WHC-EP-0520

CONUP: A Code that Calculates Tag Gas Concentrations for Reactor Components

Prepared for the U.S. Department of Energy
Assistant Secretary for Nuclear Energy

Westinghouse
Hanford Company Richland, Washington

Hanford Operations and Engineering Contractor for the
U.S. Department of Energy under Contract DE-AC06-87RL10930

DISCLAIMER

Available to the U.S. Department of Energy
and its contractors from
Office of Scientific and Technical Information
P.O. Box 62
Oak Ridge, TN 37831
(615) 576-8401

Available to the public from the U.S. Department of Commerce
National Technical Information Service
5285 Port Royal Road
Springfield, VA 22161
(703) 487-4650

Printed in the United States of America

DISCLM-1.CHP (1-91)

WHC-EP--0520

DE92 006520

WHC-EP-0520

UC-520

CONUP: A Code that Calculates Tag Gas Concentrations for Reactor Components

D. J. Hammervold
E. R. Siciliano
F. Schmittroth
R. E. Schenter

Date Published
November 1991

Prepared for the U.S. Department of Energy
Assistant Secretary for Nuclear Energy

Westinghouse Hanford Company
P.O. Box 1970
Richland, Washington 99352

Hanford Operations and Engineering Contractor for the
U.S. Department of Energy under Contract DE-AC06-87RL10930

CONUP: A CODE THAT CALCULATES TAG GAS CONCENTRATIONS FOR REACTOR COMPONENTS

D. J. Hammervold
E. R. Siciliano
F. A. Schmittroth
R. E. Schenter

ABSTRACT

The CONUP code calculates the current tag isotopic concentrations for the Tag Gas Analysis Code, SMTAG. The combined codes, CONUP and SMTAG, represent the Tag Gas System. CONUP produces tag concentrations that are decayed and transmuted over specific reactor core cycles. The calculated concentrations are used, together with measured concentrations, as input for the SMTAG code, which identifies the failed reactor components that have released tag gas. The CONUP code has two modes for calculating isotopic concentrations: absolute and incremental. In the absolute mode, the CONUP code calculates concentrations from the beginning of the reactor startup through the current cycle. In the incremental mode, the CONUP code processes concentrations from the last reactor component cycle for each component. The incremental mode saves significant processing time because the concentrations are updated only for the current cycle. A description of the underlying physical model and method of solution are presented. A description of the code and a user's guide are also given, along with example input and corresponding concentration output.

This page intentionally left blank.

CONTENTS

CONTENTS (continued)

LIST OF FIGURES

LIST OF TABLES

This page intentionally left blank.

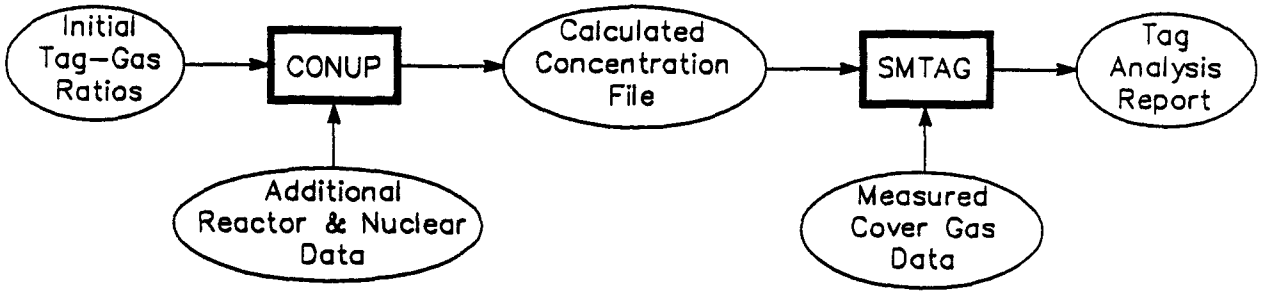
CONUP: A CODE THAT CALCULATES TAG GAS CONCENTRATIONS FOR REACTOR COMPONENTS

1.0 INTRODUCTION

The CONUP (CONcentration UPdate) code is one part of a two-part computer code system (CONUP/SMTAG) used to identify breached components in the Fast Flux Test Facility reactor. This identification is possible for those reactor components that are tagged during fabrication with a specific amount and blend of inert gas isotopes, which is referred to as tag gas. When a tagged reactor component breaches, it releases a small amount of tag gas to the reactor cover gas. The cover gas is then analyzed with a mass spectrometer to determine the measured set of isotopic ratios that uniquely define the tag gas. These measured values can differ from any of the initial values associated with the tagged components because the isotopic amounts (which, for historical reasons, are also referred to as concentrations) change as a result of irradiation. Therefore, the current isotopic concentrations in each of the tagged reactor components must be calculated, and the CONUP code performs that calculation. These calculated values are then compared with the measured values using the SMTAG code to identify the specific breached component. [For additional background information, see the companion document SMTAG and other references therein (Schmittroth 1989).]

Figure 1 is a diagram that indicates the input and output flow of the CONUP/SMTAG code system. Besides the initial tag gas ratios (introduced into the tagged components during fabrication), CONUP requires additional information so that the current isotopic concentrations can be calculated. This information consists of reactor cycle history data, neutron flux data, assembly configuration data, and nuclear cross-section and yield data.

Figure 1. The CONUP Input/Output Flow Diagram.



The CONUP code can calculate the current concentrations either incrementally from prior concentration files, or absolutely from the first reactor cycle. The advantage of incremental processing is in the overall reduction in computer processing time required to create the current cycle concentrations. The option to process over all prior cycles provides the capability to update input data parameters such as cross sections as data analysis capabilities improve.

The principal tag gas system for the Fast Flux Test Facility consists of a set of three isotopic ratios: one xenon ratio, $^{126}Xe/^{129}Xe$; two krypton ratios, $^{78}Kr/^{80}Kr$; and $^{82}Kr/^{80}Kr$. Each tag gas is represented by a point in this three-dimensional tag-ratio space. In this simple picture, the tag nearest the point representing the measured ratios identifies the failed component. In practice, two other classes of xenon and krypton isotopes are considered.

First, all stable xenon and krypton isotopes are naturally included when tag gases are manufactured; and although most tags are uniquely identified by the three principal ratios, valuable information can be obtained by considering other ratios. For example, ^{124}Xe burns out rapidly in an irradiation environment, and its concentration is therefore a measure of the burnup history of the breached component. Second, several stable isotopes of both xenon and krypton are created as fission products. The amount of these isotopes provides further valuable information on the nature of the release.

Another consideration involves tags used to identify capsules in the Materials Open Test Assembly (MOTA). These tags depend primarily on ratios of fission product xenon and krypton isotopes to eliminate possible conflicts with tag releases from driver fuel.

The complete tag gas system tracks a total of 18 xenon and krypton isotopes. These isotopes are listed in Table 1, along with three other bromine and iodine fission product isotopes that produce xenon or krypton by decay and irradiation. Thus, the CONUP code not only performs transmutation/decay calculations, but also records the evolution of 21 isotopes.

Table 1. Xenon and Krypton Isotopes Tracked in CONUP.

^{124}Xe	^{126}Xe	^{127}I	^{128}Xe	^{129}Xe
^{129}I	^{130}Xe	^{131}Xe	^{132}Xe	^{133}Xe
^{134}Xe	^{136}Xe	^{78}Kr	^{80}Kr	^{81}Kr
^{81}Br	^{82}Kr	^{83}Kr	^{84}Kr	^{85}Kr
^{86}Kr				

The following sections of this report attempt to (1) describe the underlying physical model upon which CONUP is based and (2) provide a user's guide for running CONUP. The discussion on the underlying physical model is given in Section 2.0, where a formal description of the equations used in computing the isotopic concentrations for each irradiation cycle from the initial tag gas ratio values is presented. Section 3.0 describes the structure of the CONUP code, and Section 4.0 gives a guide to its usage. If a person is already familiar with the underlying physics or desires an immediate use of CONUP, Section 3.1 and then Section 4.0 should be read. The remainder of this report can then be used as a reference, according to the needs of the reader.

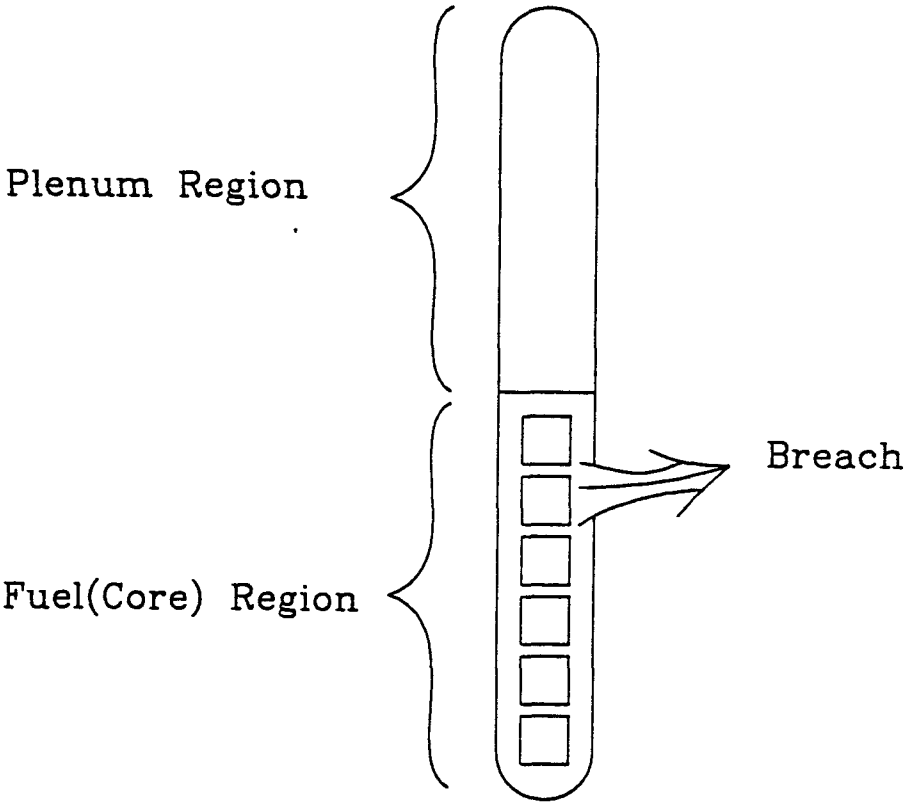
2.0 ANALYSIS OF METHODS

The purpose of this section is to present the underlying physics upon which the CONUP code is based. Section 2.1 is a general description of the model. The detailed physics equations are then given in Sections 2.2 through 2.4.

2.1 TAG GAS SYSTEM MODEL: GENERAL PHYSICS CONSIDERATIONS

An understanding of the CONUP/SMTAG code system requires a model of the gas production, transmutation, and release from a reactor component. The model used is based on the physical situation present in a reactor fuel pin at the time of a breach. This situation is shown schematically in Figure 2, in which there is a fission-gas plenum region and a fuel-pellet core region.

Figure 2. Schematic Gas Release.



At the beginning of the fuel cycle, the gases present in the pin are the initial tag gases, which are introduced by the insertion of a tag gas capsule during fabrication. This capsule contains a specific amount and blend of inert gas isotopes that is usually unique to each assembly of fuel pins. The content of the tag gas capsule is released into the plenum by puncturing the capsule at the time of final fabrication.

During reactor operation, the plenum acts as a reservoir for those gaseous fission products produced from irradiation of the core. However, during normal operation, some of the fission product gases are retained within the fuel grains and are not free to escape into the plenum. Also as a result of irradiation, the initial tag isotopes present in the plenum are transmuted.

To interpret the gas released from a fuel pin, it is convenient to identify two classes of isotopes according to their origin. The first class consists of tag gas isotopes that were in the originally manufactured tags, as well as the daughter isotopes arising from their transmutation. The second class of isotopes, the fission product isotopes, arise from the fission events in the fuel pin. Likewise, their daughters are also included in this class. The distinction between tag gas and fission product isotopes is maintained because the transmutation and transport of isotopes that originate in the fuel to the breach location will differ from isotopes that originated in the gas plenum.

The output isotopic concentration file created by the CONUP code maintains this distinction and thus has a block of fission product isotopes and a block of tag gas isotopes for each tag on the file. Again, this classification is based on the origin of the isotopes, and the same isotopes can appear in both blocks. In practice, both blocks include the complete list of isotopes.

The distinction between tag gas isotopes and fission product isotopes is maintained throughout the CONUP code. Thus, the production and subsequent burnup of these isotopes are treated separately for these two classes. For MOTA tags that do not normally contain fissionable material, the fission product (FP) block of isotopes is set to zero.

Although a tag gas usually is unique to each assembly, there are occasions when components are multiply tagged or different components have the same tag. To accommodate these cases, CONUP can support reactor components tagged with multiple tag gases as well as duplicate tags within different components. The initial concentrations are derived from the tag gas ratios provided by chemical analysis performed for each tag. Reactor core maps identify each assembly's position, and tag identifiers are associated with each assembly. The CONUP code maintains a database of assembly configurations for each irradiation cycle.

The isotopic amounts are calculated on a per-component (pin) basis. The results are saved on the concentration file (see Appendix B), where the calculated amounts for each isotope in the tag gas (TG) and FP blocks are listed for each assembly. The SMTAG code compares these values with a measured sample to determine the failed reactor component associated with the released tag gases [see Schmittroth (1989)].

In the following paragraphs, the detailed analysis associated with the previous general discussion is given.

2.2 INITIAL TAG GAS AMOUNTS

Let the symbol $n_i(t)$ denote the number of nuclei of type i present at time t. Compute the initial (t=0) values from the expression

$$n_i(0) = A_t\left(\frac{P_\alpha}{100}\right)\left(\frac{N_o}{22.4x10^3}\right)\left(\frac{r_i}{\sum_{j\epsilon S_\alpha} r_j}\right) \quad , \qquad (1)$$

where the index i denotes the specific isotope of xenon or krypton under consideration, and the label α is set equal to xenon or krypton, depending on whether i is from the set of xenon or krypton isotopes. The summation index j runs over all the isotopes in the set S_α. The remaining symbols in Equation 1 have the following definitions:

A_t = total amount of gas (in cubic centimeters) at standard temperature and pressure

P_α = percent of xenon or krypton

N_o = Avogadro's number

r_i = ratio of the amount of isotope to the amount of base isotope.

