TECHNICAL REPORT
IGE–259 R1

AN HISTORY INTERFACE BETWEEN DRAGON AND DONJON:

The HST: module

G. MARLEAU AND E. VARIN

Institut de génie nucléaire
Département de génie physique
École Polytechnique de Montréal
Décembre 2004

CONTENTS

LIST OF TABLES

1 THE DRAGON/DONJON HISTORY INTERFACE MODULE

The history interface between the codes DRAGON[1] and DONJON[2] has been written as a new module in order to facilitate the access to the GANLIB utilities that manage the required hierarchical data structures. The resulting `HST:` module can be called both by DRAGON and DONJON.

The reactor model we will consider as an example is a 3–D model with an $x = 3, y = 3$ and $z = 3$ mesh. Here we will assume that the $x - y$ plane describes fuel channels. The z plane will be associated with the so-called fuel bundles. This choice is somewhat arbitrary, however it is useful if the refueling takes place in a specific direction as in a CANDU reactor. Here, a 2-bundle shift fueling strategy will be considered. To each fresh fuel cell introduced in the core the `HST:` module will associate a unique cell number between 1 and N_c, the maximum number of cells in the reactor. Most of the information associated with the fresh fuel cells will be extracted from a DRAGON BURNUP file or defined using variable local parameters.[3] Each fresh fuel cell inserted in the core will also be associated with a specific fuel type. Each fuel type is defined as a unique initial fuel composition.

The fuel management for the reactor, including burnup and refueling will be performed by the DONJON code. Here the HST: will interact with this code via the MAP data structure.[4] Typically, each cell in the reactor will be burned inside DRAGON using the power provided in the `POWER-BUN` record and the depletion time provided in the `BURNUP-DEB` or `DEPL-TIME` records stored in the MAP structure. When refueling takes place some of the fuel cells will be extracted, other will be displaced from one position to another and finally new fresh fuel cells inserted. The fresh fuel cells properties will be extracted from the fuel types properties available on the HISTORY data structure.

In a coupled DRAGON/DONJON execution, the `HST:` module will be called at various points and for various reasons. The first call to `HST:` can be performed using:

```
MODULE       HST:   ;
*----
*  MAP data structure for initialization: MAP0
*  History data structure               : History
*----
SEQ_ASCII    MAP0 ;
XSM_FILE     History            ;
XSM_FILE     Reseau             ;
*----
*  Reactor parameters
*  ncha = nunber of channels = 9
*  nbun = nunber of bundles = 3
*  nevo = nunber of evolution = 3
*  nglo = nunber of global parameters = 1
*  nloc = nunber of local parameters = 2
*  bunl = bundle length in cm = 49.53 cm
*----
INTEGER  ncha    nbun    nevo  nglo    nloc :=
         9       3       3     1       2    ;
REAL     bunl           := 49.53            ;
*----
*  Initialize History using MAP0
*----
Reseau := MAP0                                ;
History := HST: Reseau                        ::
  DIMENSIONS GLOBAL <<nglo>> LOCAL <<nloc>>
             BUNDLES <<nbun>> <<bunl>>
             CHANNELS <<ncha>>                ;
```

Here, the HISTORY data structure will be stored in the XSM file `History`. One global and two local parameters are considered. No information about the name or the value of the global and local parameters will be available. This initialization procedure stores information only on the main level of the HISTORY data structure if the MAP data structure is not available. In this case the HISTORY is updated using a MAP data structure (in sequential ASCII file MAP0). The number of channels and bundles per channel are stored and compared with the same information in the MAP structure. For each bundle in the MAP, cell type and fuel type directories are constructed. The bundle powers and burnups available in MAP are used to generate the power rates in kW/kg and the depletion time in days required to reach the specified burnups. These values are stored in the HISTORY in the `PARAMBURNTAR` record. The fuel mass is mandatory for such calculation, thus the fuel wieght is recovered from the MAP. If the HISTPORY is in modification mode, the fuel weight is computed using the bundle lenght and the initial fuel density.

