AIM

Association for Information and Image Management

1100 Wayne Avenue, Suite 1100
Silver Spring, Maryland 20910
301/587-8202

Centimeter

Inches

MANUFACTURED TO AIM STANDARDS
BY APPLIED IMAGE, INC.

1 of 1

KEYWORDS:

Simulator
Neutronics
Modeling

RETENTION:

Lifetime

**Technical Bases of the Second Generation
SARIS Core Model (U)
(Task Number: 90-008-0)**

by

M.V. Gregory

Authorized Derivative Classifier

N.P. Baumann

November 1991

WSRC

WESTINGHOUSE SAVANNAH RIVER COMPANY

AIKEN, SC 29808

**Prepared for the U.S. Department of Energy under Contract Number
DE-AC09-80SR18035**

MASTER

DISTRIBUTION OF THIS DOCUMENT IS UNLIMITED

DOCUMENT: WSRC-RP-91-1148
TITLE: Technical Bases of the Second
Generation SARIS Core Model (U)
TASK: 90-008-0, SARIS Verification and
Validation Support

APPROVALS

R.E. Pevey, Technical Reviewer DATE: 2-4-92

C.E. Apperson, Manager-Reactor Physics Group DATE: 2-17-92

R.R. Beckmeyer, Manager-Computing Technology Group DATE: 03/03/92

M.R. Buckner, Manager-Scientific Computations Section DATE: 03/05/92

SUMMARY

A methodology has been developed to rigorously derive the constants in the Savannah River Simulator (SARIS) core model from detailed, charge-design, diffusion theory solutions. This methodology is intended to replace the ill-defined, *ad hoc* iterative process used in the past to generate these constants. Along the development path, three shortcomings of the current core model were identified and corrected. The updated core model with revised constants is termed the second generation core model. In addition, changes in the decay heat and delayed neutron precursor models are also recommended, all in the interest of improving simulator neutronics fidelity.

INTRODUCTION

The basic equation for the prompt neutrons in the current SARIS formulation is:

$$dn_i/dt = n_i(\rho_i - \beta_i)/\Lambda_i + \sum_k \lambda_k C_{ki} + \sum_j K_{ji}(n_j - n_i) \quad (1)$$

where

n_i	= neutron density in node i
ρ_i	= reactivity of node i
β_i	= delayed neutron fraction in node i
Λ_i	= prompt neutron lifetime in node i
λ_k	= decay constant for delayed precursor group k
C_{ki}	= delayed precursor concentration of group k in node i
K_{ji}	= leakage coupling constant between neighbor node j and node i; in SARIS, $K_{ji}=K_{ij}$
k	= six precursor groups are modeled
i, j	= 240 nodes make up the representation (24 in each of 10 axial levels)

There is also an associated equation for the delayed precursor concentrations that we will defer to a later section: it is identical to the usual textbook precursor equation with the addition of the node index.

The work at hand was motivated by the need to put the model constants in

Eq. 1 on a more rigorous basis. The methodology developed to provide that rigor is the subject of this report. In addition, during the course of that development, questions arose as to the correctness of the form of Eq. 1 and its application to the SRS reactor geometry. This second topic will be dealt with first since it determines the definition of the model constants. Then the methodology for deriving the model constants will be described. Finally we will close with proposed changes for two other models associated with the core model: the decay heat and delayed neutron precursor models.

FORM OF FUNDAMENTAL EQUATION

With the present formulation in SARIS, a problem arises in the behavior of the leakage term (the last term on the right hand side) in Eq. 1: neutrons are not conserved in leakage between nodes of dissimilar volumes. The best way to illustrate the problem is to provide a specific example. Assume that we are looking at only two nodes (V is the nodal volume) where we have leakage from node 1 to node 2:

$$\begin{aligned}n_1 &= 2n_2 \text{ are the initial neutron densities, and} \\V_2 &= 5V_1 \text{ are the two volumes.}\end{aligned}$$

Based on Eq. 1, we have:

$$(dn_1/dt)_{\text{leakage}} = K_{21}(n_2 - n_1) = -K_{21}n_2$$

$$(dn_2/dt)_{\text{leakage}} = K_{12}(n_1 - n_2) = K_{21}n_2$$

where we have used $K_{21} = K_{12}$, the SARIS convention (which is the key to the problem). Now, recall that n is neutron density. The above says that in timestep Δt , the neutron density in node 1 goes down by $K_{21}n_2\Delta t$ while the neutron density in node 2 goes up by a similar amount. But if the densities change by the same amount and the volumes are different, we have not conserved neutrons. That is, the neutron population in node 1 decreases by $K_{21}n_2\Delta tV_1$ while the neutron population in node 2 increases by $K_{21}n_2\Delta tV_2 = 5K_{21}n_2\Delta tV_1$. Thus five times as many neutrons show up in node 2 as left node 1. This is the inconsistency implicit in the current model. Fortunately, in the SARIS model adjacent nodes differ in volume by only tens of percent rather than by multiples of five. In addition, the

leakage coupling constants in the model are large enough that the "spurious neutrons" do not all pile up in the destination node: the gradient drives them back the other way. By definition, the resulting error must be small since large discrepancies in flux shape have not been observed.

The fix is simple enough: we just include the volume factor in the leakage term in Eq. 1 and formulate the new coupling constant K :

$$dn_i/dt = n_i(\rho - \beta)_i/\Lambda_i + \sum_k \lambda_k C_{ki} + (1/V_i) * \sum_j K_{ji}(n_j - n_i) \quad (2)$$

Then in our two-node sample problem, $(dn_1/dt)_{\text{leakage}} = -K_{21}n_2/V_1$ while $(dn_2/dt)_{\text{leakage}} = K_{21}n_2/V_2$, and the neutron balance remains consistent with node 1 losing $K_{21}n_2\Delta t$ neutrons while node 2 gains the identical amount. Our convention remains that $K_{ji} = K_{ij}$, but $K_{ji}/V_i \neq K_{ij}/V_j$ in general.

Note that if all the nodes in the model have the identical volume, then in Eq. 2 the $1/V$ factor can be moved inside the summation, and we recover the original Eq. 1 with $K_{ji} = K_{ji}/V$. It is only when the node volumes are unequal that the particular form of the leakage term in Eq. 2 needs to be used. In SARIS, all axial levels are not of the same thickness, nor do all nodes contain the same number of clusters; thus the form of Eq. 2 is the one that must be used. When we get ready to generate coupling constants in a rigorous manner, it will be in the context of Eq. 2 rather than Eq. 1.

HEXAGONAL GEOMETRY COUPLING CONSTANTS

With the form of the basic equation established, we now turn to looking at the geometry represented by the K_{ji} coupling constants. Here, another shortcoming of the present SARIS implementation turns up: the $R\theta$ geometrical model fails to capture one class of nodal coupling which is present in the actual hexagonal geometry. This can be illustrated by examining a couple of figures. Figure 1 shows the $R\theta$ geometry as conceptualized in the SARIS model. Note that each node can have at most six neighbors (e.g., node 7 is surrounded by nodes 1, 13, 8, 12, plus the node above and the node below). Thus there are only six leakage couplings defined through the K_{ji} constants in the present model in SARIS.