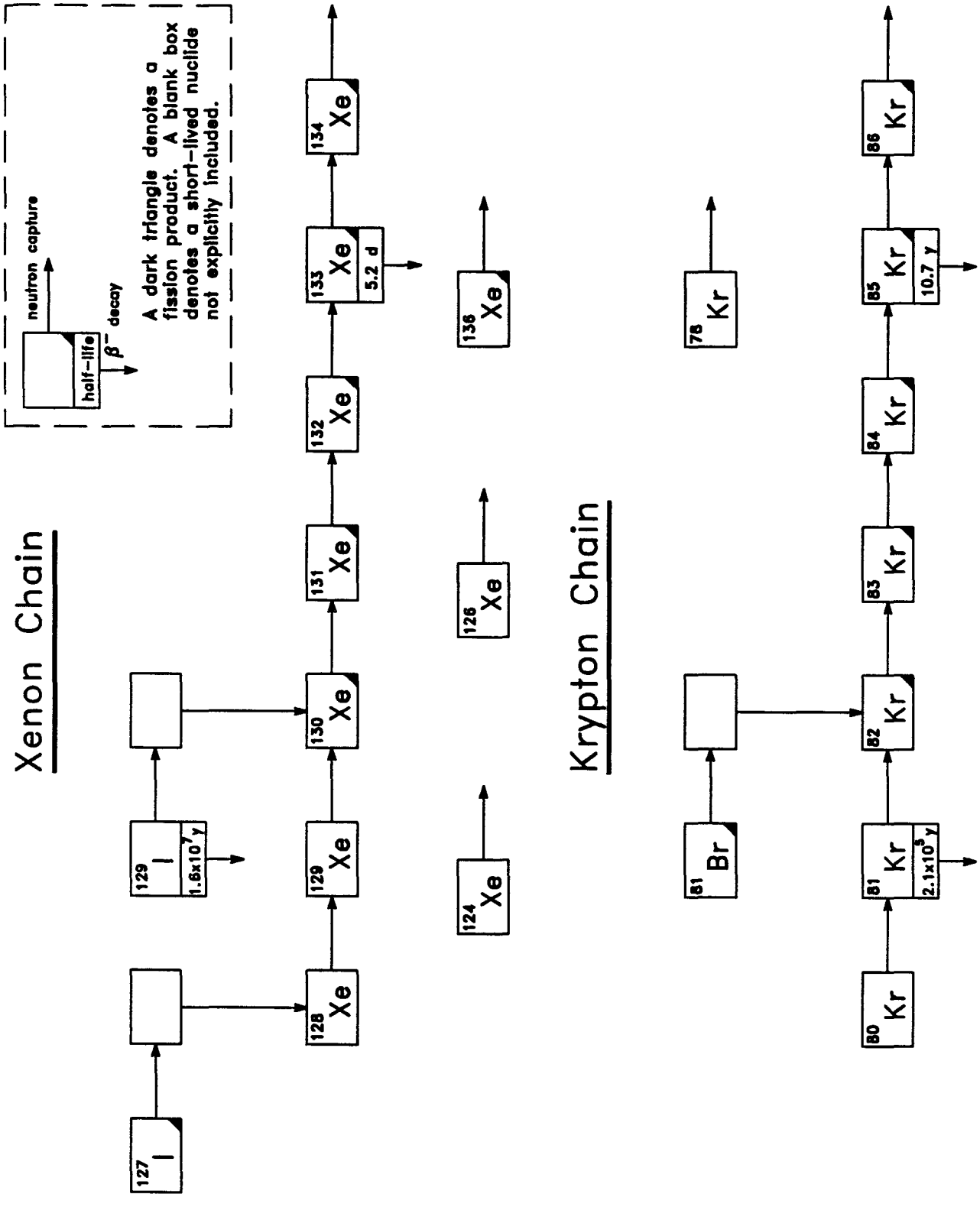
The base isotopes are ^{129}Xe and ^{84}Kr for xenon and krypton, respectively. (The values for A_t, P_α, and r_i are stored in the in the tag ratios file. See Section 4.3.6 for a description of this file.)

As previously mentioned, CONUP maintains a distinction between the fission product isotopes and the tag gas isotopes that are in the original tag composition even though some isotopes reside in both categories. This allows one to make different assumptions about the migration and burnup behavior of the two sources. The equations for these two classes are in the following sections.

2.3 BURNUP EQUATIONS FOR ISOTOPES IN THE TAG GAS BLOCK

This section describes the decay and transmutation (burnup) of the isotopes in the TG block of CONUP. Section 2.4 discusses the FP block. The pathways for production, decay, and transmutation of all 21 isotopes included in the CONUP system are illustrated in Figure 3. Transmutation induced by neutron capture is shown by a horizontal arrow. A vertical arrow with a box for the half-life identifies those isotopes that β^- decay. All other isotopes are stable. Blank boxes denote intermediate isotopes that are assumed to

Figure 3. Xenon/Krypton Isotope Chains.



decay instantaneously and are not included. The CONUP code allows at most a single daughter isotope for each parent isotope, whether that isotope was reached via transmutation or decay. Thus, for example, while both the decay and transmutation of ^{129}I are properly included, the very small amount of ^{129}Ze produced by the decay is neglected.

Isotopes that are produced as fission products are indicated in Figure 3 by a dark triangle in the lower left corner. This production process is excluded from the TG block and is discussed in the following section. For the TG block of isotopes, the source-free equation is

$$\frac{dn_i}{dt} = \sum_{j<i} \alpha_{j \to i} n_j - \beta_i n_i \quad , \tag{2}$$

where $\alpha_{j \to i}$ denotes the transition rate from parent isotope j to daughter isotope i, and β_i is the destruction rate for isotope i. The isotopic indices are ordered such that any given isotope is always transmuted to a single (daughter) isotope with a higher index. (For each parent isotope j, the daughter index i is given in the code by the array idghtr[j] that defines the transmutation chain.)

The transition rates are given by

$$\alpha_{j \to i} = \phi_l(I_\phi)\, \sigma_{c,j}(I_\sigma) \quad , \tag{3}$$

where $\phi_l(I_\phi)$ is the one-group flux at core location l, and $\sigma_{c,j}(I_\sigma)$ is the effective one-group capture cross section for isotope j. The flux and cross section are further qualified by a flux-type identifier I_ϕ = (core, plenum, mota) and a cross-section type identifier I_σ = (core, plenum, mota). This scheme allows one to establish different sets of neutron fluxes and cross sections appropriate to different reactor environments. It also provides flexibility in the code to accommodate unanticipated conditions by simply adding new flux types.

The destruction rates β_i are formed from the natural decay rate added to the sum of all transmutation rates to daughter isotopes:

$$\beta_i = \frac{\ln(2)}{T_{1/2,i}} + \sum_{j>i} \alpha_{i \to j} \quad . \tag{4}$$

(In practice, the sum in Equation 4 consists of a single term.)

The units for the flux and cross sections in the CONUP input data files are n/cm^2-s and barns, respectively. Internally, the units for the flux are converted to n/barn-day to be compatible with the cross-section unit and time in days.

The solution to Equation 2 is obtained as a power series in time t:

$$n_i(t) = \sum_{m=0}^{N_{max}} b_{im} t^m \quad , \tag{5}$$

where N_{max} is the maximum number of terms retained in the series. For the purposes of coding, it is convenient to replace the power-series coefficients $b_{i,m}$ by the related coefficients

$$A_{im} = b_{i,m-1} t^{m-1} \quad . \tag{6}$$

Equation 5 then becomes

$$n_i = \sum_{m=1}^{N_{max}+1} A_{im} \quad . \tag{7}$$

By substituting the power series into the differential equation, and by defining the diagonal matrix element $\alpha_{i \to i} = -\beta_i$, one can easily derive the following recursion relation for the coefficients: For $m \geq 2$,

$$A_{im} = \frac{t}{m-1} \sum_{k=1}^{i} \alpha_{k \to i} A_{k,m-1} \quad , \tag{8}$$

and the starting relation (m=1) is

$$A_{i1} = n_i(0) \quad . \tag{9}$$

The power-series coefficients A_{im} are calculated in the PWRSRS subroutine and used in the GASBURN subroutine. A power-series solution was chosen because, for most cases, the transmutation and decay rates are relatively slow, and a power series converges rapidly. Although it is unimportant in tag gas identification, ^{133}Xe with its short 5-day half-life is an exception. However, the power-series solution can still be used by subdividing the total time into subintervals where convergence is not a problem.

2.4 BURNUP AND PRODUCTION EQUATIONS FOR ISOTOPES IN THE FISSION PRODUCT BLOCK

The fission product isotopes (denoted by the dark triangles in Figure 3) are distinguished from the tag isotopes in two ways. Different cross-section

and flux sets are used to transmute them, and they have a source term from the fissioning of isotopes in the fuel. This source term, S_i, is simply an extra term added to the right-hand side of Equation 2, i.e.,

$$\frac{dn_i}{dt} = \sum_{j<i} \alpha_{j-i} n_j - \beta_i n_i + S_i \quad , \tag{10}$$

and is calculated from

$$S_i = \sum_k y_{ki}\, \phi_l(I_\phi)\, \sigma_{f,k}(I_{asm})\, N_{lkt} \quad , \tag{11}$$

where the summation is over all the fissionable isotopes in the fuel, and y_{ki} is the yield of isotope i from the fission of isotope k.

The other new quantities appearing in Equation 11 have the following definitions: $\sigma_{f,k}(I_{asm})$ is the effective one-group fission cross section for isotope k and is further qualified by the assembly type (e.g., inner and outer drivers). N_{lkt} is the amount of the fissionable isotope k in one fuel pin for core location l. The tag index t is used to distinguish situations where multiple tags reside in a single fuel assembly or core location.

At present, the burnup of the fissionable isotopes within a refueling cycle is ignored and S_i is treated as a constant within the cycle. This choice assumes that the amounts of fissionable isotopes are readily available on a cycle-by-cycle basis. A future alternative is to include these isotopes in the burnup calculation along with the tag and fission product isotopes.

A constant production rate S_i is equivalent to integrating a continuous series of instantaneous fissions over time. Consequently, one can easily solve Equation 2 with the constant source by integrating Equation 5 with respect to time, yielding

$$n_i(t) = \sum_{m=0}^{N_{max}} \left(\frac{t}{m+1}\right) b_{im} t^m \quad . \tag{12}$$

As before, with the alternate notation used in the code, this may be rewritten as

$$n_i = \sum_{m=1}^{N_{max}+1} \left(\frac{t}{m}\right) A_{im} \quad , \tag{13}$$

and the starting condition for the recursion relations is replaced by

$$A_{i1} = S_i \quad . \tag{14}$$

Thus, the PWRSRS routine may be used (unmodified) for the burnup and production calculation of the fission product isotopes. One simply includes the factor, (t/m), in the calculation of the sum in Equation 7.

3.0 CODE STRUCTURE

The CONUP code is the computational implementation of the physical tag gas model described in Section 2.0. This section describes the logical structure of CONUP (main module) and its subroutines so that the reader can better understand the logical and numerical functions performed to produce the tag gas concentrations. The main module is described in Section 3.1. The primary subroutines called by CONUP are described in Section 3.2, and the auxiliary subroutines are briefly described in Section 3.3.

3.1 MAIN MODULE (CONUP)

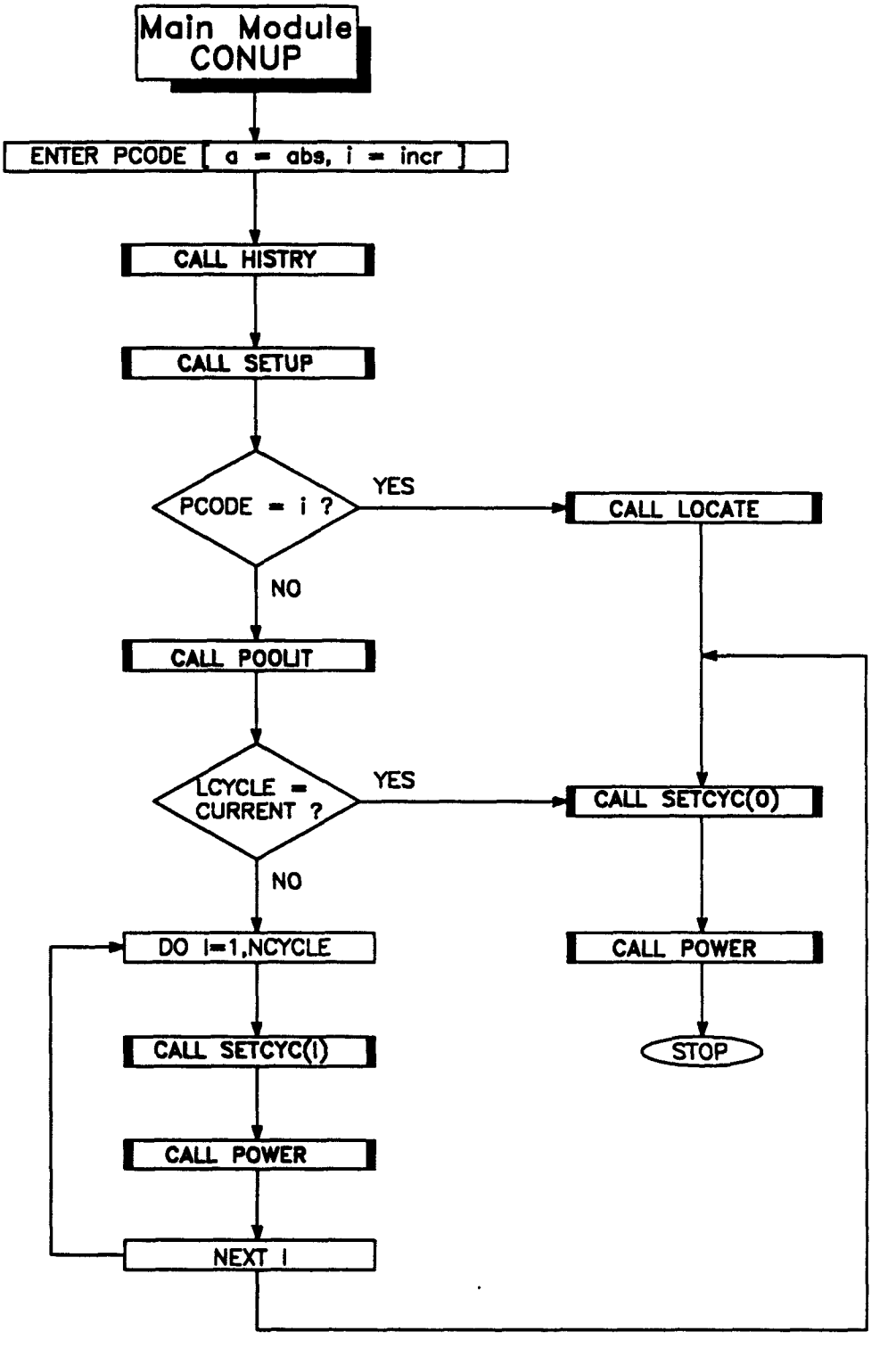
The main program module is CONUP. Figure 4 illustrates the logical organization of CONUP in flow diagram format, in which the subroutine call statements are emphasized with bold type. The CONUP calls the subroutines HISTRY, SETUP, POOLIT, LOCATE, SETCYC, and POWER; and the functions performed by these subroutines are described below.

Initially, CONUP queries the user for cycle input. The first parameter requested is the processing mode parameter PCODE. The user is prompted to enter character "a" for absolute processing or character "i" for incremental processing. If the absolute mode is selected, the code calculates the tag gas concentrations over all prior reactor cycles. If the incremental mode is selected, CONUP locates the last cycle concentrations for each assembly before processing the current cycle. This processing mode computes concentrations considerably faster, but requires the user to maintain and process all prior cycle concentration files.

The next step in CONUP processing is to call subroutine HISTRY to obtain and initialize cycle identification parameters such as cycle ID, cycle reactor power, effective full power operating days (EFPD) for the current cycle, and the most recent cycle processed. Subroutine HISTRY's function and parameters are described more fully in Section 3.2.

CONUP proceeds in the initialization process by calling subroutine SETUP. SETUP creates a set of indexes for each assembly in the current cycle. These indexes point by core location to the fluxes and cross sections that will be used for the tag gas calculations in the current cycle.

Figure 4. Flow Chart for Main Program CONUP.



After preliminary index initialization, CONUP tests the entry parameter PCODE and branches to subroutine LOCATE if the incremental mode has been selected. Subroutine LOCATE will locate the most recent concentration value for each assembly in the current cycle. If the absolute mode is selected, CONUP will call subroutine POOLIT to extract the initial tag gas ratio values and convert to isotopic concentrations. The next step in CONUP processing is to set current cycle processing parameters.

