures. This code capability was provided in order to investigate the following post-accident heat removal (PAHR) accident scenario:

1. Core debris melts through the bottom of the reactor vessel and debris and sodium spill into the reactor cavity.
2. Core debris decay power and sodium-concrete chemical reaction energy boils sodium from the pool in the reactor cavity.
3. The sodium vapor passes through ducts into an adjacent inert gas cell.
4. The sodium vapor condenses on the roof and walls and the condensate accumulates on the floor of the inert gas cell, forming a pool.
5. Sodium condensate from the pool of the inert gas cell recycles back to the reactor cavity for reboiling.

The overall effect of this scenario is to keep the core debris covered with sodium in the reactor cavity while spreading the decay power and chemical energy throughout the concrete roofs, walls, and floors of the cavity and inert gas cell. The instruction table of array RECYL includes the index of the source cell (1, 2, 3, and 4 for Cell A, B, C, and D, respectively), the

index of the destination cell, and the overflow mass setting for the source cell pool. At each call, subroutine SPRCL transfers all of the source cell pool that exceeds the overflow mass setting, L_{JK} . Of course, no sodium is transferred unless the source pool exceeds the setting. The code adjusts the node point temperatures of the sodium pool heat structure of the destination cell on the basis that the products of sodium mass and temperature are additive (the process described for subroutine SPTIN, above). The specific energy, U_J , of the transferred sodium is taken from the array ULT. SPRCL sums the increments of transferred sodium and energy and stores these sums for printing along with other cell data. The results are the evaluations of the following portions of the differential equations for the mass and energy of the sodium pool, Equations (3.2) and (3.5), both in the source cell J and in the destination cell K:

<u>Source Cell</u>	<u>Destination Cell</u>	
$\frac{d}{dt} M_{PJ} = - L_{JK}$	$\frac{d}{dt} M_{PK} = + L_{JK}$	(5.4.10)

$\frac{d}{dt} U_{PJ} = - H_J L_{JK}$	$\frac{d}{dt} U_{PK} = + H_J L_{JK}$	(5.4.11)
--------------------------------------	--------------------------------------	----------

CCALC calls subroutine NATIN to process the gas injection tables in arrays NATA, NATB, NATC, and NATD which were entered on Card Series 1201, 1301, 1401 and 1501 for Cells A, B, C and D, respectively. Each table includes the gas injection rate, $I_{N_2 + O_2}$, which can be either positive or negative as a control, the injection temperature, $T_{N_2 + O_2}$, and the oxygen content of this injected gas, F_{O_2} . The positive injection rate causes NATIN calculations that use this rate, as the reader might expect, but the negative injection rate causes a special calculation which injects enough gas to dilute the hydrogen concentration of the cell to $F_{H_2} = 0.04$ or 4 mole percent. Of course, in this special calculation, there is no gas injection unless the hydrogen concentration is greater than 4 percent. The user is warned of the potential for trouble upon restart when the hydrogen concentration is much greater than 4 percent, the special NATIN calculation will dilute the hydrogen in the first restart step. For example, if the hydrogen

concentration were 9 percent, the atmosphere mass and pressure would both increase in one step by a factor of about 9/4 and the accompanying leakage and temperature effects could upset the code operation. The special NATIN calculation with negative injection rate determines the hydrogen concentration by the ratio of the number of moles of hydrogen, n_{H_2} , to the total moles of gas in the cell, n_{tot} :

$$F_{H_2} = n_{H_2} / n_{tot} \quad (5.4.12)$$

The number of moles of gas injected to dilute the hydrogen concentration down to 4 percent is then:

$$n_{N_2 + O_2} = n_{tot} (F_{H_2} - 0.04)/0.04 \quad (5.4.13)$$

Also, the special calculation assumes that the average molecular weight of the injected gas is $w_{N_2 + O_2} = 28.85264$, as if the gas were air. The ordinary NATIN calculation with positive injection rate determines the average molecular weight of the injected gas by the composition formula,

$$w_{N_2 + O_2} = 28.016 (1 - F_{O_2}) + 32.0 F_{O_2} \quad (5.4.14)$$

and the number of moles of gas injected during the time step by the product:

$$n_{N_2 + O_2} = I_{N_2 + O_2} dt w_{N_2 + O_2} \quad (5.4.15)$$

Hereafter the ordinary and special calculations are the same. The mass of nitrogen injected is:

$$I_{N_2} dt = n_{N_2 + O_2} 28.016 (1 - F_{O_2}) \quad (5.4.16)$$

and the mass of oxygen injected is:

$$I_{O_2} dt = n_{N_2 + O_2} 32.0 F_{O_2} \quad (5.4.17)$$

NATIN uses the FORTRAN statement functions HNF and HOF to calculate the nitrogen enthalpy, H_{N_2} , and oxygen enthalpy, H_{O_2} , at the injection temperature. The injected gas adds the following energy increment to the cell:

$$\frac{dU_a}{dt} dt = H_{N_2} I_{N_2} dt + H_{O_2} I_{O_2} dt \quad (5.4.18)$$

The result is the evaluation of the following portions of the differential equations for the mass and energy of the cell atmosphere, Equations (3.1) and (3.4):

$$\frac{d}{dt} M_a = I_{N_2} + I_{O_2} \quad (5.4.19)$$

$$\frac{d}{dt} U_a = H_{N_2} I_{N_2} + H_{O_2} I_{O_2} \quad (5.4.20)$$

Again, the writer asserts the equivalence of Equations (5.4.16) and (5.4.17) with Equation (5.4.19) and of Equation (5.4.18) with Equation (5.4.20). NATIN calculates the gas injection rate and sums the increments of gas injected into the cell atmosphere. It prints this rate and sum along with other cell data. Lastly, the reader must note that gas injected by subroutine NATIN has no carbon dioxide and no water vapor.

CCALC processes the sodium-concrete chemical reaction table in array CRNACO which was entered on Card Series 2401. This table covers the chemical attacks of concrete heat structures contacted by sodium pools. In particular, there may be attacks on several heat structures which are not all in the same cell. The data table includes the number of each heat structure attacked by sodium, I , the fraction of heat structure area involved, F_{HS} , the present depth of sodium attack penetration into the structure, D , the penetration rate of the attack, dD/dt , the chemical reaction energy of the attack, Q_{conc} , the integral of the reaction energy, the sodium consumption rate of the attack, R_{conc} , and the integral of the sodium consumption. The CACECO model of this sodium-concrete reaction attack is nonmechanistic;

it assumes that the sodium pool has thermal contact with the left-hand boundary of a planar heat structure which is not altered in the course of the attack. The CCALC operations are summarized below:

1. The index to the left-hand boundary temperature, array BT1, is checked for being either 102, 105, 108, or 111 which are the temperatures of the sodium pool in Cells A, B, C, and D.
2. The array MPT is checked for having mass in the sodium pool.
3. The area of the attack is calculated from the left-hand surface area of the heat structure, $ARA(I) \cdot SR1(I)$ or A_I , and the fraction involved:

$$A_{conc} = A_I F_{HS} \quad (5.4.21)$$

This calculation is valid only for a planar structure.

4. The attack penetration depth is integrated and stored:

$$D' = D + (dD/dt) dt \quad (5.4.22)$$

5. This new penetration depth is checked against the depth of the right-hand surface of the heat structure, $RS(NN,I)$. This check assumes that the planar structure has its left-hand surface, $RS(1,I)$, at 0.0 ft depth. If the attack has penetrated the right-hand surface, this event is printed and the penetration rate is set to zero thereafter, i.e., $dD/dt = 0$.
6. The chemical reaction energy released to the sodium pool during the time step is calculated and used in the evaluation of the differential equation for pool energy:

$$\frac{d}{dt} U_p = + Q_{Na + conc} = (dD/dt) dt A_{conc} Q_{conc} \quad (5.4.23)$$

7. The integral of this chemical reaction energy is updated and stored.
8. The sodium consumption during the time step is calculated and used in the evaluation of the differential equation for pool mass:

$$\frac{d}{dt} M_p = - R_{Na} + \text{conc} = - (dD/dt) \text{ dt } A_{\text{conc}} R_{\text{conc}} \quad (5.4.24)$$

9. The integral of this sodium consumption is updated and stored.
10. These results are printed along with other cell data.

This CACECO model of the sodium-concrete reaction attack is useful when the sodium pool is large and the penetration depth is small. The obvious defects of the model are: (1) sodium is consumed without generation of reaction products and (2) the concrete structure is penetrated but not altered.

CCALC calls subroutine COOLR to process the space cooler instructions in arrays COOLA, COOLB, COOLC, and COOLD which were entered on Cards 1700, 1800, 1900, and 2000 for Cells A, B, C, and D, respectively. These instructions had been read by subroutine TABLC (an ancillary of CINPUT). The instructions included: the time the cooler starts operating, the time it stops, the temperature of the cooler fluid, T_c , the coefficients A and B of the cooler operating equation, and the minimum temperature of cooler operation. The assumed cooler operating equation is:

$$Q_{SC} = A + B (T_a - T_c) \quad (5.4.25)$$

where T_a is the cell atmosphere temperature. Subroutine TABLC calculates and stores

$$A' = A - B T_c$$

so that the effective equation for COOLR is only

$$Q_{SC} = A' + B T_a \quad (5.4.26)$$

The COOLR operations are summarized below.

1. Check that the present time is between the start and stop times.
2. Check that the cell atmosphere temperature is more than the minimum temperature for cooler operation.
3. Use Equation (5.4.26) to calculate the cooler energy removed from the cell, $Q_{SC} dt$. Note that this increment can be either positive, the normal cooling duty, or negative, as heating duty, according to the signs of the coefficients. If this increment is negative, skip the atmosphere dehydration calculations, just below.
4. If there is water vapor in the cell atmosphere, calculate the amount of water removed by the increment of cooler energy. The mass of water vapor, M_w , is in array MWT and passed to COOLR in the common variable ST02; its specific energy, U_g , is in array UGWT and passed in common variable UGW. The amount of water removed is:

$$dM_w = Q_{SC} dt / U_g \quad (5.4.27)$$

and the new mass of water vapor is:

$$M_w = M_w - dM_w = M_w - Q_{SC} dt / U_g \quad (5.4.28)$$

This step has a logical error, illustrated in Debug Cases 16 through 19 of the Analytical Validation..., ⁽²⁾ which occurs at the time M_w' goes negative; the new mass is set to zero but the excess bit of energy remains. The integral of the water removed is updated and stored. This step results in an evaluation of the following portion of the differential equation for atmosphere mass:

$$\frac{d}{dt} M_a = - C = - Q_{SC}/U_g \quad (5.4.29)$$

5. Evaluate the following portion of the differential equation for atmosphere energy:

$$\frac{d}{dt} U_a = - Q_{SC} \quad (5.4.30)$$

6. The integral of the cooler energy is updated and stored.
7. The results are printed along with other cell data.

This CACECO model of the cell space cooler is useful because it can be either a cooler-dehumidifier or a heater by choice of the input data.

The CCALC capabilities are very similar for calculating water release and carbon dioxide release as concrete is heated. These capabilities are important because concrete is the universal material of construction of LMFBR facilities; by common experience, it is a stable, durable material for floors, walls, and, where shielding is important, for roofs as well. These roof, wall, and floor structures have most of the overall heat capacity of the facility and are important for their heat absorption in accident analyses. All kinds of concrete release water when heated because water is used to make the concrete. Generally, these concrete structures will release their water content, 9 lb H₂O/ft³ concrete and more, when the concrete is heated above 200°F. Only concrete made with limestone aggregate or fines release important quantities of carbon dioxide when heated. These limestone concrete structures will release their carbon dioxide content, 50 lb CO₂/ft³ concrete and more, when the concrete is heated above 1200°F.

These water and carbon dioxide releases occur the first time the concrete is heated. They require significant amounts of energy and thereby affect the temperatures inside the concrete structures. The models of concrete water

release and of concrete carbon dioxide release are very similar and use very similar data tables which are listed below.

<u>Data Table</u>	<u>Water Release</u>	<u>Carbon Dioxide Release</u>
(1) Concrete additional heat capacity table		
(a) array	CONC	CONC
(b) input Card Series	2201	2201
(2) Release function		
(a) array	CWRD	CO2RD
(b) input Card Series	2501	2701
(3) Instructions about heat structure releases		
(a) array	SWCWR	SWC02R
(b) input Card Series	2601	2801

The CACECO model of concrete water release assumes a coherence among these data tables as follows:

1. The concrete additional heat capacity table in array CONC has the following attributes:
 - a. The material indices stored in ICONC(1) and ICONC(2) identify materials to which the additional heat capacity applies. This identification is keyed to materials used in all heat structures. The latter have thermal properties stored in array KSCS which were entered on Card Series 300001.
 - b. The additional heat capacities must be based on the water release function in array CWRD and on the carbon dioxide release function in array CO2RD.

c. If the additional heat capacity in array CONC is C_a and if the normal heat capacity for an identified material in array KSCS is C_k , then, for this particular material, the effective heat capacity used by subroutine HCALC is as follows:

i. For the first heating, $C_e = C_a + C_k$ (5.4.31)

ii. For any cooling, $C_e = C_k$ (5.4.32)

iii. For any subsequent heating that does not exceed the maximum temperature obtained by the first heating, $C_e = C_k$.

2. The concrete water release function in array CWRD is the basis for the effective heat capacities given in array CONC. For example, if this function has water release R_1 at temperature T_1 and release R_2 at higher temperature T_2 , and if the average energy effect of this release over this temperature range is Q_{12} , then the additional heat capacity over this temperature range is:

$$C_a = (R_2 - R_1) Q_{12} / (T_2 - T_1) \quad (5.4.33)$$

3. The instructions about heat structure releases in array SWCWR identify heat structures that have materials in the range of the indices stored in ICONC(1) and ICONC(2). These instructions in array SWCWR include: the structure number, the destination index for the water released from this structure, and the temperature at which the release may begin. An important use for this temperature is to designate the failure point of steel liners that cover the concrete structure and thereby prevent release until they fail.

Similarly, the CACECO model of concrete carbon dioxide release assumes a coherence among its data tables as follows:

1. The concrete additional heat capacity table in array CONC must cover both water release and carbon dioxide release. This double coverage is possible because the temperature range for water release is mainly below the temperature range for carbon dioxide release from limestone concrete, the only concrete that has both releases.
2. The concrete carbon dioxide release function in array CO2RD is the basis for the effective heat capabilities given in array CONC. The Equation (5.4.33) applies to carbon dioxide by appropriate definition of releases R_1 and R_2 .
3. The instructions about heat structure releases in array SWC02R identify the heat structures that release carbon dioxide. When the facility has limestone concrete heat structures, there must be congruence between array SWCWR and array SWC02R:
 - a. Structure numbers will be identical; each structure will release water and then, as heating continues, carbon dioxide.
 - b. The destination index for the water released and for the carbon dioxide released from each structure will be identical.
 - c. The temperature at which the releases begin should be identical.

Note that input data for basalt and magnetite concrete have the Card Series 2201, 2501, and 2601. Input data for limestone concrete have the Series 2201, 2501, 2601, 2701, and 2801. The CACECO models of water and carbon dioxide releases are based on the following assumptions:

1. The concrete heat structure is heated from the left-hand boundary which is also the inside boundary for cylindrical and spherical structures.
2. Water and carbon dioxide are released because the temperature of an interior point increases the first time.

3. The only resistance to the release of water and carbon dioxide is the conduction heating of the concrete structure.
 - a. The release from any interior point is solely dependent upon the temperature of that point. It is not dependent upon nearby temperatures nor on any pressure anywhere.
 - b. Water and carbon dioxide released from any interior point immediately passes through the left hand boundary to the destination indexed for the structure.

This simple model neglects the mass transport analysis of water and carbon dioxide movement from the interior of the concrete structure to the surface. The experience thus far with heated concrete blocks indicates that the mass transport out of the structure is rapid in comparison to the heat transfer into the structure. The CCALC operations for the water release evaluation are summarized below.

1. Set out the structure number I and the cell destination index J. Define the control IH to distinguish between the cell atmosphere and the pool as the destination.
2. Check that the maximum temperature ever attained for the left hand surface of the structure, $USM(1,I)$ or T_{max} is greater than the minimum temperature for water release, T_{min} , which was entered on Card Series 2601.
3. Define a factor, variable ST03 or F_3 , to avoid a burst of water release as the surface temperature passes the minimum release temperature. The present procedure is to increase the water release by 20 percent per degree Fahrenheit up to the limit of 100 percent release.

- a. If $T_{\max}$ is within 5°F of $T_{\min}$, then

$$(\text{ST03}) = F_3 = (T_{\max} - T_{\min})/5 \quad (5.4.34)$$

- b. If $T_{\max}$ is 5°F or more than $T_{\min}$, then $(\text{ST03}) = 1$.

4. Call subroutine CWRP to evaluate the total amount of water released from the structure up to the present time, variable CWR or R_t . This evaluation is based on the maximum temperature ever attained at each node point N; the temperatures stored in array USM(N,I). The subroutine begins at the left-hand surface, at Node 1, and sums the releases for successive node volumes throughout the structure.

- a. The water release function at Node N-1 is R_{N-1} based on its temperature, T_{N-1} .
- b. Call subroutine TABLU to interpolate the water release function at the maximum temperature, T_N of Node N, for the release, R_N .
- c. The volume of the structure between Node N-1 and Node N is the array value, XS(N,I), or V_N . The subroutine checks that the material of this volume is identified as concrete by the indices stored in ICONC(1) and ICONC(2), which were entered on Card Series 2201.
- d. The water release from this node volume is:

$$R_{tN} = 0.5 (R_{N-1} + R_N) V_N \quad (5.4.35)$$

- e. The sum of the node volume release is the structure release:

$$R_t = R_{t2} + R_{t3} + R_{t4} + \dots \quad (5.4.36)$$

5. If the structure release at the previous time was R_{told} , calculate the incremental release for this time step:

$$R_{H_2O} = (R_t - R_{told}) F_3 \quad (5.4.37)$$

6. Call subroutine WAPROP to evaluate the energy of water vapor at the present temperature of the surface, U_{H_2O} .
7. The integral of the water release is updated and stored.
8. Pass the incremental release to the cell destination set out in step (1), above.

- a. If the destination is the cell atmosphere, then evaluate the following portions of the differential equations for its mass and energy, Equations (3.1) and (3.4):

$$\frac{d}{dt} M_a = \sum_N R_{H_2O} \quad (5.4.38)$$

$$\frac{d}{dt} U_a = \sum_N U_{H_2O} R_{H_2O} \quad (5.4.39)$$

- b. If the destination is the cell sodium pool and this pool exists, then evaluate the following portions of the differential equations for its mass and energy, Equations (3.2) and (3.5):

$$\frac{d}{dt} M_p = \sum_N R_{H_2O} \quad (5.4.40)$$

$$\frac{d}{dt} U_p = \sum_N U_{H_2O} R_{H_2O} \quad (5.4.41)$$

Of course, if there is no sodium pool, the array value $MPT(J)$ is zero and the incremental release and its energy goes into the atmosphere.

9. The results of this water release are printed along with other cell data.

The CCALC operations for the carbon dioxide release evaluation are nearly identical to those described for the water release evaluation. The differences are noted below.

1. The minimum release temperature was entered on Card Series 2801.
2. This factor definition is omitted.
3. Subroutine C02F is called.
4. The incremental release for this time step is:

$$R_{CO_2} = R_t - R_{told} \quad (5.4.42)$$

5. The function statement UCF provides the energy of carbon dioxide at the present temperature of the surface, U_{CO_2} .
6. Pass the incremental release to the cell destination:
 - a. If the destination is the cell atmosphere, then these portions of Equations (3.1) and (3.4) are:

$$\frac{d}{dt} M_a = \sum_N R_{CO_2} \quad (5.4.43)$$

$$\frac{d}{dt} U_a = \sum_N U_{CO_2} R_{CO_2} \quad (5.4.44)$$

- b. If the destination is the cell sodium pool and this pool exists, then these portions of Equations (3.2) and (3.5) are:

$$\frac{d}{dt} M_p = \sum_N R_{CO_2} \quad (5.4.45)$$

$$\frac{d}{dt} U_p = \sum_N U_{CO_2} R_{CO_2} \quad (5.4.46)$$

These CACECO models of concrete water release and of concrete carbon dioxide release are simple and yet quite capable of fitting the experimental data of one-dimensional heating situations.

CCALC evaluates the leakrates between cells, and between a cell and the outside, by calling subroutines LEAKC and LEAKN. In the subroutine call statements, the indices identifying leakage source and destination are as follows:

<u>Index</u>	<u>Source and Destination Identification</u>
1	outside
2	Cell A
3	Cell B
4	Cell C
5	Cell D

Subroutines LEAKC and LEAKN both identify by index K the leakage source with the higher pressure and by index KK the leakage destination. Both subroutines use the atmosphere density of the source, ρ_K , and the differential pressure between source and destination, ΔP , to calculate the leakrate: L_c in subroutine LEAKC and L_n in LEAKN. In both subroutines, the product of the leakrate and of material concentrations from array $RATIO(J,K)$ is the material leakrate which is stored in array LEAK at location $LEAK(J,K, KK)$. The index J identifies the atmosphere material as follows.

<u>Index J</u>	<u>Leakage Materials</u>
1	Nitrogen
2	Oxygen
3	sodium vapor
4	Mass of all materials in source atmosphere
5	Hydrogen
6	Water vapor
7	Carbon dioxide

Finally, both subroutines check the leakage source against the following criteria for hydrogen-oxygen chemical reaction in the destination atmosphere:

1. The leakage source is at 500⁰F or hotter and has
2. Hydrogen and
3. Sodium vapor of 3.4×10^{-4} lb/ft³ (5.5 g/m³) concentration or greater.

When these criteria are satisfied, each subroutine will set the control HOR1(KK) to TRUE for use by subroutine EMVBAL. Subroutine LEAKC evaluates the leakrate L_c by using leakage coefficient data stored in one of nine arrays, such as array LOCOAD, which was entered on one of the card series, such as Card Series 31. The array LOCOAD holds leakage coefficients for leakage between Cell A and Cell D and this leakage can go in either direction. Subroutine TABLU finds the leakage coefficient C_1 at the differential pressure ΔP . The leakrate is then:

$$L_c = C_1 \sqrt{\Delta P \rho_K} \quad (5.4.47)$$

Subroutine LEAKN calculates the leakrate L_n by the DeLaval compressible flow nozzle equations which use the nozzle area, A_n , and nozzle coefficient C_n , data stored in one of nine arrays such as array NOZZAD. These nozzle arrays were entered on one of nine input cards such as Card 30. All

of these input cards were processed by subroutine TABLN (an ancillary of CINPUT) to form the following product:

$$C'_n = A_n C_n \sqrt{2 g 144 k / (K-1)} = A_n C_n 672,863 \quad (5.4.48)$$

where: g is the gravitational constant and

$k = 1.361$, the specific heat ratio (C_p/C_v) for nitrogen at 1000°F.

The DeLaval nozzle is a convergent-divergent nozzle that can flow subsonic, sonic, or supersonic depending upon the pressure ratio. The equations for this nozzle⁽⁷⁾ have been simplified for CACECO and the following are used in LEAKN: The source pressure, P_K , and the destination pressure, P_{KK} , are set out and the rudimental leakrate calculated:

$$L'_n = C'_n \sqrt{P_K \rho_K} \quad (5.4.49)$$

The pressure ratio R is calculated,

$$R = P_{KK} / P_K \quad (5.4.50)$$

and also the modified pressure ratio:

$$Z = R^{1/k} = R^{0.73475} \quad (5.4.51)$$

If Z is greater than 0.5353, the leakrate is subsonic and

$$L_n = L'_n \sqrt{Z (Z - R)} \quad (5.4.52)$$

Otherwise, the flow is sonic and

$$L_n = L'_n 0.246934 \quad (5.4.53)$$

LEAKN prints this calculated leakrate, and also a modified leakrate which is based on an incompressible flow equation, along with other cell data. CCALC

sums the leakrates of all of the LEAKC and LEAKN calculations and prints the sums of the mass of all materials' leak rates stored in LEAK(4,K,KK) for all K indexed sources and all KK indexed destinations. Of course, all printed leakrates are positive because they denote leakrates from the source.

CCALC uses statement functions to evaluate material enthalpies at the temperatures of the various sources and stores these enthalpies for use by subroutine LEAKR:

<u>Material Enthalpy</u>	<u>Statement Function</u>	<u>Array</u>	<u>Leakrate</u>
Carbon dioxide	HCF	HCT	LEAK(7,K,KK)
Hydrogen	HHF	HHT	LEAK(5,K,KK)
Nitrogen	HNF	HNT	LEAK(1,K,KK)
Oxygen	HOF	HOT	LEAK(2,K,KK)

CCALC then calls subroutine LEAKR to evaluate the following portions of the differential equations for the atmosphere mass and energy of Cell J which has leakage sources of index I and leakage destinations of index K.

$$\frac{d}{dt} M_a = \sum_I L_{IJ} - \sum_K L_{JK} \quad (5.4.54)$$

$$\frac{d}{dt} U_a = \sum_I U_I L_{IJ} - \sum_K U_J L_{JK} \quad (5.4.55)$$

LEAKR obtains the enthalpy of water vapor from array HWT for leakrate LEAK(6,K,KK) and of sodium vapor from array HGT for leakrate LEAK(3,K,KK). These arrays HWT and HGT are maintained for this use by subroutine EMVBAL. LEAKR prints out the net leakrate of each cell from array NRT which holds the results of Equation (5.4.54). LEAKR also prints out the sum leakrate of each cell from array SRT which holds the following results:

$$\frac{d}{dt} M_a = - \sum_K L_{JK} \quad (5.4.56)$$

Further, the integrals of the net leakage, NLEAK, and of the sum leakage, SLEAK, are updated and stored. LEAKR prints the net leakage and sum leakage along with other cell data. In these printouts of arrays NRT, SRT, NLEAK and SLEAK, the positive values are for material leaving the cell; that is, the positive leakrates and positive leakages are for material leaving the cell.

