

AN ABSTRACT OF THE THESIS OF

Marvin A. Lewallen for the degree Master of Science
(Name) (Degree)

in Nuclear Engineering presented on August 30, 1974
(Major department) (Date)

Title: COMPUTER ROUTINES FOR NUCLEAR REACTOR AFTER-HEAT
RESEARCH

Abstract approved: Redacted for privacy
Bernard I. Spinrad

Recent review of nuclear reactor emergency core cooling systems has required renewed investigations into reactor after-heat calculations. Precise energy release rate data are essential in designing and licensing ECCS equipment.

An information system was developed to aid in retrieval of pre-evaluated bibliographic data. Both information processing and report selection are performed by this system.

Three prominent decay-heat computer programs, CINDER; RIBD; and ØRIGEN were investigated and compared. RIBD and CINDER were adopted for use at OSU. The changes in and subsequent problems of each code are outlined here:

1. Correct incompatible FØRTRAN statements
2. Reduce array sizes
3. Update data libraries
4. Adapt input/output routines
5. Adjust to smaller computer word size
6. Verify sample problem results

This thesis represents the writer's participation in

a research group at OSU whose purpose is to investigate
the reactor after-heat problem.

Computer Routines for Nuclear
Reactor After-Heat Research

by

Marvin Adam Lewallen

A THESIS

submitted to

Oregon State University

in partial fulfillment of
the requirements for the
degree of

Master of Science

Completed August 30, 1974

Commencement June 1975

APPROVED:

Redacted for privacy

Professor of Nuclear Engineering

Redacted for privacy

Head of Department of Nuclear Engineering

Redacted for privacy

Dean of Graduate School

Date thesis is presented August 30, 1974

Typed by Suzanne Lewallen for Marvin A. Lewallen

Table of Contents

I. Introduction	1
II. Information System Description	8
III. Summation Theory and Comparison of CINDER, RIBD, and ØRIGEN	16
3.1 Mathematical Basis	16
3.2 Analysis of Heavy Elements and Structural Materials	24
3.3 Library Comparisons	25
3.4 Handling of Roundoff Errors	27
IV. Adaptation of RIBD and CINDER to OS-3	31
V. Summary and Conclusions	37
5.1 Information System	37
5.2 Adaptation of ENDF yields to the CINDER library	37
5.3 Comparative Library Results	38
5.4 Conclusion	39
Bibliography	40
Appendix A	42
Appendix B	47

List of Tables

I. Comparison of RIBD, CINDER, and ØRIGEN	30
II. Data Card Formats	43

List of Figures

<u>Figure</u>	<u>page</u>
I. Input Routine Logic	14
II. Sorting Routine Logic	15
III. ENDF to CINDER Transcription Logic	48

COMPUTER ROUTINES FOR NUCLEAR REACTOR AFTER-HEAT RESEARCH

I. INTRODUCTION

During the past few years certain problems of the nuclear industry have required a new critical look at reactor after-heat calculations. A major incentive for this inquiry has stemmed from power reactor emergency core cooling system (ECCS) evaluation.

ECC systems are designed to remove all decay power (after-heat) from a reactor after an emergency shutdown. This decay power is a function of time after shutdown, and also of reactor operating history and a number of other reactor design parameters. For a given reactor design, and under a specified rating of ECCS power removal capabilities, the precise determination of after-heat following a given level of operation is closely linked to the allowable power rating of the operation. The less precise the relation is, the more conservative the operating limits must be.

Unlike certain other nuclear or reactor parameters, decay power is a strong function of time. In an operating nuclear reactor the amount of fissile material present is also a function of time, but the time constant is best expressed in months or years. Decay power is a function whose time variable is generally expressed in seconds.

In ECCS evaluation, the critical time period is zero to 1000 seconds after shutdown. This small time period

presents an intrinsic difficulty in making precise predictions for ECCS design and evaluation. Decay power predictions are based on many experimentally determined parameters. A small time scale greatly increases uncertainties in these measured parameters.

Historically, nuclear reactor decay power has been estimated in the following ways:

1. By using data derived from direct measurement of released energy in irradiated samples (calorimetric experiments).
2. Starting from nuclear decay principles and fission product data it is valid to sum up the individual fission products β and γ power to arrive at an overall energy release rate.
3. By measuring β and γ activities and spectra as a function of time after a series of fission events. Total energy release rate can then be inferred from the β and γ history.

The first method, calorimeter measurements, has in the past been restricted to cooling times longer than 1000 seconds.⁽¹⁾⁽²⁾ These experiments, therefore, are not much help for direct ECCS applications but offer utility to longer decay time applications. It is worth noting that these calorimeter experiments are very useful in verifying the calculational results of method two.

Method three represents a class of experiments designed

to directly measure β and γ activity after a pulse or an extended history of fissions. Perry, Maienschein, and Vondy⁽³⁾ have recently reviewed this type of experiment with the purpose of evaluating the results as applied to ECCS evaluation and American Nuclear Society (ANS) standards.⁽⁴⁾ Perry's evaluation represents the "state of the art" as far as method three is concerned.

The β and γ experiments are usually performed as a counting type experiment. Typically a U-235 foil is irradiated in a reactor. As quickly as possible after irradiation, the foil is transferred to a sodium iodide crystal for initiation of counting. For short time periods the uncertainties are large and the results are of limited use.

Calculating reactor after-heat by the summation method (number two above) has drawn the most attention within the last five years.⁽⁵⁾⁽⁶⁾ The procedure for solution can be displayed in two separate time domains. The first domain is during reactor operation where fission product buildup is calculated as a function of time. These fission product inventories provide initial conditions for the second domain, decay. The energy release for each individual fission product is calculated, and the summation over all nuclides represents the decay power.

England⁽⁷⁾ has made a statement which expresses the validity of the summation method. He states, "Accuracy (of decay power) is only limited by basic nuclide parame-

ters." Herein lies the essence of this method. The physics of radioactive decay are easily displayed mathematically. The equations synthesized are a set of first order, non-homogenous, linear differential equations. Variations of these same equations have been displayed since the early twentieth century. Bateman⁽⁸⁾, in 1910, solved the set of equations which apply after reactor shutdown. This set of equations is homogenous as compared to the non-homogenous set during reactor operation. The solution to the non-homogenous (reactor operation) set of decay equations is the subject of some current debate.

The point of consideration is the effects of (n, λ) reactions on fission-product nuclides. This reaction represents a coupling of the differential equations across mass chains. When this coupling occurs, the set of equations needed to describe the problem becomes substantially more difficult to solve.

Recently, Shure⁽⁵⁾ has published a paper which includes a discussion and evaluation on the (n, λ) effects. His work includes a point-by-point discussion of England's results and an explanation of discrepancies in much of the (n, λ) considerations. Shure concludes that the effects of (n, λ) reactions increase decay power only at shutdown times greater than 10^7 seconds (~ 0.32 years).

As one starts investigating the specifics of the summation method, the impressive fact becomes the bulk of

material to be dealt with. Presently several computer codes are in existence that will predict reactor after-heat. Each of these codes require a very large input of all pertinent nuclear parameters. These parameters include independent fission yields, half lives, group cross sections, and all power history data. The codes which seem to be most widely used are CINDER⁽⁹⁾, ØRIGEN⁽¹⁰⁾, and RIBD⁽¹¹⁾. These codes, in general, can handle over 800 coupled nuclides for any type of power history and shutdown times.

It must be emphasized that given the same input data, these codes should calculate identical decay power. It also must be reiterated that the final decay power is only as accurate as the nuclear input parameters. Fortunately, due to the large number of pertinent nuclides, individual uncertainties are deemphasized as the entire summation is calculated. This is also fortunate since many of the nuclear parameters, primarily independent yields, must be theoretically predicted.

The evaluation of the nuclear parameters is part of the task that a group in the Nuclear Engineering Department at Oregon State University (OSU) has undertaken. Most of the uncertainty is concerned with and, hence, most of the evaluation will be done on the independent fission yields. Decay schematics will also be reviewed with some attention given to half-lives and β and γ energies.

This thesis represents a display of the work done by

the writer in conjunction with the OSU evaluation. The writer's primary responsibility was to develop computer codes for information processing and to adapt existing summation codes to the OSU computer system, OS-3.

The first area dealt with the development of a literature processing code (LPC). This code will be described in detail later. Briefly, the system was developed to efficiently sort and process any and all bibliographical data used in the OSU evaluation. Processing included keying on different parameters in order to classify and list all reports reviewed. This work was primarily done as support work to the preliminary bibliographical review (chiefly independent yields).

The second segment of this thesis addresses the current decay heat summation programs. The programs, CINDER, RIBD, and ØRIGEN, all have the capability to evaluate nuclear reactor after-heat. These three programs were developed at AEC installations for use on modern large computers. A computer such as a CDC-6600, UNIVAC-1108, or IBM-360/370 series can handle all of them with minor adjustments. This type of machine has core storage for about 80,000 separate pieces of information. The OSU-3300 can handle only 32,000 words. Consequently, to run a summation code, major programming changes had to be implemented. In later chapters general features of RIBD, ØRIGEN, and CINDER will be discussed. Exact changes to CINDER and RIBD will also be outlined.