The CONUP sets up a looping code structure to execute subroutines SETCYC and POWER for all prior cycles. Subroutine SETCYC is used to initialize all cycle parameters for the current cycle while subroutine POWER burns and transmutes each assembly's tag gas over the current reactor cycle. When processing is completed for all prior cycles, CONUP then processes the tag gas concentrations for the current cycle. When incremental processing has been selected, CONUP only processes the current cycle after locating the prior cycle assembly concentrations.

The CONUP code makes extensive use of common data blocks to share parameters between subroutines. The following section details the function of each of the primary support subroutines. Table 2 provides a list and short description of each of the primary support subroutines.

3.2 PRIMARY SUPPORT SUBROUTINES

The primary support subroutines manage, locate, set up, and process the nuclear data associated with each assembly and reactor cycle to produce the current tag gas concentrations.

3.2.1 HISTRY

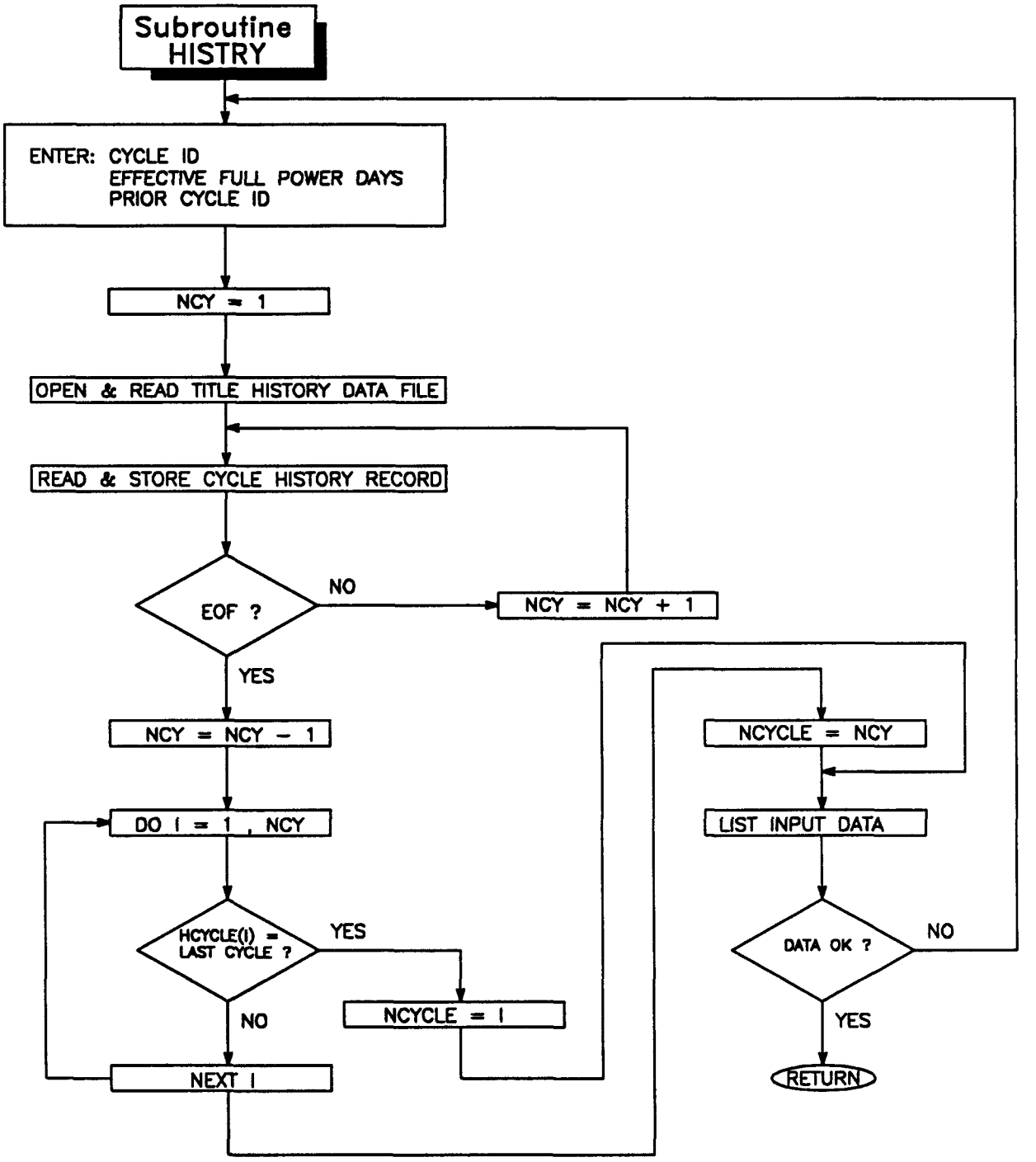
HISTRY is the first subroutine called by CONUP. The primary function of subroutine HISTRY is to query the user for cycle input and then initialize the cycle history data for all prior cycles. Figure 5 shows the logic flow of subroutine HISTRY.

After the current (user entered) cycle data is read, HISTRY opens the cycle history file (see the first file in Appendix C for an example) and saves parameter data for all prior cycles. These data will be used in the calculations of tag gas for assemblies loaded in prior cycles. The parameter NCY is the total number of prior cycles accumulated and is used to initialize the parameter NCYCLE. Once the number of prior cycles is determined, subroutine HISTRY allows the user to check and verify all cycle data input. When satisfied, the user types in "y" to the prompt ("data correct") and the subroutine returns control to the main program module CONUP. Parameter NCYCLE will be used by the main module CONUP to process all assemblies over prior cycles.

Table 2. Primary Support Subroutines.

Subroutine	Purpose
HISTRY	Inputs current cycle parameters and initializes prior history cycle parameters
SETUP	Initializes current cycle cross-reference indexes
POOLIT	Creates an initial concentration file using input from tag gas ratios and assembly configuration files
LOCATE	Creates an initial concentration file from previous cycle concentration files
SETCYC	Sets up current "BURN" cycle processing parameters and flags all assemblies in current cycle for cycle processing
POWER	Processes and "BURNS" all flagged assemblies for current process cycle
RATIO	Calculates initial tag gas amounts from the tag gas ratios for all tags of a cycle assembly
MOTA	Creates an initial concentration for each MOTA tag within a given assembly. Tags not irradiated in prior assemblies are initialized from the tag ratio file
SRATIO	Calculates isotopic amounts for an individual MOTA tag
GASBURN	Calculates tag fission product production and tag transmutation for each assembly tag
PWRSRS	Calculates the A_{im} coefficients
MPOWER	Processes and "BURNS" all MOTA tags located in prior cycle assemblies
AMTR	Converts fractional tag ratios to tag gas amounts

Figure 5. Flow Chart for Subroutine HISTRY.



3.2.2 SETUP

Subroutine SETUP initializes the current process cycle cross-reference indexes. These indexes are used to relate nuclear data input with each assembly's tag gas. Subroutine SETUP's flow diagram is shown in Figure 6. SETUP first calls subroutine CYCCNF, which reads the cycle configuration file and determines the number of assemblies. (See the second file listed in Appendix C for an example of this file.) In CYCCNF, each assembly's fabrication number is stored with its associated reactor core location.

The next function SETUP performs is to read and store all the tracked isotope indexes and their associated cross sections and yields. This is the primary function of subroutine READAT. This data is read from the nuclear data file (an example of which is given in the third file of Appendix C) and is used in the processing of each assembly over its current process cycle.

SETUP sets up parameter "i" as the main loop index to process from one (1) to the total number of assemblies (parameter NASC) in the current cycle. For each assembly, SETUP sets up parameter "j" as the assembly subindex to initialize each set of tag indexes within each assembly where NTAGS is the number of tags in the current assembly. The parameter SERIAL represents the current assembly ID and varies with each index "i". Subroutine READASM reads the assembly configuration file for a given assembly and extracts flux, cross-section, and assembly configuration data for each tag. (See the fourth file listed in Appendix C for an example of this file.) The indexes for flux, cross-section (or sigma), and assembly types are initialized for given assembly tag configuration data. When complete, all current cycle assemblies will have assigned flux and cross-section indexes for processing tag gas calculations. The main program module is now ready to extract the initial tag gas values for cycle processing.

3.2.3 POOLIT

Figure 7 shows the logic flow of subroutine POOLIT. POOLIT is called by the main program module CONUP to extract all initial tag gas values for processing in the absolute mode. These initial tag gas values are stored in the tag gas ratios file and read into POOLIT from the subroutine RATIO. (See the fifth file listed in Appendix C for an example of this input file.)

Subroutine POOLIT opens the initial concentration file for the current process cycle, processes each cycle assembly, and saves each assembly's initial tag gas values. The serial number and run date values are stored in parameters SERIAL and RDATE. If a MOTA assembly is located in the current cycle configuration, some additional processing is required. The subroutine MOTA is called to locate (if any) all prior assemblies in which the current MOTA tags are found. If found, MOTA processes those tags through all prior cycles to establish an initial set of tag gas values.

Subroutine RATIO extracts the initial tag gas values from the tag ratio file database. Subroutine WRCONX is called to write the current assembly's TG and FP data blocks to the cycle concentration file. POOLIT continues to loop through and process all assemblies in the current process cycle.

Figure 6. Flow Chart for Subroutine SETUP.

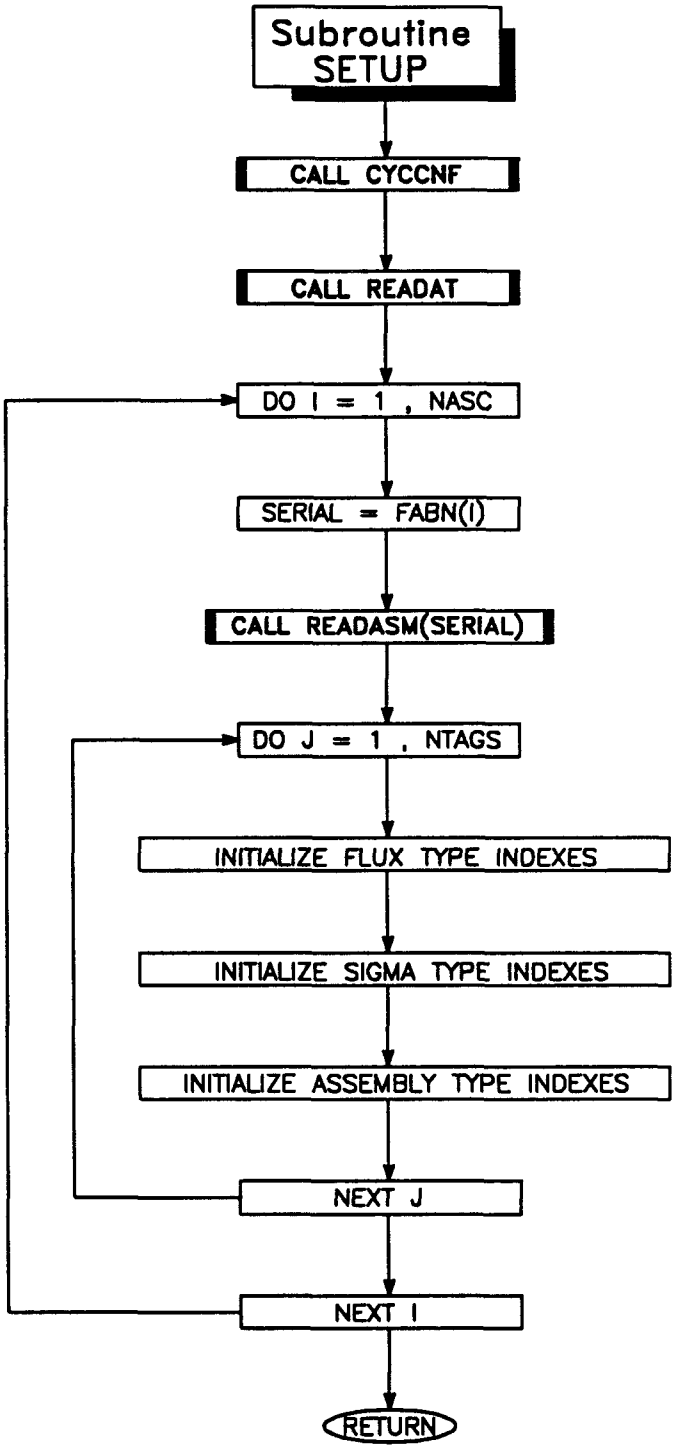
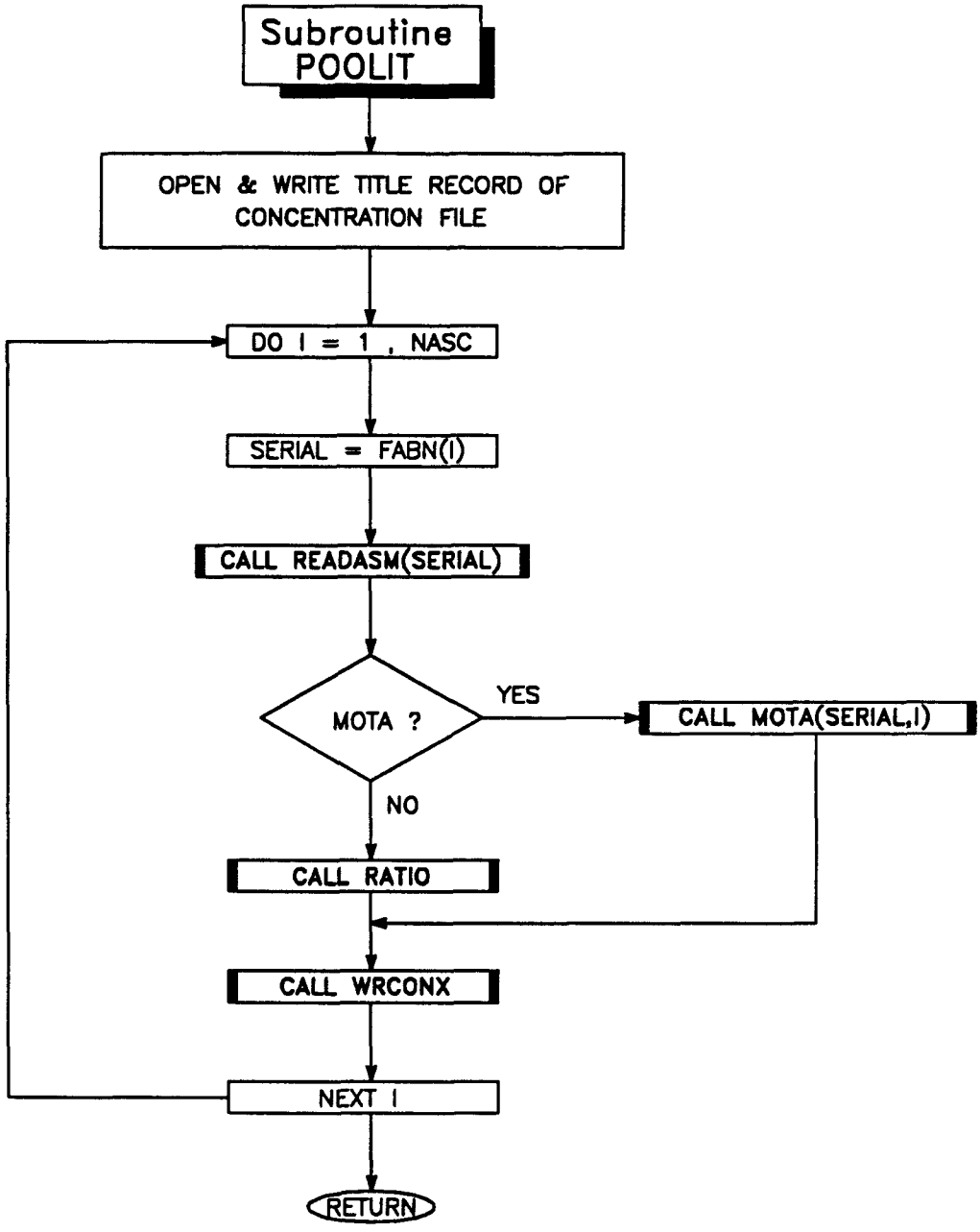


Figure 7. Flow Chart for Subroutine POOLIT.