Figure 2 shows that same geometry, but now overlaid on the actual

Figure 1. Present SARIS Core Nodalization

DIVISION OF THE K REACTOR CORE IN AXIAL AND
PLANAR COORDINATES USED IN NUCLEAR MODEL (NEUTRONICS)

Figure 2. SARIS Nodalization Superimposed on Hexagonal Geometry

hexagonal reactor face map. Now the geometry is not so straightforward. Each node contains several reactor clusters, but those clusters do not couple to other clusters in neighboring nodes in a strictly R0 fashion. Node 7 still has the same six neighbors as in the R0 geometry, but the hexagonal geometry supplies an additional leakage coupling which was absent in the pure R0 geometry: with nodes 2 and 14 as shown by the double-headed arrows.

In the context of the R0 geometry, this can be thought of as an additional "corner" leakage. Thus to properly account for the true leakage geometry, a fourth coupling coefficient (the original three were radial, azimuthal, and axial) needs to be added to the model. Implementation of the new "corner" leakage coefficient will add two new stripes to the solution matrix, which obviously requires a non-trivial change to the current algorithm: each node "i" now has more "j" neighbors expressed in the K_{ji} coupling term. The form of the fundamental equation does not change, but the data in it does. Incidentally, referring to Figure 2, the central Cluster 1 is distributed in equal portions to its neighbors (i.e., one-sixth in each of nodes 1 through 6).

CALCULATION OF K_{ji} COUPLING PARAMETERS

Now that we are satisfied with the form of the fundamental equation, we turn to the calculation of the parameters in that equation. First we examine the leakage coupling constants. Up to now, the K_{ji} terms in the current SARIS model have been defined in an *ad hoc* manner: constants for the K3.3 mixed lattice charge suggested in 1984 (ref. M.V. Gregory, DPST-84-501, "Tuning the SIMULATE Core Neutronics Model, " April 1984) have continued to be used with only minor changes, even for a Mark 22 lattice. In the initial generation of the K14 model (ref. M.V. Gregory, SCS-PMG-91019, "Pre-Restart Tuning of a K-14 Model for SARIS," January 1991), the values originally defined for the old full power K5 Mark 22 charge (which had their origin in the 1984 work) were retained with no changes. Clearly, a more rigorous methodology for generating coupling constants was needed.

That need triggered the current work. The basic idea was to link the SARIS constants to the results of a detailed, three-dimensional diffusion theory solution provided by the standard GRIMHX code: each of the individual leakage contributions in Eq. 2 could be equivalenced to the analogous leakage balance in the GRIMHX solution, thereby providing a way

to calculate the K_{ji} .

All the GRIMHX hex pairs making up one leakage boundary of the SARIS node can be identified (e.g., the 12 possible hex pairs that delineate the radial boundary between nodes 1 and 7 in Figure 2). If the GRIMHX leakage current across that boundary for one hex-pair m is J_m n/cm²/sec, with an interface area of A_m cm², then the total n/sec crossing the boundary is the sum over the m pairs: $\sum_m J_m A_m$. Assuming that the leakage balance represents an increase in the population in SARIS node k , then the change in the neutron density per unit time in that node is $(1/V_k) * \sum_m J_m A_m$. We can now postulate the direct equivalence between this GRIMHX computed quantity and the leakage term in Eq. 2:

$$(1/V_k) * K_{jk} (n_j - n_k) = (1/V_k) * \sum_m J_m A_m \quad (3)$$

where SARIS nodes j and k have the m hex pairs from the GRIMHX hexagonal geometry making up the radial boundary between them (e.g., $k=7$ and $j=1$). For simplicity, the above assumes that the axial mesh between SARIS and GRIMHX is identical. In reality, GRIMHX uses a considerably finer axial mesh than SARIS, thus there should be an additional summation over the GRIMHX axial levels spanned by the SARIS level on the right hand side of Eq. 3.

From Eq. 3, we can solve for the leakage coupling constant that precisely preserves the leakage computed by the more detailed GRIMHX model:

$$K_{jk} = (\sum_m J_m A_m) / (n_j - n_k) \quad (4)$$

Everything on the right hand side of Eq. 4 can be obtained from the GRIMHX solution:

$$J_m = (1/p) * (2/[1/D_i + 1/D_l]) * (\phi_i - \phi_l) \quad (5)$$

which is just the definition of the current based on the flux gradient and the average diffusion coefficient, where hexes i and l make up pair m , p is the hexagonal pitch, ϕ is the flux, and D is the diffusion coefficient. Note that if we are discussing an axial coupling constant, the pitch p in Eq. 5 would be replaced by the axial mesh spacing. Usually the GRIMHX calculation is for a two-group model, thus there is a sum over energy

groups implicit in Eq. 5.

The SARIS nodal neutron densities are also calculated from the GRIMHX solution from a simple arithmetic average of all hex fluxes contained in the SARIS node:

$$n_k = (1/L) \sum_l \sum_g \phi_{lg} / v \quad (6)$$

where v is the one-group velocity, g is the GRIMHX group index, and there are L hexes in the node. Note that this relates neutron density to the two group fluxes and the one-group velocity thusly: $n = (\phi^1 + \phi^2) / v$. The alternative formulation using two-group velocities: $n = \phi^1 / v^1 + \phi^2 / v^2$ was found to give bizarrely large coupling constants (recall the superscripts are just group indices). Finally, the interface area $A = p \cdot \Delta z / \sqrt{3}$ where Δz is the height of the axial segment. Again, if we are calculating the axial coupling constant, the interface area would be $A = 3p^2 / 2\sqrt{3}$.

CALCULATION OF $\rho_{ex,i}$ NODAL EXCESS REACTIVITY

One other parameter in Eq. 2 is derived from the GRIMHX solution: the nodal excess reactivity in node i , $\rho_{ex,i}$. The reactivity term ρ_i in Eq. 2 is made up of the usual contributions: control and safety rod reactivities, temperature and void coefficient reactivities, fission product reactivity, ink reactivity, boron reactivity, and excess fuel reactivity. With the exception of the last datum, other SARIS models provide the reactivities. The nodal excess reactivity (essentially the reactivity of the node when it is only occupied by cold, clean fuel and D_2O at the base temperature) is defined to be a charge-specific input quantity, defined at each of the 240 SARIS nodes. In the past it has been obtained in an *ad hoc* fashion, tuned in an artful manner to yield desired power distributions and control rod positions. In the new methodology, $\rho_{ex,i}$ will be derived in a rigorous manner from a reference GRIMHX solution, in the same spirit as the coupling constants obtained above.

The nodal excess reactivities will be computed from the steady-state form of Eq. 2:

$$0 = n_i(\rho' + \rho_{ex})_i / \Lambda_i + (1/V_i) \sum_j K_{ji}(n_j - n_i) \quad (7)$$

where the precursor terms are absent because they cancel out in the steady state, and ρ' indicates all reactivity contributions except nodal excess reactivity. We take as our reference GRIMHX solution that for a cold, clean critical state. For that known state, the ρ'_i are defined by knowing where all the control rods are positioned. For the leakage term, the K_{ji} have already been defined by Eq. 4, while the n_i can be calculated from Eq. 6 for this particular GRIMHX state. That leaves $\rho_{ex,i}$ as the only unknown in Eq. 7:

$$\rho_{ex,i} = (\Lambda_i/n_i) * (1/V_i) * \sum_j K_{ji}(n_i - n_j) - \rho'_i \quad (8)$$

The above methodology thus yields a self-consistent SARIS state which should reproduce the critical rod position, the three-dimensional neutron density distribution, and the inter-nodal leakages which were computed in the original detailed GRIMHX calculation. All this will provide a well-defined procedure for the generation of SARIS core model constants that has been absent until now.