The present version of CACECO uses the array FPT in two ways:

1. The preferred way is described in Section 5.2 and needs the odd numbered input card (Card 3, 5, 7, or 9) to precede its associated even numbered card (Card 2, 4, 6, or 8, respectively) in the input card deck. By this preferred way, the value entered as Word 4 of the odd numbered card is stored in array FPT and used for the relative humidity of the cell atmosphere. After this use, subroutine INISHL sets array FPT to zero.
2. The other way needs the odd numbered input card to follow its associated even numbered card. By this way, the value entered as Word 4 of the odd numbered card is used as an atmosphere component which has no mass and no energy but acts as a marker to track leakage from one cell to another. This tracking capability of array FPT is obtained in subroutine LEAKR. For example, if Word 4 enters the value 10.0, then the cell conditions for Cell J would show for "Fission Products" the value 0.1000000. Now if leakage from various sources increases the atmosphere mass of Cell J by the factor 7/5, the new Cell J conditions would show 0.100000 for "Fission Products." Then if leakage out of Cell J decreases its atmosphere mass by the factor 3/5, the new cell conditions would show $(3/5) \times (0.1000000) = 0.0600000$ for "Fission Products." The author hasn't used this tracking capability very often.

CCALC recalls the summed heat transfer rates of the heat structures from array SHTR and stores them in array POOL. These summed heat transfer rates were stored by subroutine HTRAN at the end of the previous time step. The summation operation sums the rates of all structures into the outside air and into each cell atmosphere, each cell pool, and each constant temperature sink. The summation is controlled by the boundary temperature indices which were entered on Card Series 1NN400 and stored in arrays BT1 and BTN. The summed rates in location POOL(J) and location SHTR(J) are identified below.

<u>Location J</u>	<u>Boundary Temperature Index</u>	<u>Identification</u>
1	100	Outside air
2	101	Cell A atmosphere
3	102	Cell A pool
4	103	Sink with index 103
5	104	Cell B atmosphere
6	105	Cell B pool
7	106	Sink with index 106
8	107	Cell C atmosphere
9	108	Cell C pool
10	109	Sink with index 109
11	110	Cell D atmosphere
12	111	Cell D pool
13	112	Sink with index 112

CCALC prints these summed heat transfer rates along with other heat structure data

CCALC calls subroutine STORE to save arrays UAT and UPT of cell energy data. Also, the present version of STORE includes but then skips the FORTRAN statements, to save the arrays of cell material data. STORE then evaluates the following portions of the differential equations for the energy of the atmosphere and of the pool by using the summed heat transfer rates stored in array POOL.

$$\frac{d}{dt} U_p = \sum_N Q_{HS} \quad (5.4.57)$$

$$\frac{d}{dt} U_p = \sum_N Q_{HS} \quad (5.4.58)$$

STORE checks and, of course, if array MPT is zero there is no sodium pool, and if array MPWT is zero, there is no water pool either and then STORE adds the summed rates for the pool energy in Equation (5.4.58) to those for the atmosphere energy in Equation (5.4.57).

Except for the chemical reactions, all portions of the differential equations for the mass and energy of the cell atmosphere, Equations (3.1) and (3.4) and of the cell pool, Equations (3.2), (3.3) and (3.5), have been evaluated. CCALC calls subroutines EMVBAL and HCALC and then again calls subroutine EMVBAL as described in Section 4 and shown in Figure 4.1. These calls complete the evaluations of the differential equations for the mass and energy of the cell atmosphere and pool. The calls integrate these equations to the end of the time step and result in the cell conditions and heat structure temperature at the end of the time step.

CCALC calls subroutine EMVBAL for the following operations which are described in detail in Section 5.5.

1. Calculate the mass and energy effects of the chemical reactions in the cell atmosphere and sodium pool, and thereby complete the differential equations for the mass and energy of the cell atmosphere and of the cell pool.
2. Integrate these differential equations to the end of the time step and predict the cell conditions at the end of the step. The important predictions are the atmosphere temperature in array TAT and the pool temperature in array TPT.

CCALC prints these predicted cell conditions along with other cell data.

CCALC calls subroutine HCALC for the following operations which are described in detail in Section 5.6.

1. Update the heat structure for each sodium pool if the pool depth has changed by more than 3 percent.
2. Integrate the differential equation for the temperatures of each heat structure to the end of the time step. This operation solves the transient heat conduction equations for the new structure temperatures. When the boundary conditions of a structure index a cell atmosphere, the operation uses the atmosphere temperature of the previous time step from array TATO and the predicted temperatures of the present time step from array TAT. Likewise for the cell pool.
3. Calculate the heat transfer rates from both surfaces of each heat structure and sum these rates into arrays POOL and SHTR.
4. The integrals of these heat transfer rates are updated and stored.
5. Print the heat structure data when control SW3 is FALSE.

CCALC prints the summed heat transfer rates and their integrals along with other heat structure data.

CCALC calls subroutine FETCH to implement the predictor-corrector method of integrating the differential equations. Also, the present version of FETCH includes, but then skips, the FORTRAN statements to recover the arrays of cell material data. In this description of the predictor-corrector method, let Q_{HD} denote a heat transfer rate such as those summed in Equations (5.4.57) and (5.4.58) at the end of the previous time step and Q'_{HS} denote the same heat transfer at the end of the present time step. The predictor-corrector method in CACECO needs three subroutine calls in sequence: HCALC,

HTRAN, and FETCH. HCALC sets array POOL to zero. HTRAN calculates Q'_{HS} and then sums the differences in array POOL: $Q'_{HS} - Q_{HS}$. FETCH completes the evaluation of the following portions of the differential equations for the energy of the atmosphere and of the pool by using the summed differences of heat transfer rates stored in array POOL.

<u>Subroutine STORE</u>	<u>Subroutine FETCH</u>	
$\frac{d}{dt} U_a + \sum_N Q_{HS} + 1/2$	$\left[\sum_N Q'_{HS} - \sum_N Q_{HS} \right]$	(5.4.59)

$\frac{d}{dt} U_p = \sum_N Q_{HS} + 1/2$	$\left[\sum_N Q'_{HS} - \sum_N Q_{HS} \right]$	(5.4.60)
--	---	----------

FETCH checks and, again, if array MPT is zero there is no sodium pool, and if array MPWT is zero there is no water pool, either, and then FETCH adds the summed rate differences for the pool energy in Equations (5.4.60) to those for the atmosphere energy in Equation (5.4.59).

CCALC calls subroutine EMVBAL the second time for the following operations which are described in detail in Section 5.5.

1. Integrate the corrected differential equations to the end of the time step and calculate the cell conditions at the end of the step. The present version of CACECO assumes either of two situations for these cell conditions at the end of the time step:
 - a. The cell has an excess of sodium and all carbon dioxide, oxygen, and water vapor are consumed in chemical reactions, or else
 - b. The cell has carbon dioxide, oxygen, and water vapor and the sodium is all consumed.

2. Evaluate and store cell data as follows:

- a. Liquid and vapor properties for the fluid in excess, either sodium or water: arrays HGT, HWT, ULT and VLT.
- b. Pool boiling properties: arrays TBOLT, UCHNG, UGBOL, ULBOL, and VLBOL.
- c. Atmosphere material concentrations: array RATIO.

CCALC prints the cell conditions at the end of the time step when the control SW4 is FALSE.

CCALC checks the convergence of the predictor-corrector method of integrating the differential equation for atmosphere energy. This check uses values of array UAT or U_a which resulted from the first call to subroutine EMVBAL and were saved in array POOL by FETCH, and, of course, values of array UAT or U'_a which resulted from the second call to EMVBAL. The convergence, C_u is the absolute value of the ratio:

$$C_u = |U_a - U'_a| / U'_a \quad (5.4.61)$$

This cell convergence is printed along with other cell conditions. CCALC sums all cell convergences into variable CONV(1) for use as a control.

CCALC saves cell data in array SUMMRY for a periodic two-page, 116-line summary of the analysis. CCALC saves this cell data for each time step up to one hour and thereafter for the first two time steps of each hour. These data are printed by a call to subroutine OUTSUM whenever 116 lines have been saved or at the end of the run. The present version of CACECO saves all cell temperatures, the hydrogen, oxygen, and water vapor compositions of Cell C atmosphere, the total leakrates from Cell A to Cell C and from Cell C to the outside, all cell pressures, the relative magnitude of the time step, variable STEP, and the sum of all cell convergences, control CONV(1). These

summary data have been useful to the author. Indeed, the author habitually checks these periodic two pages of summary data in order to validate CACECO results. This habitual check looks for the following:

1. Cell temperatures and pressure that oscillate or go on unusual transients that are not expected in the accident scenario.
2. Values of control CONV(1) larger than 1×10^{-3} or values of variable STEP smaller than 1.0 or changes in the magnitude of STEP that are not expected.

The author has modified subroutines CCALC and OUTSUM in order to save and print different data which are more convenient for special purposes, and urges the user to do likewise. CCALC saves data in array SUMMRY in two places, and both should have identical FORTRAN statements. Further, there must be congruence between the data saved by CCALC and the labels printed by OUTSUM so that the printed data are correctly labeled.

CCALC writes cell and heat structure data to the binary file PTAPE for subsequent use by the CCLST code for making summary tables and by the CCPLT code for making plots. A line of data is saved as follows:

1. at one hour intervals up to 24 hours, then
2. at two-hour intervals up to 500 hours, then
3. at four-hour intervals up to 1,000 hours, and then
4. at ten-hour intervals thereafter.

CCALC calculates an average leakrate over the period since the last data line in preparation for writing the present data line. This leakrate calculation uses arrays SLAK and TOLD (which are equivalenced into array AVGLKG) to hold the leakage and time values of the last data line. The

calculation obtains the present leakage from arrays NLEAK and SLEAK as integration by subroutine LEAKR. The present version of CACECO writes the following average leakrates.

<u>Variable</u>	<u>Leakrate</u>
AVGLKR(2)	Sodium leakrate from Cell A, lbs of sodium
AVGLKR(3)	Sodium leakrate into Cell C, lbs of sodium
AVGLKR(4)	Total atmosphere material leakrate from Cell C but converted to standard cubic feet (SCF) at 32 ⁰ F and 14.7 psia.
AVGLKR(5)	Total water leakage from Cell D, lbs of water

These particular average leakrates have been useful to the author. The present version of the code writes all cell temperatures, pressures, material inventories, and all heat structure temperatures plus carbon dioxide and water releases.

CCALC calls subroutine CHANGE to modify the length of the time step if the sum of cell convergences, control CONV(1), is outside the range of 5.0×10^{-5} to 5.0×10^{-3} . CHANGE will not modify the time step if:

1. The call is during the first ten steps of a new or newly restarted case,
2. The present step is ten times the value entered on Card Series 9001 as shown by the variable STEP with value 10.0, and
3. The present step is 0.01 times the value entered as shown by the variable STEP with value 0.01 in the two-page summary table.

If CONV(1) is less than 5.0×10^{-5} , CHANGE will lengthen the step DT by the factor 1.5848932 which is the fifth root of ten. If CONV(1) is more than 5.0×10^{-3} , CHANGE will shorten the step by the factor 1/1.5848932. CHANGE will lengthen and shorten the print intervals and increase and decrease the variable STEP by these same factors.

CCALC checks array TSPC for the proper time step and print controls to use in the next step and then returns to the top of the FORTRAN loop to begin the step and continue this marching process of integrating the differential equations for the mass and energy of the cells.

The conditions for the normal step of the FORTRAN loop were listed at the beginning of this Section 5.4 on subroutine CCALC. The normal stop provides the following printing and writing at the end of the run:

1. Control SW4 is FALSE and thereby all cell data are printed during the course of the FORTRAN loop.
2. Control SW3 is FALSE and thereby all heat structure data are printed.
3. Cell conditions are printed at the end of the run.
4. Array SUMMRY is printed with one or two pages of summary data.
5. All labeled common block and blank common data are written to the binary file DTAPE for storage until needed later for the restart of the case. CCALC prints the following statement that these data are saved:

"DATA DUMP AT STEP..."

6. CCALC returns to the main program CACECO which then calls CINPUT. CINPUT prints the values of variables DTB, DTG, DTH, DTL, DTR, and DTT which have accumulated the incremental times for various bits of the FORTRAN loop. Typically DTH is the largest time because of the effort

in HCALC to solve the heat conduction equations. CINPUT then begins to read in the cards for the next case.

In comparison, the error stop is announced in output by the CCALC printed statement:

"***** FAILED AT...TOO BAD, NO TAPE DUMP."

1. Cell conditions are printed, but these may be in error.
2. Array SUMMRY is printed to aid the user in determining the cause of the error stop.
3. CCALC returns to CACECO, which then calls CINPUT, which prints out the times and begins to read in the cards for the next case.

5.5 Subroutine EMVBAL

Subroutine EMVBAL is called by CCALC as described in Section 4.0 and shown in Figure 4.1 as follows:

1. For each cell defined by input data,
2. Twice during each time step of the marching process for integrating the differential equations of the mass and energy of the cell atmosphere and pool.

The name EMVBAL is an acronym for energy, mass and volume balance, the general process used by the subroutine in completing the differential equations, integrating them to the end of the time step, and then calculating the new cell conditions.

EMVBAL calls subroutines NAPROP, NAHPRO, and WAPROP to obtain thermodynamic properties of sodium, sodium hydride, and water, respectively, as functions of temperature and pressure. EMVBAL has FORTRAN statement functions for the

energy and enthalpy properties of the other cell materials as functions of temperature. The temperature range of interest is from 50°F to 2500°F. The reader is referred to Appendix A of the "Analytical Validation..."⁽²⁾ for representative properties values at 100°F steps from 0°F to 2000°F. The general sources for these thermodynamic properties are summarized below.

<u>Material</u>	<u>Source</u>	<u>Reference</u>
Carbon	Hougen and Watson	8
Carbon dioxide	ditto	8
Hydrogen	ditto	8
Nitrogen	JANAF	9
Oxygen	JANAF	9
Sodium	Golden and Tokar	10
Sodium hydride	Whittingham	11
Sodium hydroxide	JANAF	9
Sodium oxide	JANAF	9
Water	ASME Steam Tables	12

Further, the energy values for the chemical reactions evaluated by EMVBAL are from the JANAF tables and are usually the average energy over the temperature range of 400 K (260°F) to 1100 K (1520°F).

Subroutine NAPROP calculates the saturated liquid and vapor properties of sodium as a function of temperature and the superheated vapor properties as a function of temperature and pressure. These properties are returned through the labeled common block SODIUM. There are two ancillary subroutines to find the saturated temperature at particular conditions; both use Newton's iteration method and both call NAPROP to return the saturated properties. These two subroutines are:

1. SRCHPN finds the saturated temperature at the given total pressure.

2. SRCHUN finds the saturated temperature at the given liquid energy.

This subroutine uses the function NAUL to calculate the liquid energy.

Subroutine WAPROP calculates the saturated liquid and vapor properties of water as a function of temperature up to 300⁰F and the superheated vapor properties as a function of temperature and pressure. These properties are returned through the labeled common block WATER. There are two ancillary subroutines to find the saturated temperature at particular conditions; both use Newton's iteration method and both call WAPROP to return the saturated properties. These two subroutines are:

1. SRCHPW finds the saturated temperature at the given total pressure.

2. SRCHUW finds the saturated temperature at the given liquid energy.

This subroutine uses the function WAUL to calculate the liquid energy.

Subroutine NAHPRO calculates the hydrogen pressure and sodium hydride solubility as functions of temperature for the equilibrium condition of hydrogen, sodium, and sodium hydride. These properties are returned through the labeled common block HYDRID.

The present version of EMVBAL is organized to promote two assumptions about the chemical, transport and thermodynamic processes within each analytical cell. These assumptions are important because they limit the application of the CACECO code to various accident situations. These assumptions are as follows:

1. The atmosphere is in temperature equilibrium with the pool. That is, the atmosphere is at the same temperature as the pool and is saturated with pool material, either sodium or water, as available.

This assumption prevents immediate application of the CACECO code to sodium spray and sodium spill accident scenarios which expect the atmosphere to remain significantly cooler than the sodium pool which accumulates on the cell floor.

2. The chemical reaction sequence and reaction products are fixed. The situation that promoted each of the chemical reactions is described in Section 3.3 and need not be repeated. All but one of these chemical reactions go to completion at each time step and thereby consume all of the reactants which are not in excess. The reaction that does not go to completion is the equilibrium reaction between hydrogen, sodium, and sodium hydride, and this goes to equilibrium at each time step.

The remainder of this Section 5.5 on subroutine EMVBAL will describe its features and organization in order to aid the user who may want to modify this subroutine for his own purposes.

Certain arrays are reserved for sodium data and certain other arrays are reserved for water data, but most are used for the fluid which is in excess in the cell. For example, arrays APT and DEEP are reserved for sodium because they describe the surface area and thickness, respectively, of the heat structure that represents the sodium pool in the cell (Section 5.3). The reserved arrays and the other arrays are listed below.

<u>Application</u>	<u>Sodium</u>	<u>Water</u>	<u>Either</u>
Boiloff mass from pool			BOLT
Condensate mass from atmosphere			MLT
Mass of pool	MPT	MPWT	
Vapor pressure at pool temperature			PST
Temperature that pool would boil at the total pressure of the cell			TBOLT
Pool temperature			TPT

<u>Application</u>	<u>Sodium</u>	<u>Water</u>	<u>Either</u>
Energy of vaporization, $U_g - U_f$			UCHNG
Energy of boiloff vapor, U_g			UGBOL
Vapor energy	UGT	UGWT	
Energy of boiling liquid, U_f			ULBOL
Liquid energy	ULT	ULWT	
Total energy of pool			UPT
Vapor specific volume	VGW	VGWT	
Specific volume of boiling liquid			VLBOL
Liquid specific volume			VLT
Atmosphere volume (cell volume less the pool volume)			VOLAT

The EMVBAL operations are summarized below for the particular cell which was identified by the subroutine argument, index Z. The main operations of this summary are those listed in Section 5.4 for the subroutine CCALC calls to EMVBAL: the first call and then the second call. The author used variables instead of arrays in order to simplify most of the FORTRAN statements in EMVBAL. Sometimes these variables have multiple uses. For example, the variable MC is (a) first the mass of carbon dioxide released to the sodium pool and stored in array MCP, (b) next the mass in excess of that needed to consume all of the sodium pool, (c) then as added the mass released to the atmosphere and stored in array MCT, (d) lastly, the mass in excess of that needed to consume all of the sodium vapor which is the amount that is stored in array MCT. Generally, this summary will mention the array names and neglect the variable names.

1. First call: Calculate the mass and energy effects of the chemical reactions in the cell atmosphere and sodium pool and thereby complete the differential equations for the mass and energy of the cell atmosphere and of the cell pool.

- a. The sodium pool reactions are treated first. CCALC stored in array MCP the carbon dioxide released to the pool during the time step and in array MWP the water vapor released to the pool. EMVBAL checks array MPT to ensure that there is a sodium pool and then verifies that there is carbon dioxide or water released to this pool for reaction.
- b. The sodium pool reactions are the following:

When sodium is in excess, the carbon dioxide and water vapor are consumed and the reaction energy goes into the pool. When carbon dioxide and water vapor are in excess, the sodium is consumed in proportion to the molar concentrations of carbon dioxide and water and the reaction energy plus residual pool energy go into the atmosphere. These two reactions evaluate the following portions of the differential equations for the mass and energy of the sodium pool, Equations (3.2) and (3.5):

$$\frac{d}{dt} M_p = -R_{\text{Na} + \text{CO}_2} - R_{\text{Na} + \text{H}_2\text{O}} + R_{\text{C}} + R_{\text{H}_2} + R_{\text{Na}_2\text{O}} \quad (5.5.3)$$

$$\frac{d}{dt} U_p = Q_{\text{Na} + \text{CO}_2} + Q_{\text{Na} + \text{H}_2\text{O}} \quad (5.5.4)$$

- c. On both calls, the atmosphere-pool equilibrium assumption is begun by the following steps:
 - i. The remaining mass of the sodium pool, array MPT, is added to the mass of the sodium in the atmosphere, array MST.

- ii. The mass of the water pool, array MPWT, is added to the mass of water vapor remaining from the pool reaction and to the mass of water vapor in the atmosphere, array MWT.
- iii. The pool energy, array UPT, is added to the atmosphere energy, array UAT.
- iv. The atmosphere volume, array VOLAT, is replaced by the cell volume, array VOLT.
- v. All sodium hydride is disassociated into hydrogen and sodium.

The hydride equilibrium reaction is as follows:

The hydrogen is added to the cell hydrogen, array MGT, the sodium is added to the sodium in the atmosphere, and the hydride energy is added to the atmosphere energy. This disassociation is the reverse of the usual evaluation of the following portions of the differential equations for the mass and energy of the atmosphere, Equations (3.1) and (3.4).

$$\frac{d}{dt} M_a = - R_{\text{Na} + \text{H}_2} + R_{\text{NaH}} \quad (5.5.6)$$

$$\frac{d}{dt} U_a = Q_{\text{Na} + \text{H}_2} \quad (5.5.7)$$

- vi. Arrays MNAH, MPT, MPWT and UPT are set equal to zero.

- d. The sodium oxide reaction is treated next. EMVBAL checks that there is water vapor, that there is sodium oxide, array MNA20, and that this oxide is not associated with a sodium pool. This oxide reaction is the following deposit reaction.

<u>Deposit Reaction</u>	<u>Energy Release</u>	
$H_2O + Na_2O \rightarrow 2 NaOH$	1,500 Btu/lb Na	(5.5.8)

The oxide is consumed or the water vapor is consumed and the excess reactant, the reaction products and the reaction energy go into the atmosphere. The result is the evaluation of the following portions of the differential equations for the mass and energy of the atmosphere, Equations (3.1) and (3.4).

$$\frac{d}{dt} M_a = - R_{Na_2O + H_2O} + R_{NaOH} \quad (5.5.9)$$

$$\frac{d}{dt} U_a = Q_{Na_2O + H_2O} \quad (5.5.10)$$

- e. The sodium reactions in the atmosphere are treated next. EMVBAL checks that there is sodium, that there is carbon dioxide (perhaps left from the pool reaction), that there is oxygen, array MOT, and that there is water vapor (again, perhaps left from the pool reaction). These sodium reactions are the following:

<u>Atmosphere Reactions</u>	<u>Energy Release</u>	
$4Na + CO_2 \rightarrow 2Na_2O + C$	3,800 Btu/lb Na	(5.5.11)

$4Na + O_2 \rightarrow 2Na_2O$	5,700 Btu/lb Na	(5.5.12)
--------------------------------	-----------------	----------

$2Na + H_2O \rightarrow Na_2O + H_2$	3,400 Btu/lb Na	(5.5.13)
--------------------------------------	-----------------	----------

When sodium is in excess, the carbon dioxide, oxygen, and water vapor are consumed. When carbon dioxide, oxygen, and water vapor are in excess, there are two possibilities, as follows:

- i. Molar concentration of water vapor is greater than the molar concentration of oxygen: sodium is consumed in proportion to the molar concentrations of carbon dioxide, oxygen, and water vapor.
- ii. Concentration of oxygen is greater than that for water vapor: sodium is consumed in proportion to the concentrations of carbon dioxide and oxygen; Equation (5.5.13), the reaction with water vapor is skipped in accordance with the experimental results of Test Series E of the hydrogen recombination tests.⁽¹³⁾

These three reactions evaluate the following portions of the differential equations for the mass and energy of the cell atmosphere, Equations (3.1) and (3.4).

$$\frac{d}{dt} M_a = - R_{Na + CO_2} - R_{Na + O_2} - R_{Na + H_2O} + R_C + R_{H_2} + R_{Na_2O} \quad (5.5.14)$$

$$\frac{d}{dt} U_a = Q_{Na + CO_2} + Q_{Na + O_2} + Q_{Na + H_2O} \quad (5.5.15)$$

Each of these reactions produces sodium oxide which, if there is an excess of water vapor, will be reacted to sodium hydroxide in the next time step.

- f. The hydrogen-oxygen reaction in the atmosphere is treated last. EMVBAL checks that there is hydrogen and that the hydrogen-sodium jet conditions are satisfied for the catalytic reaction of hydrogen and oxygen. These jet conditions were developed by the Test

Series B and D of the hydrogen recombination tests.⁽¹³⁾ EMVBAL verifies that the control HOR1 is TRUE by means of the following checks in subroutines LEAKC and LEAKN about the source conditions of leakage entering the present cell:

- i. The leakage source is at 500⁰F or hotter and has
- ii. Hydrogen and
- iii. Sodium vapor of 3.4×10^{-4} lb/ft³ (5.5. g/m³) concentration or greater.