3.2.4 LOCATE

Subroutine LOCATE performs a similar function to subroutine POOLIT, except the initial tag gas values are extracted from prior cycle concentration files by subroutine GETASM. Figure 8 shows the logic flow of subroutine LOCATE.

As LOCATE loops through all the cycle assemblies, a check is made to determine if the current assembly has been located in a prior cycle. If not found, LOCATE extracts the initial tag gas values from the tag ratio file database via subroutine RATIO. Subroutine MOTA is called only when a MOTA assembly is located in current process cycle. When processing is complete, LOCATE returns to the main program module CONUP.

3.2.5 SETCYC

Subroutine SETCYC sets up the current "BURN" cycle processing parameters. The main program CONUP calls SETCYC for each reactor cycle processed. The flow diagram for subroutine SETCYC is shown in Figure 9.

SETCYC initially sets the current process cycle identifier (CYID) and cycle time (CTIME). For the specified CYID, SETCYC opens, reads, and saves the associated reactor core locations (LOCUR) and serial numbers (CFABN) from the cycle configuration file. The values of the elements in the array LOC represent the original core locations of the input process cycle. When the parameter INDEX is set to zero, all assembly locations are initialized to the current cycle values stored in LOC. SETCYC loops through all assemblies via the number of assemblies index (NASC) and sets the burn flag on if the specific assembly is in the reactor cycle CYID. All flagged cycle concentration data are written to a temporary concentration file to be processed by subroutine POWER.

The auxiliary subroutines called by SETCYC are WTRITX to write the concentration file title record, RDCONX to read an assembly concentration record from the prior cycle concentration file, and WRCONX to write an assembly concentration record.

3.2.6 POWER/MPOWER

Figure 10 shows the flow chart for subroutine POWER. POWER is the main subroutine that burns and transmutes tag gas concentrations for each flagged assembly in the specified reactor cycle. MPOWER is similar to POWER, but only processes MOTA assemblies located in prior cycle assemblies.

Subroutine POWER first calls subroutine RDDAT to read the cycle fissionable amounts file and the cycle flux file for each assembly in the current process cycle. (See the sixth and seventh files listed in Appendix C for examples of these input files.) POWER loops and processes each assembly TG block for the current reactor cycle. If the cycle time is larger than 70 EFPDs, POWER processes each TG block for total reactor cycle time (T) by dividing T into 70-day segments. This is done to improve the convergence of the power-series algorithm for the tag gas calculations.

Figure 8. Flow Chart for Subroutine LOCATE.

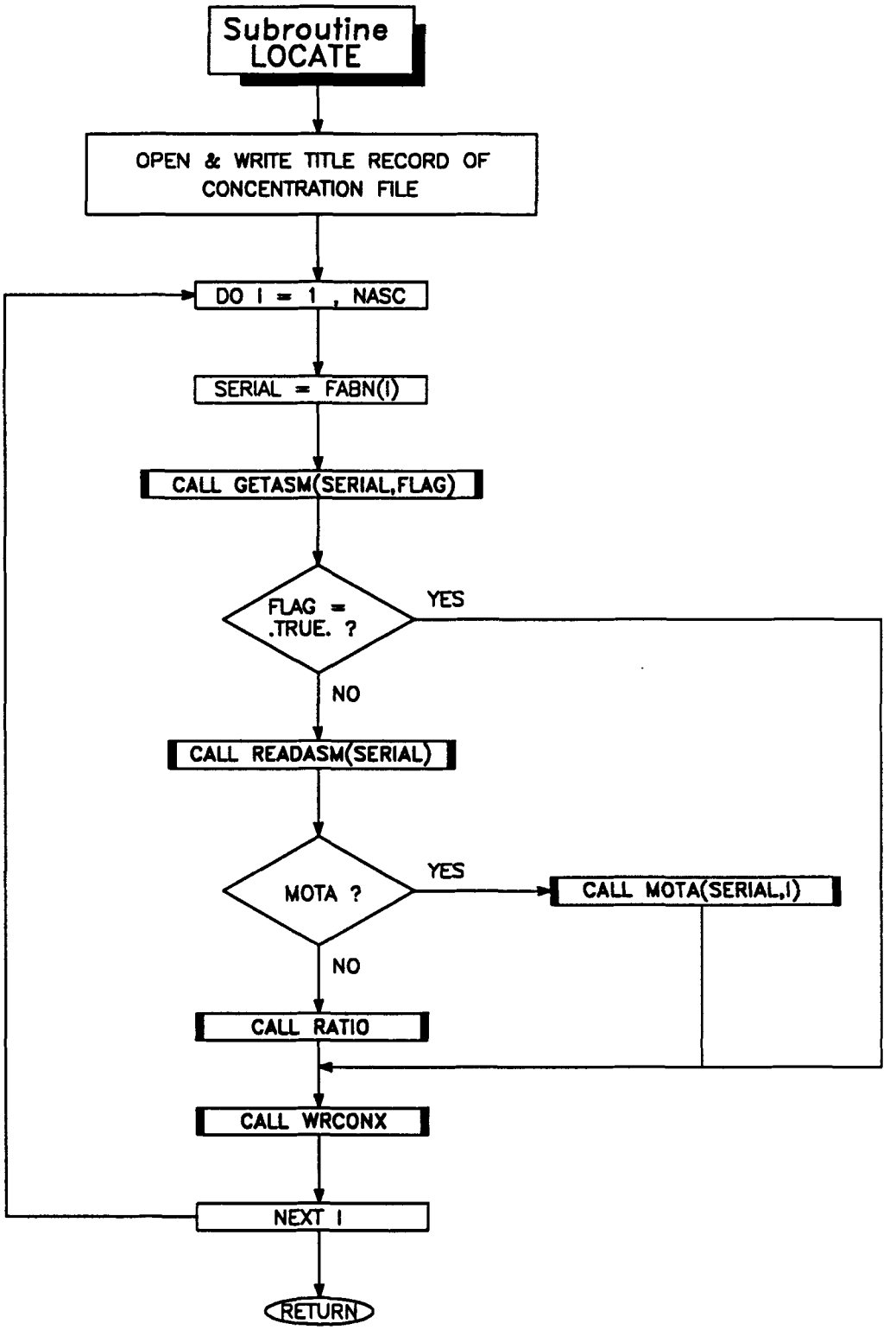


Figure 9. Flow Chart for Subroutine SETCYC.

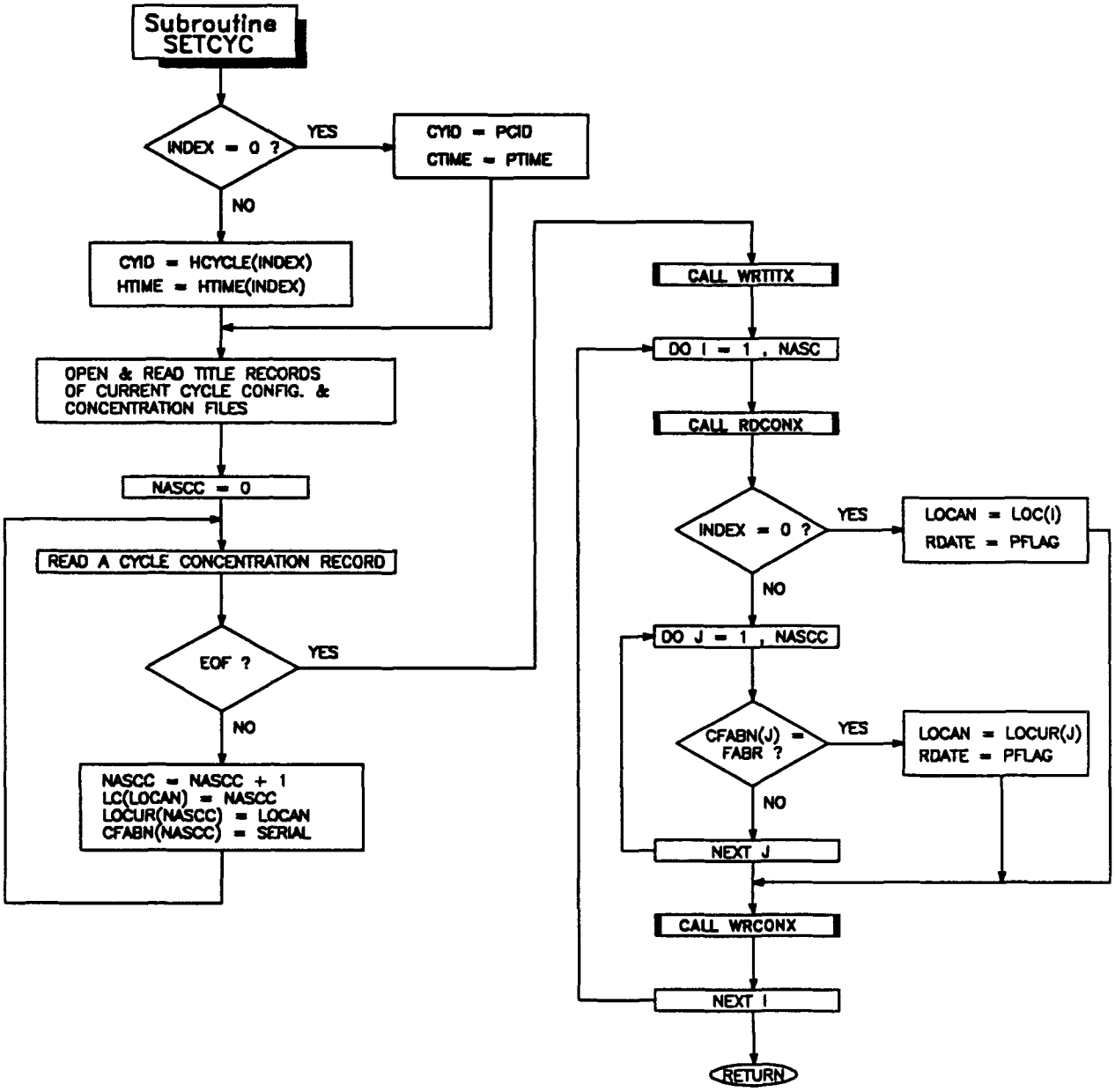
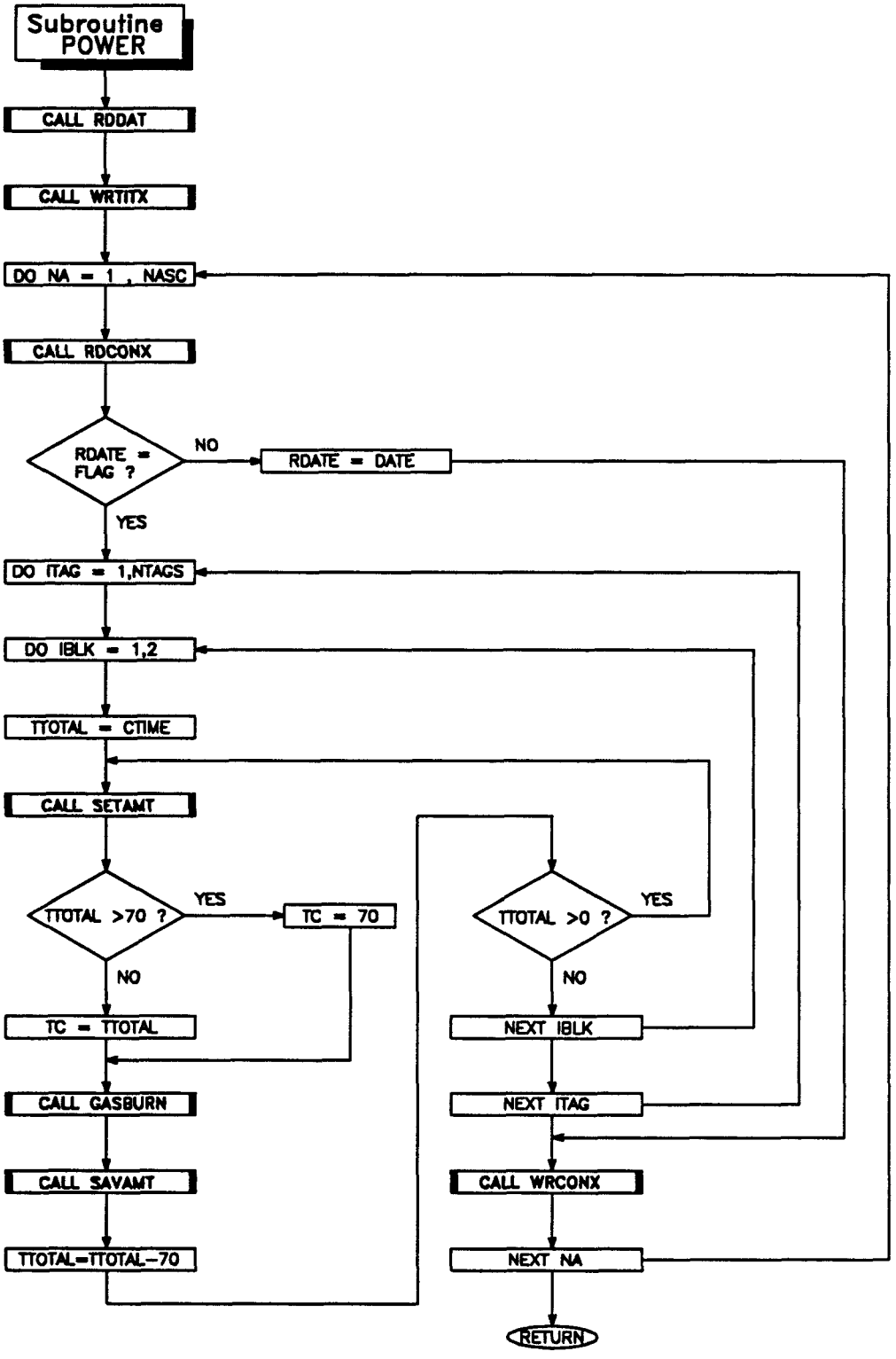


Figure 10. Flow Chart for Subroutine POWER.



The auxiliary subroutines SETAMT and SAVAMT set and save incremental tag gas concentration values for each TG data block.

3.2.7 RATIO/SRATIO

The subroutine RATIO calculates the initial tag gas amounts for all tags of a reactor assembly. The subroutine SRATIO is called by the MOTA subroutine to calculate the individual MOTA tags. The flow of logic for subroutine RATIO is shown in Figure 11.

RATIO opens and reads from the tag gas ratio file. For each tag in the given assembly, subroutine RATIO reads and locates the initial tag ratio values. Subroutine AMTR takes these ratios and converts them (via Equation 1) to individual isotope tag gas amounts. Finally, RATIO converts the tag gases to atoms per pin. The parameters FACXE and FACKR convert the xenon and krypton isotopes, respectively.

3.2.8 MOTA

The subroutine MOTA is called from subroutines POOLIT and LOCATE. MOTA creates an initial concentration for each MOTA tag within a given MOTA assembly. MOTA tags that are not irradiated in prior assemblies are initialized from the tag gas ratio file. Figure 12 shows the flow chart for subroutine MOTA.