THE SCUM CODE

The calculations of Eqs. 4-8 have been coded up in the Simulator Core Update and Modification (SCUM) program, which will be documented in a separate report. SCUM automates what would be just an intimidating book-keeping task since the typical three-dimensional GRIMHX solution contains of the order of 30,000 mesh points which need to be collapsed down to the 240 nodes in the SARIS core model. All the geometrical translations between GRIMHX and SARIS are built into the SCUM algorithm. In addition, SCUM reads the output of GRIMHX directly, thus no user intervention is required. The input to SCUM is simply limited to specifying where the GRIMHX output is to be found, defining the prompt neutron lifetime, and specifying the one-group neutron velocity. The code then generates the K_{ji} and $\rho_{ex,i}$ needed for the solution of Eq. 2 in SARIS.

Experience in testing SCUM has shown that the one loose end is the selection of which GRIMHX cases are to be used to generate the SARIS constants. The key is getting meaningful data for the right hand side of Eq. 4. The problem is that the SARIS nodes are very large compared to the finer nodalization over which the GRIMHX solution is calculated: it is entirely possible for one of the leakage sums $\sum_m J_m A_m$ to be positive

towards the node with the *higher* neutron density, leading to a *negative* K_{jk} for that coupling direction. That arises because for that particular situation, the information content in the *local* neutron densities along the interface is lost in computing the nodal *average* neutron density. The solution is to set up GRIMHX cases in which there are large flux tilts which overwhelm any local structure in the flux, thereby avoiding negative leakage coupling coefficients. In addition, simply having two neighbors with nearly identical average fluxes will result in a near-zero in the denominator of Eq. 4 with the expected effects.

The optimal strategy has turned out to be the use of three different GRIMHX cases to drive SCUM. In the first case, strong radial leakage is excited in the core by varying control rod gang settings radially: 0 vu in Gang 1, 2000 vu in Gang 2, and 4000 vu in Gang 3. This case is used to generate the radial and axial K_{jk} . A second case is used to generate the azimuthal K_{jk} : all three gangs in alternating 60 degree sectors vary sectorwise from 0 vu to 4000 vu back to 0 vu from sector to sector in order to excite azimuthal leakage. SCUM processing of these two cases yields the set of all K_{jk} employed in processing the third case. This third GRIMHX case is used to generate the nodal excess reactivities $\rho_{ex,i}$: cold, clean critical with control rods at their predicted initial critical positions. After these three GRIMHX cases have been run, all the SARIS core model data is generated in a single SCUM run.

The only other external data needed to run SCUM are the prompt neutron lifetime and the one-group neutron velocity, both of which can be obtained from Mark 22 lattice calculations. The one-group neutron velocity can be calculated by running the AVEVEL post-processor against output from a GLASS lattice calculation. The prompt neutron lifetime is already well documented (ref. E.F. Trumble, *et al.*, WSRC-RP-90-199, "Calculation of Prompt Neutron Lifetime via the Alpha Search Method," May 1990).

IMPLEMENTATION OF SECOND GENERATION CORE MODEL IN ULTRASIM

In order to distinguish it from the current SARIS core model, the implementation of Eq. 2 with the revised K_{ji} will be termed the "second generation" core model (CR2). The CR2 algorithm is derived by performing a simple semi-implicit forward differencing on Eq. 2:

$$\begin{aligned} \{n_i(t+\Delta t) - n_i(t)\} / \Delta t = \\ n_i(t+\Delta t) * [\rho(t) - \beta]_i / \Lambda_i + \sum_k \lambda_k C_{ki}(t) + \\ (1/V_i) * \sum_j K_{ji} \{n_j(t+\Delta t) - n_j(t)\} \end{aligned} \quad (9)$$

From which we get the advancement:

$$\begin{aligned} n_i(t+\Delta t) = [n_i(t) + \Delta t * \sum_k \lambda_k C_{ki}(t) + \Delta t * (1/V_i) * \sum_j K_{ji} \{n_j(t+\Delta t)\}] / \\ \{1 - \Delta t * [\rho(t) - \beta]_i / \Lambda_i + \Delta t * (1/V_i) * \sum_j K_{ji}\} \end{aligned} \quad (10)$$

Because $n_j(t+\Delta t)$ occurs on the right hand side of Eq. 10 (recall that the j 's are the neighboring nodes coupled to i), the set of 240 simultaneous equations must be solved through Gauss-Seidel iterations. Typically the SARIS core algorithm uses a fixed number of iterations (30) since a real-time application can not conveniently handle a variable number of iterations.

To make the implementation of the algorithm clearer, we can identify those terms computed from last-pass values and which do not change during the iterations:

$$\begin{aligned} AYY_i &= 1 - \Delta t * [\rho(t) - \beta]_i / \Lambda_i + \Delta t * (1/V_i) * \sum_j K_{ji} \\ CC_i &= n_i(t) + \Delta t * \sum_k \lambda_k C_{ki}(t) \end{aligned}$$

And one term that changes inside the iteration loop:

$$SUM_i = \sum_j K_{ji} \{n_j(t+\Delta t)\}$$

With those identifications, the CR2 algorithm is reduced to the simple form in which it is finally implemented:

$$n_i(t+\Delta t) = \{ \Delta t * SUM_i / V_i + CC_i \} / AYY_i \quad (11)$$

The original core model was designed for execution on an obsolescent vector processor which is being phased out, thus CR2 was written to work efficiently as a strictly scalar application. The kernel of the CR2

algorithm using the derivation of Eqs. 9-11 is given in Table 1 in the form of FORTRAN pseudo-code which will be translated into the mnemonics used in SARIS. This represents a significant simplification since the CR2 algorithm in Table 1 will entirely replace the present CRDS60, CRDS61, CRDS62, CRDS63, CRDS64, CRDS65, and CRDS66 modules, and partially the CRDS70 module. Much of the coding in those old modules was strictly designed for the vector processor application and is unwanted in a scalar application. In addition, the mechanics of setting up the CR2-related part of the SARIS database will be partially automated by creating a new module which need be only executed once. The pseudo-code for that new "one-shot" input module is given in Table 2.

TESTING OF CR2 ALGORITHM IN ULTRASIM

The CR2 algorithm has been implemented inside the ULTRASIM reactor system analysis code (ref. M.V. Gregory, DPST-89-371, "User's Guide to the ULTRASIM Code," March 1989). In fact the pseudo-code in Tables 1 and 2 has simply been taken from that ULTRASIM implementation. Extensive testing has resulted in the conclusion that CR2 is ready for implementation in the Real-Time Modeling Computer (RTMC, our "clone" of SARIS), the final demonstration step before it is migrated to SARIS.

In a consistency check, ULTRASIM/CR2 has been able to reproduce the initial critical condition state (as generated by GRIMHX for the K14.1 charge and processed by SCUM) to within 30 vu in rod position and out to the fourth significant figure in the three-dimensional neutron density distribution. Those CR2 model constants have then been used in another approach to critical by ULTRASIM with a different partial rod complement (weaks only rather than both partials): the critical rod position for this calculation agreed to within 1 vu of the predicted value. An end of cycle at-power state has also been calculated and agreed with qualitative expectations (no independent quantitative results are available for comparison).