A preliminary comparison of CINDER, ØRIGEN, and RIBD appears as a section in this thesis. The writer feels that in the process of comparing these three codes, all three should be run with the same input library on a spectrum of problems. Unfortunately, due to limitations of time, computer expenditures, and OSU's computer capabilities, this task has not yet been achieved. (RIBD and CINDER were compared on an abbreviated library. The results of this comparison appear in the summary section.)

II. INFORMATION SYSTEM DESCRIPTION

A major portion of the OSU nuclear reactor after-heat project has been a literature search performed by J. Holm, J. Liaw, and M. Shay. Initially, this search has been a review of all experimental and theoretical papers dealing with fission product yield data. In the near future it may encompass half lives, energetics, etc. To date, over 600 individual papers on isotopic yield data have been uncovered. Reports exist not only on uranium, but also on fissionable isotopes of plutonium and thorium. These reports vary in quality, experimental technique, source, nuclides reported, and date of publication: the dates of publication range from 1945 to the present; thermal and fast fission yields for all fissionable fuels have been published; methods of prediction vary from totally theoretical to mass spectrograph and radiochemical experiments. Primarily because of the number of papers and the wide range of variation, an alternative to hand indexing was deemed necessary.

With the idea of computerized sorting in mind, a number of characteristics for each report had to be devised. The aim has been to devise a set of keys such that a specific report or a group of reports could be chosen from a list of possible reports. With this philosophy in mind, five keys were chosen which could represent a report adequately.

They were:

1. The author's name.
2. The date of publication.
3. Element (~~Z~~ number).
4. Mass (A number).
5. Report status (not reviewed, good, bad, etc.).

With these criteria established the task of writing a computer program for use at OSU was undertaken.

Any combination of the above five keys should serve as input to the program. The output is all reports on the data file which meet the specified criteria.

Efficiency should be incorporated into any computer program, but especially into one like this, which is to be used repeatedly. For example, if the above five keys are all desired, two choices are available for finding the appropriate reports. One alternative is to search the entire data file five times and then compare lists. The other alternative is to sequentially search and reduce the list as the five keys are applied. That is, go through the data file checking for the first key; then, instead of undertaking another complete search, compare only the reduced list generated by the first search for the second key. Subsequent keys are handled in the same way. The former method is easier to program. However, the latter method is much more efficient and, therefore, was used in writing a literature processing code (LPC).

Since LPC was developed as a tool in the bibliographical research, it was designed for use by remote terminal. It was felt that the on line response capability would be very helpful during the review process. Real time operation would be invaluable for cross-checking isotopes and authors, or possible redundancy of reports, etc.

LPC was constructed in three parts. The first part was designed to input the information needed by the sorting routine. All materials needed to run a sort are inserted into the computer's memory by this section of the code. This input material must be in two basic segments, the user-specified parameters and the data-file inputs.

The user-specified parameters consist of the five keys displayed earlier and the input and output options.

The logic of the key input was rather straightforwardly developed. The total number of keys is input (one to five), followed by entrance into a looping routine which allows entering of each specific key (author's name, element, etc.). Each of these keys has a numerical code associated with it. These codes are used in subsequent checking at logic branches in the sorting routines.

The following step is the specific input of the chosen keys. Each key has a section designed to enter the corresponding input. Labels pertaining only to that key are listed on the remote terminal to help clarify the input. For example, if the key on a particular element's mass

number were selected for input, the following label would be written:

DESIRED MASS # =

The desired number would then be entered. Similar input sections exist with appropriate labels for the other four keys.

The remaining LPC input reads and stores the data file. At this phase of the project the data file incorporates only U-235 thermal fission yields. In the future other files will be constructed in the same way as the U-235 file.

Data file preparation and report review must be done concurrently. The information needed in a data file can only be obtained by reviewing the individual reports during review.

Besides the information labeled by the five keys, certain other parameters may also be included in the data file. The extra data may be useful to describe the overall characteristics of the report. The following list of facts may be included in the data file entry for each report:

1. Current status of report*
2. Type of paper (Experimental or Theoretical)
3. Author's name*
4. Date of publication*
5. Remarks (Includes location, ID number, etc.)
6. Number of elements for which yields were cited
7. Number of isotopes for each element

8. Chemical symbol*

9. Mass number

The facts with the asterisk (*) represent sorting criteria while the rest are auxiliary information. As each report is reviewed the above information is entered on IBM sheets and later transferred to data cards.

In LPC the entire data file is read in and stored. In Appendix A the exact format and construction of a data file is outlined and displayed. A FORTRAN listing of LPC is also included.

After the data has been read into LPC and stored in a proper format, the program enters the sorting routine. As described before, the procedure is one of successive searches and up-dates on the initial list until the only reports remaining are those with all the required characteristics. This process may be best explained by use of a flowchart. On the following pages are flowcharts of the logic used in LPC. The first flowchart is a display of the input logic, whereas the second displays the sorting routine logic.

The input logic is self-explanatory. However, the sorting logic needs some explanation since the number of programming choices has increased. The first step is to initialize all counters and set the initial list equal to the entire data file. Next the keys counter is updated and compared to the maximum. If the counter is greater than the maximum, control is transferred to the output

section and the sort is complete. If it is less, the report counter is updated and compared to its maximum. If the counter is greater, control is returned to the "update of the keys" section. If it is less, a computed control change section is entered. Control is then transferred to the appropriate checking section. If the check is positive, the list is updated. If not, control is returned to the "update of reports" section.

As was shown earlier, other logic patterns are available, but the writer felt this particular one appeared efficient and, also, relatively easy to implement and understand. If in the future more efficient or alternative logic is needed the changes to LPC need only be in the sorting section.

Figure I. Input routine logic.