The first step in processing a MOTA assembly is to locate all MOTA tags in prior reactor cycles. This is done by a call to subroutine MINIT, which initializes the MOTA tag common arrays used to identify prior cycle tag irradiation in a specific assembly. After completing this task, MOTA starts processing each tag.

When a tag is processed in the absolute mode, it is first checked for prior assembly irradiation. The flag TFLAG and associated prior cycle data have been set for each tag in subroutine MINIT. If TFLAG is true, the tag has prior cycle irradiation and subroutines SRATIO and MPOWER are called. SRATIO extracts the initial tag values from the tag ratio file while subroutine MPOWER burns and processes this tag's concentrations over these prior cycles. If TFLAG is false, the tag has no prior cycle irradiation and only subroutine SRATIO is called for tag initialization.

If the user has selected the incremental processing mode, MOTA checks to see if TFLAG has been set. If TFLAG is true, MOTA calls subroutine GETAG to extract the initial tag values from the last burned cycle. If TFLAG is false, the tag has no prior cycle irradiation and hence subroutine SRATIO is called for tag initialization.

3.2.9 AMTR

Subroutine AMTR converts the fractional tag gas ratios to isotope tag gas amounts. AMTR is called from subroutines RATIO and SRATIO and uses the conversion formula given by Equation 1.

Figure 11. Flow Chart for Subroutine RATIO.

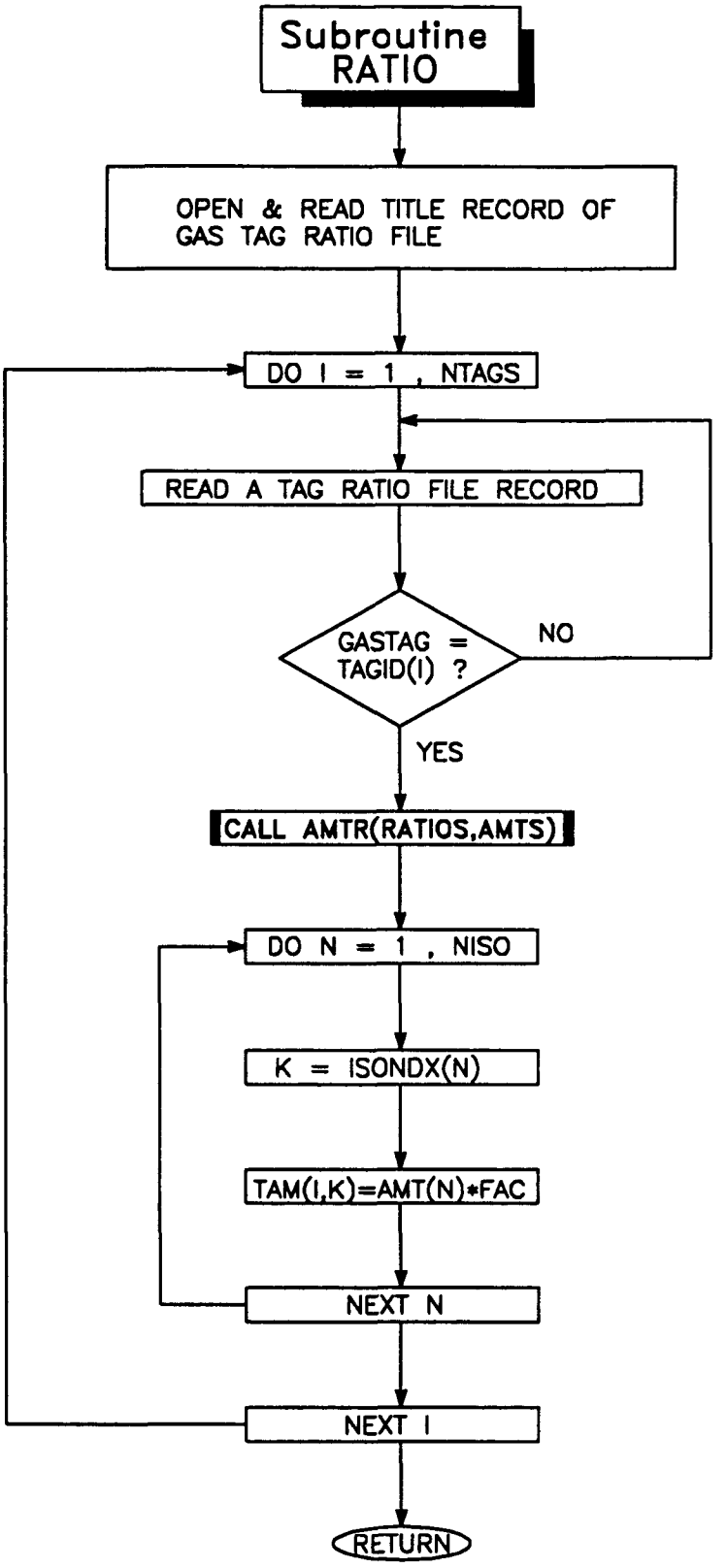
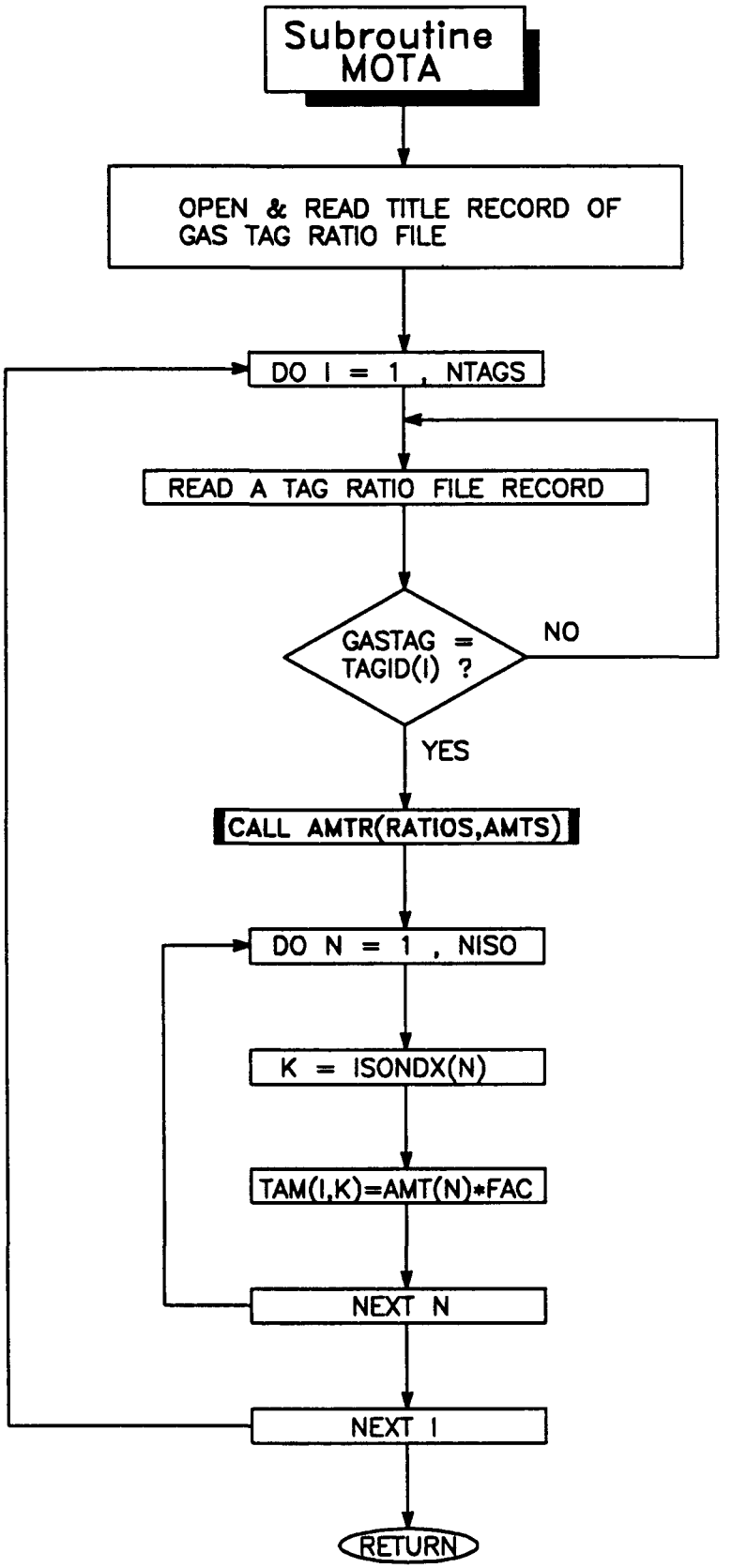


Figure 12. Flow Chart for Subroutine MOTA.



3.2.10 GASBURN

Subroutine GASBURN calculates tag gas transmutation and fission product production for a specific assembly and reactor cycle. This subroutine first calculates the transmutation and destruction rates, given by Equations 3 and 4. GASBURN then calculates the burnup equation for the tag gas isotopes (Equation 7) and the burnup and production equation for the fission product isotopes (Equation 13). The subroutine PWSRS is called by GASBURN to determine the appropriate A_{im} coefficients needed for the solution of these equations.

3.2.11 PWRSRS

The subroutine PWRSRS calculates the coefficients, A_{im}, via the recursion equation give by Equation 8. These coefficients are used to determine the the power-series solution to the decay and transmutation equations (see Equations 7 and 13).

3.3 AUXILIARY SUBROUTINES

This section contains a description of auxiliary subroutines called by CONUP's primary support subroutines. Table 3 provides a list and brief description of these subroutines. The following paragraphs expand these descriptions of CONUP's auxiliary subroutines.

3.3.1 CONCAT

The subroutine CONCAT creates a complete file name descriptor from a prefix name and specified extension. This subroutine is used extensively in the process of opening new files by cycle identification and data extension qualifier. An example of a data extension would be ".flx" for a cycle flux file.

3.3.2 CYCCNF

The subroutine CYCCNF opens, reads, and saves the current process data from the cycle configuration file. The number of assemblies in this process cycle is determined and stored in a common data block. CYCCNF is called once from subroutine SETUP.

3.3.3 CYFLUX

The subroutine CYFLUX initializes the tag flux data for a given cycle and assembly location. CYFLUX is called from subroutine MPOWER and is used to initialize tag flux data for MOTA tag calculations of tags burned in prior assembly reactor cycles.

Table 3. Auxiliary Subroutines.

Subroutine	Purpose
CONCAT	Constructs a complete file name extension for a given cycle identifier
CYCCNF	Reads and initializes current cycle assembly configuration data
CYFLUX	Initializes the flux data for a given cycle and tag assembly location
DATIME	Reads and extracts current date and time from system
GETASM	Reads the prior cycle configuration files and locates the current assembly's last process cycle
GETAG	Reads and extracts from the cycle concentration file the current MOTA tag values
MINIT	Initializes the MOTA tag common block used for identifying MOTA tags prior irradiated assemblies
RDCONX	Reads an assembly's concentration record
RDDAT	Reads fissionable amounts and plenum and core fluxes for the current cycle processing
RDTITX	Reads title record of the concentration file
READASM	Reads assembly configuration file and initializes tag data associated with a specified assembly
READAT	Reads and initializes nuclear data yields and cross sections for current cycle tag gases
SAVAMT	Temporary storage of tag gas concentrations for current irradiation cycle
SETAMT	Temporary restore of tag gas concentrations for current irradiation cycle
WRCONX	Writes out an assembly concentration record
WRTITX	Writes out title concentration record

3.3.4 DATIME

Subroutine DATIME is a run date and time subroutine that extracts and formats the date and time from the operating system of a SUN Workstation*. This information is used as a run date stamp for the concentration file and is contained in each assembly record.

3.3.5 GETASM

The subroutine GETASM reads the previous cycle configuration file and tests the current serial number to locate the current assembly's last process cycle. The order of search is incremented from the last prior cycle to the first reactor cycle so that assembly concentration data will be to the most current prior cycle. Subroutine GETASM is called from subroutine LOCATE, which is executed when the user selects the incremental processing mode.

3.3.6 GETAG

Subroutine GETAG opens, reads, and extracts from a specified cycle concentration file the current MOTA tag values. GETAG is called from subroutine MOTA and is executed if the user has selected the incremental processing mode.

3.3.7 MINIT

The subroutine MINIT creates the MOTA tag common data used for identifying prior MOTA tag irradiated assemblies. This subroutine is called from subroutine MOTA. MINIT opens and reads the assembly configuration file to locate all assemblies in which the tag was located. This information is used by MOTA to burn each tag over all prior assembly cycles.

3.3.8 RDTITX/RDCONX/WRTITX/WRCONX

The subroutines RDTITX, RDCONX, WRTITX, and WRCONX read and write records for the cycle concentration file. Subroutine RDTITX reads a title and heading record for the cycle concentration. RDCONX reads an assembly record containing all tag gases for that assembly. WRTITX writes a title and heading record for the cycle concentration file. And, lastly, WRCONX writes an assembly record for the cycle concentration file. These subroutines are used extensively throughout the CONUP program.

3.3.9 RDDAT

Subroutine RDDAT reads data needed for the current processing cycle from the cycle fissionable amounts and the cycle flux files. RDDAT is called from subroutine POWER for each reactor process cycle.

---

*SUN Workstation is a trademark of SUN Microsystems, Inc.

3.3.10 READASM

The subroutine READASM reads an assembly configuration file and initializes tag data associated with a specified assembly. READASM is called from subroutine LOCATE and POOLIT and is used to get all tags associated with a specific assembly.

3.3.11 READAT

Subroutine READAT reads from the nuclear data file and initializes the nuclear cross sections, flux types, and fission yields. This subroutine is called from SETUP and is initialized once during program execution.

3.3.12 SETAMT/SAVAMT

The subroutines SETAMT and SAVAMT are used to temporarily save and restore an assembly's tag gas concentrations when the current cycle is being processed incrementally over cycle time. These subroutines are called from the POWER and MPOWER subroutines.

4.0 USAGE

4.1 SYSTEM REQUIREMENTS

The program CONUP runs on a SUN 4/260 Sparc Workstation under the SUN UNIX* 4.1 Operating System. CONUP was written in SUN FORTRAN 77 and can be ported to other UNIX systems supporting FORTRAN 77. The filenaming convention and directory structure for CONUP are described in Section 4.2, "File Management." To run CONUP, all cycle input files need to be constructed prior to each cycle processed. The following sections describe the input files and their structures. Section 4.4 describes the actual running of the CONUP program.

4.2 FILE MANAGEMENT

This section defines the current structure of the file organization as used by the main program module CONUP.