In addition, gang rod withdrawal transients were run with ULTRASIM/CR2. Again, no other independent predictions are available (ULTRASIM/CR2 represents a unique, best-estimate computational capability); but the transients were compared against the conservative SAR response and were found to be well inside that envelope. The ULTRASIM/CR2 model was also used in a series of power ascensions to various power levels and for varying fission product states. In general the model was found to be stable and well-behaved. CR2 appears to be ready for implementation in

Table 1. Pseudo-code for CR2 Algorithm

```

C=====
C      2ND GENERATION CORE MODEL IMPLEMENTATION FOR SARIS
C      CR2 CORE SOLUTION ALGORITHM
C      FORTRAN PSEUDO-CODE
C=====
C=====DATABASE
C-----LOCAL ALGORITHM STORAGE
C      DIMENSION AYY(240),CC(240)
C-----PSEUDO-VARIABLES TO BE TRANSLATED TO SARIS MNEMONICS
C      DELTAT - TIMESTEP SIZE, SEC
C      SUMK - NEW ARRAY, NODAL COUPLING COEFFICIENT SUM (INVARIANT)
C      RHONEM - LATEST NODAL REACTIVITY
C      BETA - NODAL DELAYED NEUTRON FRACTION (INVARIANT)
C      PLAMB - NODAL PROMPT NEUTRON LIFETIME (INVARIANT)
C      ENOLD - LAST PASS NODAL NEUTRON DENSITY, N/CM3
C      ENNEW - LATEST NODAL NEUTRON DENSITY, N/CM3
C      SORC - NODAL EXTERNAL SOURCE, N/CM3/SEC (INVARIANT)
C      CLAMB - DELAYED NEUTRON DECAY CONSTANT BY GROUP, 1/SEC (INVARIANT)
C      CNEW - DELAYED NEUTRON CONCENTRATION BY GROUP AND NODE, N/CM3
C      NAYPTR - NEIGHBORS BY NODE (INVARIANT)
C      ZKUPL - COUPLING CONSTANT SET BY NODE (INVARIANT)
C      VOL - VOLUME BY NODE, CM3 (INVARIANT)
C      EFF - BETA/LAMBDA BY GROUP AND NODE (INVARIANT)
C
C      DIMENSION SUMK(240),RHONEM(240),BETA(240),PLAMB(240),ENOLD(240),
C      SORC(240),CLAMB(6),CNEW(240,6),NAYPTR(9,240),VOL(240),
C      ENNEW(240),EFF(240,6),ZKUPL(8,240)
C=====SELF TERM AND INVARIANTS IN THE ITERATION
C      DO NODE=1,240
C      AYY(NODE)=1.0 - DELTAT*(SUMK(NODE) +
C      (RHONEM(NODE)-BETA(NODE))/PLAMB(NODE))
C      CC(NODE)=ENOLD(NODE)+SORC(NODE)*DELTAT
C      DO NG=1,6
C      CC(NODE)=CC(NODE)+CLAMB(NG)*CNEW(NODE,NG)*DELTAT
C      END DO
C      END DO
C=====CARRY OUT ITERATION (30 PASSES)
C      DO ITER=1,30
C-----LOOP OVER AXIAL LEVELS
C      DO MAX=1,10
C-----LOOP OVER GANGS
C      DO NGANG=1,4
C-----LOOP OVER AZIMUTHS TO IDENTIFY PRESENT NODE
C      DO NAZM=1,6
C      NODE=MAX+10*(NAZM+6*(NGANG-1)-1)
C-----COMPUTE SUM OF COUPLING COEFFICIENT TIMES NEIGHBOR'S
C      DENSITY
C      SUM=0.000
C      NCOUNT=NAYPTR(1,NODE)
C      DO K=1,NCOUNT
C      KNAY=NAYPTR(K+1,NODE)
C      SUM=SUM+ZKUPL(K,NODE)*ENOLD(KNAY)
C      END DO
C-----NEW NEUTRON DENSITY
C      ENNEW(NODE)=((SUM*DELTAT/VOL(NODE))+CC(NODE))/AYY(NODE)
C-----UPDATE
C      ENOLD(NODE)=ENNEW(NODE)
C      END DO
C      END DO
C-----END OF LOOP OVER NODES
C      END DO
C-----END OF LOOP OVER ITERATIONS
C=====CALCULATE PRECURSOR CONCENTRATION AT END OF TIMESTEP BASED
C      ON LATEST NEUTRON DENSITY
C      DO NODE=1,240
C      DO NGRP=1,6
C      CNEW(NODE,NGRP)=((CNEW(NODE,NGRP) + DELTAT*
C      EFF(NODE,NGRP)*ENNEW(NODE)) / (1.0+DELTAT*CLAMB(NGRP)))
C      END DO
C      END DO
C      END

```

Table 2. Pseudo-code for CR2 Input Processor

```

C=====
C      2ND GENERATION CORE MODEL IMPLEMENTATION FOR SARIS
C      INPUT SETUP FOR CR2 ALGORITHM
C      FORTRAN PSEUDO-CODE
C=====
C      SET UP FOLLOWING ARRAYS FOR CR2: SUMK, NAYPTR, ZKUPL
C=====
C*****DATABASE
C-----LOCAL ALGORITHM STORAGE
C
C      DIMENSION B(8,240),NAYTYP(8,240),NAYREQ(8,240),NAYGNG(8,240),
C      IBTYPE(8)
C
C-----PSEUDO-VARIABLES TO BE TRANSLATED TO SARIS MNEMONICS
C
C      IDOIT - FLAG TO EXECUTE THIS MODULE (1=YES)
C      ZKAPPA - INPUT COUPLING CONSTANTS (SEE BELOW FOR STRUCTURE)
C      SUMK - NEW ARRAY, NODAL COUPLING COEFFICIENT SUM
C      NAYPTR - NEIGHBORS BY NODE
C      ZKUPL - COUPLING CONSTANT SET BY NODE
C      VOL - VOLUME BY NODE, CM3
C
C      DIMENSION SUMK(240),NAYPTR(9,240),ZKUPL(8,240),ZKAPPA(4,5,4),
C      VOL(240)
C
C-----INPUT COUPLING CONSTANTS ZKAPPA(NTYPE,NREQ,MOANG)
C      WHERE:
C      MOANG=(1,4) BY OANG
C      NREQ=(1,5) BY TOP, UPPER, MIDDLE, LOWER, AND BOTTOM
C      NTYPE=(1,4) BY RADIAL, AZIMUTHAL, CORNERWISE, AND AXIAL
C-----REGION NUMBER FOR EACH OF THE EIGHT B-MATRIX ENTRIES
C
C      DATA IBTYPE/4,4,2,2,1,1,3,3/
C=====RUN THROUGH THIS MODULE ONLY WHEN NEEDED (ONCE FOR SET UP)
C
C      IF(IDOIT.EQ.1) THEN
C
C*****IDENTIFY NODE NEIGHBORS
C
C      NODE=0
C-----LOOP RADially OUTWARD
C
C      DO NRING=1,4
C      DO NAZ=1,6
C-----LOOP AXIALLY DOWNWARD
C
C      DO MAX=1,10
C      NODE=NODE+1
C      AXIAL NEIGHBORS
C      NAXUP=NODE-1
C      IF(NAX.EQ.1) NAXUP=0
C      NAXDN=NODE+1
C      IF(NAX.EQ.10) NAXDN=0
C      AZIMUTHAL NEIGHBORS
C      NAZCH=NODE+10
C      IF(NAZ.EQ.6)
C      NAZCH=NODE-50
C      NAZCCW=NODE-10
C      IF(NAZ.EQ.1)
C      NAZCCW=NODE+50
C      RADIAL NEIGHBORS
C      NRADIN=NODE-60
C      NRADOT=NODE+60
C      CORNER NEIGHBORS
C      IF(NRING.EQ.1) THEN
C      NCOROT=NODE+50
C      IF(NAZ.EQ.1)
C      NCOROT=NODE+110
C      NCORIN=0
C      ELSE IF(NRING.EQ.2) THEN
C      NCOROT=NODE+70
C      IF(NAZ.EQ.6)
C      NCOROT=NODE+10
C      NCORIN=NODE-50
C      IF(NAZ.EQ.6)
C      NCORIN=NODE-110
C      ELSE IF(NRING.EQ.3) THEN
C      NCOROT=NODE+50
C      IF(NAZ.EQ.1)
C      NCOROT=NODE+110
C      NCORIN=NODE-70
C      IF(NAZ.EQ.1)
C      NCORIN=NODE-10
C      ELSE IF(NRING.EQ.4) THEN
C      NCOROT=0
C      NCORIN=NODE-50
C      IF(NAZ.EQ.6)
C      NCORIN=NODE-110
C      END IF
C
C      ZERO OUT NO NEIGHBORS
C      IF(NAXUP.LT.1) NAXUP=0
C      IF(NAXUP.GT.240) NAXUP=0
C      IF(NAXDN.LT.1) NAXDN=0
C      IF(NAXDN.GT.240) NAXDN=0
C      IF(NAZCH.LT.1) NAZCH=0
C      IF(NAZCH.GT.240) NAZCH=0
C      IF(NAZCCW.LT.1) NAZCCW=0
C      IF(NAZCCW.GT.240) NAZCCW=0
C      IF(NRADIN.LT.1) NRADIN=0
C      IF(NRADIN.GT.240) NRADIN=0
C      IF(NRADOT.LT.1) NRADOT=0