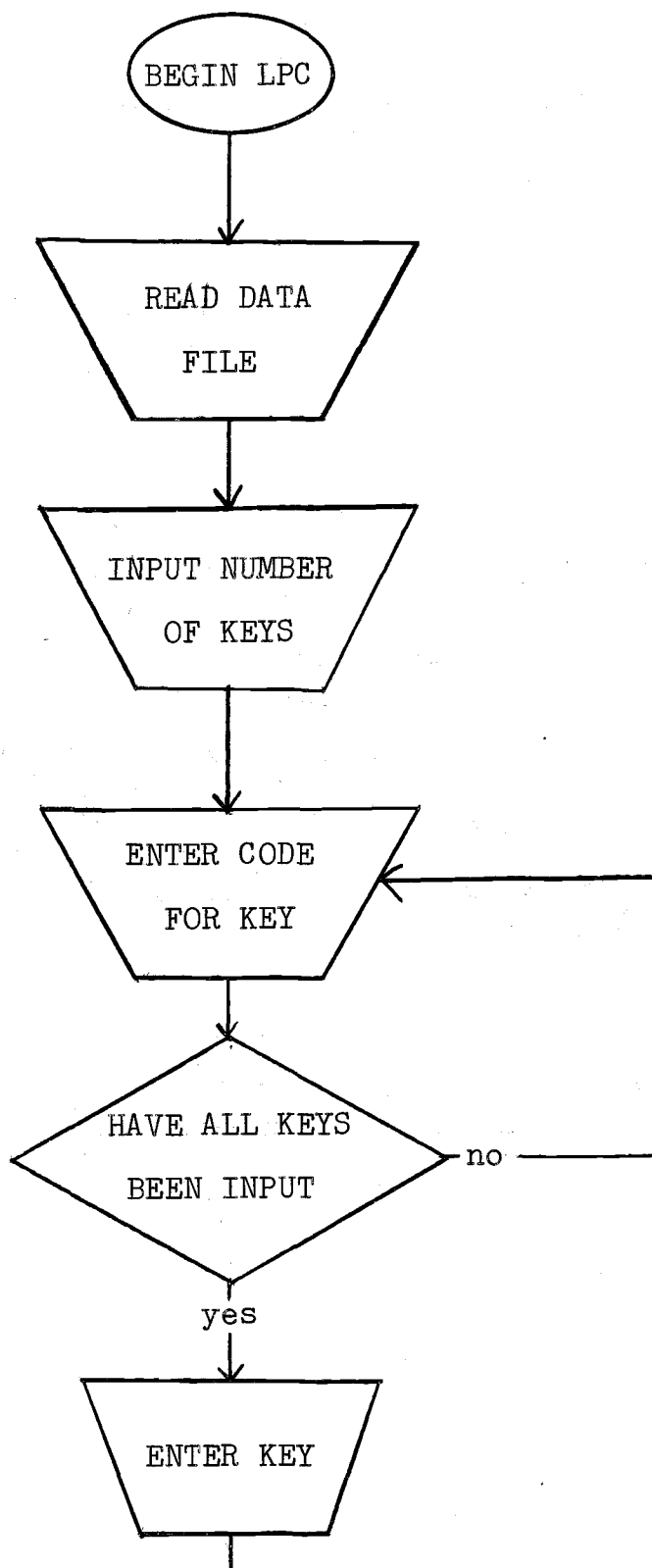
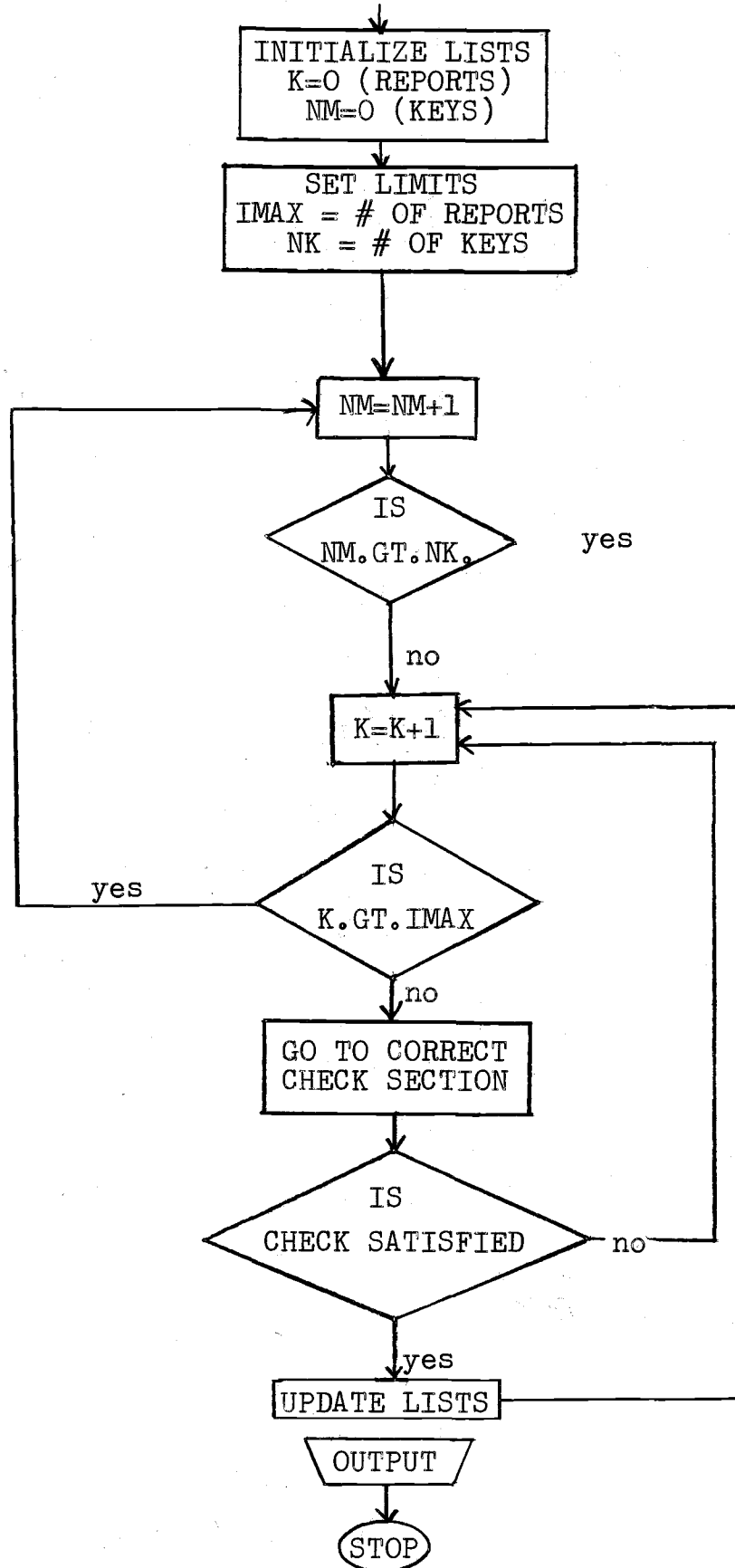


Figure II. Sorting routine logic.



III. SUMMATION THEORY AND COMPARISON OF CINDER, RIBD, AND ØRIGEN

3.1 Mathematical Basis

Many computer programs have been developed for use in fission product analysis. Of these RIBD, CINDER, and ØRIGEN are among those in most current use. Others, such as CRUNCH⁽¹²⁾, NUCY⁽¹³⁾, and ISØGEN⁽¹⁴⁾ have also seen substantial use in the past.

The goal of all fission product codes is the calculation of the fission product inventory as a function of time, both before and after reactor shutdown. After the time dependent inventory has been obtained, various "edits" are performed, the type of edit depending on the user's specific needs. These "edits" include: applications in fuel management (burnup), shielding, health physics, and reactor after-heat predictions.

In any fission product code the governing equations are developed by starting from a fission product continuity equation. The continuity equation for the i, j (i , atomic number, j , mass number index) fission product is:

$$\text{Rate of change in nuclide } i, j = \text{Rate of production of nuclide } i, j - \text{Rate of loss of nuclide } i, j$$

In an operating reactor a fission product can be formed by: 1) direct yield of a fissioning nuclide; 2) decay of a precursor ($i-1$); and 3) absorption of a neutron by the

$j-1$ nuclide. After reactor shutdown, the production includes only precursor decay. The loss rate in an operating reactor includes radioactive decay and neutron absorption. After shutdown radioactive decay is the only loss term. Substituting terms into the above equation, the following formulation is obtained (valid during reactor operation):

$$\frac{\partial N_{i,j}}{\partial t}(t, \vec{r}) = \left(\sum_{k=1}^L \gamma_{i,j,k} \sigma_{f,k} N_k \right) \phi + \lambda_{i-1} N_{i-1,j} + \phi \sigma_{c,i,j-1} N_{i,j-1} - \lambda_i N_{i,j} - \phi \sigma_{c,i,j} N_{i,j} \quad (3.1)$$

- N = number density
- i = atomic number index
- j = mass number index
- k = fissile atom index
- γ = direct fission yield
- t = time
- L = number of fissioning species
- σ_f = microscopic fission cross section (energy averaged)
- ϕ = neutron flux (energy averaged)
- λ = decay constant
- σ_c = microscopic capture cross section (energy averaged)

The above equation, when applied to all possible fission products, represents a set of coupled, first-order, non-homogeneous differential equations. These equations are linear, except for slight variations in ϕ due to the poisoning effect of fission products and burnup. Due to the small variations of ϕ and σ , average values can be used. The space dependence is involved only in the neutron flux. The solution for ϕ is a global reactor physics problem and not strongly dependent on fission product inventories. Therefore, local space and energy averaged values for ϕ can be used with no penalty. After reactor shutdown, the

neutron flux falls to zero and all flux product terms in the above equation drop out.

After shutdown the equations are partitioned into sets of four to ten coupled differential equations. With no $\delta\sigma$ perturbations, the coupling is only along isobaric chains. These isobaric equations were solved by Bateman in 1910. Since the post-shutdown solution is easily duplicated, any code with the same initial conditions and nuclide parameters should calculate identical inventories. In observing the summation codes, this seems to hold true.⁽¹⁵⁾ Discrepancies do not lie in post-shutdown configurations but in the calculation of the initial conditions. In many codes the treatment of the (n,γ) coupling is not clear. Since the exact method of solution varies from code to code, exact (n,γ) effects have not, in my opinion, been clarified.

Historically speaking, CINDER was the first of the major summation codes. It was developed in the early 1960's at Bettis Atomic Power Laboratory and its initial purpose was for depletion and burnup studies. The multi-group neutron energy structure of CINDER makes it particularly attractive for burnup applications. ORIGEN uses internally calculated cross sections, while RIBD uses collapsed one group cross sections, which, although valid, are of limited use in burnup work. The multi-group structure is of primary importance in thermal reactors, where the flux spectrum changes substantially with exposure. CINDER can be

easily adapted to use different cross sections at each time step, making coupling with a thermalization code possible.