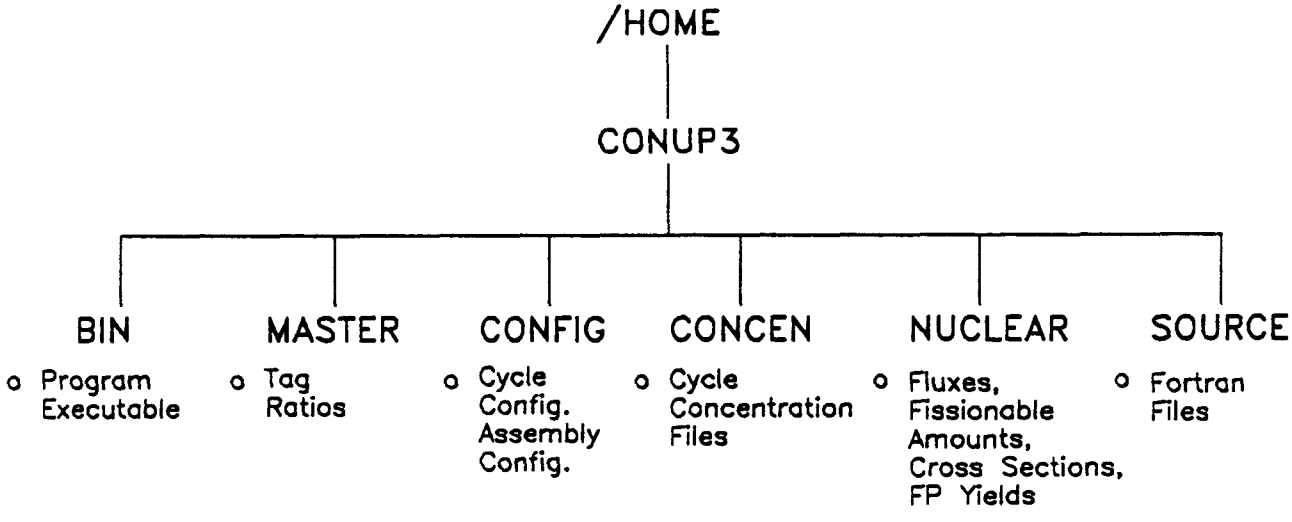
---

*UNIX is a trademark of American Telephone and Telegraph Bell Laboratories.

4.2.1 Directory Structure

CONUP's directory structure is illustrated in Figure 13. The directory structure on UNIX systems is a tree structure similar to IBM* DOS** PC. CONUP's main directory is called CONUP3, and it resides in the user's home directory. Subdirectories to CONUP3 are MASTER, BIN, CONCEN, CONFIG, NUCLEAR, and SOURCE. The CONUP concentration program file structure was established to better organize and maintain the nuclear database files required to run this system.

Figure 13. File Directory Structure.



The MASTER subdirectory contains the tag gas ratio database file along with a tag list file. The BIN subdirectory contains the CONUP executable binary program module. The CONCEN subdirectory contains all the cycle concentration files. The CONFIG subdirectory contains the cycle configuration files for each reactor cycle. The NUCLEAR subdirectory contains fluxes, fissionable amounts, and cross-section/yields files for each cycle. And lastly, the SOURCE subdirectory contains all FORTRAN 77 CONUP program and subroutine source files. The subroutine CONCAT creates the full file name extensions that allow CONUP to access these subdirectories.

4.2.2 Filenaming Convention

The CONUP program utilizes a number of files to process concentrations for each reactor irradiation cycle. Early in the programming of this code, it was decided to establish a filenaming convention for each type of input/output data file. Table 4 lists the file extensions along with descriptions of each

---

*IBM is a trademark of International Business Machines, Inc.

**DOS is a trademark of the Microsoft Corporation.

Table 4. File Name Extensions.

File extension	Description
.flx	Flux files by cycle ID
.amt	Fissionable amount files by cycle ID
.con	Concentration files by cycle ID
.cnf	Configuration files by cycle ID
.f	FORTRAN 77 source files

type of file. This type of file convention minimizes the code required for accessing these files and is good programming convention. In addition, this method aids the user in visual determination of the type of file used in each subdirectory.

4.3 INITIALIZING CONUP INPUT FILES

The main effort in running the CONUP code is in the setup and maintenance of the cycle database input files for each reactor cycle. For cycle processing, the code CONUP requires a cycle configuration file, a cycle flux file, a fissionable amounts file, a cycle history file, an assembly configuration file, a tag ratio file, and a cross-section and yields file. These seven files represent CONUP's input database for all reactor cycles.

4.3.1 Cycle Configuration File

The cycle configuration file contains the assembly identifier along with the reactor core location. This file is used by CONUP as the master assembly core loading configuration for the current irradiation cycle. An example of this file is the second file given in Appendix C. The format for this file is rigidly fixed and contains the title record(A80), cycle id(6x,A6), date(6x,4A2), time(6x,4A2), a field label record(A40), followed by the assembly serial number and core location for each assembly. The format for the serial number, core location record is (A6,2x,I4). A UNIX text editor (i.e., VI Editor) can be used to construct this file.

4.3.2 Cycle Flux File

The cycle flux file contains the core and plenum flux values for each core location. An example of this file is the seventh file contained in Appendix C. The structure of the format is rigid and contains the title record (A80), the field identification record (A80), followed by the core location for each core and plenum values. The format for this record is (2x,I4,2E15.5). The cycle fluxes represent an "Effective" axial averaged core and plenum flux.

4.3.3 Cycle Fissionable Amounts File

The cycle fissionable amounts file contains the initial and current decayed amounts of fissionable nuclides. An example of this file is the sixth file in Appendix C. The format for this file is as follows.

Record#	Parameter(s)	Format
1	Title Record	(A80)
2	Cycle ID	(6x,A6)
3	Date	(6x,4A2)
4	Time	(6x,4A2)
5	Decay Date	(6x,4A2)
6	Serial #	(6x,A6)
7	Gas Tag #	(6x,A6)
8	PU239 Initial,Current	(6x,2E12.5)
9	PU240 Initial,Current	(6x,2E12.5)
10	PU241 Initial,Current	(6x,2E12.5)
11	PU242 Initial,Current	(6x,2E12.5)
12	U 235 Initial,Current	(6x,2E12.5)
13	U 238 Initial,Current	(6x,2E12.5)
:	: : :	:
:	: : :	:

Records 6 through 13 repeat for each tag gas irradiated in a reactor process cycle. The initial value is not used in the gas production calculations, but is provided as a reference. The current value is the value at the beginning of the cycle. This value is used to calculate gas production in the FP data block for each tag gas within an assembly. These amounts are given in atoms/barn-cm. They are converted to atoms per pin in subroutine RDDAT by multiplying by a fixed effective assembly value of 11,633 cm^3, dividing by the number of pins per assembly, and multiplying by 10^{24} barns/cm^2.

This file may be updated for every cycle for each driver fuel assembly, but a constant amount is currently being used. This is a mechanism for approximately including time dependence of the amounts of fissionable nuclides. Future upgrades to CONUP will include the recalculation of fissionable amounts for the start of each reactor cycle.

4.3.4 Cycle History File

The cycle history file contains a list of all reactor cycles with their associated power levels and EFPDs. An example of this file is listed as the first file in Appendix C. The CONUP code reads this file to initialize the EFPDs associated with each reactor cycle. In the absolute processing mode, CONUP uses this information to burn each assembly over reactor cycles that each assembly has been loaded. For MOTA assemblies, CONUP calculates the initial MOTA tag gas concentrations based on the prior cycle history for which those tags were loaded. The cycle history file is updated only at the end of each reactor cycle and includes some additional date and comment parameters not used by the CONUP code. The format of the cycle history file is rigid in structure and is as follows.

Record#	Parameter(s)	Format
1	Title Record	(A80)
2	Label Record 1	(A80)
3	Label Record 2	(A80)
4	Label Record 3	(A80)
5	Label Record 4	(A80)
6	Cycle ID, Startup Date, Shutdown Date, Power, EFPD's, Comments	(A6,2x,4A2,2x, 4A2,F6.0, F7.1,3x,A37)
:	: : :	:
:	: : :	:

Record 6 repeats for each reactor cycle with the last record being the latest prior cycle to the current process cycle. The current process cycle is added to the history file when its cycle has been completed.

4.3.5 Assembly Configuration File

The assembly configuration file is the master database file for each assembly's configuration. This file contains the tag IDs along with their associated flux and cross-section types. The assembly information consists of the number of tags, the type of assembly (i.e., MOTA), and assembly description. With each tag associated with an assembly, the tag ID, number of pins, cross-section type (for both the TG and FP blocks), and flux types (for both the TG and FP blocks) are provided. An example of a partial assembly configuration file is given as the fourth file in Appendix C.

The assembly configuration file is updated only when a new assembly is manufactured for initial loading into a reactor cycle. Future assemblies can be installed in this file, but CONUP will not utilize this information until

the assembly is included in one of the cycle configuration files. The format for the assembly configuration file is as follows.

Record#	Parameter(s)	Format
1	Title Record	(A80)
2	Label Reference 1	(A80)
3	Lable Reference 2	(A80)
4	Label Reference 3	(A80)
5	Label Format 1	(A80)
6	Label Format 2	(A80)
7	Assembly ID, # Tags, Assembly Type, Load Date, Remarks	(A6,3x,I3,4x, A6,2x,4A2,3x, A30)
8	Tag#, TagID, #Pins, Tag/FP Cross Section types Tag/FP Flux types	(9x,I3,4X,A6,2x,I3, 8x,A6,2x,A6,2x, A6,2x,A6)
:	: : :	:
:	: : :	:

Record 8 repeats for each tag within an assembly, and record 7 repeats for each available assembly to be loaded into a reactor cycle.

4.3.6 Tag Gas Ratios File

The tag gas ratios file contains the initial ratios measured when the tag gas was prepared for assembly installation. The CONUP code will read this file to construct the initial isotopic concentrations for each tag isotope. An example of a partial listing of this file appears as the fifth file in Appendix C. This file is updated only when a new set of tags is created for use in current or future reactor cycles. The format for the tag gas ratios file is as follows.

Record#	Parameter(s)	Format
1	Title Record	(A80)
2	Total # Tags	(I9)
3	Label Reference 1	(A80)
4	Label Reference 2	(A80)
5	Label Reference 3	(A80)
6	Label Reference 4	(A80)
7	Tag ID, %Xe, %Kr, Amount(cc's)	(A6,2x,2F7.2,F6.1)
8	(Xe Tag Ratios(i),i=1,6)	(8x,6E12.5)
9	(Xe Ratios(i=7,8),(Kr Ratios(i), i = 1,4)	(8x,6E12.5)
10	(Kr Ratio(i),i=5)	(8x,E12.5)
:	: : :	:
:	: : :	:

Records 7 through 10 repeat for each tag gas.

4.3.7 Nuclear Data File

The nuclear data file contains the current cross sections, half-lives, and fission product yields needed to calculate the burnup and production of fission product and tag gas isotopes. It also includes values (denoted by DI) for an index array, idghtr[i], that defines the daughter isotope for each parent. An example of this file is given as the third file in Appendix C. The file formats are as follows.

Record#	Parameter(s)	Format
1	Title Record	(A80)
2	Blank Record	(/)
3	# Tag Cross Section Types	(9x,I3)
4	Label Reference 1	(A80)
5	Label Reference 2	(A80)
6	Label Reference 3	(A80)
7	Blank Record	(/)
8	Type ID (i.e. MOTA), (Isotope(i),i =1,8),	(A6, 2x,8F8.4/)
9-10	(Isotope(i),i=9,21)	2(6x,8F8.4/))
11	Blank Record	(/)
:	: : :	:
	[Records 8 through 11 repeat for each Tag type ID]	
:	: : :	:
21	FP Cross Section Title	(A80)
22	# of FP types	(9x,I3)
23	Label Record	(A80)
24	FP Type ID, (FPisotope(i),i=1,6)	(A6, 6E12.5)
:	: : :	:
	[Records 25 through 26 repeat for each FP type ID]	
:	: : :	:
27	Blank Record	(/)
28	Fluxe Types Title	(A80)
29	# of Flux Types	(9x,I3)
30	Flux Type ID	(A6)
:	: : :	:
	[Records 31 and 32 repeat for each Flux Type ID]	
:	: : :	:
33	Blank Record	(/)
34	Chain Yields Title	(A80)
35	Label Record	(A80)
36-56	(Yield(i,j), Index(i), Daughter Index(i), Half-Life(i), j=1,#FP), i = 1,21)	(6F8.4,8x,2I4,E13.6)

Neutron-capture cross sections are given by isotope for each of several cross-section types, I_σ (see Equation 3). Currently, three capture cross-section types are defined with the type identifiers: "CORE", "PLENUM",

and "MOTA". Type identifiers in the assembly configuration file then control which cross-section types are used for each assembly and tag.

The nuclear data file also contains fission cross sections for each of the six fissionable isotopes. These cross sections are also classified by type, I_{asm} (see Equation 11). The current identifiers are "IN" for inner drivers, "OT" for outer drivers, and "MT" for MOTA, although normally the MOTA assembly will not contain any fissionable isotopes important to tag gas identification. These type identifiers are synonymous with the assembly identifier also given in the assembly configuration file.

Finally, flux-type identifiers, I_g, are also used to control which fluxes are used by the calculation (see Equations 3 and 11). Three identifiers are currently used: "CORE", "PLENUM", and MOTA". These identifiers are specified by assembly and tag in the assembly configuration file and determine which fluxes from the cycle flux file are used.

For each of the 6 fissionable isotopes, j, fission-product yields Y(i,j) are given by a table of values for each of the 21 tag gas and fission product isotopes i.

The nuclear data file is read once during the CONUP setup initialization from the subroutine READAT. This file can be updated to reflect improved nuclear data; however, this step requires that all concentration files be updated for consistency.

Future code improvements to CONUP could include automating the construction of the cycle database files and utilizing a standard database management system for data management and retrieval. These improvements would minimize the data entry required for initial cycle file setup.

4.4 CYCLE PROCESSING

In Section 4.1, the process of setting up and initializing the CONUP input files was defined with parameter specification and format. This section deals with the process of actually running the CONUP code. The two modes of running the CONUP code are absolute and incremental. If the user selects the incremental mode, it is assumed that all prior cycle concentration files have been created and act as a cycle assembly concentration database for incremental processing. Otherwise, CONUP will display an error that it cannot open a previous concentration for assembly initialization.

To execute CONUP, the user must enter the fully qualified name of the CONUP execution module at the UNIX prompt. An example is shown as follows:

```
phys%> conup3/bin/conup.x
```

The bold text shown in the example represents the user entry at the UNIX prompt. Because the CONUP executable is located in subdirectory conup3/bin, the fully qualified entry is as shown. The first item CONUP prints out is a

program heading label indicating the current version and run date. At this time, the user is prompted for processing mode (see the example in Appendix A).

4.4.1 Absolute Processing

If the absolute processing mode is selected, the program prompts the user for current cycle ID, cycle EFPDs, and prior cycle ID. At this time, the program CONUP displays all input data and the user is requested to accept these data as input or reenter if desired. The CONUP program first initializes the concentration file from the tag gas ratios file and the given cycle configuration file. Then CONUP continues to process the current cycle assemblies over all prior load cycles for each assembly. When processing is complete, CONUP prints out that the current concentration file is created. An example of absolute cycle processing is given in Appendix A. This new concentration file is now ready for tag identification analysis to be performed by the program SMTAG. The new concentration files are stored in subdirectory conup3/concen and can be used for future incremental cycle processing.

4.4.2 Incremental Processing

In the incremental processing mode, CONUP requests the same input as it requests in the absolute processing mode. Upon completion of data input, CONUP creates the initial concentration file from prior cycle concentration files and only processes and burns the current cycle assemblies. An example of incremental cycle processing is included in Appendix A. Incremental cycle processing is considerably faster because only the current cycle is processed.

5.0 REFERENCE

Schmittroth, F., 1989, *SMTAG - A Code for the Sequential Analysis of Multiple Tag Gas Releases*, WHC-EP-0190, Westinghouse Hanford Company, Richland, Washington.

This page intentionally left blank.

APPENDIX A

CYCLE PROCESSING MODES

This appendix contains examples of actual CONUP cycle processing. Both incremental and absolute processing examples are included.

This page intentionally left blank.