Finally, EMVBAL checks that the molar concentration of oxygen is 11 percent or more. The hydrogen reaction is as follows:

<u>Hydrogen Recombination Reaction</u>	<u>Energy Release</u>	
$2H_2 + O_2 \rightarrow 2H_2O$	54,425 Btu/lb H ₂	(5.5.16)

When this reaction occurs, either all of the hydrogen or all of the oxygen is consumed. The user should note the potential for code upset when hydrogen has accumulated and an air injection introduces enough oxygen to satisfy the oxygen concentration check: the reaction will generate energy and thereby increase the temperature and pressure in one time step as if a hydrogen deflagration were to sweep the containment volume. This reaction evaluates the following portion of the differential equations for the mass and energy of the cell atmosphere, Equations (3.1) and (3.4).

$$\frac{d}{dt} M_a = - R_{H_2 + O_2} + R_{H_2O} \quad (5.5.17)$$

$$\frac{d}{dt} U_a = Q_{H_2 + O_2} \quad (5.5.18)$$

2. Both calls integrate these differential equations to the end of the time step and calculate the cell conditions at the end of the step. Recall the discussion in Section 5.4 about Equations (5.4.1) and (5.4.4). The author uses equations similar to Equation (5.4.1) to describe code processes even though FORTRAN statements in CCALC and EMVBAL similar to Equation (5.4.4) accomplish these code processes while integrating the differential equations. That is, the integrations have already been done and EMVBAL has the following information:

- a. The total energy of the cell, array UAT or U_a , because array UPT or U_p is zero.
- b. The masses of all cell materials: carbon, M_C ; carbon dioxide, M_{CO_2} ; hydrogen, M_{H_2} ; nitrogen, M_{N_2} ; oxygen, M_{O_2} ; sodium, M_{Na} ; sodium hydroxide, M_{NaOH} ; sodium oxide, M_{Na_2O} ; and water, M_{H_2O} .
- c. First call; the cell temperature of the previous time step, T_a ;
Second call: the cell temperature of the first call to EMVBAL, T_a .

EMVBAL iterates to find the new cell temperature T'_a which will give a calculated cell energy, the sum $U'_a + U'_p$, that matches the actual cell energy, U_a . The procedure was taken from the CONTEMPT code.⁽¹⁾ The procedure takes a trial temperature, T_a , and calculates the trial energy sum which is then compared to the actual energy. The trial temperature is decreased (or increased) as the sum $U'_a + U'_p$ is greater (or less) than U_a . The first trial temperature is the value listed in item (c), above. The initial temperature increment for each trial is $2^{\circ}F$. Successive trial temperatures are tested until either the sum $U'_a + U'_p$ brackets U_a , or four trial temperatures have been used. If four trials have been made, the increment is multiplied by four and four additional trials are made. When U_a is bracketed, a trial temperature obtained from linear interpolation of the bracketing values is tested. If convergence is not met, the search continues with one-fourth

the previous increment. Convergence is obtained when the temperature search increment, variable DELT, is less than 0.05°F or when the apparent energy error variable DECR or $U'_a + U'_p - U_a$ is less than 0.01 Btu. If the search temperature would exceed either of the present EMVBAL bounds of 50°F and $5,000^{\circ}\text{F}$, that bound is tried. The iteration is terminated with an error comment if the solution is outside these EMVBAL bounds. The energy sum $U'_a + U'_p$ is calculated as follows:

- a. Each time the temperature search increment is increased or decreased, EMVBAL calculates the total cell heat capacity for all materials except sodium, sodium hydride and water. This total heat capacity, variable CVM or $\overline{Cv}$, is calculated from the energy sum of the materials at the trial temperature, T'_a . The energy sum is:

$$\begin{aligned}
 U = & M_C U_C + M_{\text{CO}_2} U_{\text{CO}_2} + M_{\text{H}_2} U_{\text{H}_2} + M_{\text{N}_2} U_{\text{N}_2} \\
 & + M_{\text{O}_2} U_{\text{O}_2} + M_{\text{Na}_2\text{O}} U_{\text{Na}_2\text{O}} + M_{\text{NaOH}} U_{\text{NaOH}}
 \end{aligned}
 \tag{5.5.19}$$

The sodium hydroxide energy is calculated as if it were liquid and then corrected when the trial temperature is less than 623°F .

- i. If the trial temperature is less than 590°F , the heat of fusion, 68.4 Btu/lb, is subtracted from the liquid energy value.
- ii. If the trial temperature is in the range from 590°F to 623°F , a proportional amount of the heat of fusion is subtracted from the liquid energy value.

The total heat capacity is then:

$$\overline{Cv} = U/T'_a
 \tag{5.5.20}$$

- b. EMVBAL checks for sodium as the cell fluid of first choice. The task now is to partition the sodium into liquid and vapor phases and to calculate their contributions to the energy sum. The mass of all cell sodium was saved in variable SMS and is now returned as variable MS or M_{Na} . EMVBAL calls subroutine NAPROP to obtain all saturated sodium properties at the trial temperature. The specific volumes of sodium liquid, V_f and of vapor, V_g , are needed in order to split the total mass into vapor mass, variable MG or M_g , and condensate mass, variable ML or M_f , that fill the cell volume, variable VOLA or V. The process has two simultaneous equations which are also the source for the M and V of the EMVBAL acronym:

$$\text{Mass balance: } M_{Na} = M_f + M_g \quad (5.5.21)$$

$$\text{Volume balance: } V = M_f V_f + M_g V_g \quad (5.5.22)$$

The solution of these two equations gives the following EMVBAL equation for the condensate mass:

$$M_f = \frac{M_{Na} V_g - V}{V_g - V_f} \quad (5.5.23)$$

There are two possibilities for M_f as follows:

- i. M_f is positive and the vapor mass is calculated from Equation (5.5.21).
- ii. M_f is zero or negative, which means that all the sodium is vapor. In either case, M_f is set to zero. The vapor pressure is calculated from the Ideal Gas Law by using the Ideal Gas Constant of 10.7314 psia-lb/ft³-lb mole-°R and the sodium vapor molecular weight, variable ABAR, calculated by

the previous call to NAPROP. This vapor pressure, variable PS, is used for a second call to NAPROP to secure the superheated vapor properties, in particular the vapor energy, U_g .

The sodium contribution to the energy sum is then:

$$U_1 = M_g U_g + M_f U_f \quad (5.5.24)$$

This sodium partitioning has evaluated the following portions of the differential equations for the mass and energy of the cell atmosphere and pool, Equations (3.1), (3.2), (3.4) and (3.5):

$$\frac{d}{dt} M_a = - C + B \quad (5.5.25)$$

$$\frac{d}{dt} M_p = + C - B \quad (5.5.26)$$

$$\frac{d}{dt} U_a = - U_f C + U_g B \quad (5.5.27)$$

$$\frac{d}{dt} U_p = + U_f C - U_g B \quad (5.5.28)$$

- c. EMVBAL checks for hydrogen. The task now is to calculate the equilibrium quantities of hydrogen and sodium hydride and the contribution of the latter to the energy sum. This equilibrium reaction is Equation (5.5.5) and its portions; the differential equations are Equations (5.5.6) and (5.5.7). The FORTRAN statements in this section are particularly difficult to follow because of the author's propensity to use the same variable for successive different quantities. The mass of all hydrogen was saved in variable SMH and is now returned as variable MH or M_{H_2} . EMVBAL calls subroutine NAHPRO to obtain the partial pressure of hydrogen at equilibrium with sodium hydride, variable PNAHE or P_{H_2} , and the solubility of sodium hydride in sodium, variable SNAHE or

P_{NaH}^e . The partial pressure of hydrogen, variable P_{H_2} or P_{H_2} , is calculated in the atmospheric volume, $V_a = V - M_f V_f$, by using the Ideal Gas Law. This calculated partial pressure is compared to the equilibrium pressure with possibilities as follows:

- i. P_{H_2} is larger than $P_{\text{H}_2}^e$, so that there must be formation of sodium hydride solid by the mass of hydrogen in excess of that necessary to cause the equilibrium pressure. By ratio, then, the mass of hydrogen reacted to form sodium hydride, M'_{H_2} , is:

$$\frac{M'_{\text{H}_2}}{M_{\text{H}_2}} = \frac{P_{\text{H}_2} - P_{\text{H}_2}^e}{P_{\text{H}_2}} \quad (5.5.29)$$

Again, there are two possibilities:

- A. There is enough sodium to react with M'_{H_2} , in which instance the mass of hydrogen remaining, $M_{\text{H}_2} - M'_{\text{H}_2}$, the mass of sodium reacted, 22.808532, M'_{H_2} , and the mass of sodium hydride formed, $M_{\text{NaH}} = 23.808532 M'_{\text{H}_2}$, are all calculated.
- B. There is not enough sodium to react with M'_{H_2} . Now all of the sodium is consumed and its vapor pressure and energy sum, Equation (5.5.24), are set to zero. Also, the mass of hydrogen reacted is based on the mass of sodium, $M_{\text{H}_2} = M_s / 22.808532$, and again, the mass of hydrogen remaining and the mass of sodium hydride formed are calculated.

- ii. $P_{H_2}^e$ is less than P_{H_2} meaning that there is not enough hydrogen to form sodium hydride as a solid phase separate from the atmosphere and condensate phases. Instead, the sodium hydride is formed in solution with a solubility, S_{NaH} which has the following dependence with the partial pressure of the hydrogen in the atmosphere, P_{H_2} :

$$\frac{S_{NaH}}{S_{NaH}^e} = \sqrt{\frac{P_{H_2}}{P_{H_2}^e}} \quad (5.5.30)$$

This is a quadric equation for the unknown mass of hydrogen reacted, M_{H_2}' , which enters through both terms S_{NaH} and P_{H_2} :

- A. The solubility of sodium hydride, M_{NaH} , is the ratio of the mass of sodium hydride, $M_{NaH} = 23.808532 M_{H_2}'$, to the mass of sodium condensate, M_f :

$$S_{NaH} = M_{NaH} / M_f \quad (5.5.31)$$

- B. The hydrogen pressure P_{H_2} is dependent upon the mass of hydrogen remaining, $M_{H_2} - M_{H_2}'$, according to the Ideal Gas Law:

$$P_{H_2} = \frac{(M_{H_2} - M_{H_2}')10.7314 (T_a' + 459.67)}{2.016 (V - M_f V_f)} \quad (5.5.32)$$

The FORTRAN statements have equations (5.5.31) and (5.5.32) already substituted into Equation (5.5.30) and therefore go directly to the quadric formula solution of the latter for M_{H_2}' .

iii. EMVBAL completes the sodium mass balance by taking the mass of sodium reacted from the mass of sodium condensate, M_f , and if this is not enough, from the vapor mass, M_g .

iv. The sodium hydride contribution to the energy sum is the new term to Equation (5.5.24):

$$U_2 = M_{\text{NaH}} U_{\text{NaH}} \quad (5.5.33)$$

- d. EMVBAL checks for water as the cell fluid of second choice. Recall the CACECO assumption: the cell can have sodium (and may have sodium hydride), or the cell can have water, but not both simultaneously. The task now is to partition the water into liquid and vapor phases and to calculate their contributions to the energy sum. The mass of all cell water is the variable MW.

EMVBAL calls subroutine WAPROP to obtain all saturated water properties at the trial temperature. The remainder of the partition process follows that described above for sodium. The water partition uses Equations (5.5.23) and (5.5.21) and results in a water contribution to the energy sum by Equation (5.5.24). With water as the cell fluid, the differential equation for the pool mass is not Equation (5.5.26) but the following portion from Equation (3.3):

$$\frac{d}{dt} M_w = + C - B \quad (5.5.34)$$

- e. The energy sum includes the cell fluid contribution of Equation (5.5.24), possibly the sodium hydride contribution of Equation (5.5.33), and the contribution of the total heat capacity of Equation (5.5.20). EMVBAL actually calculates the apparent energy error, the variable DECR:

$$(\text{DECR}) = U'_a + U'_p - U_a = U_1 + U_2 + \overline{Cv} T'_a - U_a$$

$$= M_g U_g + M_f U_f + M_{\text{NaH}} U_{\text{NaH}} + \overline{Cv} T'_a - U_a \quad (5.5.35)$$

- f. The iteration process for the new cell temperature T'_a is counted and usually succeeds within the allowed limit of 150 trials. EMVBAL prints the salient data of trials 151 and 152 and then takes as the solution the temperature midway between the initial trial temperature, variable TAT(ZP) or T_a , and the 152nd trial temperature, variable TA of T'_a .

The result of this step is the cell temperature at the end of the time step. The cell temperature is both the atmosphere temperature, array TAT, and the pool temperature, array TPT. The first call to EMVBAL is completed but the second call continues.

3. Second call: Evaluate and store cell data as follows:

- a. Liquid and vapor properties for the fluid in excess, either sodium or water, arrays HGT, HWT, ULT and VLT.
- b. Pool boiling properties: arrays TBOLT, UCHNG, UGBOL, ULBOL and VLBOL.
- c. Atmosphere material concentrations: array RATIO.

EMVBAL also calculates the total pressure in the cell by adding the partial pressures of the gases, carbon dioxide, hydrogen, nitrogen, and oxygen, to the vapor pressure of the cell fluid, either sodium or water.

5.6 Subroutine HCALC

Heat structures represent both heat absorption and heat source in the cells. For example, one structure can represent the heat absorption of the cell roof and walls above the pool, another structure can represent the heat absorption of walls and floor submerged by the pool, and still another can represent the heat source of hot equipment, all according to the user's model which was entered for the CACECO analysis. Each heat structure is analyzed by a one-dimensional, multilayer, transient heat conduction equation in the user's choice of planar, cylindrical, or spherical geometry. In the present version of CACECO, each structure can have up to 25 temperature nodes, with as many as 20 regions of different node spacing/materials. Also, there can be none or up to 20 heat structures to model the roof, walls, floor, and equipment of the cells. The boundary conditions of each structure include convection and radiation to any cell atmosphere or pool, to any other structure, to outside air, or constant temperature heat sink.

The main process in subroutine HCALC is the integration to the end of the time step the following partial differential equation for the temperature of each heat structure.

$$g(x) \frac{\partial}{\partial t} u(x,t) = \nabla \cdot k(x) \nabla u(x,t) + S(x,t) \quad (5.6.1)$$

where

u = temperature
x = space variable
t = time variable
g = volumetric heat capacity
k = thermal conductivity
s = source term per unit volume.

This one-dimensional, transient heat conduction equation assumed in the CACECO analysis is a direct inheritance of the CONTEMPT code.⁽¹⁾ The author believes that it is one of the best features of the CACECO code because it provides the opportunity for sophisticated models of LMFBR facility structures without the complexity of two-dimensional representation. The numerical solution of this partial differential equation is described in Appendix E.

The second important process of HCALC is the calculation of heat transfer rates from both surfaces of each structure so that the energy changes for each cell atmosphere and pool can be evaluated as shown by Equations (5.4.59) and (5.4.60).

Subroutine HCALC is called by CCALC with one of two possible objectives as described in Section 4.0 and shown in Figure 4.1:

1. Objective when control SW1 is TRUE: calculate the temperatures in all heat structures at the steady-state conditions just prior to the accident transient. HCALC is called once to obtain these temperatures for each case.
2. Objective when control SW1 is FALSE: calculate the temperatures at each time step of the accident analysis. Obviously, HCALC is called many times to obtain these transient temperatures for each case.

The heat structures of an analysis are defined by input data that are read and processed by subroutine HINPUT as described in Section 5.3. The temperature node locations are illustrated in Figure 5.1. These nodes locations run from x_1 , the left-hand boundary (the inside radius of cylindrical and spherical structures) to x_N , the right-hand boundary (the outside radius of cylindrical and spherical structures). These node locations include both boundary surfaces of the structure, all interfaces between regions, and equal intervals between the interfaces and/or boundaries. A region is a structure part which contains the same material and has constant intervals

between node locations. A new region is required for a different material and/or different node intervals.

The HCALC operations use a variety of controls for the FORTRAN DO loops that accomplish array processes.

1. The DO loops that index each heat structure use the following controls:
 - a. II, the limit set in the data statement of the main program CACECO for the dimension of heat structure arrays.
 - b. NSL, the number of heat structures described for the case, the number entered on input Card 1.
 - c. NHS, the total number of structures, both described and for sodium pools, for which the heat conduction equation is solved.
2. The DO loops that index each temperature node of a particular heat structure use the following controls:
 - a. III, the limit set in the data statement of the main program CACECO for the dimension of heat structure arrays.
 - b. NON(I), the number of nodes described for the heat structure, the number entered on input Cards 1NN000.
 - c. NN, the same as NON(I).
 - d. NNN, one less than NN.

The special HCALC operations are summarized below for when control SW1 is TRUE and the CCALC call objective is to obtain the steady-state temperatures of the control NSL number of described heat structures.

1. Arrays are zeroed to hold the integrated heat transfer rates of both surfaces of the structures--arrays IHT1 and IHTN for the integrated convection rates, arrays RIHT1 and RIHTN for the integrated radiation rates, and arrays PIHT1 and PIHTN for the integrated convection rates of the sodium pool structures. In these array names, the "1" denotes the array for the left-hand boundary and the "N" denotes the array for the right-hand boundary.
2. Controls are set for iterating the steady-state temperatures. The surface temperatures of all described heat structures, arrays US1 and USN, are summed into variable START1. These surface temperatures were set to 120⁰F in the main program CACECO and then may have been reset for particular heat structures by input cards between Card 500000 and Card 600000 of the input card deck. Control ICK1 is set to 99 and then not reset to the material index entered on input Card Series 2301 to identify the concrete materials. This means that the thermal conductivities entered for these concrete materials on input Card Series 300001 are used for the steady-state calculations of the thermal conductivities that may have been entered on Card Series 2301.
3. The control NSL is used for the FORTRAN DO loop that solves the heat conduction equation for each structure because there are no pool heat structures at this time. In the course of this solution, the volumetric heat capacity of the structure, array CS, is set to zero, the boundary temperatures returned by subroutine BONDY, T1, TN, and arrays RT1 and RTN, are reduced to half value, and the structure temperatures, array US, are zeroed. This DO loop solution provides the steady-state structure temperatures. The surface temperatures, array locations US(1,I) and US(NN,I) are also stored in arrays US1 and USN, respectively.
4. The iteration loop is checked. The surface temperatures, arrays US1 and USN, are summed into variable START2. HCALC prints the iteration number, variable NSTART, and the temperature summation, variable START2,

to show the convergence of these steady-state calculations. The iteration is terminated when successive temperature summations agree to within 1°F or when five iterations have been done, whichever occurs first. The DO loop is done once more and all results are printed.

The standard HCALC operations are summarized below. These operations occur whether control SW1 is TRUE or FALSE.

1. Arrays SHTR and POOL are zeroed in preparation for summing heat transfer rates into SHTR and rate changes into POOL. These two arrays are used by CCALC and its ancillary subroutines STORE and FETCH in evaluating heat transfer effects on cell atmospheres and pools. This evaluation was discussed in conjunction with Equations (5.4.57) and (5.4.58) and also Equations (5.4.59) and (5.4.60).
2. HCALC calls subroutine DECPOW to obtain the decay power, variable POWIN, at the end of the time step. This subroutine was described in Section 5.4. The variable POWIN is averaged with variable POLD, the power at the beginning of the time step, for the volumetric heat source term of the heat conduction equation. HCALC calls subroutines TABLU and TABLG to interpolate array OATH in order to obtain the outside air temperature, array location TAT(1), and outside heat transfer coefficient, variable HOUT, which were entered on Card Series 1601. These outside air temperature and coefficient data are assumed to have a 24-hour cycle which is effected by the function call AMOD(T,24.0). The outside air data permit the user to analyze an accident that occurs at 4:00 a.m. or at any other time during the day.
3. HCALC updates the heat structure representation of the sodium pool in each cell, even though the pool may not exist or may even be a water pool. As noted before, this heat structure representation of each sodium pool has little use in the present version of the CACECO code, and the hurried reader might well skip this particular HCALC operation. HCALC checks each cell and if the sodium pool exists because

array MPT is greater than zero, then control NHS is incremented in order to include the pool heat structure in the FORTRAN DO loop that solves each heat conduction equation. HCALC calls subroutine HPOOL to update the pool heat structure description even though the pool may not exist. These HPOOL operations are summarized below.

- a. If the sodium pool does not exist, the only HPOOL operation is to set the structure surface temperature, arrays US1 and USN, to the temperature of the atmosphere.
- b. If the sodium does exist, the present depth of the pool is calculated from the pool mass, array MPT or M_p , its specific volume stored by subroutine EMVBAL, array VLT, or V_f , and its surface area entered on the odd numbered input card (Card 3, 5, 7 or 9), array APT or A_p .

$$D = M_p V_f / A_p \quad (5.6.2)$$

If this pool depth has changed by 3% or more, the pool heat structure parameters are renewed. NPOOL prints the pool heat structure title, the time, and present pool mass.

- i. The structure temperature nodes are set at intervals of $D/6$ and the surface gradient weight, array SS, the left volume gradient weight, array VS, and the right volume gradient weight, array WS are calculated.
- ii. The thermal conductivity and volumetric heat capacity at the pool temperature, array TPT, are calculated and stored in array KSCS at locations above the thermal properties entered on Card Series 300001.

HPOOL resolves the CACECO code dichotomy of having two average sodium pool temperatures as follows.

- i. One average temperature, array TPT or T_p , results from the pool energy balance and the processes of subroutine EMVBAL. The single value is taken as the correct, fiducial temperature of the well mixed pool and is the pool temperature used by all code processes that don't specifically designate the top or bottom surface temperatures of the sodium pool. In particular, this is the pool temperature designated by heat structure boundary temperature Indices 102, 105, 108, and 111 for the pools in Cells A, B, C and D, respectively.
- ii. Another average temperature is the trapezoidal average of the seven nodal temperatures of the sodium pool heat structure, array US. If the temperature in array location US(J,I) is designated T_J , the trapezoidal average is:

$$T'_p = (1/2 T_1 + T_2 + T_3 + T_4 + T_5 + T_6 + 1/2 T_7)/6 \quad (5.6.3)$$

The temperature dichotomy is resolved by shifting the heat structure temperatures so that their average matches the temperature provided by subroutine EMVBAL. For example, the shifted temperature of Node J which is stored in array location US(J,I) and used for subsequent HCALC Operations is:

$$T'_J = T_J + T_p - T'_p \quad (5.6.4)$$

This resolution, unfortunately, sometimes leads to unusual pool heat structure temperatures as illustrated by Debug Case 10 of the "Analytical Validation..."⁽²⁾

4. HCALC enters the FORTRAN DO loop that solves the heat conduction equation and calculates the heat transfer from each of the NHS number of heat structures. Actually, HCALC passes through this DO loop twice with different operations which are summarized below.

<u>Operation</u>	<u>First Passage</u>	<u>Second Passage</u>
a. Set controls for array operations	Yes	Yes
b. Set volumetric heat source, variable S, and heat capacity and thermal conductivity of structure materials, arrays CS and KS, respectively.	Yes	
c. Evaluate boundary conditions of heat transfer coefficients and temperatures for convection and radiation from all structures.	Yes	Yes
d. Set up the finite difference equations to solve the heat conduction equation. These are Equations (E9), (E11), and (E12) of Appendix E. Solve this tridiagonal set of equations shown as Equation (E13).	Yes	Yes
e. Calculate the heat transfer rates from both surfaces of each structure and sum these rates into arrays SHTR and POOL.		Yes
f. When control SW3 is FALSE, print the heat structure data.		Yes

The FORTRAN statements that accomplish the DO loop operations of (c), (d), (e), and (f), above, appear numerous only because they cover all of the various boundary conditions of the present code. These various conditions are the following:

- Two surfaces--each structure has a left-hand surface and a right-hand surface. Each surface has its own variables, arrays and subroutine calls to BONDY and HTRAN. Each surface has its own finite difference representation of the heat conduction equation..
- Two kinds of structures--there are (NSL number of) structures described by the input data and (NHS - NSL number of) structures representing sodium pools. The described heat structures have only one convection boundary condition for each surface but the pool structures may have as many as five convection boundary conditions for each surface. The arrays NP1 and NPN have the count of these convection boundary conditions for each pool structure.
- Two kinds of heat transfer--there are convection heat transfer conditions, summarized above, and there are as many as six radiation heat transfer conditions for each surface. The arrays NR1 and NRN have the count of these radiation boundary conditions for each described and pool structure.

The two (surfaces) plus two (kinds of structures) plus two (kinds of heat transfer) make six various boundary conditions. The FORTRAN statements include six calls to subroutine BONDY, three conditions for the left-hand boundary Equation (E11) plus three conditions for the right-hand boundary Equation (E12), six calls to subroutine HTRAN, two FORMAT statements for printing temperatures and coefficients, and four FORMAT statements for printing heat transfer rates and integrated rates. These DO loop operations provide new structure temperatures, in array US, and new sums of heat transfer rates, in arrays POOL and SHTR.

5. HCALC stores the surface temperatures from array locations US(1,I) and US(NN,I) into array locations US1(I) and USN(I), respectively. Also, HCALC checks the new temperature in array location US(N,I) against the previous maximum temperature at the same node in array location

USM(N,I) and if the new temperature is higher, stores it into the maximum temperature array.

HCALLC calls subroutine BONDY to evaluate the boundary conditions for each surface of each heat structure. The first argument of the FORTRAN calling statement is an integer that identifies the following boundary conditions:

<u>Argument</u>	<u>Boundary Condition</u>
1	convection at left-hand surface of one of the NSL number of described structures.
2	convection at right-hand surface of a described structure.
3	convection at either surface of a pool heat structure. The second argument, variable K, is set to values of 1, 2, 3 or 4 to identify Cells A, B, C or D as having the pool.
4	radiation at either surface of any heat structure. The common variable ARAR is set to the surface area of the particular surface attended by the call.

In BONDY, the first argument is the integer control Z. The objectives of this subroutine are to use the integer indices of arguments BH and BT in order to evaluate the following:

<u>Item</u>	<u>Name</u>
H	boundary heat transfer coefficient, an argument
TN	either: the sum of the old boundary temperature (of previous time step) and of the new temperature (predicted by subroutine EMVBAL) or else twice the value of a constant boundary temperature.