Now, let us assume that a DRAGON calculation was performed for the cell located in bundle j=1 of channel i=1. We will also assume that these cells contain a single type of fuel. Here the moderator temperature `TMod` is a global parameter while the fuel (`TComb`) and coolant (`TCalo`) temperatures are considered local parameters. We assume that after the cell flux calculation a BURNUP data structure was generated using the following instructions:

```
*----
*  Procedures for cell calculation: CellCalc
*----
PROCEDURE    CellCalc                ;
*----
*  Global parameter: Tmod for moderator temperature
*  Local parameters: TComb for fuel temperature
*                    TCalo for coolant temperature
*----
REAL   TMod         := 345.66         ;
REAL   TComb TCalo  := 941.29 560.66  ;
*----
*  Initial burnup options for cell calculation
*----
REAL   Power DeltaT := 31.9713  5.0   ;
*----
*  Local data structures
*----
LINKED_LIST  Burnup Edition           ;
*----
*  Execution control parameters
*  icha = channel number = 1
*  ibun = bundle number = 1
*----
INTEGER  icha    ibun  := 1 1         ;
*----
*  Perform cell calculation
*----
Burnup Edition := CellCalc Burnup ::
 <<TComb>> <<TCalo>> <<TMod>>
 <<Power>> <<DeltaT>>                 ;
```

Then, assuming that the history structure `HistXSM` was created using the options above, we can use

```
*----
*  Update history structure
*----
History := HST: History Burnup        ::
```

```
  GET TMod <<TMod>>
  CELLID <<icha>> <<ibun>>
  GET TComb <<TComb>> TCalo <<TCalo>> ;
```

where no *idfuel* is given (see Table 2), thus we have used the default value for *idfuel*=1 to store in HISTORY the general information associated with fuel channel 1 and bundle 1. Here, the initial properties associated with fuel type 1 will be generated from the initial isotope densities in the BURNUP. For the `CELLID`, here `icha`=1 and `ibun`=1, the bunrup information, isotope densities, depletion parameters and initial fuel density are stored in a {/celldir/} directory. Moreover the power rate 31.9713 kW/kg and the depletion time 5.0 days are kept in the `PARAMBURNTAR` record.

A HISTORY data structure that contains the initial cell information can be updated using a MAP data structure:

```
*----
*  MAP data structure for refueling: MAP1
*----
SEQ_ASCII    MAP1 ;
*----
*  Refuel
*----
Reseau := MAP1                  ;
History := HST: History Reseau  ;
```

Here, new burnup power ratings will be stored in the HISTORY data structure reflecting the power distribution in the DONJON calculation. The refueling information available in the MAP structure will also be used to redistribute the fuel in the HISTORY structure at various cell location.

Finally the last option is to recover this information in DRAGON to perform a new series of cell calculations:

```
*----
*  Local parameters
*  Initial burnup options for cell calculation
*  *A is after refueling
*  *B is before refueling
*----
REAL   TCombA  TCaloA  TCombB  TCaloB  ;
REAL   PowerA  DeltaTA PowerB  DeltaTB ;
Burnup := HST: History                        ::
  PUT TMod >>TMod<<
  CELLID <<icha>> <<ibun>>
  PUT BREFL  BURN >>DeltaTB<< >>PowerB<<
             TComb >>TCombB<< TCalo >>TCaloB<<
      AREFL  BURN >>DeltaTA<< >>PowerA<<
             TComb >>TCombA<< TCalo >>TCaloA<< ;
IF DeltaTB 0.0 > THEN
*----
*  Burn before refueling
*----
  Burnup Edition := CellCalc Burnup ::
    <<TCombB>> <<TCaloB>> <<TMod>>
    <<PowerB>> <<DeltaTB>>                    ;
  Edition := DELETE: Edition                  ;
ENDIF ;
*----
*  Burn after refueling
*----
```

```
Burnup Edition := CellCalc Burnup ::
  <<TCombA>> <<TCaloA>> <<TMod>>
  <<PowerA>> <<DeltaTA>>                        ;
*----
*  Update History
*----
History := HST: History Burnup                  ::
  CELLID <<icha>> <<ibun>>                       ;
```

Note that here, there are two sets of local parameters that can be extracted from the history data structure, namely the before (`BREFL`) and the after (`AREFL`) refueling information. In the case of fresh fuel (single fuel description or a refueled bundle) extracting the before information is not required. However, if one uses the general procedure described above to extract the before and after information, one will be able to identify the new fuel bundles as well as the bundle that have not been moved in the core by the fact that $\Delta t = 0$ for burnup before refueling. For bundles that have been displaced in the core during refueling then $\Delta t > 0$.