```

Table 2 (cont). Pseudo-code for CR2 Input Processor

```

C          IF(NRADOT.GT.240) NRADOT=0
C..... IDENTIFY THE 8 NODE NEIGHBORS
C          B(1,NODE)=NAXUP
C          B(2,NODE)=NAXDN
C          B(3,NODE)=NAZCW
C          B(4,NODE)=NAZCCW
C          B(5,NODE)=NRADIN
C          B(6,NODE)=NRADOT
C          B(7,NODE)=NCORIN
C          B(8,NODE)=NCOROT
C          END DO
C          END DO
C          END DO
C=====SET UP NEIGHBOR POINTING ARRAYS (ALL LOCAL EXCEPT NAYPTR)
C          NAYPTR(9,240) - POINTS TO NEIGHBOR NODE NUMBERS,
C                          INDEX 1=NUMBER OF NEIGHBORS
C                          INDICES 2-9=NEIGHBOR NODE NUMBERS
C          NAYTYP(8,240) - COUPLING TYPE TO NEIGHBOR,
C                          1=RADIAL, 2=AZIMUTHAL, 3=CORNER, 4=AXIAL
C          NAYREG(8,240) - REGION TYPE OF NEIGHBOR,
C                          1=TOP, 2=UPPER, 3=MIDDLE, 4=LOWER, 5=BOTTOM
C          NAYNGO(8,240) - GANG NUMBER OF NEIGHBOR
C-----LOOP OVER AXIAL LEVELS
C          DO NAX=1,10
C-----LOOP OVER GANGS
C          DO NGANG=1,4
C-----LOOP OVER AZIMUTHS TO IDENTIFY PRESENT NODE
C          DO NAZM=1,6
C          NODE=NAX+10*(NAZM+6*(NGANG-1)-1)
C          NCOUNT=0
C-----LOOP OVER NEIGHBORS OF PRESENT NODE
C          DO I=1,8
C          IF(B(I,NODE).NE.0) THEN
C          NCOUNT=NCOUNT+1
C          IDENTIFY NEIGHBOR
C          NAYPTR(NCOUNT+1,NODE)=B(I,NODE)
C          IDENTIFY COUPLING TYPE
C          NAYTYP(NCOUNT,NODE)=I*TYPE(I)
C          IDENTIFY REGION TYPE
C          NREG=3
C          IF(I.EQ.1) THEN
C          IF(NAX.EQ.1) NREG=4
C          IF(NAX.EQ.9) NREG=2
C          IF(NAX.EQ.10) NREG=1
C          ELSE IF(I.EQ.2) THEN
C          IF(NAX.EQ.1) NREG=3
C          IF(NAX.EQ.2) NREG=4
C          IF(NAX.EQ.10) NREG=2
C          ELSE
C          IF(NAX.EQ.1) NREG=5
C          IF(NAX.EQ.10) NREG=1
C          END IF
C          NAYREG(NCOUNT,NODE)=NREG
C          IDENTIFY NEIGHBOR GANG
C          IF(I.EQ.5.OR.I.EQ.7) THEN
C          NGANGN=NGANG-1
C          ELSE
C          NGANGN=NGANG
C          END IF
C          NAYNGO(NCOUNT,NODE)=NGANGN
C          END IF
C          END DO
C          NAYPTR(1,NODE)=NCOUNT
C          END DO
C          END DO
C          END DO
C=====SET UP SUM OF COUPLING COEFFICIENTS AT EACH NODE.
C          THIS EVEN INCLUDES THE COUPLING TO ZERO FLUX BOUNDARIES!!
C          DO NAX=1,10
C          NREGR=5
C          IF(NAX.EQ.1) NREGR=5
C          IF(NAX.EQ.10) NREGR=1
C          DO NGANG=1,4
C          DO NAZM=1,6
C          NODE=NAX+10*(NAZM+6*(NGANG-1)-1)
C          SUMK(NODE)=0.000
C-----LOOP OVER EXISTING NEIGHBORS
C          NC=NAYPTR(1,NODE)
C          DO K=1,NC
C          NTYPE=NAYTYP(K,NODE)
C          NREG=NAYREG(K,NODE)
C          NGANGN=NAYNGO(K,NODE)
C          ZKUPL(K,NODE)=ZKAPPA(NTYPE,NREG,NGANGN)
C          SUMK(NODE)=SUMK(NODE)+ZKUPL(K,NODE)
C          END DO
C-----ACCOUNT FOR ZERO FLUX BOUNDARY CONDITION "NODES"
C          AXIALLY OUT THE BOTTOM
C          IF(NAX.EQ.1)
C          SUMK(NODE)=SUMK(NODE)-ZKAPPA(4,5,NGANG)
C          AXIALLY OUT THE TOP
C          IF(NAX.EQ.10)
C          SUMK(NODE)=SUMK(NODE)-ZKAPPA(4,1,NGANG)
C          RADIALLY OUT THE PERIPHERY
C          IF(NGANG.EQ.4)
C          SUMK(NODE)=SUMK(NODE)-ZKAPPA(1,NREGR,4)
C-----DIVIDE BY VOLUME
C          SUMK(NODE)=SUMK(NODE)/VOL(NODE)
C          END DO
C          END DO
C          END DO
C=====END OF IF TEST FOR RUNNING THROUGH THIS MODULE
C          END IF
C          END

```

the SARIS coding residing on the RTMC.

IMPLEMENTATION OF CR2 IN SARIS

There remains one final complication associated with the SARIS core model: it is actually implemented in terms of a *normalized power density* rather than a *neutron density* (which is what all our derivations up to this point have been based on). In their implementation, Singer-Link made the assumption that the former could simply be directly replaced with the latter in all the equations. As we will show, that is not correct and is the source of the third error in the current SARIS model: we will rigorously transform the equations from neutron density to power density and show that an overlooked term arises.