EXAMPLE OF INCREMENTAL CYCLE PROCESSING

```
**********************************************************
*  CONUP (Ver 3.0)  --  Date: 10/25/91  Time: 09:54:37  *
**********************************************************

Enter P Code [a=abs,i=incr]      : i
Enter Current Cycle ID [a6]      : 7A
Enter Current Cycle Time [EFPD] : 105.1
Enter Last Cycle ID [a6]         : 6D

Current Cycle: 7A      Last Cycle   : 6D      Time[EFPD]:  105.1

Included Burn Cycles are listed below:

ATP      CIRL     HPPT     8DAY     PWRMLT   LOWPWR
1A       1B       2A       2B       3        4
5A       5B       6A       6B       6C       6D
7A

Is this Correct [y/n]? y

Creating Initial Concentration File ...
Initial Concentration File Created  ...
Current Cycle = 7A       Processing ...
Concentration File = 7A     Created ...
```

EXAMPLE OF ABSOLUTE CYCLE PROCESSING

```
**********************************************************
*  CONUP (Ver 3.0)  --  Date: 10/25/91  Time: 09:26:05  *
**********************************************************

Enter P Code [a=abs,i=incr]      : a
Enter Current Cycle ID [a6]      : 6D
Enter Current Cycle Time [EFPD] : 14.3
Enter Last Cycle ID [a6]         : 6C

Current Cycle: 6D      Last Cycle   : 6C      Time[EFPD]:   14.3

Included Burn Cycles are listed below:

ATP      CIRL     HPPT     8DAY     PWRMLT   LOWPWR
1A       1B       2A       2B       3        4
5A       5B       6A       6B       6C       6D

Is this Correct [y/n]? y

Creating Initial Concentration File ...
Initial Concentration File Created  ...
Current Cycle = ATP      Processing ...
Current Cycle = CIRL     Processing ...
Current Cycle = HPPT     Processing ...
Current Cycle = 8DAY     Processing ...
Current Cycle = PWRMLT   Processing ...
Current Cycle = LOWPWR   Processing ...
Current Cycle = 1A       Processing ...
Current Cycle = 1B       Processing ...
Current Cycle = 2A       Processing ...
Current Cycle = 2B       Processing ...
Current Cycle = 3        Processing ...
Current Cycle = 4        Processing ...
Current Cycle = 5A       Processing ...
Current Cycle = 5B       Processing ...
Current Cycle = 6A       Processing ...
Current Cycle = 6B       Processing ...
Current Cycle = 6C       Processing ...
Current Cycle = 6D       Processing ...
Concentration File = 6D    Created ...
```

APPENDIX B

CONCENTRATION FILE

This appendix contains a description of CONUP's concentration file parameters, file format structure, and an example of a concentration file produced by CONUP.

The CONUP cycle concentration file contains the amounts for all the isotopes tracked by CONUP (see Table 1), and these amounts are used as input to the SMTAG program. The format for this file is rigid and must follow the structure given below. The amounts of isotopes are given in numbers of atoms.

The concentration file contains a title record, number of isotopes, isotope name list, and data for each assembly included in this file. The isotope-named values are provided in both the TG block and the FP block of data. The assembly data header record contains the six character assembly ID, the assembly ITYPE, number of tags in the assembly, and several other variables not used by SMTAG.

Record#	Parameter(s)	Format
1	Title Record	(A80)
2	Number of Isotopes(niso)	(10x,I5)
3-5	(Isoname(i),i=1,niso)	(9x,8(1x,A6))
6	Number of Assemblies	(10x,I5)
7	Assembly ID, ITYPE, #tags, other	(4x,A6,20x,2I4,15x,E13.5)
8-17	Tag ID, #Pins, (TAGvalue(i),i=1,niso), (FPvalue(i),i=1,niso)	(4x,A6,I5,5E13.6/(15x,5E13.6)) (15x,5E13.6)
:	:	:
:	:	:

Records 8 through 17 repeat for each tag within an assembly, and this group of records repeats for each assembly over the total number of assemblies in the concentration file. See the following partial listing of a concentration file for an example of this format structure.

This page intentionally left blank.

PARTIAL LISTING OF A CYCLE CONCENTRATION FILE

```
CONCENTRATION CYCLEID: 6D                    Run Date: 10/25/91  Time: 09:26:05
           21
         XE-124 XE-126 I -127 XE-128 XE-129 I -129 XE-130 XE-131
         XE-132 XE-133 XE-134 XE-136 KR-078 KR-080 KR-081 BR-081
         KR-082 KR-083 KR-084 KR-085 KR-086
           90
    16407            1201  IN   0   1   0.000000E+00 0.703946E+23  134.010/25/91
    7       217 0.167261E+18 0.913687E+17 0.000000E+00 0.908435E+18 0.935901E+19
                0.000000E+00 0.117534E+19 0.511292E+19 0.600810E+19 0.200548E+14
                0.220920E+19 0.185479E+19 0.167412E+18 0.734087E+18 0.700179E+15
                0.636605E+12 0.319323E+19 0.309097E+19 0.150912E+20 0.135488E+16
                0.460940E+19
                0.000000E+00 0.000000E+00 0.499030E+20 0.791707E+18 0.285076E+16
                0.151233E+21 0.209071E+19 0.390186E+21 0.586621E+21 0.430107E+20
                0.807989E+21 0.760168E+21 0.000000E+00 0.000000E+00 0.000000E+00
                0.141268E+20 0.179717E+18 0.342690E+20 0.573495E+20 0.138588E+20
                0.908303E+20
    16445            1202  IN   0   1   0.000000E+00 0.192335E+24  366.210/25/91
    34      217 0.346473E+18 0.167329E+18 0.000000E+00 0.103556E+19 0.895334E+19
                0.000000E+00 0.113304E+19 0.493407E+19 0.610281E+19 0.213658E+14
                0.226860E+19 0.193653E+19 0.357830E+18 0.138248E+19 0.377653E+16
                0.985030E+13 0.387986E+19 0.310273E+19 0.139697E+20 0.352002E+16
                0.418808E+19
                0.000000E+00 0.000000E+00 0.132692E+21 0.576757E+19 0.563923E+17
                0.405029E+21 0.138074E+20 0.975895E+21 0.165284E+22 0.446814E+20
                0.219376E+22 0.207693E+22 0.000000E+00 0.000000E+00 0.000000E+00
                0.380168E+20 0.106569E+19 0.919505E+20 0.158281E+21 0.365377E+20
                0.248823E+21
    16425            1301  IN   0   1   0.000000E+00 0.133912E+24  268.310/25/91
    50      217 0.340180E+18 0.163690E+18 0.000000E+00 0.102238E+19 0.894769E+19
                0.000000E+00 0.111995E+19 0.496409E+19 0.609871E+19 0.189587E+14
                0.228711E+19 0.193787E+19 0.445487E+18 0.141218E+19 0.251060E+16
                0.425798E+13 0.385999E+19 0.307862E+19 0.139026E+20 0.229933E+16
                0.418429E+19
                0.000000E+00 0.000000E+00 0.935940E+20 0.282875E+19 0.193141E+17
                0.284708E+21 0.693758E+19 0.708603E+21 0.113517E+22 0.417044E+20
                0.153157E+22 0.144607E+22 0.000000E+00 0.000000E+00 0.000000E+00
                0.266620E+20 0.551720E+18 0.645770E+20 0.109677E+21 0.258461E+20
                0.173026E+21
    533              1302  CR  10   1   0.000000E+00 0.117920E+24  256.710/25/91
    88      217 0.435436E+18 0.209445E+18 0.000000E+00 0.123230E+19 0.944536E+19
                0.000000E+00 0.106449E+19 0.469003E+19 0.575588E+19 0.120809E+14
                0.218877E+19 0.186127E+19 0.565590E+18 0.281772E+19 0.323669E+16
                0.354474E+13 0.811230E+19 0.360872E+19 0.920611E+19 0.984564E+15
                0.257346E+19
                0.000000E+00 0.000000E+00 0.000000E+00 0.000000E+00 0.000000E+00
                0.000000E+00 0.000000E+00 0.000000E+00 0.000000E+00 0.000000E+00
                0.000000E+00 0.000000E+00 0.000000E+00 0.000000E+00 0.000000E+00
                0.000000E+00 0.000000E+00 0.000000E+00 0.000000E+00 0.000000E+00
                0.000000E+00
```

This page intentionally left blank.

APPENDIX C

INPUT FILE SAMPLES

This appendix contains samples of input files used by CONUP. The format for these files is not free format, and all fields must adhere to a strict structure. These samples are listed to provide the reader with insight to the structure and format of the CONUP code.

This page intentionally left blank.

EXAMPLE OF CYCLE HISTORY FILE

FFTF POWER HISTORY FOR CONUP3 PROGRAM DATE: 10/25/91

CYCLE ID	STARTUP DATE	SHUTDOWN DATE	EXP. (mw)	CYCLE EFPD's	AS-BUILT BEGIN DB	AS-BUILT END DB	CYCLE COMMENTS
ATP		12/23/80	400.	9.0	ipa3d		3D As-Built DB's
CIRL		03/16/81	400.	4.7	nat3d		* = 3D
HPPT		09/19/81	400.	9.8	hpp3d		*
8DAY		11/23/81	400.	8.6	epr		also epr3d *
PWRMLT		12/18/81	400.	0.3			
LOWPWR	01/16/82	01/17/82	400.	0.0	lpcdb		
1A	04/14/82	05/24/82	400.	34.0	boc1db		BOC1DB *
1B	08/27/82	11/12/82	400.	67.9	cm140		CM140 * AB-1 REPL
2A	01/19/83	02/19/83	400.	26.6	cy2a		BOC2, CY2A *
2B	02/28/83	05/22/83	400.	73.9	cy2b		CY2B * INSERT D-9
3	07/07/83	10/23/83	400.	101.5	b3235		
4	01/02/84	04/23/84	400.	109.5	b4336		
5A	06/21/84	08/17/84	400.	53.6	boc5		RPL=REPLACE
5B	08/24/84	11/03/84	400.	69.1	boc5b		RPL DE9 LKR, AAD6
6A	12/26/84	01/26/85	400.	28.9	b6569		
6B	02/21/85	04/21/85	400.	53.8	m6598		RPL W8018(WF) FO1
6C	04/27/85	06/04/85	400.	37.0	m6652		RPL ACN-1 LKR
6D	06/08/85	06/24/85	400.	14.3	m6699		RPL MW-4(LKR)
7A	08/15/85	12/07/85	400.	105.1	b7703		
7B	12/09/85	12/13/85	400.	3.3	b7b808		RPL D9-2(LKR)
7C	12/16/85	01/03/86	400.	14.3	b7c811		RPL 16393LKR
8A	02/06/86	04/25/86	400.	54.9	b8a825		
8B	06/04/86	06/14/86	400.	6.1			TESTS REMOVED
8C	06/30/86	07/18/86	400.	2.0	b8b880		GEM TEST
9A1	09/11/86	10/13/86	291.	29.4	b9a888		
9A2	10/16/86	02/05/87	291.	108.3	s9a917		RPL PO-2(LKR)2066
9B	03/04/87	06/20/87	291.	106.4	b9b1026		
9C	07/02/87	10/10/87	291.	97.7	b9c1132		
10A1	11/18/87	01/26/88	291.	66.3	b10a1230		
10A2	02/01/88	02/23/88	291.	20.3	s10a1296		FSP-1 INSERTED
10A3	02/25/88	03/04/88	291.	5.8	s10a1317		RPL D1PRESS 16514
10A4	03/07/88	05/07/88	291.	59.4	s10a1322	e10A1382	RPL PO-1 LKR 2065
10B	06/10/88	10/17/88	291.	126.7	b10b1382	e10b1509	
10C1	11/10/88	01/08/89	291.	56.9	b10c1509		RPL MOTA-1F FSP-1
10C2	01/16/89	01/20/89	291.	2.9	s10c1566		RPL PO-4 LKR PO-5
10C3	01/23/89	03/13/89	291.	47.0	s10c1568		
11A1	05/03/89	05/16/89	291.	11.3	b11a1615		
11A2	05/21/89	06/02/89	291.	10.5	s11a1626		Insert MIP
11A3	06/14/89	09/19/89	291.	94.6	s11a1637		Take out MIP
11B1	01/04/90	04/08/90	291.	85.5	b11b1732.4		
11B2		10/30/90	291.	132.7			
11C		03/25/91	291.	81.5			
12A1		07/23/91	291.	52.8			
12A2		09/24/91	291.	48.6			

PARTIAL LISTING OF A CYCLE CONFIGURATION FILE

```
TITLE Cycle 6D Configuration Data
CYID  6D
DATE  06/07/90
TIME  10:00:13
SERIAL# LOCAN REMARKS
16407    1201
16445    1202
16425    1301
533      1302
16482    1303
16398    1304
16487    1401
16484    1402
2140     1403
16501    1404
16432    1405
90003    1406
8230     1501
2074     1503
548      1504
8252     1505
2068     1506
8196     1507
8219     1508
4194     1601
2070     1602
4208     1603
4198     1604
4175     1605
4112     1606
4184     1607
4192     1608
4104     1609
8211     1610
16411    2101
16523    2201
16497    2202
16457    2301
534      2302
16402    2303
16400    2304
16443    2401
16442    2402
16505    2403
2141     2404
16462    2405
16417    2406
8203     2501
538      2502
8255     2503
531      2504
8213     2505
8261     2506
4199     2605
```

SAMPLE NUCLEAR DATA FILE

NUCLEAR DATA 10/25/91

ISOTOPE CROSS SECTIONS BY TYPE
NTYPES = 3

TYPEID	XE-124	XE-126	I -127	XE-128	XE-129	I -129	XE-130	XE-131
	XE-132	XE-133	XE-134	XE-136	KR-078	KR-080	KR-081	BR-081
	KR-082	KR-083	KR-084	KR-085	KR-086			
CORE	1.0000	0.2000	0.4500	0.1540	0.4030	0.3300	0.1000	1.5000
	0.4000	2.0000	0.1100	0.0001	0.0900	0.1100	0.4000	0.2500
	0.1000	0.3000	0.0100	0.2990	0.0010			
PLENUM	20.000	1.4000	0.0020	0.5000	4.3000	0.0030	0.2000	11.100
	0.1300	1.5000	0.0500	0.0050	0.8400	2.1000	4.0000	0.0010
	2.7000	3.2000	0.2000	1.0000	0.0040			
MOTA	0.7800	0.4300	0.5000	0.2100	0.4000	0.6000	0.2000	0.3000
	0.1000	0.7000	0.0300	0.0030	0.3700	0.2200	0.9000	0.5000
	0.1000	0.2000	0.0130	0.6000	0.0030			

FISSION CROSS SECTIONS
MTYPES = 4

TYPEID		PU239	PU240	PU241	PU242	U 235	U 238
IN	0	0.17730E+01	0.47050E+00	0.22550E+01	0.34700E+00	0.17140E+01	0.05970E+00
OT	0	0.17730E+01	0.47050E+00	0.22550E+01	0.34700E+00	0.17140E+01	0.05970E+00
CR	10	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
MT	20	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00