TINEW left hand boundary temperature for subroutine HTRAN, a
 labeled common variable.

TNNEW right-hand boundary temperature for subroutine HTRAN, another
 labeled common variable.

The BONDY operations are summarized below:

- a. The boundary temperature is evaluated from its index argument BT. Note that the author's nomenclature for these temperatures is confusing: this boundary temperature is denoted in Appendix B as the "bulk temperature" because in many analysis models it connotes the bulk atmosphere temperature or bulk pool temperature. Further, the structure surface temperature is denoted in Appendix B as the "boundary temperature."
 - i. An argument value of -99 to -1 or of +1 to +99 indexes the left-hand surface temperature, array USL, or the right-hand surface temperature, array USN, respectively, of the same numbered heat structure.
 - ii. An argument value of 100 indexes the outside air temperatures, array locations TAT(1) and TATO(1), which had been entered on input Card Series 1601.
 - iii. An argument value of 101 to 112 indexes the atmosphere temperature of a particular cell, arrays TAT and TATO; indexes the pool temperature, arrays TPT and TPTO, if this pool exists because of sodium mass, array MPT, or of water mass, array MPWT; or indexes the constant temperature of a particular sink. These possibilities are shown below.

<u>Boundary Temperature</u>	<u>Cell A</u>	<u>Argument BT Values</u>		<u>Cell D</u>
		<u>Cell B</u>	<u>Cell C</u>	
Cell atmosphere	101	104	107	110
Cell pool	102	105	108	111
Constant sink	103	106	109	112

- b. BONDY evaluates the boundary coefficient from its index argument BH:
- i. An argument value of 0 denotes the insulated surface and actually is evaluated first because the boundary conditions are easy: the coefficient is zero and the temperature is taken to be the structure's own surface temperature.
 - ii. An argument of +1 to +99 indexes a constant coefficient of array HTC which was entered on input Card Series 400001.
 - iii. An argument value of 100 indexes the outside air heat transfer coefficient, variable HOUT, which was entered on input Card Series 1601.
 - iv. An argument value of 101 through 109 indexes the time/temperature dependent coefficients of array NTCT which were entered on input Card Series 410001 through 490001. BONDY evaluates the dependency (time or temperature) value and stores it in the labeled common variable ST01 to be used as the argument for subroutine TABLU to interpolate the coefficient value needed. Again note that the author's nomenclature for these temperature dependencies is confusing. The boundary temperature here is denoted in Appendix B as the "bulk temperature" and the surface temperature here is denoted as the "boundary temperature." There are the following kinds of dependency in the present version of the code.

<u>Kind</u>	<u>Dependency of Coefficient</u>
1	time, variable T.
2	average of surface temperature, labeled common variable TSUR, and boundary temperature of BONDRY operation (a), above.
3	difference between surface and boundary temperatures.
4	same as Kind 3.
5	surface temperature.
6	boundary temperature.

- v. An argument value of 110 indexes the coefficients of array HPA. This array is set to zero in CCALC. The presently inactive subroutine POOLR would evaluate the heat transfer coefficient between the sodium pool and atmosphere of each cell and store this coefficient in array HPA for retrieval by BONDRY.

4. BONDRY calculates the radiation equivalent heat transfer coefficient. This calculation uses the boundary temperature, variable TN or T_1 , which was evaluated by BONDRY Operation (a), the surface temperature, variable TSUR or T_2 , and the total interchange area, variable H or $A_2 F_{21}$, which was evaluated by BONDRY Operation (2). The total interchange area⁽¹⁴⁾ is shared by heat structure surface area A_1 and by surface area A_2 , both. The total interchange area depends upon the shape of the cell enclosure and the emissivity and absorptivity of the source and sink surfaces. The CACECO code assumes that the source Surface J and sink Surface K are both "grey," so that:

$$A_J \epsilon_{JK} = A_K \epsilon_{KJ} \quad (5.6.5)$$

Further, the CACECO code input assumes that the same total interchange area value will be indexed by both heat structure Surface J and heat structure Surface K. These total interchange areas are entered the same as heat transfer coefficients:

- on Card Series 1601 for array OATH in order to secure a 24-hour cycle time dependence
- on Card Series 400001 for array HTC for constant values, the usual entry used by the author, and
- on Card Series 410001 through 490001 for array HTCT in order to secure a time or temperature dependence.

The calculation of the radiation equivalent heat transfer coefficient uses the following equation which assumes that the temperatures, T_1 and T_2 , are in degrees Rankin and that the Stefan-Boltzmann constant is 0.1713×10^{-8} Btu/hr-ft²-°R⁴:

$$H_2 = 0.1713 \times 10^{-8} A_2 \mathcal{F}_{21} (T_1^2 + T_2^2)(T_1 + T_2)/A_2 \quad (5.6.6)$$

HCALC calls subroutine HTRAN to evaluate the heat transfer rates for each surface of each heat structure and to sum these rates into arrays POOL and SHTR. HTRAN evaluates the heat transfer rate for each convection and radiation boundary condition. All of these evaluated rates are based on: the surface area of the structure, labeled common variable ARAR or A_s , the heat transfer coefficient, argument H, the surface temperature just calculated by HCALC, labeled common variable TSUR or T_s , and the boundary temperature, argument TN or T_b . The CACECO code defines the heat transfer rate as follows, so that the positive value means heat loss from a structure into the atmosphere or pool:

$$Q_{HS} = A_s H (T_s - T_b) \quad (5.6.7)$$

6.0 HELPFUL HINTS

The general procedure for running a CACECO code case begins by setting up the input card data deck for the first run. This input data describes those portions of the LMFBR facility considered important in the analysis model for the accident scenario and the initial accident effects of the scenario. For example, the affected sodium equipment cell may be sufficient in the model for the scenario of a small sodium spill. In comparison, this same cell, the adjacent cell, and the reactor containment building may be needed in the model for the scenario of a large sodium spill that involves failure of cell containment and leakage from one cell to the next, to the building with chemical reaction effects, and perhaps leakage to the outside. Section 2.3 noted that this task of setting up the input card deck requires time for efforts such as: reading drawings and memorandums, interrogating authorities, copying data of earlier cases, calculating new data, and, perhaps, guessing remaining data needed by the code for the scenario. Again, the user is urged to include comments freely in order to identify data sources, the accident scenario, and its unique features.

The next step is to debug the input data deck by attempting to run the case for short times. If the user has been careless or the user's supervisor wanted CACECO results yesterday, this debug step can be heartbreaking. The code input subroutines, CINPUT and its ancillaries, choke on unseen card punch errors and at guileless data errors--the author is usually exhilarated when the deck runs. However, these input checks are very elementary and the user must scrutinize the data again to be sure that they are perfect.

Complicated cases, cases with extended running times, expensive cases--all are best accomplished by a sequence of short computer runs which use the CACECO restart capability to continue and to modify the analysis model at each restart during the course of the analysis. Indeed, this restart capability permits very complicated scenarios with details and twists that cannot be predicted in advance. These cases have the aspect of a major laboratory experiment with daily chores of operating, checking, correcting,

and maintaining during the course of the analysis. A recent case needed 37 runs made over a 54-day period of part-time attention--and these 37 were just the runs which were considered worth saving and continuing on from. The author's usual experience with extended cases and with cases having delicate scenarios is that more than half the runs are faulty and must be scrapped because of restart data errors and values that should have been larger or smaller to obtain the desired scenario result; further iteration is needed to find the proper values, so that the case can be continued. The author limits each case run to less than five minutes of computer time and expects to check the run results in order to decide whether to modify and re-run or to restart and continue.

The CACECO restart capability uses the binary file DTAPE which must be preserved from run to run. The code summary results of each run are saved on the binary file PTAPE and these separate run results must be assembled for subsequent input to auxiliary programs that print summary reports and plot report figures. The saving, reuse, and assembling chores on the DTAPE and PTAPE files should be done by standardized computer system routines that are used by habit, so that all checking, thinking, and inventing efforts are focused onto the CACECO input and results.

There are a variety of data phenomena that cause CACECO subroutines to choke and sputter. Unfortunately, the CACECO printed results do not always show that malfeasance has occurred. Further, when occurrence is noted, the printed results do not make obvious any remedy so that the re-run can be successful. The data phenomena of most concern are those that cause oscillations. The usual method for detecting oscillations is by inspection of the periodic two-page, 116-line cell summary of array SUMMRY, printed by subroutine OUTSUM. The clues are unexpected changes in the variables STEP and CONV(1), strange values of cell temperatures and pressures, and, perhaps, oscillating values of these variables. Some of the offending data phenomena are listed below.

1. Step changes in the values of a data table entered on some card series. Code upsets and oscillations have been caused by step changes in the following data tables:
 - a. Accident effects
 - sodium sprays of Card Series 401 through 701
 - sodium spills of Card Series 801 through 1101
 - air injections of Card Series 1201 through 1501
 - outside air temperatures of Card Series 1601
 - space cooler operations of input Cards 1700 through 2000
 - b. Facility properties
 - leakage coefficients of Card Series 11 through 91
 - water and carbon dioxide releases of Card Series 2501 and 2701
 - c. Heat transfer coefficients and total interchange areas
 - outside air coefficients of Card Series 1601
 - time and temperature dependent coefficients and areas of Card Series 410001 through 490001
2. Step changes in the values of input data entered by a restart. The code capability to modify nearly all of the input data provides easy opportunity to upset code operations at each restart.

In addition to the data tables, listed above, the following input data have initiated upsets and oscillations by being introduced at a restart.

a. facility properties:

leakage nozzle areas of Cards 10 through 90

b. heat transfer coefficients and total interchange areas:

constant coefficients and areas of Card Series 400001

The usual remedy for the upsets and oscillations caused by step changes is to convert the step to a ramp change with a slope of 2% to 10% per time step. In many cases, the user must resort to trial and error to determine input data that work best.

Heat transfer rates can be so large that they permit mortal oscillations. Heat transfer rates are controlled by input values of heat transfer coefficients and by input values of total interchange areas. Coefficients and areas that are too large permit temperature and energy oscillations that are unstable even though the implicit method of solving the heat conduction equation, described in Appendix E, is stable for any size time step. The operative definition for stability in this solution method is true; a perturbation causes oscillations that die away. The unstable oscillations occur when a cell and a structure scuffle or when two structures tussle over the location of a swing quantity of energy that moves back and forth at each time step. The only remedy for these oscillations is to reduce the magnitude of the offending heat transfer coefficient or total interchange area.

A common heat transfer situation with a potential for vexing, possibly unstable, temperature oscillations, has a cell heated by one or more hot structures and cooled by one or more cold structures. There is no concern when the cell heat capacity is large compared to the heat transfer rates. There is no concern when the heating rate far exceeds the cooling rate and

vice versa. There should be concern when the cell acts as intermediary to an overall energy flow from the hot structures to the cold structures because a small perturbation in cell temperature is magnified by the resulting heat transfer rates. The author has used either of two remedies: the first minimized the importance of the overall energy flow by reducing the heat transfer coefficients; the second invents radiation heat transfer paths between the structures that bypass the cell atmosphere and pool.

Leakrates can be so large that they permit mortal oscillations. These leakrates are controlled by input values of nozzle areas and by input values of leakage coefficients. Areas and coefficients that are too large permit atmosphere and energy oscillations because there is no flow friction in the present version of the code and the pressure of two cells may never equalize because material and energy can flow back and forth between the cells with each time step. Chemical reactions in one cell, or the other, or in both, only magnify the potential for leakage oscillations. A common leakage situation with this potential has hot sodium vapor in a cell and air in the building--cell leakage to the air reacts and may raise the building temperature and pressure enough to drive air leakage into the sodium vapor which reacts and may raise the cell temperature and pressure enough to begin the leakage cycle again with the next time step. The two remedies for these leakage oscillations are, first, to reduce the offending nozzle area and leakage coefficient and second, modify subroutines LEAKC and LEAKN so as to eliminate the reverse leakage.

Concrete heat structures can be modeled with temperature node intervals so large that water (and possibly carbon dioxide) is released to bothersome surges as the structure is heated. The potential for surges depends upon the node intervals, the concrete heating rate over these intervals, and the water release function. The usual release function for basalt concrete, shown in Appendix F as the input data on Card Series 2501, has more than half of the water being released as the concrete heats from 214⁰F to 300⁰F. This small temperature range has great leverage upon the water release from any incremental volume between two temperature nodes of the

concrete structure. The CACECO code method for calculating water release, described in Section 5.4 in conjunction with Equation (5.4.35), makes 214⁰F a threshold temperature for water release from basalt concrete; very little water is released until the left-hand temperature of the incremental node volume, temperature T_{N-1} , exceeds 214⁰F. Then this left-hand temperature exerts leverage upon water release from the whole node volume. The result can be a sudden surge of water out of the structure to the destination specified by data on Card Series 2601; perhaps the atmosphere or sodium pool of a sodium equipment cell involved by a sodium spill accident and then the mass and energy effects of the water surge become amplified by chemical reaction effects and the overall result may annoy the code user. The usual remedy for troublesome surges is to remodel the structure to have smaller node intervals.

The author urges good sense in setting up and running each CACECO case because the code cannot default to an operable data set. The code results will be no better than the analysis model, and won't surpass the accident scenario predicted.

Lastly, the Users' Guide has been prepared to explain the code processes and methods and many FORTRAN comment statements have been incorporated into the code, to aid the user who wants to modify the code to suit his particular needs. There have been many users who have changed the code to suit new purposes, with the author's encouragement. The code can be improved. Of course, each user must test the results, to be sure that his code alteration hasn't introduced new errors. The author welcomes users' improvements for incorporation into the next version of the CACECO containment analysis code.

7.0 REFERENCES

1. L. C. Richardson, L. J. Finnegan, R. J. Wagner, and J. M. Waage, CONTEMPT, a Computer Program for Predicting the Containment Pressure - Temperature Response to a Loss-of-Coolant Accident, IDO-17220, Phillips Petroleum Company, Idaho Falls, Idaho, June 1967.
2. R. D. Peak, Analytical Validation of the CACECO Containment Analysis Code, HEDL-TME 79-2, Hanford Engineering Development Laboratory, Richland, WA, May 1979.
3. D. M. Young and R. T. Gregory, A Survey of Numerical Mathematics, Volume I, Addison-Wesley Publishing Company, Reading, Massachusetts, 1972. Euler Method: Equation 5.8 on page 442. Modified Euler Method: Equation 6.8 on page 451.
4. P. Beiriger, et al., SOFIRE-II User Report, AI-AEC-13055, Atomics International, Canoga Park, CA, March 30, 1973.
5. L. Baurmash, et al., HAA-3 User Report, AI-AEC-13088. Atomics International, Canoga Park, CA, 1973. (HAA-3C denotes the current version in use at HEDL).
6. G. W. Spangler, et al., Description of the COMRADEX Code, AI-67-TDR-108, Atomics International, Canoga Park, CA, 1967. (COMRADEX-H denotes the current version in use at HEDL).
7. R. H. Perry and C. H. Chilton, editors, Chemical Engineer's Handbook, 5th Edition, McGraw Hill Book Company, New York, 1973; p. 5-29.
8. O. A. Hougen and K. M. Watson, Chemical Process Principles, Part One, Material and Energy Balances, John Wiley & Sons, Inc., New York, 1943, pp 214 to 218.
9. JANAF Thermochemical Tables: (first edition) Thermal Research Laboratory, Dow Chemical Company, Midland, Michigan, August 1965; Second Edition, National Bureau of Standards, NSRDS-NBS37, June 1971, Superintendent of Documents, U.S. Government Printing Office, Washington D.C., 20402 (Catalog No. C 13.48:37)
10. G. H. Golden and J. V. Tokar, Thermophysical Properties of Sodium, ANL-7323, Argonne National Laboratory, Argonne, IL, August 1967.
11. A. C. Whittingham, "An Equilibrium and Kinetic Study of the Liquid Sodium-Hydrogen Reaction and Its Relevance to Sodium-Water Leak Detection in LMFBR Systems," J. of Nuclear Materials, Vol. 60, pp. 119-129, 1976.

12. R. B. McClintock and G. J. Silvestri, Formulations and Iterative Procedures for the Calculation of Properties of Steam, American Institute of Mechanical Engineers, 345 East 47th Street, New York, N.Y., 1968. (These are the formulas for the 1967 ASME steam tables.)
13. R. W. Wierman, Experimental Study of Hydrogen Formation, Jet Ignition, and Jet Extinguishment Under Postulated Breeder Reactor Accident Conditions, HEDL-TME 78-80, Hanford Engineering Development Laboratory, Richland, WA, May 1979; Test Series B, D and E.
14. R. H. Perry and C. H. Chilton, op cit; p. 10.54.

APPENDIX A

FLOW DIAGRAM FOR THE CACECO CODE

APPENDIX A

FLOW DIAGRAM FOR THE CACECO CODE

Note that the following subroutines are machine or system dependent and must be checked during conversion of the code to another installation.

<u>Subroutine</u>	<u>Purpose</u>
CVIRDP	Reads integer and real number data in free field format on cards. Requires reread (Univac) or decode (CDC) capability. There are IBM versions of this subroutine existent.
DOY	Obtains date from computer.
ETIMEF	Obtains computer processing time.
FLD	Obtains certain bits in a computer word.
FLDL	Moves certain bits in one computer word to another.
FPTCK	Verifies that a number is real.
ITLIMIT	Obtains the job/time limit.
MOVE	Moves characters by using the FLDL function.
RTCHEK	Checks job time remaining, calls TCHECK.
TCHEK	Checks job time remaining, calls ITLIMIT.
TOD	Obtains clock time from computer.

APPENDIX B

CACECO INPUT FORMAT

APPENDIX B

CACECO INPUT FORMAT

B.1 Introduction

The case title card, all structure title cards and comment cards can have any alphabetic-numeric information desired because these are reproduced in the output. All other cards carry numerical data in the free field CVIRDP format described in Appendix C. In this free field format, each entry on a card is considered to be a numerical value and, except for the card number, is a real (floating point) value unless specified otherwise. The card number is the first number on each card.

Generally, if the data of any section are not required for the case, the cards of that section are omitted from the input data deck. The mandatory cards and data are listed below.

<u>Section</u>	<u>Card Number</u>	<u>Data</u>
B.2		Case title card.
B.4	1	TFNL, NSL and IPOOL(4).
B.27	9999 or 10000	End of tables card.
B.37	500000	End for the CINPUT subroutine.
B.39	600000	End of steady state temperature input.
B.41	1000000	Final card for new or restarted case.

B.2 Case Title Card

A dollar sign (\$) or an asterisk (*) is required in Column 1 of this card. The remaining 79 columns can have any alphabetic, numeric or special characters desired because the information on this card will be printed as the heading of each page of output data. At the start of the card input, cards are checked until one is encountered with a dollar sign or asterisk in Column 1. This card is taken to be the title card for the case data that follows. This feature allows the user to stack several short cases in succession and the code will start the next case after encountering an input error.

B.2 Case Title Card (cont'd)

Note that the CVIRDP subroutine (described in Appendix C) which is used to read input data cards, will print out cards having an alphabetic C (C) or asterisk in Column 1. This feature permits comments to be interspersed among the data cards without affecting code input.

The author tends to use the dollar sign to denote the case title card and the alphabetic C to denote a comment card. Examples follow.

CARDS ILLUSTRATING INPUT DATA FOR CACECO CODE.

```
C
$THE DOLLAR SIGN IN COLUMN 1 MAKES THIS A TITLE CARD.
*THE ASTERISK IN COLUMN 1 MAKES THIS A TITLE OR COMMENT CARD.
CTHE ALPHABETIC C IN COLUMN 1 MAKES THIS A COMMENT CARD.
C COMMENT CARDS ARE REPRODUCED IN OUTPUT.
```

The CVIRDP subroutine stops reading input data on a card once it encounters an asterisk or alphabetic C in Columns 2 through 80. This feature permits card identification comments to be placed on the card without affecting code input and output. Examples of these local comments follow.

```
      * THE ASTERISK OR ALPHABETIC C IN COLUMNS 2 THRU 80
      C DENOTE COMMENTS THAT ARE NOT REPRODUCED IN OUTPUT.
      * COMMAS AND/OR SPACES SEPARATE NUMBER WORDS.
1,12,123 . 1 12 123    1 12 123  * IDENTICAL INTEGER TRIPLETS    ;
1.0,12.0,123.0  1.0 1.2E+1 1.23E+2    * IDENTICAL REAL VALUES
              1+1 12+2 123+3    * IDENTICAL REAL VALUES
```

B.3 Card 0, Cell Labels

Word	Description
------	-------------

1	0
---	---

This card signals the CINPUT subroutine to read the next card for new cell labels using the format: 4(4X,A6). An example follows:

0	CEL AA	CEL BB	CEL CC	CFL DD	* CARD SIGNALS NEW CELL LABELS ON NEXT CARD.
					* NEW CELL LABELS.

B.4 Card 1, Final Time, etc.

<u>Word</u>	<u>FORTTRAN</u>	<u>Description</u>
1		1
2	TFNL	Final time, the time at which the case will end (hours).
3	NSL	The number of heat conducting structures (integer). Presently NSL can have a maximum value of 20. If the value of NSL is zero, no heat conduction calculations are made and the cards of Sections B.28 through B.37 are omitted. NSL can have a maximum value of (II-4) or of 95, whichever is smaller. The value of II is set by a data statement in the CACECO element to reflect the dimension of variable NON(24), etc. of COMDECK HEAT. Presently II = 24.
4	IPOOL(4)	Has two possible values (integer): 0 Denotes a new case starting at 0.0 hours, a new final time or a new number of heat structures. 1 Denotes a case restart with the code continuing from when it left off.

Generally two Card 1's are required for the restart of a case. On the first Card 1, the Word 4 signals the CINPUT subroutine to read the restart data stored on the DTAPE file at the end of the previous case run. On the second Card 1, the Word 4 must be zero to enable the CINPUT subroutine to read the new final time or new number of heat structures for the restarted case.

Examples follow.

1	1.2	3	0	* RUN TO 1.2 HOURS WITH 3 STRUCTURES.

1	12.3	5	1	* RESTART FROM END OF LAST RUN.
1	24.6	5	0	* RUN ON TO 24.6 HOURS WITH 5 STRUCTURES.

B.5 Cards, 2, 4, 6 and 8, Cell Data

The code provides for the four cells: Cell A, Cell B, Cell C, and Cell D. There is a minimum of specialization among these cells and a case can use any or all of these cells as necessary. The author prefers to use Cell A for the accidental sodium spray or spill and Cell C for the reactor containment building.

B.5 Cards 2, 4, 6 and 8, Cell Data

Cards 2, 4, 6, and 8 are quite similar. Each inputs conditions for a particular cell.

<u>Word</u>	<u>Cell A Description</u>
1	2
2	Atmosphere volume of Cell A (ft ³).
3	Pressure of Cell A atmosphere at outset of run (psia).
4	Temperature of Cell A atmosphere at outset of run (°F).
5	Oxygen content of Cell A atmosphere at outset of run (volume percent).
6	Constant temperature (°F) value for bulk temperature Index 103. This value is available as a constant temperature source or sink boundary condition as described in Section B.33 for the heat structures.

In the FORTRAN variable, the subscript J ranges 1, 2, 3 and 4 for Cells A, B, C and D, respectively. Likewise, subscript JP ranges 2, 3, 4 and 5.

<u>Word</u>	<u>FORTRAN Variable</u>	<u>Cell B Description</u>	<u>Cell C Description</u>	<u>Cell D Description</u>
1		4	6	8
2	VOLT(J)	Gas Volume	Gas Volume	Gas Volume
3	PTT(JP)	Pressure	Pressure	Pressure
4	TAT(JP)	Temperature	Temperature	Temperature
5	MOT(J)	Oxygen Content	Oxygen Content	Oxygen Content
6	TCONT(J)	Index 106 for bulk temperature	Index 109 for bulk temperature	Index 112 for bulk temperature

Examples of these cards follow Section B.6.

B.6 Cards 3, 5, 7 and 9, More Cell Data

Cards 3, 5, 7 and 9 are quite similar. Each inputs conditions for a particular cell.

<u>Word</u>	<u>Cell A Description</u>
1	3
2	(lbs) Net positive value: quantity of sodium introduced into Cell A at outset of run; net negative value: quantity of water introduced.
3	Energy introduced into Cell A at outset of run (Btu).
4	Relative humidity of Cell A atmosphere at outset of the run (percent of saturated water content).
5	Surface area of the sodium or water pool that may form in Cell A (ft ²). At outset of accident, the code assumes that Cell A has no pool. This area is for the pool that could form by condensation or spillage during the course of the accident.
6	Thermal radiation emissivity-view factor product for heat transfer between the surface of the sodium pool and the atmosphere in Cell A (dimensionless).

In the FORTRAN variable, the subscript J ranges 1, 2, 3 and 4 for Cells A, B, C and D, respectively.

<u>Word</u>	<u>FORTRAN Variable</u>	<u>Cell B Description</u>	<u>Cell C Description</u>	<u>Cell D Description</u>
1		5	7	9
2	MST(J)	Sodium or water introduced	Sodium or water introduced	Sodium or water introduced
3	UAT(J)	Energy introduced	Energy introduced	Energy introduced
4	FPT(JP)	Relative humidity	Relative humidity	Relative humidity
5	APT(J)	Pool area	Pool area	Pool area
6	EMT(J)	Emissivity-view factor product	Emissivity-view factor product	Emissivity-view factor product

Examples of these cards follow.