2 USERS GUIDE FOR THE HST: MODULE

The `HST:` module has been designed to manage a full reactor execution in DONJON using explicit DRAGON calculations for each cell. This module saves in an HISTORY data structure the information available in BURNUP data structures generated by DRAGON. It can also read MAP data structure generated by DONJON to prepare the HISTORY data structure for a new series of cell calculations in DRAGON. The HISTORY data structure can also be used to update the MAP data structure. Finally, the module `HST:` can be used to create an initial BURNUP data structure that can be used to evolve the cell another time step in DRAGON.

The `HST:` module can be used to create or update an HISTORY data structure. The possible options are:

Table 1: Updating an HISTORY structure using a MAP structure

HISTORY `:=` `HST:` [*HISTORY*] *MAP* [`::` [**(hstdim)**] [`GET` **(hstpar)**]] `;`

Table 2: Updating an HISTORY structure using a BURNUP structure

HISTORY `:=` `HST:` [*HISTORY*] [*BURNUP*] [`::` [**(hstdim)**]
[`GET` **(hstpar)**] [`CELLID` *icha ibun* [*idfuel*] [`GET` **(hstpar)**]]] `;`

It can also be used to create a BURNUP data structure from the information available on an HISTORY:

Table 3: Updating a BURNUP structure using an HISTORY structure

BURNUP `:=` `HST:` *HISTORY* [`::` [**(hstdim)**]
[`PUT` **(hstpar)**]
`CELLID` *icha ibun*
[`PUT` { `BREFL` **(hstbrn) (hstpar)** `AREFL` **(hstbrn) (hstpar)** | [`AREFL`] **(hstbrn) (hstpar)** }]] `;`

It can also be used to update a MAP data structure from the information available on anHISTORY:

Table 4: Updating an HISTORY structure using a MAP structure

MAP `:=` `HST:` *MAP HISTORY* `;`

where

HISTORY `character*12` name of an HISTORY data structure.

BURNUP `character*12` name of a BURNUP data structure.

MAP `character*12` name of a MAP data structure.

(hstdim) structure containing the dimensions for the HISTORY data structure.

`CELLID` keyword to identify the cell for which history information is to be processed.

icha channel number for which history information is to be processed.

ibun bundle number for which history information is to be processed.

idfuel fuel type number associated with this cell. One can associate to each fuel cell a different fuel type. By default a single fuel type is defined and it fills every fuel cell. Only the initial properties of each fuel type are saved. These properties are used for refueling.

`GET` keyword to specify that the values of the parameters selected in **(brnpar)** will be read from the input stream or CLE-2000 local variables and stored on the HISTORY data structure.

`PUT` keyword to specify that the values of the parameters selected in **(brnpar)** will be read from the HISTORY data structure and transferred to local CLE-2000 variables.

`BREFL` to specify that the information to extract from the HISTORY data base is related to the properties of the cell before refueling takes place.

`AREFL` to specify that the information to extract from the HISTORY data base is related to the properties of the cell after refueling took place.

(hstbrn) structure containing the burnup options.

(hstpar) structure containing the local parameters options.

The **(hstdim)** input structure is required for general dimensioning purpose. It is generally used only when creating the HISTORY data structure. However, the number of global and local parameters used in a HISTORY data structure can be increased at all time. The number of channels, bundles and the refueling scheme must be defined at the creation of the HISTORY data structure. This information can be provided manually or extracted from a MAP data structure. The general form of the **(hstdim)** input structure follows:

Table 5: Structure **(hstdim)**

[`EDIT` *iprint*]
[`DIMENSIONS` [`GLOBAL` *nglo*] [`LOCAL` *nloc*] [`BUNDLES` *nbun bunl*] [`CHANNELS` *ncha*]]

where

`EDIT` keyword used to modify the print level *iprint*.

iprint index used to control the printing in this module. It must be set to 0 if no printing on the output file is required.

`DIMENSIONS` keyword used to indicate that the general dimensioning of the HISTORY data structure will be modified.

`GLOBAL` keyword used to modify the number of global parameters on the HISTORY data structure.

nglo the number of global parameters. Note that the history module will use the maximum value between the current *nglob* and the value, if any, defined on the HISTORY data structure.

`LOCAL` keyword used to modify the number of local parameters on the HISTORY data structure.

nloc the number of local parameters. Note that the history module will use the maximum value between the current *nloc* and the value, if any, defined on the HISTORY data structure.