FLUX TYPES
KTYPES = 2
CORE
PLENUM

CHAIN YIELDS PER 100 FISSIONS ENDF/B-6 VER J 4/19/89							I	DI	Half-Life
PU239	PU240	PU241	PU242	U235	U238		-	--	----------
0.0	0.0	0.0	0.0	0.0	0.0	XE-124	1	0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	XE-126	2	0	0.0
.5172	.4211	.3318	.2911	.3078	.1355	I -127	3	4	0.0
0.0	0.0	0.0	0.0	0.0	0.0	XE-128	4	5	0.0
0.0	0.0	0.0	0.0	0.0	0.0	XE-129	5	7	0.0
1.500	1.038	1.166	1.292	.9184	.9454	I -129	6	7	.5844E+10
.0036	.0020	.0001	.0001	.0000	.0000	XE-130	7	8	0.0
3.875	3.777	3.364	3.167	3.220	3.282	XE-131	8	9	0.0
5.317	5.577	4.711	4.484	4.665	5.155	XE-132	9	10	0.0
6.971	6.889	6.462	6.603	6.720	6.772	XE-133	10	11	5.25
7.379	7.865	7.550	7.397	7.662	7.675	XE-134	11	0	0.0
7.051	6.763	6.691	6.850	6.234	6.946	XE-136	12	0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	KR-078	13	0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	KR-080	14	15	0.0
0.0	0.0	0.0	0.0	0.0	0.0	KR-081	15	16	.767025E+08
.1350	.1282	.09239	.08734	.2145	.1084	BR-081	16	17	0.0
.0006	.0003	.0001	.0001	.0002	.0000	KR-082	17	18	0.0
.3131	.2770	.2035	.2090	.5771	.3929	KR-083	18	19	0.0
.4954	.3907	.3568	.3147	1.026	.8203	KR-084	19	20	0.0
.12896	.1040	.08528	.08528	.2808	.15392	KR-085	20	21	.391548E+04
.7838	.7031	.5992	.5909	1.943	1.286	KR-086	21	0	0.0

PARTIAL LISTING OF ASSEMBLY CONFIGURATION FILE

ASSEMBLY DATA CONFIGURATION FILE							08/22/91
Serial#	#Tags	Type	Load-Date	Remarks			
	Tag#	TagID	#Pins	Xsec-Types		Flux-Types	
				Tag	FP	Tag	FP
xxxxxx	xxx	xxxxxx	xx/xx/xx	xxxxxxxxxxxxxxxxxxxxxxxxxxxxxx			
	xxx	xxxxxx	xxx	xxxxxx	xxxxxx	xxxxxx	xxxxxx
90003	25	MT 20		MOTA-1C			
	1	M5	5	MOTA	CORE	CORE	CORE
	2	M6	5	MOTA	CORE	CORE	CORE
	3	M7	5	MOTA	CORE	CORE	CORE
	4	M16	5	MOTA	CORE	CORE	CORE
	5	M17	5	MOTA	CORE	CORE	CORE
	6	M19	5	MOTA	CORE	CORE	CORE
	7	M20	5	MOTA	CORE	CORE	CORE
	8	M21	5	MOTA	CORE	CORE	CORE
	9	M31	5	MOTA	CORE	CORE	CORE
	10	M33	5	MOTA	CORE	CORE	CORE
	11	M35	5	MOTA	CORE	CORE	CORE
	12	M37	5	MOTA	CORE	CORE	CORE
	13	M39	5	MOTA	CORE	CORE	CORE
	14	M41	5	MOTA	CORE	CORE	CORE
	15	N1	5	MOTA	CORE	CORE	CORE
	16	N3	5	MOTA	CORE	CORE	CORE
	17	N5	5	MOTA	CORE	CORE	CORE
	18	N20	5	MOTA	CORE	CORE	CORE
	19	N21	5	MOTA	CORE	CORE	CORE
	20	N22	5	MOTA	CORE	CORE	CORE
	21	N23	5	MOTA	CORE	CORE	CORE
	22	N24	5	MOTA	CORE	CORE	CORE
	23	N25	5	MOTA	CORE	CORE	CORE
	24	N27	5	MOTA	CORE	CORE	CORE
	25	N29	5	MOTA	CORE	CORE	CORE
16435	1	IN 0					
	1	66	217	PLENUM	CORE	PLENUM	CORE
16432	1	IN 0					
	1	30	217	PLENUM	CORE	PLENUM	CORE
16431	1	IN 0					
	1	226	217	PLENUM	CORE	PLENUM	CORE
16430	1	IN 0					
	1	218	217	PLENUM	CORE	PLENUM	CORE
16429	1	IN 0					
	1	248	217	PLENUM	CORE	PLENUM	CORE
16428	1	IN 0					
	1	20	217	PLENUM	CORE	PLENUM	CORE
16427	1	IN 0					
	1	242	217	PLENUM	CORE	PLENUM	CORE
16426	1	IN 0					
	1	42	217	PLENUM	CORE	PLENUM	CORE
16425	1	IN 0					
	1	50	217	PLENUM	CORE	PLENUM	CORE
16424	1	IN 0					
	1	66	217	PLENUM	CORE	PLENUM	CORE
16422	1	IN 0					
	1	8	217	PLENUM	CORE	PLENUM	CORE
16421	1	IN 0					
	1	215	217	PLENUM	CORE	PLENUM	CORE

PARTIAL LISTING OF TAG GAS RATIOS FILE

TAG GAS RATIOS FILE UPDATED: 02/06/91

390 Total # of Gas Tags

TagID	%Xe	%Kr	Amt(cc)				
	Xe124/129	Xe126/129	Xe128/129	Xe130/129	Xe131/129	Xe132/129	Xe134/129
	Xe136/129	Kr 78/84	Kr 80/84	Kr 82/84	Kr 83/84	Kr 86/84	
1	0.00	0.00	0.0				
	0.25200E-01	0.14600E-01	0.11090E+00	0.13100E+00	0.64930E+00	0.80770E+00	0.31030E+00
	0.26390E+00	0.49600E-01	0.15830E+00	0.28130E+00	0.21700E+00	0.30170E+00	
2	0.00	0.00	0.0				
	0.20000E-01	0.12200E-01	0.10710E+00	0.12690E+00	0.59680E+00	0.72700E+00	0.27680E+00
	0.23420E+00	0.72700E-01	0.12430E+00	0.23440E+00	0.20790E+00	0.30480E+00	
3	0.00	0.00	0.0				
	0.21300E-01	0.10000E-01	0.86600E-01	0.14650E+00	0.74000E+00	0.93000E+00	0.36050E+00
	0.30540E+00	0.16380E-01	0.91100E-01	0.34010E+00	0.24170E+00	0.30110E+00	
4	0.00	0.00	0.0				
	0.26400E-01	0.12200E-01	0.91900E-01	0.14020E+00	0.69200E+00	0.85900E+00	0.32960E+00
	0.27890E+00	0.17110E-01	0.93700E-01	0.34270E+00	0.24210E+00	0.30100E+00	
5	0.00	0.00	0.0				
	0.33450E-01	0.14970E-01	0.93700E-01	0.14520E+00	0.74200E+00	0.93400E+00	0.36280E+00
	0.30840E+00	0.16750E-01	0.92300E-01	0.34330E+00	0.24360E+00	0.29940E+00	
6	0.00	0.00	0.0				
	0.41180E-01	0.18100E-01	0.10320E+00	0.13610E+00	0.67800E+00	0.84900E+00	0.32810E+00
	0.27800E+00	0.17000E-01	0.93300E-01	0.33960E+00	0.24070E+00	0.30280E+00	
7	0.00	0.00	0.0				
	0.18000E-01	0.97500E-02	0.96900E-01	0.12340E+00	0.54800E+00	0.63800E+00	0.23560E+00
	0.19780E+00	0.11100E-01	0.48700E-01	0.21190E+00	0.20490E+00	0.30550E+00	
8	0.00	0.00	0.0				
	0.25400E-01	0.12500E-01	0.98900E-01	0.12720E+00	0.58200E+00	0.69400E+00	0.25940E+00
	0.21880E+00	0.10200E-01	0.44800E-01	0.20640E+00	0.20310E+00	0.30700E+00	
9	0.00	0.00	0.0				
	0.32100E-01	0.15200E-01	0.10390E+00	0.13090E+00	0.62700E+00	0.77000E+00	0.29620E+00
	0.25150E+00	0.10200E-01	0.45100E-01	0.20910E+00	0.20570E+00	0.30490E+00	
10	0.00	0.00	0.0				
	0.39400E-01	0.18600E-01	0.11460E+00	0.12110E+00	0.55400E+00	0.66500E+00	0.25410E+00
	0.21400E+00	0.10000E-01	0.44600E-01	0.20570E+00	0.20260E+00	0.30830E+00	
11	0.00	0.00	0.0				
	0.21380E-01	0.10150E-01	0.86000E-01	0.14450E+00	0.73700E+00	0.92700E+00	0.35670E+00
	0.30650E+00	0.89600E-02	0.50100E-01	0.23400E+00	0.21170E+00	0.30360E+00	
12	0.00	0.00	0.0				
	0.26960E-01	0.12530E-01	0.92300E-01	0.14010E+00	0.69600E+00	0.86700E+00	0.33370E+00
	0.28400E+00	0.96700E-02	0.53350E-01	0.24300E+00	0.21480E+00	0.30360E+00	
13	0.00	0.00	0.0				
	0.33600E-01	0.14790E-01	0.92900E-01	0.14410E+00	0.73400E+00	0.93300E+00	0.36170E+00
	0.30840E+00	0.90100E-02	0.50600E-01	0.23460E+00	0.21180E+00	0.30430E+00	

PARTIAL LISTING OF A CYCLE FISSIONABLE AMOUNTS FILE

```
TITLE Cycle 6D     Atom Densities
CYID  6D
DATE  11/29/90
TIME  13:24:52
DECY  05/31/90
SN    16407
GAST  7
PU239  0.15628E-02 0.13416E-02
PU240  0.21172E-03 0.26140E-03
PU241  0.21020E-04 0.19695E-04
PU242  0.35500E-04 0.42000E-05
U 235  0.10880E-04 0.84000E-05
U 238  0.53581E-02 0.51682E-02
SN    16445
GAST  34
PU239  0.15628E-02 0.13416E-02
PU240  0.21172E-03 0.26140E-03
PU241  0.21020E-04 0.19695E-04
PU242  0.35500E-04 0.42000E-05
U 235  0.10880E-04 0.84000E-05
U 238  0.53581E-02 0.51682E-02
SN    16425
GAST  50
PU239  0.15628E-02 0.13416E-02
PU240  0.21172E-03 0.26140E-03
PU241  0.21020E-04 0.19695E-04
PU242  0.35500E-04 0.42000E-05
U 235  0.10880E-04 0.84000E-05
U 238  0.53581E-02 0.51682E-02
SN    533
GAST  88
PU239  0.15628E-02 0.13416E-02
PU240  0.21172E-03 0.26140E-03
PU241  0.21020E-04 0.19695E-04
PU242  0.35500E-04 0.42000E-05
U 235  0.10880E-04 0.84000E-05
U 238  0.53581E-02 0.51682E-02
SN    16482
GAST  38
PU239  0.15628E-02 0.13416E-02
PU240  0.21172E-03 0.26140E-03
PU241  0.21020E-04 0.19695E-04
PU242  0.35500E-04 0.42000E-05
U 235  0.10880E-04 0.84000E-05
U 238  0.53581E-02 0.51682E-02
SN    16398
GAST  58
PU239  0.15628E-02 0.13416E-02
PU240  0.21172E-03 0.26140E-03
PU241  0.21020E-04 0.19695E-04
PU242  0.35500E-04 0.42000E-05
U 235  0.10880E-04 0.84000E-05
U 238  0.53581E-02 0.51682E-02
```

PARTIAL LISTING OF A CYCLE FLUX FILE

CYCLEID = 6D FLUX DATA UPDATED: 06/19/90

LOCAN	CORE FLUX	PLENUM FLUX
1201	6.08025E+15	3.92475E+13
1202	6.07887E+15	4.11643E+13
1301	5.77678E+15	3.65500E+13
1302	5.31675E+15	2.46758E+13
1303	5.59553E+15	3.62889E+13
1304	5.43224E+15	3.90358E+13
1401	4.86191E+15	2.87735E+13
1402	4.25820E+15	2.30146E+13
1403	4.73940E+15	2.83862E+13
1404	4.96388E+15	3.45771E+13
1405	4.73636E+15	3.54063E+13
1406	4.90591E+15	3.56776E+13
1501	4.20696E+15	2.82170E+13
1502	3.49262E+15	1.85915E+13
1503	3.16981E+15	1.86415E+13
1504	3.42839E+15	1.83969E+13
1505	4.03911E+15	2.75151E+13
1506	4.17791E+15	3.07287E+13
1507	3.87721E+15	2.97770E+13
1508	4.21265E+15	3.13115E+13
1601	3.07543E+15	2.14907E+13
1602	2.66776E+15	1.76703E+13
1603	2.24919E+15	1.61361E+13
1604	2.66653E+15	1.76732E+13
1605	3.01984E+15	2.11867E+13
1606	3.29153E+15	2.51970E+13
1607	3.20587E+15	2.59312E+13
1608	2.75637E+15	2.36834E+13
1609	3.22561E+15	2.60706E+13
1610	3.33293E+15	2.56014E+13
1701	2.23216E+15	1.74169E+13
1702	2.00412E+15	1.49102E+13
1703	1.68753E+15	1.26609E+13
1704	1.28896E+15	1.01203E+13
1705	1.71721E+15	1.28746E+13
1706	2.02418E+15	1.50957E+13
1707	2.23324E+15	1.74472E+13
1708	2.26615E+15	1.85741E+13
1709	2.04513E+15	1.75849E+13
1710	1.57582E+15	1.41400E+13
1711	2.04863E+15	1.76173E+13
1712	2.28260E+15	1.87058E+13
1801	1.12925E+15	9.13364E+12
1802	9.80254E+14	7.83074E+12
1803	7.69300E+14	6.36493E+12
1804	5.29529E+14	4.70208E+12
1805	8.24207E+14	6.92785E+12
1806	1.05689E+15	8.65596E+12
1807	1.20817E+15	1.00344E+13
1808	1.26520E+15	1.08465E+13
1809	1.19458E+15	1.06228E+13
1810	9.78960E+14	9.14731E+12

This page intentionally left blank.

DISTRIBUTION

Number of copies

OFFSITE

2 Argonne National Laboratory East
9700 South Cass Avenue
Argonne, IL 60439

K. Gross
J. B. Lambert

1 Rockwell International
6623 Canoga Ave.
Canoga Park, CA 91303

V. Kesheshian

France

1 PHENIX
J. P. Coulon
Bolte Postale 171
30200 BAGNOLS-sur-CEZE
France

Japan

Power Reactor and Nuclear Fuel Development Corporation

2 Hiroshi Hara
SAN-KAI-DOH Building
9-13, 1-Chome, Akasaka,
Minato-Ku, Tokyo, Japan

2 Toshiba Corporation

Tak. Sudo
8 Shinsugita Isogoku Yokohama
Japan

Yuji Sano
4-1, Ukishima-Cho, Kawasaki-Ku,
Kawasaki, 210, Japan

DISTRIBUTION (cont)

Number of copies

ONSITE

3	U.S. Department of Energy Field Office, Richland	
	R. A. Almquist	A6-55
	O. A. Farabee	A6-55
	Reading Room	A1-65
2	Pacific Northwest Laboratory	
	M. W. Goheen	P7-22
	Technical Files	K1-11
37	Westinghouse Hanford Company	
	R. B. Baker	L5-01
	L. R. Campbell	N2-32
	D. C. Corrigan	N1-70
	J. W. Daughtry (2)	N2-32
	K. D. Dobbin	N3-32
	M. R. Gilbert	N2-32
	J. Greenborg	H0-35
	D. J. Hammervold (5)	H0-36
	S. F. Kessler	N2-32
	M. K. Korenko	L5-56
	D. M. Lucoff	N1-36
	J. A. Rawlins	H0-36
	R. F. Richard	N2-32
	R. E. Schenter (2)	H0-36
	F. Schmittroth (5)	H0-36
	E. R. Siciliano (5)	H0-36
	E. L. Tang	N2-32
	A. E. Waltar	H0-32
	D. W. Wootan	H0-35
	Central Files	L8-04
	Information Release Administration (3)	H4-17