England, CINDER's author, displays n coupled nuclides

as:

$$\begin{aligned} \frac{dN_1}{dt} &= \sum_{K=1}^L \gamma_{1K} \phi \sigma_K - (\lambda_1 + \sigma_{1c} \phi) N_1 \\ \frac{dN_2}{dt} &= \sum_{K=1}^L \gamma_{2K} \phi \sigma_K - (\lambda_2 + \sigma_{2c} \phi) N_2 + \gamma_1 N_1 \\ &\vdots \\ &\vdots \\ &\vdots \\ \frac{dN_n}{dt} &= \sum_{K=1}^L \gamma_{nK} \phi \sigma_K - (\lambda_n + \sigma_{nc} \phi) N_n + \gamma_{n-1} N_{n-1} \end{aligned} \quad (3.2)$$

By defining;

$$\begin{aligned} \gamma_i &= \sigma_{i-1} \phi \text{ or } \gamma_i = \lambda_i \text{ (depending on the specific chain)} \\ \beta_i &= s + \lambda_i + \sigma_{ic} \phi \\ C_i &= \frac{\sum_{K=1}^L \gamma_{iK} \phi \sigma_K + N_i(0)}{s} \end{aligned}$$

and s as the Laplace transform variable, the Laplace-transformed equations can be written in matrix form as:

$$\begin{vmatrix} \beta_1 & & & & \\ \gamma_1 & \beta_2 & & & 0 \\ & \gamma_2 & \ddots & & \\ & & \ddots & \ddots & \\ 0 & & & \gamma_{n-1} & \beta_n \end{vmatrix} \begin{vmatrix} N_1 \\ N_2 \\ \vdots \\ \vdots \\ N_n \end{vmatrix} = \begin{vmatrix} C_1 \\ C_2 \\ \vdots \\ \vdots \\ C_n \end{vmatrix}$$

This set of equations is easily solved by using Cramer's rule. The determinant of the matrix is simply $\beta_1 \beta_2 \dots \beta_n$. On replacing the n^{th} column by (C_i) , the determinant is $\gamma_1 \gamma_2 \dots \gamma_{n-1} C_1 + \beta_1 \gamma_2 \gamma_3 \dots \gamma_{n-2} C_2 + \beta_1 \beta_2 \gamma_3 \gamma_4$

... $\delta_{n-1} C_3 + \beta_1 \beta_2 \dots \beta_{n-2} \delta_{n-1} C_{n-1} + \beta_1 \beta_2 + \beta_{n-1} C_n$. Using the values of the two determinants, the solution, expressed in terms of the transformed variable is:

$$N_n(s) = \sum_{m=1}^n \frac{\delta_m \delta_{m+1} \dots \delta_{n-1} C_m}{\beta_m \beta_{m+1} \dots \beta_n}$$

Expanding this into partial fractions and transforming back into the time domain yields:

$$N_n(t) = \sum_{m=1}^n \frac{1}{\delta_n} \prod_{k=m}^n \delta_k \left[\sum_{k=1}^L \delta_k \delta \sigma_{\epsilon k} \left[\frac{1}{\prod_{l=m}^n (\lambda_l - \delta \sigma_{\epsilon l})} \right] \right. \\ \left. - \sum_{j=m}^n \frac{\exp -(\lambda_j + \delta \sigma_{\epsilon j}) t}{(\lambda_j + \sigma_{\epsilon j} \delta) \prod_{l=m}^n (\lambda_l + \sigma_{\epsilon l} \delta - \lambda_j - \sigma_{\epsilon j} \delta)} \right] \\ + N(0) \left(\sum_{j=m}^n \frac{\exp -(\lambda_j + \sigma_{\epsilon j} \delta) t}{\prod_{l=m}^n (\lambda_l + \sigma_{\epsilon l} \delta - \lambda_j - \sigma_{\epsilon j} \delta)} \right) \quad (3.3)$$

Although somewhat complicated to derive, England's display of the fission product solution is entirely valid when used with averaged values of δ and σ over a time step.

The computer code RIBD⁽¹¹⁾ (Radio-Isotope Buildup and Decay) was originally developed for radiological studies. Its "edits" are designed to provide lists of radiologically important elements (noble gases, halogens, etc.).

The fission product continuity equation, in the notation adopted by the author's of RIBD, takes the following form:

$$\frac{dI}{dt} = P_1 Y_{1i} + P_2 Y_{2i} + \sum_{\text{all } j} Y_{ji} A_j - L_i I \quad (3.4)$$

Where:

$$P_1 = P_1^0 e^{-\lambda t}$$

$$P_2 = P - P_1 = P - P_1^0 e^{-\lambda t}$$

- $P_1^0 = P / (1+\delta)$
 δ = initial ratio of first species fissions to second species fissions
 $X = T / \theta$
 P = gross fission rate for time interval
 P_1 = fission rate of first species
 P_2 = fission rate of second species
 P_1^0 = initial fission rate of first species
 T = constant applicable to final fission rate for first species
 θ = time interval length, days
 I = concentration of nuclide, i
 L_i = loss factor for nuclide, $L_i = \phi \sigma_{ai} + \lambda_i$
 σ_{ai} = absorption cross section of nuclide, i
 λ_i = decay constant of nuclide, i
 $\phi = X / \sigma$, $\sigma = \sigma_{a235}$
 Y_{1i} = fission yield (%) of nuclide, i , from first species
 Y_{2i} = fission yield (%) of nuclide, i , from second species
 A_j = concentration of direct precursor, j , of nuclide, i
 Y_{ji} = yield operator on path from precursor, j to nuclide, i . This parameter is either the precursor λ multiplied by the appropriate branching ratio or $\phi \sigma_{aj}$ if (n, δ) coupling is involved.

Although the RIBD general equation describes the same physical system as the CINDER general equation, there are certain specific differences. While the CINDER formulation assumes constant power from each fissioning species over a time step, RIBD expresses individual power as an exponential depletion. This is a more complicated non-homogeneity than that encountered in CINDER's formulation. However, a closed form of the solution to the fission product continuity equation is still obtainable. The solution is closed form in the sense that a series expansion is used, but the expansion represents the exact solution not a numerical approximation. Consequently, the solution is not a numerical

integration over time.

Using techniques similar to those in the CINDER formulation, the solution to the RIBD general equation is displayed as:

$$I = I_0 e^{-L_i \theta} + P Y_2 (1 - e^{-L_i \theta}) / L_i + P_1 (Y_1 - Y_2) (e^{-L_i \theta} - e^{-L_j \theta}) / (L_i - L_j) + \sum_{n=1}^{\infty} \left\{ (-1)^{n-1} F_n(L_i \theta) / (L_i)^n \sum_{\text{all } j} Y_{ji} D^{(n-1)}(A_j) \right\} \quad (3.5)$$

Where

$$F_n(L \theta) = e^{-L \theta} \sum_{m=0}^{\theta} (L \theta)^m / (m!)$$

and $D^{(n)}(A_j)$ is the n^{th} derivative of the concentration of precursor, j , at time θ . The linearity of the set of equations allows the solution to be expressed as a sum over all ancestors. This contribution is represented by the second sum in the solution equation.

The newest summation code, developed at ORNL, is \emptyset RIGEN⁽¹²⁾. It uses a very different method for solving the fission product continuity equation than does RIBD or CINDER. The \emptyset RIGEN expression for production and loss of fission products is:

$$\frac{dX_i}{dt} = \sum_{j=1}^N \rho_{Lj} \lambda_j X_j + \emptyset \sum_{k=1}^N f_{ik} \sigma_k X_k - (\lambda_i + \emptyset \sigma_i) X_i \quad (i = 1 \dots N) \quad (3.6)$$

X_i = atom density of nuclide, i

λ_i = decay constant of nuclide, i

σ_i = spectrum-averaged neutron absorption cross-section of nuclide i

ρ_{Lj} = decay branching ratio

f_{ik} = absorption branching ratio

\emptyset = position- and energy-averaged neutron flux

Bell, the author of \emptyset RIGEN, chooses to express the solution to his general equation as: $\dot{X} = \bar{A} \bar{X}$

(The reader will note that equation 3.6 is homogeneous; therefore not applicable during reactor operation. In fact, this equation is not applicable during shutdown either, as the \emptyset terms should not be present. Bell has displayed this equation to present his method of solution. The reader is referred to papers by Ball⁽¹⁶⁾ and Pease⁽¹⁷⁾ for a complete discussion to the solution method used in \emptyset RIGEN.) The solution to equation (3.6) is displayed as:

$$\bar{X} = \exp(\bar{A}t)\bar{X}(0)$$

where $\bar{X}(0)$ is a vector of initial atom densities and \bar{A} is a transition matrix containing the rate coefficients for decay and neutron capture. The function $\exp(\bar{A}t)$ in the above equation is a matrix exponential function. This is a matrix of dimension N, which is defined to be:

$$\exp(\bar{A}t) = \bar{I} + \bar{A}t + \frac{(\bar{A}t)^2}{2!} + \dots = \sum_{m=0}^{\infty} \frac{(\bar{A}t)^m}{m!} \quad (3.7)$$

The matrix exponential method is, in the writer's opinion, a much more convenient method of displaying the set of fission product continuity equations. The logic and mathematics are straightforward and not prohibitively complex. However, the actual calculation of the exponential matrix for a given physical system is difficult. In the \emptyset RIGEN library 813 isotopes are represented. In calculating an exponential matrix for this library, 813 x 813 or 660,969 words would have to be stored at once in the computer's memory. This amount of storage is prohibitive in all

presently available computers. Finding the appropriate exponential matrix can be done, but it is certainly not straightforward and requires many storage shortcuts and a choice of correct calculational branches.