B.6 Cards 3, 5, 7 and 9, More Cell Data (cont'd)

```

C FFTF REACTOR CAVITY WITH 1 PERCENT OXYGEN IN DRY NITROGEN.
4 22000.0 14.7 120.0 1.0 80.0 * FFTF REACTOR CAVITY
-----
C FFTF REACTOR BUILDING WITH DRY AIR. CARDS OF IDENTICAL DATA.
6 1.4E+6 14.7 80.0 21.0 120.0 * FFTF REACTOR BUILDING
6 1.4E+6 14.7 80.0 21.0 120.0 * FFTF REACTOR BUILDING
6,1.4E+6,14.7,80.0,21.0,120.0 * FFTF REACTOR BUILDING
-----
7 0.0 0.0 40.0 1225.0 0.0 * 40% HUMIDITY, POOL OF 1225 FT2
6 1.4E+6 14.7 80.0 21.0 120.0 * FFTF R.BLDG, HUMID AIR
-----
C FFTF REACTOR BUILDING HAS 300 LB SODIUM SPRAY AT 1050 DEG.F.
7 0.0 0.0 40.0 0.0 0.0 * 40% HUMIDITY
6 1.4E+6 14.7 80.0 21.0 120.0 * FFTF R.BLDG, HUMID AIR
7 300.0 141564.0 0.0 1225.0 0.12 * 300 LB SODIUM SPRAY.
-----
C FFTF REACTOR BUILDING HAS 300 LB SODIUM SPILL AT 1050 DEG.F.
7 300.0 141564.0 40.0 1225.0 0.12 * 300 LB SODIUM SPILL
6 1.4E+6 14.7 80.0 21.0 120.0 * FFTF R.BLDG, HUMID AIR
-----
C FFTF REACTOR BUILDING HAS 300 LB WATER SPILL AT 250 DEG.F.
7 -300.0 65544.0 40.0 1225.0 0.0 * 300 LB WATER SPILL
6 1.4E+6 14.7 80.0 21.0 120.0 * FFTF R.BLDG, HUMID AIR

```

B.7 Cards 10, 20, 30, 40, 50, 60, 70, 80 and 90, Nozzle Data

Cards 10, 20, 30, 40, 50, 60, 70, 80 and 90 are quite similar. Each inputs data for the nozzle between different cells or between a cell and the outside air. Each nozzle permits atmosphere to flow from the higher pressure cell to the lower pressure cell or outside. If the differential pressure reverses, the flow reverses. The atmosphere is the mixture of carbon dioxide, hydrogen, nitrogen, oxygen, sodium vapor and/or water vapor present in the higher pressure cell (or outside).

Word	Card 10, Cell A to Cell B Nozzle Data Description
1	10
2	Nozzle flow area (ft ²).
3	Nozzle flow coefficient (dimensionless).

B.7 Cards 10, 20, 30, 40, 50, 60, 70, 80 and 90, Nozzle Data (cont'd)

Particular Directions for Nozzle Flow (Card Numbers)

<u>From</u>	<u>to Cell A</u>	<u>to Cell B</u>	<u>to Cell C</u>	<u>to Cell D</u>	<u>to Outside</u>
Cell A	--	10	20	30	40
Cell B	10	--	50	60	70
Cell C	20	50	--	80	90
Cell D	30	60	80	--	--
Outside	40	70	90	--	--

Examples follow.

C CELL A SODIUM BOILOFF, 1234 LB/HR AT 1620 DEG.F AND 0.1 PSIG.
20 0.144 0.60 * CELL A TO CELL C BOILOFF OF 1234 LB/HR.

C CELL C VENT OF 20,000 SCFM PURGE AIR AT 60 DEG.F AND 0.1 PSIG.
90 3.7 0.817 * CELL C TO OUTSIDE VENT OF 20,000 SCFM AIR.

B.8 Card Series 11, 21, 31, 41, 51, 61, 71, 81 and 91, Leakage Data

The 11, 21, 31, 41, 51, 61, 71, 81 and 91 card series are quite similar. Each inputs coefficient data for leakage between different cells or between a cell and the outside air. Each coefficient permits atmosphere to flow from the higher pressure cell (or outside) to the lower pressure cell or outside. If the differential pressure reverses, the flow reverses. The atmosphere is the mixture of carbon dioxide, hydrogen, nitrogen, oxygen, sodium vapor and/or water vapor present in the higher pressure cell (or outside).

The leakage coefficients are taken to be a function of differential pressure and are input as a table of differential pressure vs. coefficient. The pressure values must start with a differential of zero and end with a value larger than the maximum differential anticipated. Otherwise, the case terminates with an error notice when the transient differential pressure exceeds the range of the table. The TABLU subroutine provides linear interpolation between the differential pressure entries in each table.

B.8 Card Series 11, 21, 31, 41, 51, 61, 71, 81 and 91, Leakage Data (cont'd)

Word Card Series 11, Cell A--Cell B Leakage Data
Description

- 1 XX, XX = 11, 12, 13, ..., 19 as required.
- 2 Differential pressure in ascending order (psid).
- 3 Total leakage coefficient (lbs/hr- psid-lb/ft³)

Additional pairs of values on this card and on additional cards
as needed up to a maximum of 10 pairs.

Leakage Flow Connections (Card Series Numbers)

<u>From</u>	<u>to Cell A</u>	<u>to Cell B</u>	<u>to Cell C</u>	<u>to Cell D</u>	<u>to Outside</u>
Cell A	--	11	21	31	41
Cell B	11	--	51	61	71
Cell C	21	51	--	81	91
Cell D	31	61	81	--	--
Outside	41	71	91	--	--

Examples follow.

```
C CELL A SODIUM BOILOFF, 1234 LB/HR AT 1620 DEG.F AND 0.1 PSIG.  
21 0.0 3.0E+4 12.3 3.0E+4 * CEL A-CEL C BOILOFF, 1234 LB/HR  
-----  
C CELL C LEAKAGE OF 0.1 PERCENT/DAY AT 10 PSIG FROM 1.4E+6 CU.FT.  
C CONTAINMENT. LEAKAGE ABOVE 9.9 PSIG BY RELIEF VALVE OPENING.  
91 0.0 6.5 9.9 6.5 10.4 6.5E+2 * CELL C LEAKAGE, 0.1 %/DY  
92 20.0 6.5E+4 * WITH RELIEF VALVE FLOW ABOVE 9.9 PSIG.
```

B.9 More Tables

Sections B.10 through B.16 describe the tabular input of histories of decay power, factors for apportioning decay power among the cell atmospheres and pools, sodium sprays, sodium spills, air injection and outside air conditions. For these history tables, the first entry of each set of tabular values is time. These time values must start with time equal to zero and end with a time value larger than the final time of the case. Otherwise, the case terminates with an error notice. Note that the TABLU subroutine provides linear interpolation between the time entries in all of these histories except the DECAYP table of decay power.

Sections B.17 and B.18 describe cell space coolers and sodium recycle, respectively. Sections B.19 through B.24 cover various tables of concrete data and Section B.26 describes the specification of time step and printing.

B.10 Card Series 101, Decay Power History

The decay power history combines heat sources such as fission product decay, activated isotope decay and electric motor losses. Other input data can specify the release of this power in any cell atmosphere or pool and as volumetric heat source in any heat structure. The history is stored in the DECAYP array and the TIMEDP subroutine provides logarithmic interpolation between the pairs of time and power values.

<u>Word</u>	<u>Description</u>
-------------	--------------------

1	1XX; XX = 01, 02, 03, ..., 25 as required.
2	Time in ascending order (hour).
3	Decay power (Btu/hr).

Additional pairs of values on this card and on additional cards as needed up to a maximum of 25 pairs.

Example follows Section B.12.

B.11 Card Series 201, Factors Apportioning Decay Power Among Atmospheres

This history of factors is stored in the APABCD array and the TABLU subroutine provides linear interpolation between the time entries.

<u>Word</u>	<u>Description</u>
-------------	--------------------

1	2XX; XX = 01, 02, 03, ..., 10 as required.
2	Time in ascending order (hour).
3	Decay power factor for Cell A atmosphere (dimensionless).
4	Decay power factor for Cell B atmosphere.
5	Decay power factor for Cell C atmosphere.
6	Decay power factor for Cell D atmosphere.

Additional quintuplets of values on this card and on additional cards as needed up to a maximum of 10 quintuplets.

Example follows Section B.12.

B.12 Card Series 301, Factors Apportioning Decay Power Among Pools

This history of factors is stored in the PPABCD array.

Word	Description
------	-------------

1	3XX; XX = 01, 02, 03, ..., 10 as required.
2	Time in ascending order (hour).
3	Decay power factor in Cell A sodium or water pool (dimensionless).
4	Decay power factor in Cell B pool.
5	Decay power factor in Cell C pool.
6	Decay power factor in Cell D pool.

Additional quintuplets of values on this card and on additional cards as needed up to a maximum of 10 quintuplets.

Example follows.

```

C FFTF DECAY POWER AFTER 400 MW(T) OPERATION OF EQUILIBRIUM CORE.
101 0.0 72.59E+6 1.0 16.05E+6 2.0 12.61E+6 * FFTF D.POWER
102 4.0 10.17E+6 8.0 8.42E+6 16.0 6.91E+6 * FFTF D.POWER
-----
201 0.0 0.0 0.0629 0.0988 0.0138 * NOTHING INTO CELL A,
202 1.0 0.0 0.0430 0.2043 0.0623 * NOBLE GASES INTO CEL B
203 4.0 0.0 0.0466 0.2090 0.0983 * HALOGENS INTO CELL C
204 16.0 0.0 0.0368 0.1980 0.1447 * 293 KW MOTOR IN CELL D
-----
301 0.0 0.8383 0.0 0.0 0.0 * ALL REMAINING DECAY POWER
302 1.0 0.7527 0.0 0.0 0.0 * GOES INTO CELL A POOL.
303 4.0 0.7444 0.0 0.0 0.0 16.0 0.7652 0.0 0.0 0.0
  
```

B.13 Card Series 401, 501, 601 and 701, Sodium Sprays

The 401, 501, 601 and 701 card series are quite similar. Each inputs the history of sodium spray into the atmosphere of Cell A, Cell B, Cell C and Cell D, respectively.

Word	Card Series 401, Cell A Sodium Spray Description
------	---

1	4XX; XX = 01, 02, 03, ..., 10 as required.
2	Time in ascending order (hour).
3	Sodium spray rate into Cell A atmosphere (lb/hr).
4	Total energy (enthalpy) of this sodium (Btu/lb).

Additional triplets of values on this card and on additional cards as needed up to a maximum of 10 triplets.

B.13 Card Series 401, 501, 601 and 701, Sodium Sprays (cont'd)

<u>Card Series</u>	<u>Cell Atmosphere</u>	<u>FORTTRAN Variables</u>		
		<u>Array</u>	<u>Limit</u>	<u>Location</u>
401	A	SATA	I14	J14
501	B	SATB	I15	J15
601	C	SATC	I16	J16
701	D	SATD	I17	J17

Example follows:

```

C THE SODIUM SPRAY INTO CELL B ATMOSPHERE STARTS AT 0.5 HOURS.
C IT BEGINS AS LIQUID AT 400 DEG.F, CHANGES TO VAPOR AT 500 DEG.F
C AND THENCE INTO LIQUID AT 600 DEG.F. SPRAY STOPS AT 1.5 HOURS.
501 0.0 0.0 0.0 0.5 0.0 0.0 * CELL B SPRAY AT 400F
502 0.5 1.2 272.06 1.0 2.3 2204.82 * THEN VAPOR AT 500F
503 1.5 3.4 335.01 1.5 0.0 0.0 * THEN LIQUID AT 600F
504 199.0 0.0 0.0 * CASE RUNS TO 198 HRS

```

B.14 Card Series 801, 901, 1001 and 1101, Sodium Spills

The 801, 901, 1001 and 1101 card series are quite similar. Each inputs the history of sodium spill into the sodium pool of Cell A, Cell B, Cell C and Cell D, respectively.

Word Card Series 801, Cell A Sodium Spill Description

- 1 8XX; XX = 01, 02, 03, ..., 10 as required.
- 2 Time in ascending order (hour).
- 3 Liquid sodium spill rate into Cell A sodium pool (lb/hr).
- 4 Temperature of this sodium (°F). Note, if the temperature is positive, each sodium addition increases the volume of the cell to avoid an artificial pressurization of the atmosphere. That is, a positive value implies that the sodium spilled from a tank within the cell. If the temperature is negative, the sodium properties are based on its absolute value and the cell volume remains constant.

Additional triplets of values on this card and on additional cards as needed up to a maximum of 10 triplets.

B.14 Card Series 801, 901, 1001 and 1101, Sodium Spills (cont'd)

<u>Card Series</u>	<u>Cell Pool</u>	<u>FORTTRAN Variables</u>		
		<u>Array</u>	<u>Limit</u>	<u>Location</u>
801	A	SPTA	I18	J18
901	B	SPTB	I19	J19
1001	C	SPTC	I20	J20
1101	D	SPTD	I21	J21

Example follows.

```

1101  0.0  0.0  0.0      0.5  0.0  0.0  * CELL D SPILL AT 400F
1102  0.5  1.2  400.0    1.0  2.3  500.0 * HEATS TO 500F AT 1 HR
1103  1.5  0.0  600.0      * HEATS TO 600F AND STOPS AT 1.5 HR
1104  9999.0  0.0  0.0      * CASE RUNS TO 90 HOURS OR MORE.

```

B.15 Card Series 1201, 1301, 1401 and 1501, Air Injection

The 1201, 1301, 1401 and 1501 card series are quite similar. Each inputs the history of air injection into the atmosphere of Cell A, Cell B, Cell C and Cell D, respectively.

<u>Word</u>	<u>Card Series 1201, Cell A Air Injection Description</u>
-------------	---

- 1 12XX; XX = 01, 02, 03, ..., 10 as required.
- 2 Time in ascending order (hour).
- 3 Positive value; air injection rate into the Cell A atmosphere (lb/hr).
Negative value: signal to the NATIN subroutine to calculate the air injection required to reduce the hydrogen concentration to 4 volume percent and if the hydrogen concentration is already 4 percent or less, no air injection is made. For positive and negative values, the injected air is dry with zero relative humidity.
- 4 Temperature of this air (°F).
- 5 Oxygen content of this air (volume percent).

Additional quadruplets of values on this card and on additional cards as needed up to a maximum of 10 quadruplets.

<u>Card Series</u>	<u>Cell Atmosphere</u>	<u>FORTTRAN Variables</u>		
		<u>Array</u>	<u>Limit</u>	<u>Location</u>
1201	A	NATA	I22	J22
1301	B	NATB	I23	J23
1401	C	NATC	I24	J24
1501	D	NATD	I25	J25

B.15 Card Series 1201, 1301, 1401 and 1501, Air Injection (cont'd)

Example follows.

```
C START CELL A INERT GAS PURGE OF 20 SCFM AT 63 HOURS.
1201  0.0  0.0  0.0  0.0  63.0  0.0  0.0  0.0  * CELL A INERT
1202  63.0  91.7  60.0  1.0      * GAS PURGE OF 20 SCFM AT 60F.
1203  199.0  91.7  60.0  1.0      * CASE RUNS TO 198 HR.
-----
C INJECT AIR WHEN EVER CELL C HYDROGEN EXCEEDS 4 PERCENT.
1401  0.0  -1.2  80.0  21.0      * INJECT AIR AT 80F WHEN EVER
1402  999.0  -1.2  80.0  21.0      * CELL C HYDROGEN EXCEEDS 4%.
```

B.16 Card Series 1601, Outside Air Conditions

The 1601 card series provides the history of outside air temperature and heat transfer coefficient throughout the 24 hour day. This history is used as a 24-hour cyclic table because the search in this table is for time, modulus 24 hours. Thus, only time values to 24 hours are needed. The temperature and coefficient are indexed for boundary conditions of heat structures. The history is stored in the OATH array.

Word	FORTTRAN Name In Use	Description
1		16XX; XX = 01, 02, 03,..., 25 as required.
2		Time in ascending order to 24 hours (hours).
3	TAT(1)	Temperature of outside air (°F).
4	HOUT	Heat transfer coefficient to outside air (Btu/hr-ft ² -°F).

Additional triplets of values on this card and on additional cards as needed up to a maximum of 25 triplets.

Example follows.

```
C CASE STARTS AT 0.0 HOURS WHEN ACCIDENT OCCURS AT 8 AM.
1601  0.0  70.0  1.3  4.0  90.0  2.2  * ACCIDENT AT 8 AM.
1602  8.0  100.0  3.3  12.0  90.0  2.2  * NOON AT 90 F,
1603  16.0  70.0  1.3  20.0  60.0  1.1  * 100F WINDY AT 4 PM
1604  24.0  70.0  1.3      * 60F CALM AT 4 AM
```

B.17 Cards 1700, 1800, 1900 and 2000, Space Cooler Data

The 1700, 1800, 1900 and 2000 cards are quite similar. Each describes the space cooler in Cell A, Cell B, Cell C and Cell D, respectively. In each cell, the space cooler dehydrates and cools the atmosphere. Each space cooler cools according to the following operating equation.

B.17 Cards 1700, 1800, 1900 and 2000, Space Cooler Data

$$R = A + B(T_a - T_c)$$

where: R = cooling rate (Btu/hr),

T_a = cell temperature (°F), pool in Cell A, atmosphere in Cells B, C and D,

T_c = cooler fluid temperature (°F).

When the calculated cooling rate, R, is positive, energy is removed from the cell. However, the code condenses and removes water vapor (thereby removing energy) before it effects the atmosphere temperature. When the cooling rate is negative, energy is added to the cell.

Word Card 1700, Cell A Space Cooler
Description

- 1 1700
- 2 Time cooler in Cell A starts operating (hour).
- 3 Time cooler in Cell A stops (hour).
- 4 Temperature of cooler fluid, T_c , above (°F).
- 5 Coefficient A of cooler operating equation (Btu/hr).
- 6 Coefficient B of cooler operating equation (Btu/hr-°F).
- 7 Minimum temperature of cooler operation (°F).

Card Number	Cell	FORTRAN Variable	
		Array	Switch
1700	A	COOLA	N17
1800	B	COOLB	N18
1900	C	COOLC	N19
2000	D	COOLD	N20

Examples follow.

```

C CELL A ATM COOLER HAS 200 KW COOLING WHEN ATM IS AT 400 DEG.F
C                      AND 1,000 KW COOLING WHEN ATM IS 1000 DEG.F.
C COOLER STARTS AT 8 HRS, MINIMUM ATM TEMPERATURE IS 250 DEG.F.
1700  8.0  999.0  100.0  -6.83E+5  4.55E+4  250.0  * CELL A ATM
-----
C CELL C ATMOSPHERE COOLER WITH CONSTANT 1.0 MW DUTY.
C COOLER CONDENSES 3.2E+3 LB/HR WATER BEFORE EFFECTING THE
C ATMOSPHERE TEMPERATURE.
1900  24.0  2345.0  60.0  3.41E+6  0.0  80.0  * CELL C ATM.COOLER

```

B.18 Card 2100, Sodium Recycle

The 2100 card permits the sodium pool of one cell to overflow into another cell. This overflow is a form of sodium recycling. This data is stored in the RECYL array.

<u>Word</u>	<u>Description</u>
-------------	--------------------

1	2100
2	Index of source cell (integer). Allowable values for this index are as follows.

<u>Cell</u>	<u>Index</u>
A	1
B	2
C	3
D	4

3	Index of destination cell (integer).
4	Source cell pool overflow setting (lbs). The SPRCL subroutine will transfer to the destination cell all sodium that exceeds the source cell overflow setting.

Example follows.

```
C SODIUM POOL IN CELL B BOILS UP AND VAPOR PASSES INTO CELL D
C WHERE IT CONDENSES. CONDENSATE ACCUMULATES IN CELL D TO THE
C LIMIT OF 12,345 LBS AND THEN RECYCLES BACK TO CELL B.
2100 4 2 1.2345E+4 * CELL D RECYCLES TO CELL B
```

B.19 Card Series 2201, Concrete Additional Heat Capacity

The heat conduction analyses of the heat structures normally use the constant values of heat capacity provided on the 300001 card series (Section B.34). Concrete, an important structural material, exhibits an effective higher heat capacity upon being heated above 200°F as water is driven off. Limestone concrete exhibits an effective higher heat capacity upon being heated above 1500°F as carbon dioxide is driven off. These higher heat capacities are not exhibited during cooling or subsequent reheating. This 2201 card series provides heat capacity data as a function of temperature that add to the fixed values provided on the 300001 card series during the first temperature rise of the concrete structure. The data are stored in the CONC array.

B.19 Card Series 2201, Concrete Additional Heat Capacity (cont'd)

Word Description

- 1 22XX; XX = 01, 02, 03,..., 20 as required.
- 2 Material number that starts sequence of concrete materials (real).
- 3 Material number that ends sequence (real).
- 4 Temperature, in ascending order (°F).
- 5 Volumetric heat capacity effect due to water and/or carbon dioxide release from the concrete (Btu/ft³-°F). These values are added to the fixed values input for the sequence of concrete materials on the 300001 card series.

Additional pairs of values on this card and on additional cards as needed up to a maximum of 19 pairs.

Example follows:

```

2201  5.0  6.0      * CONCRETE MATERIALS 5 THRU 6 HAVE VARIABLE CP
2202   50.0   0.0   192.0   0.0   192.0  13.1 * AVG. D1,D2,D4
2203  214.0  13.1   214.0  61.2   300.0  61.2 * AVG. D1,D2,D4
2204  300.0  13.0   400.0  13.0   400.0   3.4 * AVG. D1,D2,D4
2205 1200.0   3.4  1200.0  15.4  1382.0  15.4 * CO2 BGNS 1200F
2206 1382.0  73.4  1505.0  73.4  1505.0  128. * H2O ENDS 1500F
2207 1652.0 128.  1652.0  957.  1710.0  957. * CO2 ENDS 1710F
2208 1710.0   0.0                      * MAXIMUM OF 19 PAIRS LIMITS TABLE

C
C LIMESTONE CONCRETE (145 LB/FT3).
C   0-1500 DEG.F, WATER RELEASE LATENT HEAT BASED ON RESULTS OF
C   HEDL DRYING TESTS D1, D2 AND D4 WITH WATER
C   ENERGY TAKEN AS UGW AT TEMPERATURE PER WAPROP
C   SUBROUTINE.
C 1200-1710 DEG.F, CARBON DIOXIDE RELEASE DURING CONCRETE
C   CALCINATION REPORTED IN WARD-D-0141 FOR CRBRP
C   CONCRETE. CALCITE 60 PERCENT, DOLOMITE 40 AND
C   DOLOMITE CONVERTS TO MGO+CALCITE BELOW 1505 F.
  
```

B.20 Card Series 2301, Concrete Thermal Conductivity

The heat conduction analyses of the heat structures normally use the constant values of thermal conductivity provided on the 300001 card series (Section B.34). Concrete, an important structural material exhibits a lower conductivity upon being heated because water is driven off and other effects. This 2301 card series provided thermal conductivity as a function of temperature that replace the fixed values provided on the 300001 card series. The data are stored in the CONK array.

B.20 Card Series 2301, Concrete Thermal Conductivity (cont'd)

Word Description

- 1 23XX; XX = 01, 02, 03,..., 20 as required.
- 2 Material number that starts sequence of concrete materials (real).
- 3 Material number that ends sequence (real).
- 4 Temperature, in ascending order (°F).
- 5 Concrete thermal conductivity (Btu/hr-ft-°F). These values replace the fixed values input for the sequence of concrete materials on the 300001 card series.

Additional pairs of values on this card and on additional cards as needed up to a maximum of 19 pairs.

Example follows.

```

2301  6.0  7.0      * CONCRETE MATERIALS 6 THRU 7 HAVE VARIABLE K
2302   50.0 0.80    200.0 0.70   400.0 0.59 * HARMATHY-ALLEN
2303 1000.0 0.42   1600.0 0.37  2000.0 0.23 * HARMATHY-ALLEN
C
C CONCRETE THERMAL CONDUCTIVITIES PER HARMATHY AND ALLEN.
C AM. CONCRETE INSTITUTE JOURNAL, VOL. 70, PP. 132-142 (1973).
C VALUES FROM THERMAL DIFFUSIVITY ASSUMING CONSTANT HEAT CAPACITY
C OF 0.21 BTU/LB-F.

```

B.21 Card Series 2401, Sodium-Concrete Reaction Data

Sodium attacks concrete and other structural materials. This 2401 card series describes the attack of a heat structure that must be in contact with a sodium pool. The attack model in the CCALC subroutine checks for (a) the cell in thermal contact with the left (inside) boundary of the heat structure, (b) presence of sodium in the cell pool and (c) structure depth not yet penetrated. This attack model does not alter the heat structure. The data are stored in the CRNACO array. Example follows.

```

2401  3.0  1.0  0.0  0.0417  63400.0  0.0  110.0  0.0 * CEL A FL
C
C STRUCTURE 3 INCLUDES THE FAILED LINER AND LIMESTONE CONCRETE OF
C THE CENTER OF CELL A FLOOR DIRECTLY UNDER THE REACTOR VESSEL.
C THE CONCRETE IS ATTACKED BY SODIUM FOR 4 HOURS. THEN THE
C REACTION PRODUCTS STOP FURTHER ATTACK.
C SODIUM-CONCRETE REACTION ENERGIES PER R.E. WITKOWSKI ET AL,
C "A CHEMICAL INVESTIGATION OF SODIUM-CONCRETE REACTIONS",
C WESTINGHOUSE RESEARCH LABORATORIES, PITTSBURGH, PENNSYLVANIA,
C JANUARY 10, 1977, PAGE 52.
C      CONCRETE      DENSITY      REACTION ENERGY
C      HANFORD BASALT 149 LB/FT3  195 CAL/GM = 52,200 BTU/FT3
C      OAK RIDGE LIMESTONE 145      249      63,400
C      MAGNETITE      225      360      146,000

```

B.21 Card Series 2401, Sodium-Concrete Reaction Data (cont'd)

Word	Description
------	-------------

1	24XX; XX =01, 02, 03,... 06 as required.
2	Heat structure number (real). The attack occurs on the lefthand or inside surface of the structure.
3	Fraction of heat structure area involved (dimensionless with range of 0.0 to 1.0).
4	Present penetration of sodium attack (feet).
5	Penetration rate of sodium attack (ft/hr). The attack on a structure ceases when the penetration rate is set to zero.
6	Energy release of sodium attack (Btu/ft ³ of structure).
7	Present total energy of sodium attack (Btu).
Additional sextuplets of values on this card and on additional cards as needed up to a maximum of 6 sextuplets.
8	Sodium consumption rate of attack (lbs/ft ³ of structure).
9	Present total consumption of sodium attack (lbs).