We start off with the neutron density equation which we repeat for convenience from Eq. 2 to here:

$$dn_i/dt = n_i(\rho - \beta)_i/\Lambda_i + \sum_k \lambda_k C_{ki} + (1/V_i) \sum_j K_{ji}(n_j - n_i) \quad (12)$$

And for the delayed neutron precursors we have:

$$dC_{ki}/dt = n_i \beta_{ki}/\Lambda_i - \lambda_k C_{ki} \quad (13)$$

where β_{ki} = delayed neutron fraction for group k in node i.

Now Eqs. 12 and 13 are not the exact form in which the SARIS neutronics equations are solved. In particular, Eq. 12 is transformed into terms of a normalized power density Q_i simply by replacing $n \rightarrow Q$ everywhere. It is not obvious that this is correct, especially because of the coupling created by the K_{ji} leakage term. A rigorous derivation will show in fact that such a simple transformation is not correct.

To start off, let's convert the neutron density n_i (neutrons/cm³) to the power density in the node q_i (MW/cm³). But first we relate the power to the flux:

$$P_i = \gamma V_i \sum_f f_i \phi_i, \text{ MW in node } i$$

where Σ_{fi} = macroscopic fission cross-section in node i
(1/cm)
 ϕ_i = flux in node i (neutron/cm²/sec)
 γ = conversion factor, (Mev/fission)*(MW-sec/Mev)
or (MW-sec/fission)

Further, we have $\phi_i = n_i v$, where v is the neutron velocity (cm/sec). Thus we have:

$$P_i = \gamma V_i \Sigma_{fi} n_i v$$

Or

$$q_i = P_i / V_i = \gamma \Sigma_{fi} n_i v.$$

Finally, because we know where SARIS is headed, we define the normalized "dimensionless" power density Q_i :

$$Q_i = q_i / P_0 = (\gamma \Sigma_{fi} v / P_0) * n_i$$

where P_0 = some arbitrary normalization power (MW).

For convenience, let's define $T_i = \gamma \Sigma_{fi} v / P_0$ so we have very simply that:

$$Q_i = T_i n_i.$$

Now we go back to Eq. 12, simply multiply through by T_i , use the definition of Q_i , and get:

$$dQ_i/dt = Q_i(\rho - \beta)_i / \Lambda_i + T_i * \sum_k \lambda_k C_{ki} + (1/V_i) * \sum_j K_{ji} (\epsilon_{ij} Q_j - Q_i) \quad (14)$$

where $\epsilon_{ij} = T_i / T_j$.

Note that transformation in the current SARIS model (i.e., $n \rightarrow Q$) holds only if $\epsilon_{ij} = 1$ (or $T_j = T_i$), which is true only if every node is of the same composition.

We continue by making the same T-transformation in the precursor concentration equation:

$$C_{ki} = T_i C_{ki}.$$

We multiply Eq. 13 through by T_i and use the definitions of Q_i and C_{ki} in both Eqs. 13 and 14. We end up with the final, *normalized* forms of fundamental equations for implementation in SARIS:

$$dQ_i/dt = Q_i(\rho - \beta)_i/\Lambda_i + \sum_k \lambda_k C_{ki} + (1/V_i) * \sum_j K_{ji} (Q_j \Sigma_{fi}/\Sigma_{fj} - Q_i) \quad (15)$$

$$dC_{ki}/dt = Q_i \beta_{ki}/\Lambda_i - \lambda_k C_{ki} \quad (16)$$

where Q_i = normalized fission power density in node i
(i.e., $P_0 Q_i$ is fission MW in the node)

C_{ki} = normalized delayed precursor power density in
node i due to delayed group k

We see that there is no change in the form of the delayed precursor equation in the transformation $C \rightarrow C$, but the $n \rightarrow Q$ transformation results in a significantly different leakage coupling term. The erroneous original formulation $\sum_j K_{ji} (Q_j - Q_i)$ is replaced by $(1/V_i) * \sum_j K_{ji} (Q_j \Sigma_{fi}/\Sigma_{fj} - Q_i)$ in the second generation core model.

The latest form of the leakage coupling term necessitates the calculation of one more SARIS datum from the GRIMHX output. The average fission cross-section in SARIS node i is obtained by flux weighting the two-group fission cross-sections in GRIMHX:

$$\Sigma_{fi} = \sum_l \sum_g \phi_l \Sigma_{fgl} / \sum_l \sum_g \phi_l \quad (17)$$

where there are l GRIMHX hexes contained in the i-th SARIS node. This is one other piece of data generated by SCUM and used to modify Eq. 8 and Σ_{mi} in Eq. 11 when they are transformed into the Q variable.

Thus we are left with two options in implementing CR2 in SARIS. Either we can keep the model fundamentally based on power density while making the transformations in the basic equations to be consistent with Eqs. 15 and 16. Or we can keep the form of the original basic equations but now base them on neutron density. Both approaches will be

investigated, and the more efficient strategy will be implemented in SARIS.

DELAYED NEUTRON DATA

Delayed neutron data are handled the same way between the old and new core models. However, the numerical values in that data need to be updated. This became apparent during the testing of the Reactivity Computer in SARIS during development of the K14.1 restart procedures: the delayed neutron parameters in the Reactivity Computer model did not agree with those in SARIS. Examination quickly revealed that the delayed neutron parameters in SARIS have always been those for U^{235} alone, while those in the Reactivity Computer have folded into them the photoneutron characteristics of a D_2O system. Plus SARIS used a prompt neutron lifetime that was different than the currently accepted value for Mark 22 charges (6.5×10^{-4} sec where the accepted value is 1.27×10^{-4} sec). For better transient fidelity, SARIS needs to use the latest Mark 22/D2O set of data.

The latest values are given in Table 3 and are based on collapsing 10 groups down to six (ref. N.P. Baumann, SCS-RPG-91075, "On Methods for Collapsing Delayed Neutron Groups," March 1991). This data will be implemented as part of CR2. It should correct a long standing problem with SARIS predicting a far faster, long-term drop-off in flux after a scram than that observed in the actual plant.

Table 3. Delayed Neutron Data for CR2

__group__	___ β_i ___	___ λ_i ___
1	2.838e-6	4.093e-6
2	2.627e-5	1.776e-4
3	6.170e-5	3.270e-3
4	1.754e-3	2.644e-2
5	4.255e-3	2.293e-1
6	1.001e-3	1.693e-0

$$\beta_{\text{total}} = 0.0071$$

$$\Lambda_{\text{prompt}} = 1.27e-4 \text{ sec}$$

DECAY HEAT MODEL FOR CR2

During the course of developing CR2, it was also noticed that the decay heat model in SARIS under-predicted the at-power decay heat contribution by at least a factor of two and did not show the correct transient response in the first three minutes after a scram. As with the above delayed neutron data change, this issue has nothing to do with the CR2 algorithm as such. But the implementation of the new CR2 model makes for a convenient point at which the decay heat model could be revised.