3.2 Analysis of Heavy Elements and Structural Materials

Although not applicable in fission product studies, energy release and inventories of actinides are important in decay heat studies and health physics. In certain reactor types (chiefly HTGR) actinide heating is particularly significant. This effect is noticed primarily at long shutdown times after extended exposures. Therefore, its importance is in spent fuel handling and safety studies. CINDER does not have the capability to investigate actinide build-up or structural material activation. RIBD has the capability to predict actinide heating but not the prediction of structural material activations. ØRIGEN has extensive library data for all thermal reactor structural materials. Resonance integrals are calculated for epithermal absorption and used in predicting absorption rates. Not only (n,γ) reactions are investigated, but also (n,α) , (n,p) , and $(n,2n)$ reactions. Similar calculations are done for actinides in the ØRIGEN code. Resonance integrals are used for (n,f) reactions with calculated cross sections used for (n,f) , (n,γ) , $(n,2n)$, and $(n,3n)$ reactions. It is apparent that for actinide and structural material

problems ØRIGEN has the best calculational capability.

One interesting point on which to compare RIBD, CINDER, and ØRIGEN is their treatment of the fissioning nuclides. CINDER can handle up to seven heavy nuclides, contributing to reactor power. RIBD, on the other hand, can only calculate two. ØRIGEN has the capability to simulate more than two, depending on the specific reactor type (LMFBR, LWR, HTGR).

In a LWR, especially at the end of a fuel cycle, fissions take place not only in U-235 but also in U-238 and Pu-239 (and, to a slight extent, Pu-241). Consequently, to obtain a realistic solution to the decay heat problem a minimum of three fissioning species must be used. RIBD, although it only has two fissioning species, is considered adequate for fast reactors, since U-238 and Pu-239 are usually considered sufficient to represent the problem.

3.3 Library Comparisons

One major difference between RIBD and CINDER is in the way the data libraries are compiled. (ØRIGEN is not considered here, since the ØRIGEN library has not yet been obtained at OSU.) The CINDER library, like England's general solution, is set up to calculate one coupled set of fission products at a time. For example, a linear chain, starting with Ge-75M is treated as follows:



This chain is treated in CINDER as an entity by itself. All parameters, such as decay constants, cross sections, etc., are input in consecutive chain order. Nuclide parameters are input in the order in which they appear in the chain. Any branching in the chains is taken into account by formation of two separate chains whenever branching takes place. The result is that, although CINDER only treats 93 mass chains, 204 "pseudo" chains are utilized to model the problem. (The extra chains are due to (n, γ) effects and chain branching.) Similarly in the OSU CINDER library, 1157 nuclides are calculated but only 738 of these nuclides are non-redundant. The CINDER library contains yield data on U-235, U-238, Pu-239, Th-232, and U-233 fissions. Decay energies are included, but only take the form of average β and γ energies per disintegration. As discussed earlier in the chapter, all cross sections are in the four group form with library values being thermal reactor representative ones. All data is recorded in 80 character card image form.

The CINDER library is formed in three distinct parts. Part one contains decay constants, coupling parameters, and absorption cross sections. Part two lists fission product yields for U-235, U-238, Pu-239, Th-232, and U-233. All yields are given in fraction per fission, rather than percent as reported in some libraries. The last part of the library contains average β and γ energies in MeV per disintegration.

Since the RIBD code is a "grid processor", nuclide data is not input in any specific chain order. All nuclide parameters are input at one entry in the library. The library is ordered in increasing mass and atomic number. During calculation, RIBD internally sets up all chains.

3.4 Handling of Roundoff Errors

In all three codes, RIBD; CINDER; and ØRIGEN, the loss of significant figures during calculation is a serious problem. Each code handles the potential problem in a different way. RIBD attempts to handle the problem by expressing the solution to the fission product continuity equation in three different forms, depending on the magnitude of the arguments of the exponential functions needed. For small arguments (<4), the exponential product solution is expanded into series with precancellation of almost equal terms. For large arguments (>79), a similar series is employed to avoid machine overflow. Intermediate arguments are handled by the standard solution displayed earlier (Eqn. 3.5). Checks are used in RIBD to verify that the general form of the solution has converged in the ancestor calculation. If convergence has not been reached the calculation is redone. If the calculation fails again, error diagnostics are printed and the ancestor calculation is bypassed.

In my opinion, the CINDER code has a more detailed

roundoff checking routine than RIBD. In evaluating England's fission product solution (Eqn. 3.3), two specific types of errors are checked for.

One type of error is the lack of significance in calculating contributions to a nuclide from possible precursors. For long chains and very small time steps, the contribution to a nuclide at the end of the chain from precursors at the beginning of the chain can be negligible. CINDER checks for this lack of significance and, if it exists, the calculation of precursor contributions is neglected. This routine not only checks the loss of significance in coupling, but also reduces computer running time since it obviates insignificant calculations.

The second type of error is more serious than the first. It may occur in exponential terms divided by a term subtracting almost equal parameters. CINDER checks for this problem and, if it exists, the calculation is terminated. If this type of error was not eliminated serious underflow problems could develop.

The ØRIGEN code has not been run at OSU. However, in investigating reference literature, ØRIGEN employs the same sort of roundoff checking that RIBD does. That is, anticipating where problems may exist, it routes calculations to an alternative path. Certain subroutines and segments of the calculation are done in double precision arithmetic to avoid any loss of significant figures.

Table I, on the following page, serves as a display of comparisons of CINDER, RIBD, and ØRIGEN.

TABLE I. COMPARISON OF RIBD, CINDER, AND ØRIGEN

<u>BASIS</u>	<u>CINDER</u>	<u>RIBD</u>	<u>ØRIGEN</u>
Date of initial development	1962-1965	1966-1968	-1973
Developer	T.R. England	R.O. Gumprecht	M.J. Bell
Place of origin	Bettis, U of Wisconsin	Douglas United, Battelle NW	Oak Ridge NL
No. of γ energy groups	4	1	12
No. of neutron groups	4	1	1
No. of fissile nuclides contributing to power	7	2	3
No. of library yield sets	5	14	6
Actinide heating?	No	Yes	Yes
Structural material heating?	No	No	Yes
Initial purpose	Depletion	Fission product inventories	Depletion
Remarks	Initially developed for depletion. Later adapted for decay heat.	Initially used for radiological studies. Historically used primarily for fast reactors.	Newest code. Used for actinide and structural material analysis.

IV. ADAPTATION OF RIBD AND CINDER TO OS-3

The adaptation of RIBD and CINDER to the Oregon State University computer system (OS-3) included the following six types of operations:

1. Rewriting incompatible FØRTRAN statements.
2. Reducing array sizes to allow running on a smaller core computer.
3. Changing data libraries to allow use of arrays of smaller dimensions.
4. Adapting input/output routines to utilize smaller data libraries.
5. Correcting any problems of smaller computer word size.
6. Verifying sample problem results.

Operation one was carried out in adapting both CINDER and RIBD. In CINDER, incompatible FØRTRAN was encountered in the program declaration statement, "logical if" statements, common block statements, and the punched output routine. In each case the OS-3 form for each statement type was substituted for the incompatible form. In RIBD, incompatible FØRTRAN was found in the program declaration statement, data statement formatting, the use of logical operators and double precision, input/output statements, and format statements. In most cases OS-3 FØRTRAN equivalents were used. However, for double precision and certain

input/output sections, direct equivalents did not exist and other alternatives had to be developed.

The data library input section of RIBD was unusual. In the original form, RIBD read "card image" data, then converted this data to binary, and finally reread the binary. This type of input/output is not available on OS-3, so a new program input was developed. Fortunately, all data input is handled in a single subroutine. This subroutine was replaced with one which read all data in the "card image" form, with subsequent data transfer to the main program. Part of the original RIBD input was in the "NAME-LIST" format. This format does not exist on OS-3. Therefore it was replaced with an equivalent data file.

The original version of RIBD utilized double precision variables in some calculations. On OS-3, if any variable in a routine is declared double precision all variables must be double precision. Rather than declare all variables in all RIBD double precision calculations as double precision (which would severely limit storage), all double precision was removed. The consequences of double precision removal became obvious only when the sample problem was calculated. The loss of significant figures is apparent when the OS-3 sample problem results are compared with the original results.

The next task in adapting RIBD and CINDER to OS-3 was to reduce array sizes to "fit" the program on the CDC-3300.

The CDC-3300 has a memory capacity of 32,000 words. To run a program on OS-3, the program, all data, and any library functions (Exp, Sin, Alog, etc.) must fit into the 32,000 word memory at the same time.

In the process of reducing array size, the "REAL 1" function was used in both CINDER and RIBD. This function is used to expand the CDC-3300 memory. In declaring real dimensioned variables as "REAL 1", the way in which the array is stored in the computer's memory is altered. Part of the value is stored in the upper memory bank and the rest in the lower bank. (The upper bank is usually used for auxiliary storage.) When a particular value is needed, the lower and upper portions must be matched and joined to form the needed value. Ordinarily, the entire value is stored in lower memory and no matching of upper and lower memory is needed. The disadvantage in the "REAL 1" declaration is that the memory matching process adds significantly to the program running time.