Example follows.

```
-----  
----- FOUR HOURS LATER ----- FOUR HOURS LATER -----  
2401 3.0 1.0 0.1667 0.0 0.0 3.657E+6 0.0 6345.0 * CEL A  
C  
C STRUCTURE 3 INCLUDES THE FAILED LINER AND LIMESTONE CONCRETE OF  
C THE CENTER OF CELL A FLOOR DIRECTLY UNDER THE REACTOR VESSEL.  
C THE CONCRETE HAS BEEN ATTACKED BY SODIUM FOR 4 HOURS. THE  
C ACCUMULATION OF REACTION PRODUCTS STOPS FURTHER ATTACK.
```

B.22 Card Series 2501, Concrete Water Release

This 2501 card series provides the temperature vs. water release function used by the CWRP subroutine for determining water release from each concrete heat structure as the concrete is heated. The water is released only from materials having numbers in the sequence of Words 2 thru 3 of the 2201 card series (see Section B.19), only as each structure is heated the first time, and only for structures listed in the 2601 card series (see Section B.23). These data are stored in the CWRD array.

B.22 Card Series 2501, Concrete Water Release (cont'd)

Word Description

- 1 25XX; XX 01, 02, 03,..., 20 as required.
- 2 Temperature in ascending order (°F).
- 3 Water that would be released upon heating concrete to this temperature (lbs water/ft³ of concrete).

Additional pairs of values on this card and on additional cards
as needed up to a maximum of 20 pairs.

Example follows.

2501	50.0	0.00	192.0	0.00	214.0	0.27	* AVG. D1,D2,D4
2502	276.0	3.63	300.0	5.10	400.0	6.26	* AVG. D1,D2,D4
2503	500.0	6.55	750.0	7.26	830.0	7.77	* AVG. D1,D2,D4
2504	1500.0	9.07	2000.0	9.07			* AVG. D1,D2,D4

C
C LIMESTONE CONCRETE (145 LB/FT³). CONCRETE WATER RELEASE BASED
C ON RESULTS OF HEDL DRYING TESTS D1, D2 AND D4.

B.23 Card Series 2601, Structures that Release Concrete Water

The 2601 card series describes the heat structures that will release concrete water upon heating. These data are stored in the SWCWR array.

Word Description

- 1 26XX; XX = 01, 02, 03,... 13 as required.
- 2 Structure number (real).
- 3 Index destination for water released from the structure (real). Allowable values of this index are as follows.

<u>Destination</u>	<u>Index Values</u>	
	<u>Atmosphere</u>	<u>Pool</u>
Cell A	1.0	-1.0
Cell B	2.0	-2.0
Cell C	3.0	-3.0
Cell D	4.0	-4.0
Condensor-drain	5.0	

- 4 Temperature at which release begins (°F). No water is released until the lefthand (inside) surface temperature of the structure exceeds this value. This temperature could be the failure value of plastic-paint coating or of steel liners that cover the concrete structure.

Additional triplets of values on this card and on additional cards
as needed up to a maximum of 13 triplets.

B.23 Card Series 2601, Structures that Release Concrete Water (cont'd)

Example follows.

```
2601  2.0  +1.0  350.0  3.0  -1.0  120.0  * CEL A WALL AND FLOOR
2602  5.0   3.0   70.0                * CELL C FLOOR
```

B.24 Card Series 2701, Concrete Carbon Dioxide Release

This 2701 card series provides the temperature vs. carbon dioxide release function used by the CO2F subroutine for determining carbon dioxide release from each concrete heat structure as the concrete is heated. The carbon dioxide is released only from materials having numbers in the sequence of Words 2 thru 3 of the 2201 card series (see Section B.19), only as each structure is heated the first time, and only for structures listed in the 2801 card series (see Section B.25). These data are stored in the CO2RD array.

<u>Word</u>	<u>Description</u>
-------------	--------------------

1	27XX; XX = 01, 02, 03,..., 20 as required.
2	Temperature in ascending order (°F).
3	Carbon dioxide that would be released upon heating concrete to this temperature (lbs carbon dioxide/ft ³ of concrete).

Additional pairs of values on this card and on additional cards as needed up to a maximum of 20 pairs.

Example follows.

```
2701    50.0    0.0    1200.0    0.0    1382.0    2.3 * CO2 RELEASE
2702   1505.0   11.0   1652.0   21.6   1710.0   52.6 * CO2 RELEASE
2703   2000.0   52.6                        * CO2 RELEASE
```

C
C LIMESTONE CONCRETE (145 LB/FT³). CARBON DIOXIDE RELEASE PER
C WARD-D-0141, TB.7, PG.51, DECEMBER 1976. DECOMPOSITION OF
C DOLOMITE TO 1505 DEG.F AND OF CALCITE TO 1710 DEG.F.

B.25 Card Series 2801, Structures that Release Concrete Carbon Dioxide

The 2801 card series describes the heat structures that will release concrete carbon dioxide upon heating. These data are stored in the SWCO2R array.

B.25 Card Series 2801, Structures that Release Concrete Carbon Dioxide (cont'd)

Word Description

- 1 28XX; XX = 01, 02, 03, ..., 13 as required.
- 2 Structure number (real).
- 3 Index destination for carbon dioxide released from the structure (real).
Allowable values of this index are as follows.

<u>Destination</u>	<u>Index Values</u>	
	<u>Atmosphere</u>	<u>Pool</u>
Cell A	1.0	-1.0
Cell B	2.0	-2.0
Cell C	3.0	-3.0
Cell D	4.0	-4.0

- 4 Temperature at which release begins (°F). No carbon dioxide is released until the lefthand (inside) surface temperature of the structure exceeds this value. This temperature could be the failure value of steel liners that cover the concrete structure.

Additional triplets of values on this card and on additional cards as needed up to a maximum of 13 values.

Example follows.

2801 3.0 -1.0 120.0 * CELL A FLOOR HAS LIMESTONE CONCRETE

B.26 Card Series 9001, Time Step and Print Control

The 9001 card series provides some control over (a) the time step used by the code in its marching analysis and (b) the frequency and type of output data. The code uses a given time step as long as the convergence (reported each hour and reported with cell data) falls in the range 5.0E-3 to 5.0E-5. When the convergence gets below 5.0E-5, the time step is increased by a factor of 1.58 and the analysis continues. When the convergence gets above 5.0E-3, the time step is decreased by a factor of 0.63 and the analysis continues. In any case, the time step will not exceed 10 times nor recede 1/100 times the value input on these cards. The heat structure data include heat rates, boundary conditions and the node temperature of each structure. The cell data include decay energy distribution, leakage, and both predicted and final cell conditions. The cell data are printed with the structure data and, also, can be printed more often. The control data are stored in the TSPC array.

B.26 Card Series 9001, Time Step and Print Control (cont'd)

<u>Word</u>	<u>Description</u>
-------------	--------------------

1	90XX; XX = 01, 02, 03,..., 50 as required.
2	Time, in ascending order, up to which the following time step and print controls are in force (hours).
3	Time step (hours).
4	Print-out-frequency for heat structure data (number of time steps).
A print-out-frequency of 500 means that the data will be printed every 500 of the Word 3 time steps
5	Print-out-frequency for cell data (number of time steps).

Additional quadruplets of values on this card and on additional cards as needed up to a maximum of 50 quadruplets.

Examples follow.

9001	9.0	0.0025	1.0	1.0	* INPUT STEP 9 SEC, PRINT EACH STEP

9001	0.2	0.0005	200.0	100.0	* STRUCTURES 0.1HR, DATA 0.05HR
9002	1.0	0.001	400.0	200.0	* STRUCTURES 0.4HR, DATA 0.20HR
9003	2.0	0.002	500.0	250.0	* STRUCTURES 1.0HR, DATA 0.50HR
9004	12.	0.002	1000.0	500.0	* STRUCTURES 2.0HR, DATA 1.00HR

9001	24.63	0.005	1200.0	1200.0	* STRUCTURES+DATA AT 6.0 HRS
9002	99.0	0.005	1.0	1.0	* EVERY STEP AFTER 24.63 HRS

B.27 Cards 9999 and 10000, End of Tables

The 9999 and 10000 cards signal the CINPUT subroutine the end of input by Card Numbers 0 to 2100 and by Card Series 11 to 9001. Either the 9999 or 10000 card must be used.

<u>Word</u>	<u>Description</u>
-------------	--------------------

1	9999
This card means that heat structure data follow. The heat structure data are described in Sections B.28 thru B.33 that follow. |
|---|--|

<u>Word</u>	<u>Description</u>
-------------	--------------------

1	10000
2	Structure number (integer).
3	Heat transfer surface multiplier (ft ² for planer geometry, ft for cylindrical and dimensionless for spherical geometry). Additional pairs of values on this card as desired to change the surface multipliers of heat structures input by previous run.

B.27 Cards 9999 and 10000, End of Tables (cont'd)

Examples follow.

9999

10000

10000 2 1.2 7 1953.0 * NEW SURFACE MULTIPLIER FOR 2 AND 7.

B.28 Heat Structure Title Card

Heat structure data follow the 9999 card in CACECO code input. However, if no heat structures are needed in the case, then NSL will be zero on the 1 card (see Section B.4) and the 9999 card is followed by the 500000 card.

There is NSL number of heat structures and each structure must be defined by a set of data as described in Sections B.28 through B.33.

The title card is the first card of the data set for each structure. This title card can have any alphabetic, numeric or special characters desired to identify the structure. The title card is read by the HINPUT subroutine using Format 8A10 and is stored in the TLHS array.

The remaining cards of the data set have a six digit card number with the second and third digit represented here by YY to denote 01 for the first heat structure, 02 for the second, 03 for the third, etc. Also, the following sections use this nomenclature.

I = structure number; I = 1, 2, 3, ..., NSL.

J = region of the structure; J = 1, 2, 3, ..., NOR(I).

N = node number or the interval between node points of the structure.

Examples of heat structure data follow Section B.33.

B.29 Card 1YY000, Heat Structure General Information

<u>Word</u>	<u>FORTTRAN Variable</u>	<u>Description</u>
1		1YY000
2	NON(I)	Total number of node points (integer). This quantity is one larger than the number of node point intervals. Its maximum value is III which is presently set at 25.
3	NOR(I)	Number of regions (integer). Each region covers one size of node point interval and/or one material. A maximum of 20 regions is allowed in each heat structure.
4	IGEOM	Geometry type (integer). Allowable values are as follows. 0 planar geometry 1 cylindrical 2 spherical
5	RS(1,I)=AS(1)	Coordinate of left hand boundary for planar geometry and inside radius for cylindrical and spherical geometry (feet).
6	SPF(I)	Power factor, p_f , of Equation (E6)* with normal units of $(1/\text{ft}^3)$ when decay power (101 card series, Section B.10) has units of (Btu/hr).
7	SDT(I)	Delay time (hour). If positive, it is the time when volumetric heat generation starts. If negative, it is the time when volumetric heat generation stops.
8	ARA(I)	Heat transfer surface multiplier as follows. <u>Geometry</u> <u>Units</u> <u>Comment</u> Planar (ft^2) surface area Cylindrical (ft) height of cylinder Spherical (dimensionless) fraction of spherical surface

B.30 Card Series 1YY100, Heat Structure Node Locations

<u>Word</u>	<u>FORTTRAN Variable</u>	<u>Description</u>
1		1YY1XX XX = 00, 01, 02, ..., NOR(I), at most.
2	NS(J,I)	Number of intervals in the Jth region (integer).
3	RS(N,I)=POL	Coordinate (in ascending order) of the right hand boundary or outside radius of the Jth region. For the remaining regions, the number and coordinate pairs of values on this card and on additional cards as needed to the limit of NOR(I) pairs.

B.31 Card Series 1YY200, Heat Structure Material Numbers

These material numbers refer to the materials described in the 300001 card series (see Section B.34).

<u>Word</u>	<u>FORTTRAN Variable</u>	<u>Description</u>
1		1YY2XX; XX = 00, 01, 02,..., NOR(I), at most.
2	MN(J,I)	Material number for Region J; J = 1, 2, 3,..., NOR(I) (integer). For the remaining regions, the material numbers on this card and on additional cards as needed to the limit of NOR(I) numbers.

B.32 Card Series 1YY300, Heat Structure Volumetric Heat Distribution

Volumetric heat is based on the decay power input on the 101 card series (see Section B.10). This decay power is factored by SPF(I) and is started or stopped by SDT(I) of the 1YY000 card (Section B.29). The distribution of volumetric heat generation in the structure is input now.

<u>Word</u>	<u>Description</u>
1	1YY300
2	Type of volumetric heat distribution (integer). Allowable values are as follows. <ul style="list-style-type: none">0 Volumetric heat generation is constant over the entire structure and Word 3 is the heat source value. The 1YY301 card series is not used.1 Heat generation is constant in each region but can differ in successive regions. The 1YY301 card series is required.2 Heat generation can differ in successive node intervals. The 1YY301 card series is required.
3	Constant value for volumetric heat generation if type, above, is zero (dimensionless).

The structure data set includes the 1YY301 card series only if the distribution type is nonzero.

B.32 Card Series 1YY300, Heat Structure Volumetric Heat Distribution (cont'd)

<u>Word</u>	<u>Description</u>
-------------	--------------------

1	1YY3XX; XX = 01, 02, 03,..., to NOR(I) or to NON(I)-1, at most.
2	Volumetric heat generation fraction for the Jth region, J = 1, 2, 3,..., NOR(I), or for the Nth interval, N = 1, 2, 3,..., NON(I)-1, (dimensionless).
Additional fraction values on this card and on additional cards as needed for each of the remaining regions or intervals.

B.33 Card Series 1YY400, Heat Structure Boundary Conditions

The heat structure boundary conditions involves the bulk temperature and the coefficient for conduction-convection heat transfer at the left hand (inside) and right hand (outside) surfaces of the structure. The bulk temperature can be the adjacent atmosphere, pool, structure, etc. The boundary conditions may involve the bulk temperature and the total interchange area for radiation heat transfer at one or both of these same surfaces.

Every heat structure data set needs the 1YY400 card to index the conduction-convection conditions at the left and right hand boundaries. Only structures having thermal radiation need the 1YY401 card series to index the radiation conditions at these same boundaries.

<u>Word</u>	<u>FORTTRAN Variable</u>	<u>Description</u>
1		1YY400
2	BH1(I)	Index to left hand boundary heat transfer coefficient (integer). Allowable indices are described below.
3	BT1(I)	Index to left hand bulk temperature (integer). Allowable indices are described below.
4	BHN(I)	Index to right hand boundary heat transfer coefficient (integer).
5	BTN(I)	Index to right hand bulk temperature (integer).
6	NCH	Number of other structures radiating to the Ith structure (blank or integer). If NCH has an integer value, then there are NCH number of cards in the 1YY401 card series.

B.33 Card Series 1YY400, Heat Structure Boundary Conditions (cont'd)

The allowable indices for the heat transfer coefficient control are as follows (Words 2 and 4).

- 0: The coefficient is zero indicating a symmetrical or insulated boundary.
- 1, 2, 3, ..., 95: The coefficient is taken from the HTC array as HTC(1), HTC(2), HTC(3), ..., HTC(95), respectively. The HTC array is input on the 400001 card series (Section B.35).
- 100: The coefficient is the time-dependent value interpolated from the OATH array which is input on the 1601 card series (Section B.16).
- 101, 102, ..., 109: The coefficient is the time or temperature-dependent value interpolated from the HTCT array which is input on the 410001, 420001, ..., 490001 card series, respectively (Section B.36).

Note that Index 110 cannot be input; it is code generated to reference the coefficient calculated by P00LR subroutine for heat transfer from the sodium pool to the atmosphere.

The boundary conditions for this Ith structure can involve heat transfer with Structure K. The allowable indices for bulk temperature control are then as follows (Words 3 and 5).

- 0: The bulk temperature is the boundary temperature of the Ith structure itself (adiabatic condition).
- K: Left hand (inside) boundary of Structure K.
- +K: Right hand (outside) boundary of Structure K.

Note, the Kth structure can be a sodium pool with indices listed below (NSL datum input on the 1 card, Section B.4).

<u>Surface</u>	<u>Cell A</u>	<u>Cell B</u>	<u>Cell C</u>	<u>Cell D</u>
Top	-(NSL+1)	-(NSL+2)	-(NSL+3)	-(NSL+4)
Bottom	+(NSL+1)	+(NSL+2)	+(NSL+3)	+(NSL+4)

B.33 Card Series 1YY400, Heat Structure Boundary Conditions (cont'd)

100: The bulk temperature is the time dependent value interpolated from the OATH array which is input on the 1601 card series (Section B.16).

101 to 112: Index refers to cell conditions, the cell atmosphere temperature, sodium/water pool temperature and the constant source-sink temperature. The source-sink temperature was input on the 2, 4, 6 and 8 card (Section B.5).

<u>What</u>	<u>FORTTRAN Variable</u>	<u>Cell A</u>	<u>Cell B</u>	<u>Cell C</u>	<u>Cell D</u>
Atmosphere	TAT(JP)	101	104	107	110
Sodium/Water pool	TPT(J)	102	105	108	111
Constant source-sink	TCONT(J)	103	106	109	112

The structure data set includes the 1YY401 card series only if radiation heat transfer is desired. Each surface of the Ith structure can have heat radiation with as many as 6 other structures.

<u>Word</u>	<u>FORTTRAN Variable</u>	<u>Description</u>
1		1YY4XX; XX = 01, 02, 03,..., NCH where NCH was input as Word 6 of the 1YY400 card. Below, L = XX.
2	RBH1(L,I)	Index (integer) to total interchange area for radiation heat transfer off the left hand (inside) boundary of the Ith structure. Allowable indices include: 1, 2, 3,..., 95 for interchange areas from the HTC array, 100 for interchange area from the OATH array, and 101, 102,..., 109 for interchange areas from the HTCT array.
3	RBT1(L,I)	Index (integer) to left hand bulk temperature. Allowable indices include: -(NSL+4) to -(NSL+1) for top surface of sodium pools, -K to +K for left hand to right hand boundaries of Structure K, respectively, 100 for outside temperature from the OATH array, and 101 through 112 cell conditions.
4	RBHN(L,I)	Index (integer) to total interchange area off the right hand (outside) boundary of the Ith structure.
5	RBTN(L,I)	Index (integer) to right hand bulk temperature.

B.33 Card Series 1YY400, Heat Structure Boundary Conditions (cont'd)

Several examples follow.

```
CELL C ROOF, 34200FT2 AREA, 1.1-IN. STEEL+1.0-IN. URETHANE INSULATN
102000 7 2 0 0.0 0.0 0.0 3.42F+4 * PLANER GEOMETRY
102100 2 0.0917 4 0.175 * 2 INTERVALS THRU STEEL, 4 URETHANE
102200 1 3 * STEEL AND URETHANE
102300 0 0.0 * NO VOLUMETRIC HEAT
102400 108 107 100 100 * CELL C LEFT, OUTSIDE RIGHT
```

-----EXAMPLES ASSUME 8 HEAT STRUCTURES (NSL = 8 ON CARD 1)-----

```
CEL B ROOF, 600FT2, EMBDMT IN 6.0FT CONCRETE, CEL B LEFT, C RIGHT
106000 24 7 0 0.0 0.0 0.0 600.0 * PLANER GEOMETRY
106100 3 0.25 3 0.75 3 1.75 * 1, 2, 4 INCH INTERVALS
106101 5 4.25 * 6 INCH INTERVALS
106102 3 5.25 3 5.75 3 6.00 * 4, 2, 1 INCH INTERVALS
106200 5 6 6 6 6 6 * EMBEDMENT AND THEN CONCRETE
106300 0 0.0 * NO VOLUMETRIC HEAT
106400 101 104 108 107 2 * CONVECTION TO CELL B AND CELL C
106401 3 -7 0 0 * RADIATION TO CELL B WALL
106402 4 -9 0 0 * RADIATION TO CELL B SODIUM POOL
CELL B WALL, IA=4750FT2, EMBEDMENT IN 5.3FT. CONCRETE, ATM. AT LEFT
107000 18 5 1 18.0 0.0 0.0 42.0 * CYLINDRICAL, 42FT HIGH
107100 3 18.25 3 18.50 3 19.0 * 1, 1, 2 INCH INTERVALS
107101 3 20.0 5 23.333333 * 4 AND 8 INCH INTERVALS
107200 5 6 6 6 6 * EMBEDMENT AND CONCRETE
107300 0 0.0 * NO VOLUMETRIC HEATING
107400 101 104 0 0 2 * CONVECTION FROM CELL B ATMOSPHERE
107401 3 -6 0 0 * RADIATION TO CELL B ROOF
107402 5 -9 0 0 * RADIATION TO CELL B POOL SURFACE
CELL B FLOOR, 600FT2, 4-IN. CORE DEBRIS IN 9FT. CONCRETE UNDER POOL
108000 19 6 0 0.0 0.0050 3.5 600.0 * HEAT 200FT3, 3.5HRS
108100 4 0.33 2 0.50 3 1.00 * 1, 1, 2 INCH INTERVALS
108101 3 2.00 3 4.00 3 8.00 * 4, 8, 16 INCH INTERVALS
108200 4 5 6 6 6 6 * CORE DEBRIS, EMBEDMENT AND CONCRETE
108300 1 0.0 * VOLUMETRIC HEAT BY REGION
108301 1.0 0.0 0.0 0.0 0.0 0.0 * HEAT ONLY CORE DEBRIS
108400 102 +9 2 100 * CONVECTION FROM POOL, CONDUCTION-AIR
```

B.34 Card Series 300001, Heat Structure Material Data

The 300001 card series should be skipped when there are no structures. It provides thermal conductivity and heat capacity data for all materials in the described heat structures. Each of the NSL number of structures may be composed of more than one material and each material may be used in more than one structure. The order of the material data in these cards is the material number which was input on the 1YY200 card series (Section B.31) for each structure. The conductivity and capacity data are constant, independent of temperature.

B.34 Card Series 300001, Heat Structure Material Data

These material data can be modified for concrete materials. The constant conductivity values will be replaced by the temperature dependent values input on the 2301 card series (Section B.20) for one sequence of material numbers. The constant capacity values will be increased by the temperature dependent values input on the 2201 card series (Section B.19) for another sequence of materials.

These 300001 card series data are stored in the KSCS array.

<u>Word</u>	<u>Description</u>
-------------	--------------------

1	3000XX; XX = 01, 02, 03,..., 20 as required.
2	Thermal conductivity for material Number 1 (Btu/hr-ft-°F).
3	Volumetric heat capacity for material Number 1 (Btu/ft ³ -°F). The volumetric heat capacity is the product of density and heat capacity.

Additional pairs of values are input on this card and on additional cards for material Numbers 2, 3,..., etc. A maximum of 20 materials are allowed.

Examples follow that are consistent with the heat structure examples.

C	NO.	MATERIALS
C	1	CELLULAR URETHANE FOAM INSULATION AT 80 DEG.F.
C	2	CALCIUM SILICATE INSULATION AT 420 DEG.F.
C	3	CARBON STEEL.
C	4	CORE DEBRIS MIXTURE OF CONCRETE, STEEL AND SODIUM.
C	5	3-INCH DEEP EMBEDMENT OF STEEL IN CONCRETE, FIXED K
C	6	LIMESTONE CONCRETE, 145 LB/FT ³ , CO ₂ AND H ₂ O RELEASE
C	7	LIMESTONE CONCRETE, NO CO ₂ OR H ₂ O RELEASE IF HEATED
300001	0.013 0.60 0.039 2.42	* URETHANE, CALCIUM SILICATE
300002	25.0 72.0 6.0 40.02	* CARBON STEEL, CORE DEBRIS
300003	1.3 31.0	* 3-IN. DEEP EMBEDMENT OF STEEL IN CONCRETE
300004	0.80 30.4	* LIMESTONE CONCRETE WITH CO ₂ +H ₂ O RELEASE.
300005	0.80 30.4	* LIMESTONE CONCRETE, NO CO ₂ -H ₂ O RELEASE.

B.35 Card Series 400001, Constant Heat Transfer Coefficients and Total Interchange Areas

The 400001 card series should be skipped when there are no structures. It provides fixed values of heat transfer coefficients and total interchange areas that are indexed in the boundary conditions for the heat structures. The order of coefficient-area data in these cards is the index number which was input on the 1YY400 card series (Section B.33) for various structures. These data are stored in the HTC array.