SARIS now uses a two-group decay heat model. If we let H_{ij} be the MW of decay heat in node j for decay group i :

$$dH_{ij}/dt = Y_i \cdot Q_j - \lambda_i H_{ij} \quad (18)$$

where $i=[1,2]$, $j=[1,240]$, the total decay heat is $H_j=H_{1j}+H_{2j}$, Q_j is the MW fission power in node j , the Y_i are the decay heat yields in the two groups, and the λ_i are the decay constants. Presumably, the two-group values of λ are formulated so that the model can compute two different time responses (i.e., a bimodal response): one that dies off rapidly, plus one that changes only slowly. However, the SARIS model does not achieve that desired response. Closer examination of the model suggested that a minimum of a three-group decay heat model was required so that the total decay heat in node j is $H_j=H_{1j}+H_{2j}+H_{3j}$.

A three-group model has been developed which agrees reasonably with the response given by the standard HMTABLE code used by SRL for LOCA power limit analyses (ref. P.B. Parks *et al.*, DPST-85-828, Addendum 1, "Revised Decay Heat Tables for Mark 16B, Mark 22 and Mark 31 Assemblies for Use in ECS Analyses," June 1987). The three-group model uses the following constants in Eq. 18:

$$\begin{array}{ll} \lambda_1 = 8.0e-2 \text{ sec}^{-1} & Y_1 = 2.184e-3 \\ \lambda_2 = 1.4e-3 \text{ sec}^{-1} & Y_2 = 2.058e-5 \\ \lambda_3 = 8.0e-5 \text{ sec}^{-1} & Y_3 = 8.400e-7 \end{array}$$

The various responses are shown in Figure 3, where decay heat as a fraction of initial operating power is plotted versus time after scram. Note that the correct at-power decay heat fraction is 0.05, which the current decay heat model computes as 0.02. The reference HMTABLE plot

(Curve 1) is for a Mark 22 at a very low exposure (2 days at an assembly power of 0.0015 MW). The three-group model (Curve 2) agrees with the reference data for the at-power state and does a reasonable job in tracking the initially rapid drop in decay heat, which the two-group model (Curve 3) does not see. Long term, beyond 1500 sec, all three curves are in agreement. The data in Figure 3 was generated using the solution of Eq. 18 with $Q=0$ (i.e., post-scrum) as coded in the Appendix.

The three-group model will be implemented along with the CR2 changes by modifying the CRDS20 and CRDS80 modules, and redefining the CRKDECAYHEAT1 and CRKDECAYHEATYLD1 constants.

CR2 MODEL EXTENSIBILITY

One of the benefits of the CR2 formulation is that the model can be easily extended to a true hexagonal geometry representation. The only constraint is the available CPU horsepower. The newly defined K_{ji} coupling constants are already in the correct form for a hexagonal geometry (i.e., six neighbors in the plane). Thus the CR2 algorithm itself would require no changes. The core model could be extended to a cluster-type geometry rather than the nodal geometry in Figure 2 simply by modifying the input processor in Table 2 and increasing the maximum number of nodes. Assuming the computational power were available, this would take the model directly from 24 nodes in the plane to 91 clusters in the plane (i.e., from a 240 to a 910 node model). A host of approximations currently being made would become unnecessary (e.g., fixed power shape interpolation within a multi-cluster node). Obviously, such an extension is beyond the current capabilities of the hardware. Nevertheless, the software will already be in a convenient form to handle such a future update.

Figure 3.

FIG. 3. DECAY HEAT PWR / OPERATING PWR SARIS VS. HMTABLE

CONCLUSION

What began as an exercise to put the constants used in the SARIS core model on a sounder footing has ended up as a detailed verification of the implementation of that model. During that process, four needed changes were identified in the current core model:

1. The leakage coupling term must be modified to correctly account for variations in nodal volume and composition.
2. The effect of hexagonal geometry leakage components must be accounted for.
3. The decay heat model needs to be extended to three groups.
4. The delayed neutron precursor data needs to be modified to incorporate the photoneutron contributions typical of a D₂O reactor.

Plus a methodology for rigorous determination of the model constants has been successfully developed.