Originally the CINDER dimensions totaled 37,700. By using the "REAL 1" function and reducing the dimensions of the variables used to store nuclide parameters, the program could fit on the CDC-3300 with total dimensions of 25,000. RIBD was originally dimensioned for 35,800. To fit on OS-3 this figure was reduced to 19,300. The array size could be increased somewhat by cutting out superfluous FØRTRAN statements and reducing unused array sizes. With increased use

the advantage of increased array size may promote this work.

Since the array sizes of CINDER and RIBD were reduced, the size and method of input of their respective data libraries had to be altered. The CINDER code is especially amenable to this problem. The chain structure of the CINDER library can be split up into several parts with no intrinsic loss of accuracy. Since these chains are "pseudo" chains, they are not coupled to each other. The solution, therefore does not depend on the order or the starting and stopping points of the data library.

England, while at the University of Wisconsin, had to split his data file into three separate sections. In essence, this is comparable to running three separate problems and adding the results together to obtain the overall answer. At OSU, five different sections were needed to run the CINDER sample problem.

In the RIBD code the data library was split into two distinct parts. The RIBD code is a "grid processor" since it internally sets up the chain structure. Therefore, the separation of the library includes the possibility of error. Fortunately, the library was split in the center of the fission yield "valley". The error in the lack of cross coupling is negligible (especially for thermal reactors). Globally, the effect of the decay heat summation cannot be seen.

After operations one through four (described above), were completed, RIBD and CINDER were successfully compiled.

The sample problems were then ready to be calculated. When the sample problems were run for the first time, the problem of finite computer word length appeared in both CINDER and RIBD. The problem came from terms of the following form:

$$\frac{x_i - y_i}{x_i}$$

When the term $(x-y)/x$ is less than 6×10^{-11} , any divisions by the quantity $(x-y)$ will result in an overflow. Any subsequent calculations with $(x-y)$ are meaningless.

On the CYBER computer installation (at Richland, WA) where the RIBD code was obtained, the standard word is longer, and therefore the roundoff error is much less than on OS-3 ($\sim 10^{-14}$). Therefore, calculations with the RIBD data library on the CYBER installation may be significant, while the same calculation on OS-3 may yield erroneous results. Unfortunately, this seems to have been the case.

In the RIBD code, overflow occurred in the sample problem (the RIBD sample problem was a .05 second irradiation at 4000 MW). The loss of significant figures occurred in the ancestor summation calculation for particular nuclides. After much investigation, the section of the code which was causing the problem was isolated. Checks were inserted to truncate terms in which the loss of significance had occurred. The OS-3 RIBD results then matched sample problem results out to three to five significant figures for each nuclide.

The roundoff problem in CINDER was much less severe than in the RIBD code. Whereas the RIBD "blowup" occurred in the buildup phase, the CINDER problem developed in the decay portion. After investigation of the decay calculation portion of the CINDER code, the exponential cut-off criterion was varied. Originally, any power of "e" less than -1200 was internally set to zero. This limit was changed to -200 , which truncated an increased number of insignificant calculations. After this change the OS-3 CINDER sample problem results agreed perfectly with the supplied results.

V. SUMMARY AND CONCLUSIONS

5.1 Information System

A computerized information system has been devised and is in operation for retrieving fission yield information.

5.2 Adaptation of ENDF Yields to the CINDER Library

One special task performed by the writer for the decay heat research was the adaptation of ENDF (Evaluated Nuclear Data File) direct fission yields into the CINDER library. This work included transcribing ENDF yields (ordered by increasing mass and atomic number) into the CINDER library's chain structure. Three computer programs, CVERT; ENCINA; and ENCINB, were written to perform the adaptation in two distinct phases. The first task (using CVERT) was to re-establish the ENDF data in the CINDER FORTRAN format. The second phase was to write a computerized dictionary to transform CINDER labels (alpha-numeric) into ENDF labels (integers). ENCINA and ENCINB were written to: 1) transcribe a CINDER label into the ENDF form, 2) find the appropriate entry in the ENDF file, and 3) place the ENDF direct yield values into the CINDER library.

The only irregularity came in replacing CINDER nuclides which were the start of a chain. This was handled by summing all possible precursor yields in the ENDF file and using this value in the CINDER library.

In Appendix B, the logic pattern of the ENDF-to-CINDER transcription is displayed. FORTRAN listings of CVERT, ENCINA, and ENCINB are also included.

5.3 Comparative Library Results

After the sample problems were verified, RIBD and CINDER were run with an identical abbreviated library. This comparison revealed that, although RIBD gave correct answers for its sample problem, a realistic reactor problem caused RIBD to yield invalid results. Unfortunately, the problem was created by the roundoff checks which were inserted into RIBD to truncate underflowed calculations. In the RIBD sample problem, the irradiation time was too short (.05 second) to saturate any fission products. Therefore, when precursor contributions were truncated (due to overflow), the effect was negligible. However, in the CINDER comparison case, the irradiation was much longer (2 hours). Many nuclides saturated in this time period, so the truncation of precursor terms involved significant error.

The CINDER sample problem contains an extended irradiation (10,000 hours). Due to CINDER's handling of precursor contributions and the longer time scale, CINDER will yield valid results down to small time steps of a few minutes (a very short time, as in the RIBD sample problem, cannot be evaluated). CINDER, from a roundoff problem standpoint, yields correct values and can be used for parameter review and nuclear reactor after-heat predictions.

5.4 Conclusion

After adaptation of the summation codes, RIBD and CINDER, to the OSU computer system, OS-3, sample problem results were verified for both. The RIBD code displayed more roundoff problems than did CINDER, with sample problem discrepancies displaying this fact. In the final account, the CINDER code represents a useable tool in the OSU after-heat project. Unfortunately, the RIBD code has, at this time (August, 1974), incurable roundoff problems on the OSU computer and will not give any viable results to a practical reactor problem. Since it would be useful to validate that RIBD and CINDER do, indeed give equivalent results for the same input, their comparison should be pursued on another, larger computer. Assuming this validation to be correct, it is recommended that CINDER, for reasons cited, be used as the primary computation tool in OSU studies.

BIBLIOGRAPHY

1. R.A. Day and C.J. Cannon, Direct Calorimetric Study of Fission-Product Decay in Active Slugs, P/41, Radiochemical Studies: The Fission Products, National Nuclear Energy Series, Div. IV, Vol. 9, p. 422; McGraw-Hill, New York (1951). (Includes also results of other early experiments by Borst and Wheeler, Lane and Brady, Turkevich and Brady.)
2. K. Johnston, A Calorimetric Determination of Fission-Product Heating in Fast Reactor Plutonium Fuel, J. Nucl. Energy, 19, 527 (1965).
3. A.M. Perry, F.C. Maienschein, and D.R. Vondy, Fission-product Afterheat—A Review of Experiments Pertinent to the Thermal-Neutron Fission of U-235, ORNL-TM-4197, Oak Ridge National Laboratory (1973).
4. American Nuclear Society, Proposed ANS Standard Decay Energy Release Rates Following Shutdown of Uranium-Fueled Thermal Reactors, Draft Standard ANS 5.1, approved by subcommittee ANS-5 of the ANS Standards Committee, October 1971.
5. K. Shure, ²³⁵U Fission-Product Decay Energy—1972 Reevaluation, WAPD-TM-1119 (October 1972).
6. Testimony, Exhibit 1113, AEC Hearing on LWR ECCS Acceptance Criteria, (1972).
7. T.R. England, An Investigation of Fission-Product Behavior and Decay Heating in Nuclear Reactors, PhD Thesis, University of Wisconsin (1969).
8. H. Bateman, The Solution of a System of Differential Equations Occurring in the Theory of Radio-Active Transformations, Proc. Cambridge Phil. Soc., 16, 423 (1910).
9. T.R. England, CINDER—A One-Point Depletion and Fission-Product Program, WAPD-TM-334 (Rev.), (1964).
10. M.J. Bell, ØRIGEN—The ORNL Isotope Generation and Depletion Code, ORNL-4628, Oak Ridge National Laboratory (May 1973).
11. R.O. Gumprecht, Computer Code RIBD, DUN-SA-94, Douglas United Nuclear (May 1969).

12. M.P. Leitzke and H.C. Claiborn, CRUNCH-An IBM-704 Code for Calculating N Successive First Order Reactions, ORNL-2958, Oak Ridge National Laboratory (October 1960).
13. D.R. Vondy, Development of a Explicit Solution to the Nuclide Chain Equations for Digital Machine Calculations, ORNL-TM-361 (October 1962).
14. H.H. Van Tuyl, ISØGEN-A Computer Code for Radioisotope Generation Calculations, HW-83785 (1964).
15. R.E. Schenter, Personal Communication, Hanford Engineering and Development Laboratory, Richland, Washington, (July 1974).
16. S.J. Ball and R.K. Adams, MATEXP, a General Purpose Digital Computer Program for Solving Ordinary Differential Equations by the Matrix Exponential Method, ORNL-TM-1933, Oak Ridge National Laboratory (August 1967).
17. L. Pease, DEEMS, A FØRTRAN Program for Solving the First Degree Coupled Differential Equations by Expansion in Matrix Series, TDSJ-49 (October 1963).