B.35 Card Series 400001, Constant Heat Transfer Coefficients and Total Interchange Areas (cont'd)

Word Description

- 1 4000XX; XX = 01, 02, 03, ..., 95 as needed.
- 2 Heat transfer coefficient (Btu/hr-ft²-°F) or total interchange area (ft²) for index Number 1.
- 3 Coefficient or area for index Number 2.

Additional values are given on this and other cards as needed for all coefficients and/or areas indexed. A maximum of 95 values are allowed.

Examples follow that are consistent with the heat structure examples.

C	NO.	HEAT TRANSFER COEFFICIENTS		
C	1	OVERALL FROM CELL C, THRU WALL INTO ATTACHED ROOMS		
C		AND THEN THRU THEIR ROOFS+WALLS TO OUTSIDE AIR.		
C	2	OVERALL FROM CELL B FLOOR THRU GROUND TO OUTSIDE AIR		
C				
C	NO.	STRUCTURES TOTAL INTERCHANGE AREAS.		
C	3	-6 TO -7	CELL B ROOF TO CELL B WALLS.	
C	4	-6 TO -9	CELL B ROOF TO SURFACE OF POOL ON FLOOR	
C	5	-7 TO -9	CELL B WALL TO SURFACE OF POOL ON FLOOR	
400001	0.17	0.24	* 1 AND 2, HEAT TRANS. COEFFICIENTS	
400002	133.0	3.4	90.0	* 3 THRU 5, TOTAL INTERCHANGE AREAS

B.36 Card Series 410001 thru 490001, Variable Heat Transfer Coefficients and Total Interchange Areas

The 410001 thru 490001 card series should be skipped when there are no heat structures. Each of these nine series provides a set of time or temperature dependent values of heat transfer coefficients or total interchange areas that can be indexed in the boundary conditions for the structures. The index Numbers 101, 102, 103, ... 109 correspond to data input on the 410001, 420001, ..., 490001 card series, respectively.

The kind of coefficient or area dependence is described by Word 2 with allowable values as follows.

B.36 Card Series 410001 thru 490001, Variable Heat Transfer Coefficients and Total Interchange Areas (cont'd)

<u>Kind</u>	<u>Description</u>
1	Time dependent coefficient or area.
2	Temperature average of boundary and bulk temperatures.
3	Temperature difference (absolute value) between boundary and bulk temperatures.
4	Same as Kind 3 for steady state calculations but the coefficient is multiplied by the value of FACTOR (presently set at 1.0) for all transient calculations.
5	Boundary temperature dependent coefficient or area.
6	Bulk temperature dependent coefficient or area.

In the above, the boundary temperature is the surface temperature of the Ith structure and the bulk temperature is that indexed as being an air, atmosphere or pool temperature or even the surface temperature of the adjacent Kth structure. The TABLU subroutine provides linear interpolation between the time or temperature entries of these data sets. These data are stored in the HTCT array.

Word Description

- 1 4100XX; XX = 01, 02, 03,..., 15 as needed for index Number 101 coefficients or areas.
...
4900XX; as needed for index Number 109 coefficients or areas.
- 2 1, 2, 3,... 6; the kind of time or temperature dependence (integer).
This kind of control is input only once, on the first card of the series.
- 3 Time (hours, beginning at 0.0 hours) or temperature (°F). Time or temperature must be in ascending order.
- 4 Heat transfer coefficient (Btu/hr-ft²-°F) or total interchange area (ft²) associated with time or temperature. The coefficient or area can be zero but cannot be negative.

Additional pairs of values are input on this card and on additional cards as needed up to a maximum of 15 pairs.

Examples follow that are consistent with the heat structure examples.

B.36 Card Series 410001 thru 490001, Variable Heat Transfer Coefficients and Total Interchange Areas (cont'd)

C TYPE 101 COMBINE RESISTANCES ON BOTH SIDES OF THE STEEL LINER. ON THE ATMOSPHERE SIDE IS THE RESISTANCE OF NATURAL CONVECTION IN THE CELL R. ON THE CONCRETE SIDE IS THE RESISTANCE OF CONDUCTION PLUS RADIATION ACROSS THE 1/8 INCH GAP BETWEEN THE LINER AND CONCRETE WALL. THE EMISSIVITIES ARE 0.7 FOR STEEL AND 0.8 FOR CONCRETE. NOTE THAT THE LINER RESISTANCE AND HEAT CAPACITY HAS BEEN NEGLECTED. KIND 5 DENOTES DEPENDENCE UPON CONCRETE SURFACE TEMPERATURE.

410001	5	50.0	0.8	200.0	0.8	500.0	0.9	800.0	1.0
410002		1100.0	1.4	1400.0	5.6	1700.0	18.0		

C TYPE 102 ARE HEAT TRANSFER COEFFICIENTS FOR SODIUM POOL-TO-FLOOR. KIND 1 DENOTES DEPENDENCE ON TIME TO ELIMINATE TENDENCY OF TEMPERATURE OSCILLATIONS AS THE POOL AND FLOOR TEMPERATURES CONVERGE.

420001	1	0.0	1.1	3.0	200.0	24.0	200.0	60.0	100.0
420002		480.0	50.0	2345.0	25.0				

C TYPE 108 ARE NATURAL CONVECTION HEAT TRANSFER COEFFICIENTS FOR AIR. KIND 3 DENOTES DEPENDENCE ON TEMPERATURE DIFFERENCE.

480001	3	0.0	1.1	100.0	1.1	300.0	1.6	500.0	2.1
480002		700.0	2.4	900.0	2.8	1100.0	3.3	1300.0	4.0

B.37 Card 50000, Final Card for Subroutine CINPUT

The 500000 card ends the containment data, heat structure data and table data that describe the physical and accidental situation. This card is required in the deck that restarts a case as well as in the deck for a new case.

Word Description

1 500000

B.38 Card YY, Heat Structure Steady State Temperatures

The code calculates steady state temperatures in all of the heat structures before beginning the transient analysis. When the boundary conditions include radiation heat transfer this calculation may be helped by inputting appropriate boundary temperatures for the structures involved.

The TINPUT subroutine stores the left hand (inside) boundary temperatures in the US1 array and the right hand (outside) boundary temperature in the USN array. This same subroutine will store any other temperature in the US and USM arrays as described below.

B.38 Card YY, Heat Structure Steady State Temperatures (cont'd)

<u>Word</u>	<u>Description</u>
-------------	--------------------

1	YY = I; the number of the Ith structure (integer) with allowable values of I = 1, 2, 3, ..., NSL + 4.
2	N; the number of the Nth node of the Ith structure (integer) with allowable values of N = 1, 2, 3, ..., NON(I).
3	Temperature for the Nth node of the Ith structure (°F).

Additional temperatures are input on this card in nodal order:

Word 4 for Node N+1, Word 5 for Node N+2, etc.

Special values for Word 2 operate as follows.

N = 1 The temperature is stored in the left hand (inside) boundary array at US1(I) and in the nodal arrays at US(1,I) and USM(1,I).

N $\geq$ 2 The temperature is stored in the right hand (outside) boundary array at USN(I) and in the nodal arrays at UN(N,I) and USM(N,I).

N $\leq$ -2 The temperature is stored in the right hand (outside) boundary array at USN(I) and at several locations in US and USM arrays. Let N_{i-1} be the node number of the Ith structure on the previous card and $N_i = |N|$, the node number of the same structure on the present card. Possibly $N_{i-1} = 0$. Then the temperature is stored in the sequence of locations of $1+N_{i-1}$, $2+N_{i-1}$, ..., N_i in the US and USM arrays.

If I is negative, then all structure temperatures including the left and right hand boundary temperatures are set equal to the value of Word 3. Examples follow Section A.41.

B.39 Card 600000, End of Steady State Temperature Input

The 600000 card ends the input data required to calculate steady state temperatures in all the heat structures. This card is not required in the deck that restarts a case.

<u>Word</u>	<u>Description</u>
-------------	--------------------

1	600000

B.40 Card YY, Heat Structure Initial Temperatures

Various structures may have boundary conditions for the transient analysis which cause the steady state calculations to generate wrong initial temperatures. The correct initial temperatures can be input now through the TINPUT subroutine (see Section B.38 for complete description).

<u>Word</u>	<u>Description</u>
-------------	--------------------

1	YY = I; the number of the Ith structure (integer).
2	N; the number of the Nth node of the Ith structure (integer).
3	Temperature for the Nth node of the Ith structure (°F).

B.41 Card 1000000, Final Card of Input Deck

The 1000000 card ends the card deck for a new or restarted case.

<u>Word</u>	<u>Description</u>
-------------	--------------------

1	1000000

Examples follow that are consistent with the heat structure examples.

```
500000      * END OF CINPLT SUBROUTINE CARDS.
  6  1  120.0  * CELL B SIDE OF ROOF.
  6 24  80.0  * CELL C SIDE OF CELL B ROOF.
-7  1  120.0  * CELL B WALL AT 120 DEG.F THROUGHOUT.
  8  1  120.0  * CELL B FLOOR SURFACE.
  9  1  120.0  * CELL B SODIUM POOL (FLOOR BEFORE SPILL).
600000      * END TINPUT CARDS BEFORE STEADY STATE TEMPERATURES
  8  1  2600.0 * NODES 1 TO 5 REPRESENT CORE DEBRIS AT 2600 F.
  8 -5  2600.0 * NODES 6 TO 19 REPRESENT FLOOR BEFORE THE SPILL.
  8  6  119.0  118.0  117.0  116.0  115.0  113.4  111.6
  8 13  110.0  107.0  103.0  100.0  93.0   86.0   80.0
1000000      * END OF CARD DECK FOR THIS RUN.
```

APPENDIX C

DATA FORMAT

APPENDIX C

DATA FORMAT

C.1 Introduction

The CVIRDP subroutine reads a card and converts the BCD information to the integer or real (floating point) binary information used by the computer; the conversion type is indicated by the BCD information itself.

C.2 Usage

An example of the CVIRDP subroutine use might be two cards punched as follows.

105 +318 105+2 -14 -25-4	* FIRST CARD
10,25.2.,1E-3,-1.4+5,245.65	C SECOND CARD

The subroutine would convert the first card BCD numbers 105 and 318 to their binary integer equivalents (105 and 318), the BCD number 105+2 to the binary real equivalent (10.5), -14 to the negative integer equivalent (-14) and -25-4 to the negative real equivalent (-0.000025). The asterisk (*) and alphabetic C (C) denote the end of the BCD informations to be converted and the remainder of the card can be used for comments that are not reproduced in output.

The second card contains the two integer numbers (10 and 25) and the four real values (2.0, 0.0001, -140,000 and 245.65).

As illustrated by this example, the following rules govern the conversion.

- (1) Information to be converted is grouped into fields, one field for each number. A field is terminated by either a blank [note one exception in (5)], a comma, or an asterisk. Any number of blanks may precede the next field.
- (2) An asterisk or alphabetic C terminates the scan of the card for fields to be converted, allowing the remainder of the card for comments. The asterisk or alphabetic C can also terminate the last field to be converted.
- (3) The sign of the number if present occupies the first column of the field. A plus sign is assumed if no sign is present.

C.2 Usage (cont'd)

- (4) An integer number is indicated if none of the following is present:
 - (a) a decimal point
 - (b) a sign following a digit in a field
 - (c) an "E" (with or without a following sign) following a digit in a field.
- (5) A real quantity is indicated if any of the conditions listed in (4) is present. The digits preceding the exponent sign or "E" are the fractional part of the real number. If no decimal point is present, it is assumed to be in front of the first digit of the fraction. The exponent sign and number are the power of ten by which the fraction is multiplied. The exponent field need not be present if a decimal point is used. To allow conversion of cards punched by the FORTRAN punch routine, one blank immediately following an "E" will be treated as if it were a plus sign. This is the only exception to the rule that a blank indicates the end of a field.
- (6) Note that the card format is such that cards punched by FORTRAN routines using I, E, or F conversion can be read as long as at least one blank or comma separates each field. Also this format is compatible with the input conventions of FORTRAN except
 - (a) the restrictions of no blanks between characters in the number
 - (b) the convention of an assumed decimal point at the beginning of the fractional part of a number if no decimal point is present
 - (c) the placement of data on the card at arbitrary spacing.
- (7) The subroutine should detect all errors in formatting the information on the card. These would include multiple decimal points, a decimal point in the exponent field, sign placement errors, multiple "E", and no digits in the fraction or exponent fields. A comma may be used only to terminate a field; a comma preceded by a blank is an error. Commas punched in adjacent columns may not be used to enter a zero. A blank card or a card in which the first nonblank column is an asterisk or alphabetic C is not considered an error, but is treated as a card containing no information.

C.2 Usage (cont'd)

- (8) The integer or the fractional part of a real quantity with the decimal point removed must be less than 2^{35} or so according to the computer.
- (9) An error causes a printout of the card, and a statement of the column of the card causing the error.

APPENDIX D

LIST OF CACECO COMMON BLOCK VARIABLES

APPENDIX D LIST OF CACECO COMMON BLOCK VARIABLES

<u>Variable</u>	<u>Type</u>	<u>Common Block</u>	<u>Variable Use</u>
ABAR	REAL	SODIUM	Average molecular weight of sodium vapor
APABCD	REAL	TABLES	Decay power apportioning factors to atmospheres
APT	REAL	CELL1	Cell pool area, ft ²
ARA	REAL	blank	Heat structure surface area factor
ARAR	REAL	BHEAT	Surface area of structure, ft ²
AS	REAL	blank	Composite variable used in HCALC
AVGLKR	REAL	LEAK	Average leak rate, lbs/hr
BHN	INTEGER	blank	Boundary heat transfer index
BH1	INTEGER	blank	
BOLT	REAL	CELL2	Boilup of pool from EMVBAL, lbs
BS	REAL	blank	Volumetric heat factors used in HCALC
BTN	INTEGER	blank	
BT1	INTEGER	blank	Boundary temperature index
CCRT	REAL	ZCALC	
CCRZ	REAL	ZCALC	Total carbon dioxide release, lbs
CONC	REAL	TABLES	Initial carbon dioxide release, lbs
			Variable volumetric specific heat table, Btu/ft ³ -°F
CONK	REAL	TABLES	Variable thermal conductivity table, Btu/hr-ft-°F
COOLA	REAL	TABLES	Cell space cooler tables
COOLB	REAL	TABLES	
COOLC	REAL	TABLES	
COOLD	REAL	TABLES	
CO2RD	REAL	TABLES	Concrete carbon dioxide release table, lbs/ft ³
CRNACO	REAL	TABLES	Sodium-concrete reaction tables
CS	REAL	blank	Volumetric heat capacity used in HCALC
CWRD	REAL	TABLES	Concrete water release table
CWRT	REAL	ZCALC	Total water release, lbs
CWRZ	REAL	ZCALC	Initial water release, lbs
DECAYP	REAL	TABLES	Decay power table, Btu/hr
DEEP	REAL	CELL3	Depth of pool, ft
DHAT	REAL	ZCALC	Decay heat added to cell atmosphere, Btu

Appendix D (cont'd)

<u>Variable</u>	<u>Type</u>	<u>Common Block</u>	<u>Variable Use</u>
DHPT	REAL	ZCALC	Decay heat added to cell pool, Btu
DT	REAL	TIMET	Time step, hours
DOV2	REAL	TIMET	Half time step, hours
EMT	REAL	ZINPT	Pool emissivity-view product
ERROR	LOGICAL	SWITCH	Error switch
FACTOR	REAL	BHEAT	Sodium pool surface heat transfer factor, SOFIRE-CACECO code correlation
FPT	REAL	CELL1	Relative humidity in cell, input as volume percent
GL3	REAL	POLSUR	Part of Grashof number for pool surface
GS	REAL	blank	Composite variable used in HCALC
HCT	REAL	FLODAT	Energy of carbon dioxide, Btu/lb
HEAD	REAL	INOUT	Printout heading, case title
HFNAH	REAL	CELL2	Enthalpy of sodium hydride formation
HGN	REAL	SODIUM	Enthalpy of sodium vapor, Btu/lb
HGT	REAL	CELL2	Enthalpy of vapor, Btu/lb
HGW	REAL	WATER	Enthalpy of water vapor, Btu/lb
HHT	REAL	FLODAT	Enthalpy of hydrogen, Btu/lb
HLN	REAL	SODIUM	Enthalpy of sodium liquid, Btu/lb
HLW	REAL	WATER	Enthalpy of water liquid, Btu/lb
HNT	REAL	FLODAT	Enthalpy of nitrogen, Btu/lb
HOR1	LOGICAL	FLODAT	} Hydrogen and oxygen reaction switches
HOR2	LOGICAL	FLODAT	
HOT	REAL	FLODAT	Enthalpy of oxygen, Btu/lb
HOUT	REAL	BHEAT	Heat transfer coefficient to outside air Btu/hr-ft ² -°F
HPA	REAL	POLSUR	Heat transfer coefficient at surface of sodium pool, Btu/hr-ft ² -°F
HTC	REAL	THEAT	Heat transfer coefficient and total interchange area table
HTCT	REAL	THEAT	Heat transfer coefficient tables, Btu/hr-ft ² -°F
HTN	REAL	blank	} Boundary heat transfer coefficients used in HCALC
HT1	REAL	blank	
HWT	REAL	FLODAT	Enthalpy of water vapor, Btu/lb
ICCU	REAL	ZCALC	Energy of integrated carbon dioxide release
ICONC	INTEGER	TABLES	First and last materials with variable specific heat
ICONK	INTEGER	TABLES	First and last materials with variable thermal conductivity

Appendix D (cont'd)

<u>Variable</u>	<u>Type</u>	<u>Common Block</u>	<u>Variable Use</u>
ICWU	REAL	ZCALC	Energy of integrated water release, Btu
IHTN	REAL	blank	Integrated heat transfer from structures
IHT1	REAL	blank	
II	INTEGER	CHEAT	Maximum number of heat structures
III	INTEGER	CHEAT	Maximum number of temperature nodes in heat structures
IRCYL	INTEGER	TABLES	Source and destination of recycled sodium
ISTO2	INTEGER	INOUT	Number of input data words on card
ISUM	INTEGER	SUMRYT	Subscript on SUMMARY
ISUMRY	INTEGER	SUMRYT	Size of SUMMARY table
ISWC02	INTEGER	TABLES	Concrete carbon dioxide release source and destination
ISWCWR	INTEGER	TABLES	Concrete water release source and destination
IT	INTEGER	THEAT	
I1	INTEGER	POINTT	Limit of data input to data tables
I11	INTEGER	POINTT	
I12	INTEGER	POINTT	
I13	INTEGER	POINTT	
I14	INTEGER	POINTT	
I15	INTEGER	POINTT	
I16	INTEGER	POINTT	
I17	INTEGER	POINTT	
I18	INTEGER	POINTT	
I19	INTEGER	POINTT	
I2	INTEGER	POINTT	
I20	INTEGER	POINTT	
I21	INTEGER	POINTT	
I22	INTEGER	POINTT	
I23	INTEGER	POINTT	
I24	INTEGER	POINTT	
I25	INTEGER	POINTT	
I26	INTEGER	POINTT	
I27	INTEGER	POINTT	
I3	INTEGER	POINTT	

Appendix D (cont'd)

<u>Variable</u>	<u>Type</u>	<u>Common Block</u>	<u>Variable Use</u>
I32	INTEGER	POINTT	
I33	INTEGER	POINTT	
I34	INTEGER	POINTT	
I35	INTEGER	POINTT	
I36	INTEGER	POINTT	
I37	INTEGER	POINTT	
I38	INTEGER	POINTT	
I4	INTEGER	POINTT	
I5	INTEGER	POINTT	
I6	INTEGER	POINTT	
I7	INTEGER	POINTT	
I8	INTEGER	POINTT	
I9	INTEGER	POINTT	
J1	INTEGER	POINTT	
J11	INTEGER	POINTT	
J12	INTEGER	POINTT	
J13	INTEGER	POINTT	
J14	INTEGER	POINTT	
J15	INTEGER	POINTT	
J16	INTEGER	POINTT	
J17	INTEGER	POINTT	
J18	INTEGER	POINTT	
J19	INTEGER	POINTT	
J2	INTEGER	POINTT	Location index to data tables
J20	INTEGER	POINTT	
J21	INTEGER	POINTT	
J22	INTEGER	POINTT	
J23	INTEGER	POINTT	
J24	INTEGER	POINTT	
J25	INTEGER	POINTT	
J26	INTEGER	POINTT	
J27	INTEGER	POINTT	
J3	INTEGER	POINTT	

Appendix D (cont'd)

<u>Variable</u>	<u>Type</u>	<u>Common Block</u>	<u>Variable Use</u>
J32	INTEGER	POINTT	
J33	INTEGER	POINTT	
J34	INTEGER	POINTT	
J35	INTEGER	POINTT	
J36	INTEGER	POINTT	
J37	INTEGER	POINTT	
J38	INTEGER	POINTT	
J4	INTEGER	POINTT	
J5	INTEGER	POINTT	
J6	INTEGER	POINTT	
J7	INTEGER	POINTT	
J8	INTEGER	POINTT	
J9	INTEGER	POINTT	
KS	REAL	blank	Thermal conductivity used in HCALC
KSCS	REAL	THEAT	Thermal conductivity and volumetric heat capacity table
KT	INTEGER	THEAT	Kind of HTCT heat transfer table
LABLE	INTEGER	LABLET	Cell labels for output
LEAK	REAL	FLODAT	Leakage between cells, Btu/hr
LINE	INTEGER	INOUT	Output line number
LOCOAB	INTEGER	LEAK	Leakage coefficient data tables
LOCOAC	INTEGER	LEAK	
LOCOAD	INTEGER	LEAK	
LOCOAO	INTEGER	LEAK	
LOCOBC	INTEGER	LEAK	
LOCOBD	INTEGER	LEAK	
LOCOBO	INTEGER	LEAK	
LOC OCD	INTEGER	LEAK	
LOCOCO	INTEGER	LEAK	
LT	REAL	POLSUR	Pool surface linear dimension, ft
MAT	REAL	FLODAT	Mass of atmosphere, lbs
MCB	REAL	CELL2	Mass of carbon, lbs

Appendix D (cont'd)

<u>Variable</u>	<u>Type</u>	<u>Common Block</u>	<u>Variable Use</u>
MCP	REAL	CELL2	Mass of carbon dioxide added to sodium pool, lbs
MCT	REAL	CELL2	Mass of carbon dioxide, lbs
MGT	REAL	CELL2	Mass of sodium vapor, lbs
MHT	REAL	CELL2	Mass of hydrogen, lbs
MLCT	REAL	CELL2	Moles of carbon dioxide, lb-moles
MLHT	REAL	CELL2	Moles of hydrogen, lb-moles
MLNT	REAL	CELL2	Moles of nitrogen, lb-moles
MLOT	REAL	CELL2	Moles of oxygen, lb-moles
MLST	REAL	CELL2	Moles of sodium vapor, lb-moles
MLT	REAL	CELL2	Mass of condensate to pool from EMVBAL, lbs
MLWT	REAL	CELL2	Moles of water vapor, lb-moles
MN	INTEGER	blank	Region material number used in HCALC
MNAC03	REAL	CELL2	Mass of sodium carbonate (not used), lbs
MNAH	REAL	CELL2	Mass of sodium hydride, lbs
MNAOH	REAL	CELL2	Mass of sodium hydroxide, lbs
MNA2O	REAL	CELL2	Mass of sodium oxide, lbs
MNT	REAL	CELL2	Mass of nitrogen, lbs
MOLT	REAL	CELL2	Total moles in cell, lb-moles
MOT	REAL	CELL1	Mass of oxygen, lbs
MPT	REAL	CELL2	Mass of sodium pool, lbs
MPWT	REAL	CELL2	Mass of water pool, lbs
MST	REAL	CELL1	Mass of sodium in atmosphere, lbs
MWP	REAL	CELL2	Mass of water in sodium pool, lbs
MWT	REAL	CELL2	Mass of water vapor, lbs
NATA	REAL	TABLES	Gas injection to cell atmosphere tables
NATB	REAL	TABLES	
NATC	REAL	TABLES	
NATD	REAL	TABLES	
NERR	INTEGER	ZINPT	Number of errors
NLEAK	REAL	FLODAT	Net leakage, lbs
NLKSW	LOGICAL	ZCALC	Leakage switch if TRUE, no leakage for cell
NON	INTEGER	blank	Number of temperature nodes in heat structure, used in HCALC

Appendix D (cont'd)

<u>Variable</u>	<u>Type</u>	<u>Common Block</u>	<u>Variable Use</u>
NOR	INTEGER	blank	Number of regions in heat structure used in HCALC
NOZZAB	REAL	NOZZLE	Nozzle data talks
NOZZAC	REAL	NOZZLE	
NOZZAD	REAL	NOZZLE	
NOZZAO	REAL	NOZZLE	
NOZZBC	REAL	NOZZLE	
NOZZBD	REAL	NOZZLE	
NOZZBO	REAL	NOZZLE	
NOZZCD	REAL	NOZZLE	
NOZZCO	REAL	NOZZLE	
NPAGE	INTEGER	INOUT	Page number of output
NPN	INTEGER	PHEAT	Number of boundary conditions on pool used in HCALC
NP1	INTEGER	PHEAT	
NRN	INTEGER	RHEAT	
NRT	REAL	LEKRT	Net leak rate, lbs/hr
NR1	INTEGER	RHEAT	Number of boundary conditions for radiation, used in HCALC
NS	INTEGER	blank	Number of mesh points in a heat structure region
NSL	INTEGER	CHEAT	Number of heat structures
N17	LOGICAL	POINTN	Space cooler logical switches
N18	LOGICAL	POINTN	
N19	LOGICAL	POINTN	
N20	LOGICAL	POINTN	Sodium recycle logical switch
N21	LOGICAL	POINTN	
N1	LOGICAL	POINTN	Nozzle logical switches
N2	LOGICAL	POINTN	
N3	LOGICAL	POINTN	
N4	LOGICAL	POINTN	
N5	LOGICAL	POINTN	
N6	LOGICAL	POINTN	
N7	LOGICAL	POINTN	
N8	LOGICAL	POINTN	
N9	LOGICAL	POINTN	