APPENDIX. Solution of Post-Scram Decay Heat Equation

```

C=====COMPUTE 3-GROUP DECAY HEAT MODEL RESPONSE
C
C      DIMENSION DHYLD(3),DHDKY(3),ZTIME(1000),ZPOWER(3000),HTIME(30),
C            MFRAC1(30),MFRAC2(30)
C
C-----MHTABLE REFERENCE DATA
C
C.....TIME AFTER SHUTDOWN
DATA HTIME /0.0,1.0,1.5,1.4,1.5,1.6,1.8,2.0,2.2,2.4,2.6,2.8,3.0,
M      5.0,10.0,20.0,30.0,40.0,50.0,60.0,70.0,80.0,90.0,
M      100.0,110.0,500.0,1000.0,2000.0,5000.0,10000.0/
C.....FRACTION OF OPERATING POWER - 3.0E-5 MMDAYS
DATA MFRAC1 /0.50401,0.046191,0.045936,0.045665,0.045402,
M      0.045147,0.044856,0.044619,0.044379,0.044138,
M      0.043897,0.043656,0.043417,0.043178,0.042937,0.042697,
M      0.042456,0.042216,0.041976,0.041736,0.041496,0.041256,
M      0.041016,0.040776,0.040536,0.040296,0.040056,0.039816,
M      0.039576,0.039336,0.039096,0.038856,0.038616,0.038376,
M      0.038136,0.037896,0.037656,0.037416,0.037176,0.036936,
M      0.036696,0.036456,0.036216,0.035976,0.035736,0.035496,
M      0.035256,0.035016,0.034776,0.034536,0.034296,0.034056,
M      0.033816,0.033576,0.033336,0.033096,0.032856,0.032616,
M      0.032376,0.032136,0.031896,0.031656,0.031416,0.031176,
M      0.030936,0.030696,0.030456,0.030216,0.029976,0.029736,
M      0.029496,0.029256,0.029016,0.028776,0.028536,0.028296,
M      0.028056,0.027816,0.027576,0.027336,0.027096,0.026856,
M      0.026616,0.026376,0.026136,0.025896,0.025656,0.025416,
M      0.025176,0.024936,0.024696,0.024456,0.024216,0.023976,
M      0.023736,0.023496,0.023256,0.023016,0.022776,0.022536,
M      0.022296,0.022056,0.021816,0.021576,0.021336,0.021096,
M      0.020856,0.020616,0.020376,0.020136,0.019896,0.019656,
M      0.019416,0.019176,0.018936,0.018696,0.018456,0.018216,
M      0.017976,0.017736,0.017496,0.017256,0.017016,0.016776,
M      0.016536,0.016296,0.016056,0.015816,0.015576,0.015336,
M      0.015096,0.014856,0.014616,0.014376,0.014136,0.013896,
M      0.013656,0.013416,0.013176,0.012936,0.012696,0.012456,
M      0.012216,0.011976,0.011736,0.011496,0.011256,0.011016,
M      0.010776,0.010536,0.010296,0.010056,0.009816,0.009576,
M      0.009336,0.009096,0.008856,0.008616,0.008376,0.008136,
M      0.007896,0.007656,0.007416,0.007176,0.006936,0.006696,
M      0.006456,0.006216,0.005976,0.005736,0.005496,0.005256,
M      0.005016,0.004776,0.004536,0.004296,0.004056,0.003816,
M      0.003576,0.003336,0.003096,0.002856,0.002616,0.002376,
M      0.002136,0.001896,0.001656,0.001416,0.001176,0.000936,
M      0.000696,0.000456,0.000216,0.000000/
C.....FRACTION OF OPERATING POWER - 30 MMDAYS
DATA MFRAC2 /0.051889,0.048277,0.047420,0.047130,0.046887,
M      0.046632,0.046411,0.046191,0.045970,0.045750,0.045529,
M      0.045309,0.045088,0.044868,0.044647,0.044427,0.044206,
M      0.043986,0.043765,0.043545,0.043324,0.043104,0.042883,
M      0.042663,0.042442,0.042222,0.042001,0.041781,0.041560,
M      0.041340,0.041119,0.040899,0.040678,0.040458,0.040237,
M      0.040017,0.039796,0.039576,0.039355,0.039135,0.038914,
M      0.038694,0.038473,0.038253,0.038032,0.037812,0.037591,
M      0.037371,0.037150,0.036930,0.036709,0.036489,0.036268,
M      0.036048,0.035827,0.035607,0.035386,0.035166,0.034945,
M      0.034725,0.034504,0.034284,0.034063,0.033843,0.033622,
M      0.033402,0.033181,0.032961,0.032740,0.032520,0.032299,
M      0.032079,0.031858,0.031638,0.031417,0.031197,0.030976,
M      0.030756,0.030535,0.030315,0.030094,0.029874,0.029653,
M      0.029433,0.029212,0.028992,0.028771,0.028551,0.028330,
M      0.028110,0.027889,0.027669,0.027448,0.027228,0.027007,
M      0.026787,0.026566,0.026346,0.026125,0.025905,0.025684,
M      0.025464,0.025243,0.025023,0.024802,0.024582,0.024361,
M      0.024141,0.023920,0.023699,0.023479,0.023258,0.023038,
M      0.022817,0.022597,0.022376,0.022156,0.021935,0.021715,
M      0.021494,0.021274,0.021053,0.020833,0.020612,0.020392,
M      0.020171,0.019951,0.019730,0.019510,0.019289,0.019069,
M      0.018848,0.018628,0.018407,0.018187,0.017966,0.017746,
M      0.017525,0.017305,0.017084,0.016864,0.016643,0.016423,
M      0.016202,0.015982,0.015761,0.015541,0.015320,0.015100,
M      0.014879,0.014659,0.014438,0.014218,0.013997,0.013777,
M      0.013556,0.013336,0.013115,0.012895,0.012674,0.012454,
M      0.012233,0.012013,0.011792,0.011572,0.011351,0.011131,
M      0.010910,0.010690,0.010469,0.010249,0.010028,0.009808,
M      0.009587,0.009367,0.009146,0.008926,0.008705,0.008485,
M      0.008264,0.008044,0.007823,0.007603,0.007382,0.007162,
M      0.006941,0.006721,0.006500,0.006280,0.006059,0.005839,
M      0.005618,0.005398,0.005177,0.004957,0.004736,0.004516,
M      0.004295,0.004075,0.003854,0.003634,0.003413,0.003193,
M      0.002972,0.002752,0.002531,0.002311,0.002090,0.001870,
M      0.001649,0.001429,0.001208,0.000988,0.000767,0.000547,
M      0.000326,0.000106,0.000000/
C-----ORIGINAL SAR'S DATA
C
DATA DHYLD/ 0.11397E-4, 0.52075E-6, 0.0 /
DATA DHDKY/ 1.0E-3, 0.58686E-4, 1.0 /
C-----TRIAL-AND-ERROR TESTING: 1-RAPID DIE OFF, 2-SLOW
C
DHDKY(1)=8.0E-2
DHDKY(2)=1.4E-3
DHDKY(3)=8.0E-5
DHYLD(1)=1.05*0.026*DHDKY(1)
DHYLD(2)=1.05*0.014*DHDKY(2)
DHYLD(3)=1.05*0.010*DHDKY(3)
C
DELTAT=0.250
TIME=0.0
FRAC1=DHYLD(1)/DHDKY(1)
FRAC2=DHYLD(2)/DHDKY(2)
FRAC3=DHYLD(3)/DHDKY(3)
FRAC1=FRAC1+FRAC2+FRAC3
WRITE(6,10) FRAC1,FRAC2,FRAC3,FRAC1
10 FORMAT(' DECAY HEAT FRACTIONS W.R.T. FISSION: ',/,
M      ' GROUP 1: ',G12.5,/,
M      ' GROUP 2: ',G12.5,/,
M      ' GROUP 3: ',G12.5,/,
M      ' TOTAL: ',G12.5,/)
FRAC1=FRAC1/FRAC1
FRAC2=FRAC2/FRAC1
FRAC3=FRAC3/FRAC1
WRITE(6,20) FRAC1,FRAC2,FRAC3
20 FORMAT(' RELATIVE DECAY HEAT FRACTIONS: ',/,
M      ' GROUP 1: ',G12.5,/,
M      ' GROUP 2: ',G12.5,/,
M      ' GROUP 3: ',G12.5,/)
C-----MARCH FORWARD IN TIME
H1=FRAC1
H2=FRAC2
H3=FRAC3
NEDIT=3
NINC=1
ZTIME(NINC)=TIME
ZPOWER(NINC)=FRAC/(FRAC+1.0)
DO 100 NSTEP=1,20000
NEDIT=NEDIT+1
TIME=TIME+DELTAT
C-----IMPLICIT
C
H1=H1/(1.+DELTAT*DHDKY(1))
H2=H2/(1.+DELTAT*DHDKY(2))
C-----EXPLICIT
H1=H1*(1.-DELTAT*DHDKY(1))
H2=H2*(1.-DELTAT*DHDKY(2))
H3=H3*(1.-DELTAT*DHDKY(3))
C
HT=(H1+H2+H3)*FRAC
HT=HT/(HT+1.0)
H1=H1/FRAC1
H2=H2/FRAC2
H3=H3/FRAC3
IF(NEDIT.LT.25) GO TO 100
NEDIT=0
NINC=NINC+1
ZTIME(NINC)=TIME
ZPOWER(NINC)=HT
CCC WRITE(6,30) NSTEP,TIME,H1,H2,H3,HT
30 FORMAT(' N,TIME,H1,H2,H3,HT>> ',I4,2X,5G12.5)
100 CONTINUE
NTMAX=NINC
C=====CREATE PLOT FILE
C
NFIG=1
WRITE(69,9000) NFIG
9000 FORMAT(' FIG. ',I2,'. DECAY HEAT MODELS')
WRITE(69,9001)

```

APPENDIX (cont). Solution of Post-Scram Decay Heat Equation

```
9001 FORMAT(' TIME, SEC ')
      WRITE(69,9002)
9002 FORMAT(' INPUT DATA.',T73,'12345678')
      C
      NTHMT=30
      WRITE(69,9013) NTHMT
9013 FORMAT(' HMT#1',T73,I3)
      WRITE(69,9004) (HTIME(I),HFRAC1(I),I=1,NTHMT)
9004 FORMAT(3(F10.3,F14.4))
      C
      WRITE(69,9014) NTHMT
9014 FORMAT(' HMT#2',T73,I3)
      WRITE(69,9004) (HTIME(I),HFRAC2(I),I=1,NTHMT)
      C
      WRITE(69,9003) NTMAX
9003 FORMAT(' SARIS',T73,I3)
      WRITE(69,9004) (ZTIME(I),ZPOWER(I),I=1,NTMAX)
      C
      END
//GO.FT69F001 DD DISP=SHR,LABEL=(,,OUT),
//          DSN=T4148.MIKE.DATA(DKAYPLOT)
//
/*
```

**DATE
FILMED**

8 / 16 / 93

END