APPENDICES

APPENDIX A

Table II, on the following page, is a display of the data card formats for LPC. The "A" type card is a header card, present for each report. The "B" type card contains all the isotopic data for one particular element.

A data file will consist of one type "A" card for each report followed by a number of type "B" card specified on column 45 of the "A" card. The type "B" cards must be in the order of reference on the "A" card (column 47 and on).

At the end of this appendix is a FORTRAN listing of LPC.

TABLE II. DATA CARD FORMATS

Type A

<u>COLUMN</u>	<u>DEFINITION</u>
1	Blank.
2	* unreviewed, + good, - bad, etc.
3	Blank.
4	E experimental, T theoretical.
5	Blank.
6-13	Author's name.
14	Blank.
15-18	Date of publication (4 numbers).
19	Blank.
20-43	ID code and remarks.
44	Blank.
45	Number of elements cited (1-10).
46	Blank.
47	Number of isotopes in element 1 (1-5).
48	Blank.
49	Number of isotopes in element 2 (1-5).
.	
.	
.	
.	

Type B (one per element)

<u>COLUMN</u>	<u>DEFINITION</u>
1-6	Blank.
7-8	Elements 2-letter chemical symbol.
9	Blank.
10-12	Isotopes mass number.
13	Blank for ground state, "M" for meta stable state.
14	Blank.
15-17	Isotopes mass number.
18	Blank for ground state, "M" for meta stable state.
.	
.	
.	
.	

```

PROGRAM LPC
DIMENSION KEY (5),KBT(175),IDATE(175),
1XDR(175,3),IELM(175),NISØ(175,5),A(175,10,5),LIST(175)
2,IS(175),Q(175,10,5)
REAL NAME, IAUT
INTER A, DLM
C READ ALL DATA *****
I=0
101 I=I+1
READ(30,550)KBT(I),IS(I),IAUT(I),IDATE(I),XDR(I,1)
1,XDR(I,2),XDR(I,3),ITEMP
2,(NISØ(I,L),L=1,ITEMP)
IF(EØF(30))GØ TO 200
IELM(I)=ITEMP
DØ 110 L=1,ITEMP
MEMP=NISØ(I,L)+1
READ(30,511)A(I,L,1),(A(I,L,K),Q(I,L,K),K=2,MEMP)
110 CØNTINUE
GØ TO 101
550 FØRMAT(1X,A1,1X,A1,A8,1X,3A8,I2,8(1X,I1))
511 FØRMAT(6X,A2,8(1X,I3,A1))
200 IMA=LAL=I-1
WRITE(61,506)
506 FØRMAT(' KEY=1, AUTHOR'/' KEY=2, DATE'/' KEY=3,ELEME
1NT'/' KEY=4, MASS #'/' KEY=5, STATUS')
99 NK=TTYIN(4H# OF,4H KEY,4HS = )
IMAX=LAL
DØ 20 I=1,NK
20 KEY(I)=TTYIN(4HKEY ,4H = )
I=1
30 CONTINUE
GØ TØ (50,60,70,80,90,),KEY(I)
50 CØNTINUE
WRITE(61,508)
508 FØRMAT(' ENTER FIRST 8 DIGITS ØF AUTHØRS NAME'/' )
READ(60,509)NAME
509 FØRMAT(A8)
GØ TØ 95
60 CØNTINUE
ISTRD=TTYIN(4HSTAR,4HTING,4H DAT,4HE = )
ISTØD=TTYIN(4HSTØP,4HING ,4HDATE,4H = )
GØ TØ 95
70 CØNTINUE
WRITE(61,71)
71 FØRMAT(' ENTER DESIRED ELEMENTS 2 LETTER CHEMICAL'
1' SYMBOL '/')
READ(60,72)ELM
72 FØRMAT(A2)
GO TO 95
80 CØNTINUE

```

```

      MASS=TTYIN(4HDESI,4HRED ,4HMASS,4H # ,4H= )
      WRITE(61,81)
81  FØRMAT(' ENTER BLANK FØR GRND STATE, M FOR METAST. ')
      READ(60,92)B
      GØ TØ 95
90  CØNTINUE
      WRITE(61,91)
91  FØRMAT(' * NØT REVIEWED, + GØØD, - BAD   ENTER CODE' /)
      READ(60,92)KCØD
92  FØRMAT(A1)
C   END INPUT *****
95  I=I+1
      IF(I.GT.NK)GØ TØ 100
      GØ TØ 30
C   INITIAL LISTS *****
100 CØNTINUE
      DØ 21 I=1,IMAX $ 21   LIST(I)=I
      N=0
220 N=N+1
      IF(K.GT.IMAX)GØ TØ 300
      MTEMP=LIST(K)
C   START COMPARISONS *****
      GØ TØ (1,2,3,4,5),KEY(N)
1   IF(IAUT(MTEMP).EQ.NAME)GØ TØ 153
      GØ TØ 151
2   IF(IDATE(MTEMP).GT.ISTOD)GØ TØ 151
      IF(IDATE(MTEMP).LT.ISTRD)GØ TØ 151
      GØ TØ 153
3   ITEMP=IELM(MTEMP)
      DØ 973 L=1,ITEMP
      IF(A(MTEMP,L,1).EQ.ELM)GØ TØ 153
973 CØNTINUE
      GØ TØ 151
4   ITEMP=IELM(MTEMP)
      DØ 971 L=1,ITEMP
      KTEMP=NISØ(MTEMP,L)+1
      DØ 971 LL=2,KTEMP
      IF(A(MTEMP,L,LL).EQ.MASS.AND.Q(MTEMP,L,LL).EQ.B)
1  GØ TØ 153
971 CØNTINUE
      GØ TØ 151
5   IF(KBT(MTEMP).EQ.KCØD)GØ TØ 153
      GØ TØ 151
153 NM=NM+1 $ CØDE=1.
      LIST(NM)=LIST(K)
      KMAX=NM
      GØ TØ 151
300 CØNTINUE
C   END OUTER LØØP *****
C   ØUTPUT
      IMAX=KMAX

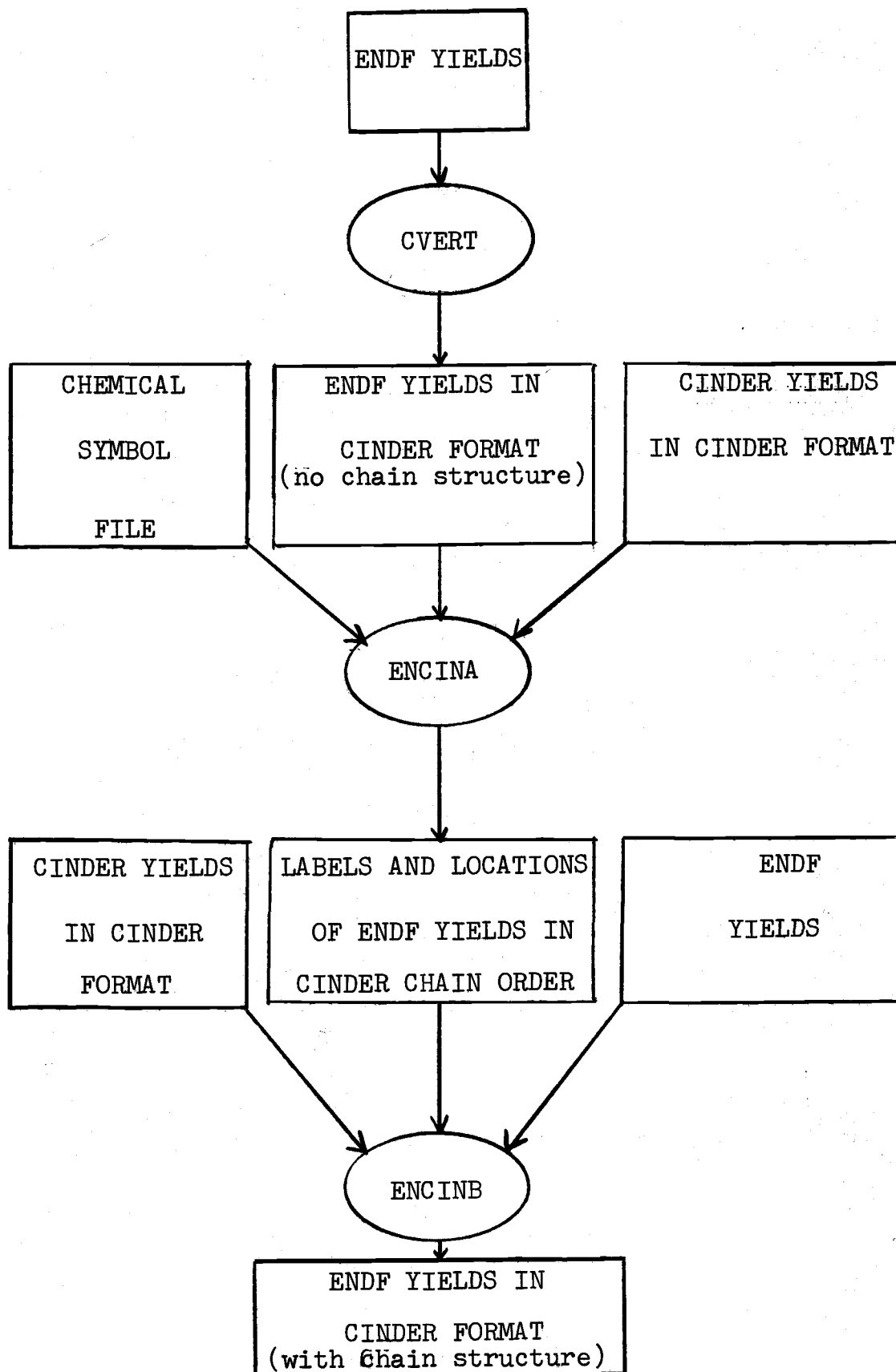
```

```
GØ TØ 220
400 IF(CØDE.EQ.0.)GØ TØ 460
WRITE(61,401)
401 FØRMAT(////30(1H*),2X,' SØRT RESULTS',2X,30(1H*))//)
DØ 450 L=1,IMAX
IT=LIST(L)
ITT=IELM(IT)
450 WRITE(61,445)KBT(IT),IS(IT),IAUT(IT),IDATE(IT)
1,(XDR(IT,K),K=1,3),(A(IT,K,1),K=1,ITT)
445 FØRMAT(3X,A1,2X,A1,A8,2X,I4,2X,3A8,8(3X,A2))
GØ TØ 510
460 WRITE(61,461)
461 FØRMAT(' NØ REFERENCE FØUND'//)
CØNTINUE
510 WRITE(61,630)
630 FØRMAT(////' FØR ANØTHER SØRT, LET C=1.')
C=TTYIN(4HC= )
IF(C.EQ.1)GØ TØ 99
END
```