Appendix D (cont'd)

Variable	Type	Common Block	Variable Use
OATH	REAL	TABLES	Outside air conditions table
PARN	REAL	PHEAT	} Pool surface areas
PAR1	REAL	PHEAT	
PBHN	INTEGER	PHEAT	} Pool boundary heat transfer coefficient
PBH1	INTEGER	PHEAT	
PBTN	INTEGER	PHEAT	} Pool boundary temperature index
PBT1	INTEGER	PHEAT	
PHN	REAL	PHEAT	Pool boundary heat transfer coefficient
PH1	REAL	PHEAT	} Pool boundary heat transfer rates, Btu/hr
PHTN	REAL	PHEAT	
PHT1	REAL	PHEAT	
PIHTN	REAL	PHEAT	} Integrated pool boundary heat transfer, Btu
PIHT1	REAL	PHEAT	
POOL, IPOOL	REAL	INOUT	Input data temporary storage
PPABCD	REAL	TABLES	Decay power apportioning factors to pools
PRNTC	REAL	PRINTT	Print out control
PRNTFC	REAL	PRINTT	Containment data print control
PRNTFH	REAL	PRINTT	Heat structure data print control
PRNTH	REAL	PRINTT	Print out control
PSIA	REAL	SODIUM	Sodium vapor pressure, psia
PSIW	REAL	WATER	Water vapor pressure, psia
PST	REAL	CELL2	Vapor pressure, psia
PTN	REAL	PHEAT	} Pool boundary temperature index
PT1	REAL	PHEAT	
PTT	REAL	CELL1	Cell atmosphere pressure, psia
PTTO	REAL	FLODAT	Old cell atmosphere pressure, psia
PURFLO	REAL	LEAK	Average purge flow, SCFM
RATIO	REAL	FLODAT	Mass ratios for leakage calculations
RBHN	INTEGER	RHEAT	} Boundary radiation heat transfer coefficient index
RBH1	INTEGER	RHEAT	
RBTN	INTEGER	RHEAT	} Boundary radiation temperature index
RBT1	INTEGER	RHEAT	
REACT	REAL	POLSUR	Heat due to pool surface reaction

Appendix D (cont'd)

<u>Variable</u>	<u>Type</u>	<u>Common Block</u>	<u>Variable Use</u>
RECYL	REAL	TABLES	Sodium cell to cell recycle table
RESTAR	LOGICAL		Logical variable indicating restart run
RHN	REAL	RHEAT	Boundary radiation heat transfer coefficient
RH1	REAL	REHEAT	
RHOT	REAL	CELL2	Atmosphere density, lbs/ft ³
RHTN	REAL	RHEAT	Boundary radiation heat transfer rates, Btu/hr
RHT1	REAL	RHEAT	
RIHTN	REAL	RHEAT	Integrated radiation heat transfer, Btu
RIHT1	REAL	RHEAT	
RS	REAL	blank	Heat structure temperature node location, ft
RTN	REAL	RHEAT	Boundary radiation temperatures, °F
RT1	REAL	RHEAT	
SATA	REAL	TABLES	Sodium sprays to cell atmosphere tables
SATB	REAL	TABLES	
SATC	REAL	TABLES	
SATD	REAL	TABLES	
SDT	REAL	blank	Volumetric heat source delay time, hrs
SGN	REAL	SODIUM	Entropy of sodium vapor, Btu/hr-°R
SHTR	REAL	EHEAT	Sum of heat transfer rates, Btu/hr
SIHTR	REAL	EHEAT	Sum of integrated heat transfer, Btu
SKT	REAL	LEKRT	Step sum of leakage, lbs
SLEAK	REAL	FLODAT	Total leakage, lbs
SLN	REAL	SODIUM	Entropy of sodium liquid, Btu/hr-°R
SPF	REAL	blank	Volumetric heat power factor, used in HCALC
SPTA	REAL	TABLES	Sodium spills to cell pool tables
SPTB	REAL	TABLES	
SPTC	REAL	TABLES	
SPTD	REAL	TABLES	
SRN	REAL	blank	Heat structure surface geometry factor used in HCALC
SR1	REAL	blank	
SRT	REAL	LEKRT	Step sum of leakrates, lbs/hr
SS	REAL	blank	Heat structure geometry factors used in HCALC

Appendix D (cont'd)

<u>Variable</u>	<u>Type</u>	<u>Common Block</u>	<u>Variable Use</u>
STEP	REAL	PRINTT	Time step increase factor
ST01	REAL	RICH5	} Temporary variables
ST02	REAL	RICH5	
ST03	REAL	RICH5	
SUMMRY	REAL	SUMRYT	Data to be printed by OUTSUM
SW	REAL	SWITCH	Switch indicating cell exists
SWCO2R	REAL	TABLES	Structures with carbon dioxide release table
SWCWR	REAL	TABLES	Structures with water release table
SW1	REAL	SWITCH	Initial conditions switch
SW3	REAL	SWITCH	Print heat structure data if FALSE
SW4	REAL	SWITCH	Print containment data if FALSE
SW5	REAL	SWITCH	Switch variable
SW6	LOGICAL	ZCALC	No leakage calculations if TRUE
T	REAL	TIMET	Time, hours
TAT	REAL	CELL1	Cell atmosphere temperature, °F
TATO	REAL	BHEAT	Old cell atmosphere temperature, °F
TBOLT	REAL	CELL2	Temperature pool would boil, °F
TCONT	REAL	BHEAT	Constant heat sink temperature, °F
TF	REAL	SODIUM	Temperature for sodium properties, °F
TFNL	REAL	TIMET	End of run time, hours
TFW	REAL	WATER	Temperature used for water properties, °F
TLHS	REAL	blank	Heat structure titles
TNNEW	REAL	BHEAT	Temperature for heat transfer calculation
TPT	REAL	CELL2	Pool temperature, °F
TPTO	REAL	BHEAT	Old pool temperature, °F
TR	REAL	SODIUM	Temperature used for sodium properties, °R
TRW	REAL	WATER	Temperature used for water properties, °R
TSPC	REAL	TABLES	Time step, print control table
TSUR	REAL	BHEAT	Surface temperature used by BONDRY
TINEW	REAL	BHEAT	Temperature for heat transfer calculation
UAT	REAL	CELL1	Cell atmosphere energy, Btu
UCHNG	REAL	CELL3	Energy of vaporization, Btu/lb

Appendix D (cont'd)

<u>Variable</u>	<u>Type</u>	<u>Common Block</u>	<u>Variable Use</u>
UGBOL	REAL	CELL3	Energy of boiling vapor, Btu/lb
UGN	REAL	SODIUM	Energy of sodium vapor, Btu/lb
UGT	REAL	CELL3	Energy of sodium vapor, Btu/lb
UGW	REAL	WATER	Energy of water vapor, Btu/lb
UGWT	REAL	CELL3	Energy of water vapor, Btu/lb
ULBOL	REAL	CELL3	Energy of boiling liquid, Btu/lb
ULN	REAL	SODIUM	Energy of sodium liquid, Btu/lb
ULT	REAL	CELL3	Energy of sodium liquid, Btu/lb
ULW	REAL	WATER	Energy of liquid water, Btu/lb
UPT	REAL	CELL2	Energy of pool, Btu
US	REAL	blank	Heat structure temperatures, °F
USM	REAL	blank	Maximum heat structure temperature, t, °F
USN	REAL	blank	} Heat structure surface temperatures, °F
US1	REAL	blank	
VGN	REAL	SODIUM	Specific volume of sodium vapor, ft ³ /lb
VGT	REAL	CELL3	Specific volume of sodium vapor ft ³ /lb
VGW	REAL	WATER	Specific volume of water vapor, ft ³ /lb
VGWT	REAL	CELL3	Specific volume of water vapor, ft ³ /lb
VLBOL	REAL	CELL3	Specific volume of boiling liquid, ft ³ /lb
VLN	REAL	SODIUM	Specific volume of liquid sodium, ft ³ /lb
VLT	REAL	CELL2	Specific volume of liquid, ft ³ /lbs
VLW	REAL	WATER	Specific volume of liquid water, ft ³ /lb
VOLAT	REAL	CELL3	Atmosphere volume, ft ³
VOLT	REAL	CELL1	Cell volume, ft ³
VS	REAL	blank	} Heat structure node geometry factors used in HCALC
WS	REAL	blank	
XS	REAL	blank	Heat structure element volume, ft ³
YS	REAL	blank	Heat structure variable for HCALC

APPENDIX E

SOLUTION OF THE HEAT CONDUCTION EQUATION

APPENDIX E SOLUTION OF THE HEAT CONDUCTION EQUATION

The temperature distribution through each heat structure is determined by the solution to the one-dimensional, multi-region heat conduction equation. This solution was developed by R. J. Wagner as the HEAT1 program in 1963 (see IDO-16867) and used in the CONTEMPT program in 1967 (see IDO-17220). This solution is one of the best features of the CACECO code because it provides the opportunity for sophisticated models of LMFBR facility structures without the complexity of two-dimensional representation. A summary of this solution, used in HCALC, is given below.

The partial differential equation for which a numerical approximation is derived is:

$$g(x) \frac{\partial}{\partial t} u(x,t) = \nabla \cdot k(x) \nabla u(x,t) + S(x,t) \quad (E1)$$

where

u = temperature

x = space variable

t = time variable

g = volumetric heat capacity

k = thermal conductivity

S = source term per unit volume.

Approximations are derived for planar, cylindrical, and spherical geometries in one space dimension. Figure E1 illustrates the placement of node points at which the temperatures will be calculated.

Node points are placed on the external boundaries of the structure, at the interfaces between materials, and at equal intervals between the interfaces and/or boundaries. A region is a structure part which contains the same material and has a constant spacing between node points. A new region is required for a different material and/or a different node point spacing.

FIGURE E1. Node Point Layout

Figure E2 represents three typical node points. The h 's indicate the node point spacings, and these are not necessarily equal. Subscripts designate space indexes and, in particular, subscripts ℓ and r designate quantities to the left and right, respectively, of the node point. Between node points k , g , and S are assumed constant, but $k_{\ell n}$ is not necessarily equal to k_{rn} , and similarly for g and S .

FIGURE E2. Typical Node Points

To obtain the spatial difference approximation for the n th node point, Equation (E1) is integrated over an incremental volume as indicated by the dashed line in Figure E2. Since the code is one dimensional, the other

dimensions of this volume are set to one. For planar geometry, the volume is a rectangular solid. For cylindrical geometry, the volume is a cylindrical annulus; and for spherical geometry, the volume is spherical shell. For convenience, the following volume and surface gradients are defined with superscripts, v and s, to denote these volume and surface-gradient weights. The surface weight h_n^b is used only at exterior boundaries. All of these are calculated by HINPUT for use by HCALC.

For planar geometry,

$$\begin{aligned} h_{\ell n}^v &= \frac{h_{\ell n}}{2} & h_{rn}^v &= \frac{h_{rn}}{2} \\ h_{\ell n}^s &= \frac{1}{h_{\ell n}} & h_{rn}^s &= \frac{1}{h_{rn}} \\ h_n^b &= 1. \end{aligned} \tag{E2}$$

For cylindrical geometry,

$$\begin{aligned} h_{\ell n}^v &= 2\pi \frac{h_{\ell n}}{2} \left(x_n - \frac{h_{\ell n}}{4} \right) & h_{rn}^v &= 2\pi \frac{h_{rn}}{2} \left(x_n + \frac{h_{rn}}{4} \right) \\ h_{\ell n}^s &= \frac{2\pi}{h_{\ell n}} \left(x_n - \frac{h_{\ell n}}{2} \right) & h_{rn}^s &= \frac{2\pi}{h_{rn}} \left(x_n + \frac{h_{rn}}{2} \right) \\ h_n^b &= 2\pi x_n. \end{aligned} \tag{E3}$$

For spherical geometry,

$$\begin{aligned} h_{\ell n}^v &= \frac{4}{3} \pi \left\{ x_n^3 - \left(x_n - \frac{h_{\ell n}}{2} \right)^3 \right\} & h_{rn}^v &= \frac{4}{3} \pi \left\{ \left(x_n + \frac{h_{rn}}{2} \right)^3 - x_n^3 \right\} \\ h_{\ell n}^s &= \frac{4\pi}{h_{\ell n}} \left(x_n - \frac{h_{\ell n}}{2} \right)^2 & h_{rn}^s &= \frac{4\pi}{h_{rn}} \left(x_n + \frac{h_{rn}}{2} \right)^2 \\ h_n^b &= 4\pi x_n^2 \end{aligned} \tag{E4}$$

For all geometries for node N of structure I, the volume weight $h_{\ell n}^v$ is stored in FORTRAN array VS(N,I), the volume weight h_{rn}^v is stored in array WS(N,I), and the surface gradient weight h_{rn}^s is stored in array SS(N,I). The surface weight at the left hand (inside) exterior boundary, h_n^b , is stored in array SRl(I) and the surface weight at the right hand (outside) exterior boundary is stored in array SRN(I). Also for all geometries, the heat capacity of the node is the sum of left hand and right hand parts,

$$D_n = g_{\ell n} h_{\ell n}^v + g_{rn} h_{rn}^v \quad (E5)$$

The space dependence and the time dependence of the source term are assumed separable, i.e.,

$$S(x,t) = P_f P_t \lambda(t) Q(x) \quad (E6)$$

where P_f is a power factor, $P_t \lambda(t)$ is the time-dependent part, and $Q(x)$ is the space dependent part. The time dependent part consists of two functions, P_t and $\lambda(t)$. The function P_t is defined by:

$$\begin{aligned} P_t &= 1 \text{ when volumetric heat is on as defined by Word 7 of the} \\ &\quad \text{1YY000 card for each heat structure and} \\ P_t &= 0 \text{ when volumetric heat is off. This control is stored in} \\ &\quad \text{array SDT(I).} \end{aligned}$$

The function $\lambda(t)$ is the decay power value taken from the decay power table (Card series 101). The space-dependent part, $Q(x)$, has left and right hand parts denoted by subscripts ℓ and r as described previously. The term sum, $(Q_{\ell n} h_{\ell n}^v + Q_{rn} h_{rn}^v)$, is stored in array YS(N,I).

The finite difference approximation to the integral form of Equation (E1) is the basic difference equation for the nth node point at times t_m and t_{m+1} . This approximation is written as follows.

$$\begin{aligned} \frac{(u_n^{m+1} - u_n^m)}{(t_{m+1} - t_m)} D_n &= - (u_n - u_{n-1}) k_{\ell n} h_{\ell n}^s + (u_{n+1} - u_n) k_{rn} h_{rn}^s \\ &+ P_f P_t \lambda (Q_{\ell n} h_{\ell n}^v + Q_{rn} h_{rn}^v) \end{aligned} \quad (E7)$$

The symbol u_n^m indicates the node temperature at time t_m , and the symbol u_n^{m+1} indicates the temperature at time t_{m+1} . Equation (E7) can be written when the right-hand side is given by the symbol δ_n ,

$$\frac{(u_n^{m+1} - u_n^m) D_n}{\Delta t} = \delta_n.$$

Thus far, the time superscript for δ_n has been omitted. For steady state,

$$\delta_n = 0$$

and no time superscripts are needed.

For the time-dependent case, the average of the value of δ_n at the beginning and the end of the time interval is used, giving the following implicit equation.

$$\frac{(u_n^{m+1} - u_n^m) D_n}{\Delta t} = \frac{\delta_n^{m+1} + \delta_n^m}{2} \quad (E8)$$

In this equation, the u^m temperatures are known because they are either the results of the steady state calculation or the results of the last time step. The determination of u^{m+1} temperatures requires the solution of simultaneous equations.

Writing Equation (E8) out in full, the difference approximation for the n th interior node point for transient and steady state cases is

$$a_n u_{n-1}^{m+1} + b_n u_n^{m+1} + c_n u_{n+1}^{m+1} = d_n \quad (E9)$$

where

$$a_n = -\frac{k_{ln} h_{ln}^S \Delta t}{2} \quad c_n = -\frac{k_{rn} h_{rn}^S \Delta t}{2} \quad b_n = D_n - a_n - c_n$$

$$d_n = -a_n u_{n-1}^m + (D_n + a_n + c_n) u_n^m - c_n u_{n+1}^m \\ + \Delta t P_f P_t \left(\frac{\lambda^{m+1} + \lambda^m}{2} \right) (Q_{ln} h_{ln}^V + Q_{rn} h_{rn}^V)$$

For node N of structure I, the term a_n is stored in FORTRAN array AS(N), the term b_n is stored in array BS(N), the term c_n is stored in array AS(N+1), and term d_n is stored in array GS(N). The term D_n is the temporary variable ST01 in HCALC.

The CACECO code assumes heat transfer by convection at the surfaces or exterior boundaries of each heat structure. There are two such boundaries for each structure. At each boundary, the code also allows heat transfer by radiation to six other temperatures which may represent an atmosphere or pool and other structures. Each radiation heat transfer exchange is handled as if it were another convection heat transfer exchange with an effective heat transfer coefficient that takes into account the Stefan-Boltzmann constant and other radiation principles. That is, with convection and radiation both, the boundary (surface) can exchange heat with more than one "bulk" temperature and each exchange will use its own heat transfer coefficient. These CACECO boundary conditions are written as follows.

$$-k \frac{\partial u}{\partial \bar{x}} = \sum_i H_i(u,t) [u - u_{Bi}(t)] \quad (E10)$$

where u is the exterior boundary (surface) temperature

$u_{Bi}(t)$ is the i 'th bulk temperature

$\bar{x}$ is a vector in the direction of out of the heat structure

k is the thermal conductivity of the structure material

$H_i(u,t)$ is the heat transfer coefficient for the exchange with the i 'th bulk temperature

The specification of $H_i(u,t)$ and $u_{Bi}(t)$ for each boundary are done by input data indices.

The method described above is again used to obtain the difference approximation at the exterior boundaries, but with the volume of integration as indicated in Figure E3.

FIGURE E3. Boundary Node Points.

The complete equation for the node point at $x = x_1$ becomes

$$\left(\frac{u_1^{m+1} - u_1^m}{\Delta t} \right) g_{r1} h_{r1}^v = + \sum_i H_{i1} (u_{Bi1} - u_1) h_1^b$$

$$+ (u_2 - u_1) k_{r1} h_{r1}^s + P_f P_t \lambda Q_{r1} h_{r1}^v$$

Approximations for the boundary at $x = x_n$ are derived in a similar fashion. These equations for the boundary mesh points are converted to the implicit formula in the same manner as for the interior points. Thus, for $x = x_1$,

$$b_1 u_1^{m+1} + c_1 u_2^{m+1} = d_1 \quad (E11)$$

where

$$c_1 = - \frac{k_{r1} h_{r1}^s \Delta t}{2}, \quad b_1 = g_{r1} h_{r1}^v + \sum_i H_{i1} h_1^b \frac{\Delta t}{2} - c_1$$

$$d_1 = c_1 u_2^m + (g_{r1} h_{r1}^v - \sum_i H_{i1} h_1^b \frac{\Delta t}{2} + c_1) u_1^m$$

$$+ \sum_i H_{i1} \left(u_{Bi1}^{m+1} + u_{Bi1}^m \right) h_1^b \frac{\Delta t}{2} + \Delta t P_f P_t \left(\frac{\lambda^{m+1} + \lambda^m}{2} \right) Q_{r1} h_{r1}^v$$

For $x = x_N$,

$$a_N u_{N-1}^{m+1} + b_N u_N^{m+1} = d_N \quad (E12)$$

where

$$a_N = - \frac{k_{\ell N} h_{\ell N}^S \Delta t}{}, \quad b_N = g_{\ell N} h_{\ell N}^V + \sum_i H_{iN} h_N^b \frac{\Delta t}{2} - a_N ,$$

$$d_N = - a_N u_{N-1}^m + \left(g_{\ell N} h_{\ell N}^V - \sum_i H_{iN} h_N^b \frac{\Delta t}{2} + a_N \right) u_N^m \\ + \sum_i H_{iN} \left(u_{BiN}^{m+1} + u_{BiN}^m \right) h_N^b \frac{\Delta t}{2} + \Delta t P_f P_t \left(\frac{\lambda^{m+1}}{2} + \frac{\lambda^m}{2} \right) Q_{\ell N} h_{\ell N}^V$$

The difference approximations for the mesh points, Equations (E9), (E11), and (E12) lead to a tridiagonal set of $N + 1$ equations.

$$\begin{array}{cccc|c|cccc} b_1 & c_1 & 0 & 0 & & u_1^{m+1} & & d_1 \\ a_2 & b_2 & c_2 & 0 & 0's & u_2^{m+1} & & d_2 \\ 0 & a_3 & b_3 & c_3 & & u_3^{m+1} & & d_3 \\ & . & . & . & & . & & . \\ & & . & . & & . & & . \\ & & . & . & & . & & . \\ & & . & . & & . & & . \\ 0's & a_{N-1} & b_{N-1} & c_{N-1} & & u_{N-1}^{m+1} & & d_{N-1} \\ & 0 & a_N & b_N & & u_N^{m+1} & & d_N \end{array} \quad \times \quad = \quad (E13)$$

With the boundary conditions of Equation (E10) the coefficient matrix is always symmetric.

The solution to the tridiagonal set of equations is obtained by Gaussian elimination as follows.

$$(1) \text{ Form: } E_1 = \frac{c_1}{b_1} \text{ and } F_1 = \frac{d_1}{b_1} .$$

$$(2) \text{ Form: } E_j = \frac{c_j}{b_j - a_j E_{j-1}} \text{ and } F_j = \frac{d_j - a_j F_{j-1}}{b_j - a_j E_{j-1}}$$

for $j = 2, 3, \dots, N-1$

(3) The boundary temperature is: $u_N^{m+1} = \frac{d_N - a_N F_{N-1}}{b_N - a_N E_{N-1}}$

(4) The other temperatures are: $u_j^{m+1} = -E_j u_{j+1} + F_j$

for $j = N-1, N-2, \dots, 2, 1$.

A list of corresponding CACECO variables is provided below:

<u>Variable</u>	<u>FORTTRAN Variable</u>	<u>Meaning</u>
a_n	AS(I)	Composite variables used for each heat structures, Equations E9, E11 and E12.
b_n	BS(I)	
c_n	AS(I)	
d_n	GS(I)	
g_{rn}, g_{ln}	CS(I)	Volumetric heat capacity, Btu/ft ³ -°F
h_{rn}^v, h_{rn}^s	VS(J,I), SS(J,I)	Node point geometry weights of Equations E2, E3, and E4
h_{ln}^v, h_{ln}^s	WS(J,I), SS(J,I)	
h_N^b	SRN(I)	Surface boundary geometry weights of Equations E2, E3, and E4
h_1^b	SR1(I)	
k_{ln}, k_{rn}	KS(I)	Thermal conductivity, Btu/hr-ft-°F
u_n^m	US(J,I)	Node point temperatures, °F
x_n	RS(J,I)	Node point locations, ft
$H(u,t)$	H1,HN	Boundary heat transfer coefficient, Btu/hr ft ² °F
u_B	T1,TN	Bulk Temperature, °F
u	TSUR,US1(I),USN(I)	Boundary Temperature, °F
D_n	ST01	Composite variable with heat capacity.

<u>Variable</u>	<u>FORTTRAN Variable</u>	<u>Meaning</u>
E_j	AS(J)	Temporary variable used in Gaussian elimination solution
F_j	GS(J)	
P_f	SPF(I)	Volumetric heat power factor, $1/\text{ft}^3$
P_t	-----	Volumetric heat control stored in SDT(I)
$Q(x)$	YS(J,I)	Volumetric heat distribution
$\lambda(t)$	POWIN	Volumetric heat generation rate, Btu/hr
Δt	DT	Time step, hours
$\Delta t/2$	DTOV2	Half time step, hours

APPENDIX F

CACECO CODE CASE 731

Appendix F - CACECO CODE CASE 731

This example of CACECO code input describes analysis of a hypothetical reactor vessel melt-through incident in a 2500 MWt LMFBR containment typical of pipe loop LMFBRs. This example was discussed in Section 2.0 of the Users' Guide. The fourteen pages of this example are printed on one microfiche in order to reduce the bulk of this report. A card copy of Case 731 input data is available from the author.

DISTRIBUTION

UC 79b (253)

DOE/RL

Chief Patent Attorney

DOE/FFTFO (5)

Director

DOE/RRT-HQ (2) B-107

Program Division Director

HEDL (29)

G. R. Armstrong	W/C-78
J. M. Atwood	W/C-45
G. R. Franz	W/C-76
J. W. Hagan	W/C-78
J. P. Hale	W/C-76
T. A. Mangelsdorf	W/B-63
H. C. Martin	W/C-76
W. J. McShane	W/B-12
L. D. Muhlestein	W/221T
A. Padilla, Jr.	W/F-126C
R. D. Peak (2)	W/C-75
M. G. Piepho	W/C-78
W. F. Sheely	W/C-44
D. E. Simpson	W/C-80
D. D. Stepnewski	W/C-75
E. J. Walter	W/B-17
Central Files (10)	W/C-110
Publ Svcs (2)	W/C-115

WESTINGHOUSE ELECTRIC CORP. (3)

Advanced Reactors Div Lib (2)

Box 158

Madison, PA 15663

L. E. Strawbridge (1)