`BUNBLES` keyword used to specify the number of bundles per channels for the reactor model considered in the HISTORY data structure.

nbun the number of bundles per channels for the reactor model. Note that if *nbun* is different from the value already defined on the HISTORY data structure or the MAP data structure, the execution will be aborted.

bunl bundle length in cm. This information is required to compute inital fuel weight.

`CHANNELS` keyword used to specify the number of fuel channels for the reactor model considered in the HISTORY data structure.

ncha the number of fuel channels for the reactor model. Note that if *ncha* is different from the value already defined on the HISTORY data structure or the MAP data structure, the execution will be aborted.

The **(hstbrn)** serves a unique purpose, mainly to extract from the HISTORY file the information required to process a burnup evaluation in DRAGON using the `EVO:` module. The information must be stored inside CLE-2000 variables. The general form of this output structure is:

Table 6: Structure **(hstbrn)**

`BURN` *period power*

where

`BURN` keyword to indicate that burnup information follows.

period the burnup period (in days) that will be transferred to a real CLE-2000 variable.

power the power density (in kW/kg) that will be transferred to a real CLE-2000 variable.

The **(hstpar)** serves two purposes. First, it is used to define the names of the local and global parameters that may be used in our calculations as well as the values of these local parameters. In can also be used to extract from a HISTORY data structure the values of these parameters. The general form of this structure is:

Table 7: Structure **(hstpar)**

[[*NAMPAR valpar*]]

where

NAMPAR name of a local or global parameter to process. The parameters specified before the keyword *CELLID* is read will be considered global otherwise they will be considered local.

valpar real value for the local or global parameter to process. In the case where the `GET` option is activated, the history module will extract this parameter from the input data stream. In the case where the `PUT` option is activated, the history module will try to transfer this information into a real CLE-2000 variable.

3 CONTENTS OF THE HISTORY DATA STRUCTURE

This data structure contains the information required to ensure a smooth coupling of DRAGON with DONJON when an history based full reactor calculation is to be performed.

3.1 The Main /history/ Directory

On its first level, the following records and sub-directories will be found in the /history/ directory:

Table 8: Main records and sub-directories in /history/

Name	Type	Units	Comment
`SIGNATURE␣␣␣`	C∗12		Signature of the data structure (SIGNA).
`BUNDLELENGTH`	R(1)	cm	Bundle lenght L_z.
`STATE-VECTOR`	I(40)		Vector describing the various parameters associated with this data structure $\mathcal{S}_i^h$.
`NAMEGLOBAL␣␣`	C($\mathcal{S}_1^h$)∗12		Names $\mathcal{G}_j$ of global parameter j
`PARAMGLOBAL␣`	R($\mathcal{S}_1^h$)		Values G_j of global parameter j
`NAMELOCAL␣␣␣`	C($\mathcal{S}_2^h$)∗12		Names $\mathcal{L}_j$ of local parameter j
`CELLID␣␣␣␣␣␣`	I($\mathcal{S}_3^h, \mathcal{S}_4^h$)		Array containing an identification number $C_{i,j}$ associated with bundle i, channel j stored in the data structure.
`FUELID␣␣␣␣␣␣`	I($\mathcal{S}_3^h, \mathcal{S}_4^h$)		Array containing the fuel type $F_{i,j}$ associated with bundle i, channel j stored in the data structure.
{*fueldir/*}	Dir		List of $\mathcal{S}_{11}^h$ sub-directories with names $\mathcal{F}_{i,j}$ that contain the properties associated with the fuel type $F_{i,j}$.
{*celldir/*}	Dir		List of $\mathcal{S}_4^h \times \mathcal{S}_3^h$ sub-directories with names $\mathcal{C}_{i,j}$ that contain the properties associated with the cell $C_{i,j}$.