APPENDIX B

On the following pages are found a display of the ENDF-to-CINDER yield transcription logic and a FORTRAN listing of all computer programs used.

Figure III. ENDF to CINDER transcription logic.




```

PROGRAM CVERT
REAL A,B
DIMENSION A(5,3,260),B(5,3,260)
MNL=0
99 I=0
MNL=,ML+1
1 I=I+1
DO 10 J=1,255
READ(60,5)(A(I,K,J),K=1,3),(B(I,K,J),K=1,3)
5 FORMAT(6E11.4)
10 CONTINUE
IF(I.EQ.5)GO TO 7
GO TO 1
7 J=0
8 J=J+1
IF(A(1,2,J).EQ.1.)KP=1HM
KP=1H
WRITE(30,100)A(1,1,J),KP,(A(L,3,J),L=1,5)
IF(B(1,2,J).EQ.1.)KP=1HM
KP=1H
WRITE(30,100)B(1,1,J),KP,(B(L,3,J),L=1,5)
IF(J.EQ.255)GO TO 101
GO TO 8
100 FORMAT(F5.0,A1,5(E12.8))
101 IF(MNL.GT.1)GO TO 234
GO TO 99
234 CONTINUE
END

```

```

PROGRAM ENCINA
DIMENSION LAB(1200,5),KAL(75),KEN(3,1200),IN(1200,2)
KP=1
DO 10 I=1,1157
READ(1,100)LAB(I,1),LAB(I,3)
100 FORMAT(A1,4X,A1)
10 CONTINUE
REWIND 1
READ(2,101)(KAL(I),I=26,70)
101 FORMAT(24(1X,A2))
DO 30 L=1,1157
IF(LAB(L,1).EQ.1H*)GO TO 11
IF(LAB(L,1).EQ.1H/)GO TO 12
IF(LAB(L,1).EQ.1H=)GO TO 12
IF(LAB(L,1).EQ.1H-)GO TO 12
IF(LAB(L,1).EQ.1H')GO TO 12
IF(LAB(L,1).EQ.1H )GO TO 12
LAB(L,1)=0
GO TO 15
11 LAB(L,1)=2
GO TO 15
12 LAB(L,1)=1
15 IF(LAB(L,3).EQ.1HM)GO TO 13
LAB(L,3)=1H
LAB(L,4)=4
GO TO 30
13 LAB(L,4)=1
30 CONTINUE
DO 50 L=1,1157
J=LAB(L,4)+LAB(L,1)
GO TO(61,62,62,63,64,64),J
61 READ(1,91)LAB(L,2),LAB(L,5)
91 FORMAT(A2,1X,I2)
GO TO 49
63 READ(1,92)LAB(L,2),LAB(L,5)
93 FORMAT(A2,2X,I2)
GO TO 49
64 READ(1,94)LAB(L,2),LAB(L,5)
94 FORMAT(1X,A2,1X,I2)
49 IF(LAB(L,5).LT.72)LAB(L,4)=LAB(L,5)+100
50 CONTINUE
DO 80 L=1,1157
DO 70 J=26,70
IF(LAB(L,2).EQ.XAL(J))GO TO 76
70 CONTINUE
GO TO 85
80 CONTINUE
GO TO 86
85 WRITE(61,79)L,LAB(L,2)
79 FORMAT('ERROR STOP AT LABEL',14,2X,'READ ',A2)
GO TO 500
86 DO 110 l=1,1019

```

```

110 READ(3,111)KEN(1,I),KEN(2,I),KEN(3,I)
111 FØRMAT(I2,I3,A1)
    KMIN=1 $ KMAX=500
    DØ 200 L=1,1157
    DØ 195 j=KMIN,KMAX
    IF(LAB(L,2).EQ.KEN(1,J).AND.LAB(L,5).EQ.KEN(2,J))GØ
1GØ TØ 196
    GØ TØ 195
196 IF(LAB(L,3).EQ.2)IN(L,2)=1
    GØ TØ 195
197 IN(L,1)=J
    IF(LAB(L,1).EQ.2)IN(L,2)=1
    GØ TØ 199
195 CØNTINUE
    WRITE(61,198)LAB(L,2),LAB(L,5),LAB(L,3),L
198 FØRMAT(' ERRØR STØP ',I2,I3,A1,' CINDER LABEL',1X,I4)
    GØ TØ 200
199 KP=-1*KP
    IF(KP.LT.0)GØ TØ 217
    GØ TØ 216
217 KMIN=KMIN+1
216 KMAX=KMAX+1
    IF(KMAX.GT.1019)KMAX=1019
200 CØNTINUE
    STØP=0 $ K=0
210 L=K+1
    K=L+10
    IF(K.GT.1157)GØ TØ 201
    GØ TØ 202
201 K=1157
    STØP=1.
202 WRITE(10,203)(IN(I,1),LAB(I,1),I=L,K)
    IF(STØP.EQ.1)GØ TØ 500
    GØ TØ 210
500 CØNTINUE
203 FØRMAT(11(I4,1X,I1))
    END

```

```

PROGRAM ENCINB
REAL C,E
DIMENSION IN(1200,2),C(7,1200),IEN(3,1100),E(1100,5)
DO 10 I=1,1157
10 READ(1,2)(C(J,I),J=1,7)
   2 FORMAT(A6,5(E12.8),6X,A3)
   STOP=0 $ K=0
210 L=K+1
   K=1+10
   IF(K.GT.1157)GO TO 201
   GO TO 202
201 K=1157
   STOP=1
202 READ(2,203)(IN(I,1),IN(I,2),I=L,K)
   IF(STOP.EQ.1.)GO TO 500
   GO TO 210
203 FORMAT(I1(I4,1X,I1))
500 DO 107 I=1,1019
107 READ(3,4)IEN(1,I),IEN(2,I),IEN(3,I),(E(I,J),J=1,5)
   4 FORMAT(I2,I3,A1,5(E12.8))
   DO 155 I=1,1157
   II=IN(I,1)
   T=C(2,I)+C(3,I)+C(4,I)+C(5,I)+C(6,I)
   IF(T.EQ.0.)GO TO 155
   IF(IN(I,2).EQ.TO) GO TO 140
   DO 5 J=2,6
   5 C(J,I)=E(II,J-1)
   GO TO 155
140 DO 6 J=2,6
   6 C(J,I)=E(II,J-1)
   DO 145 J=1,250
   IF(II-J)155,155,141
141 L=II-J
   IF(IEN(2,L).EQ.IEN(2,II))GO TO 142
   GO TO 145
142 DO 143 K=2,6
143 C(K,I)=C(K,I)+E(L,K-1)
145 CONTINUE
155 CONTINUE
   DO 300 I=1,1157
   KT=IN(I,1)
300 WRITE(30,301)(C(K,I),K=1,7),IEN(1,KT),IEN(2,KT),
1 IEN(3,KT)
301 FORMAT(A6,5(E12.8),6X,A3,1X,I2,I3,A1)
   END

```