The signature variable for this data structure must be SIGNA=`L_HISTORY␣␣␣`. The dimensioning parameters stored in the state vector $\mathcal{S}_i^h$ represent:

- number of global parameters $N_g = \mathcal{S}_1^h$
- number of local parameters $N_l = \mathcal{S}_2^h$
- number of bundles per channel $N_b = \mathcal{S}_3^h = \mathcal{S}_1^M$
- number of channels in the core $N_c = \mathcal{S}_4^h = \mathcal{S}_2^M$
- type of depletion solution used $T_s = \mathcal{S}_6^h = \mathcal{S}_1^{\mathrm{BURNUP}}$
- type of burnup considered $T_b = \mathcal{S}_7^h = \mathcal{S}_2^{\mathrm{BURNUP}}$
- number of isotopes $N_I = \mathcal{S}_8^h = \mathcal{S}_4^{\mathrm{BURNUP}}$
- number of transport groups $G = \mathcal{S}_9^h = \mathcal{S}_5^{\mathrm{BURNUP}}$
- number of regions $N_r = \mathcal{S}_{10}^h = \mathcal{S}_6^{\mathrm{BURNUP}}$

where $\mathcal{S}_i^{\mathrm{BURNUP}}$ contains the information stored on the `STATE-VECTOR` for a DRAGON BURNUP structure[3] and $\mathcal{S}_i^M$ the information stored in `RDATA` for the MAP structure[4] of DONJON.

The fuel directory name $\mathcal{F}_{i,j}$ associated with fuel type $F_{i,j}$ is composed using the following FORTRAN instruction:

`WRITE(`$\mathcal{F}_{i,j}$`,'(A4,I8.8)') 'FUEL',`$F_{i,j}$

This directory will contain the initial isotopic content of this fuel type. The cell directory name $\mathcal{C}_{i,j}$ associated with $C_{i,j}$ is composed using the following FORTRAN instruction:

`WRITE(`$\mathcal{C}_{i,j}$`,'(A4,I8.8)') 'CELL',`$C_{i,j}$

This directory will contain the value of the local parameters associated with cell $C_{i,j}$ as well as the current isotopic content of this cell.

The identification number $C_{i,j}$ associated with channel j and bundle i can be seen as the serial number of the bundle located at a position in space identified by (i, j). It is automatically defined within the `HST:` module. For a fresh core $C_{i,j} = n$ where n represents the cell order definition in the input file. Upon refueling, some bundles in channel k of the core are displaced from region (l, k) to (m, k), new bundles are introduced at location (l, k) and old bundles removed from location (m, k). If one assumes that C^{NEW} and C^{OLD} represents the value of C after and before refueling then we will have:

$$C_{m.k}^{\mathrm{NEW}} = C_{l,k}^{\mathrm{OLD}}$$
$$C_{l,k}^{\mathrm{NEW}} = C_{m,k}^{\mathrm{FRESH}}$$

where $C_{m,k}^{\mathrm{FRESH}}$ represent a fresh fuel cell. The local parameters and burnup power density of the fuel cell previously located at (m, k) are preserved and the fresh fuel isotopic densities is that provided in $F_{m,k}$, the fuel type associated with $C_{m,k}^{\mathrm{FRESH}}$.

3.2 The Fuel Type Sub-directory

Each fuel sub-directory $\mathcal{F}_{i,j}$ contains the following information

Table 9: Fuel type sub-directory

Name	Type	Units	Comment
`FUELDEN-INIT`	R(2)		Vector giving the initial volumic density of heavy element in the fuel ρ_f (g/cm^3) and the initial linear density of heavy element in the fuel m_f (g/cm).
`ISOTOPESNAME`	C(N_I)*12		Name of isotopes used in this fuel type
`ISOTOPESMIXT`	I(N_I)		Mixture associated with each isotopes in this fuel type.
`ISOTOPESDENS`	R(N_I)	(cm b)$^{-1}$	Isotopic densities ρ_i for each of the isotopes.

3.3 The Cell Type Sub-directory

Each cell isotopic sub-directory $\mathcal{C}_{i,j}$ contains the following information

Table 10: Cell sub-directory

Name	Type	Units	Comment
FUELDEN-INIT	R(2)		Vector giving the initial volumic density of heavy element in the fuel ρ_f (g/cm^3) and the initial linear density of heavy element in the fuel m_f (g/cm).
PARAMLOCALBR	R(N_l)		Values V_l of local parameter L_l before refueling.
PARAMLOCALAR	R(N_l)		Values V_l of local parameter L_l after refueling.
PARAMBURNTBR	R(2)		Depletion time (T_1 in days) and power rate (P_1 in kW/kg) before refueling.
PARAMBURNTAR	R(2)		Depletion time (T_2 in days) and power rate (P_2 in kW/kg) after refueling.
DEPL-PARAM$\sqcup\sqcup$	R(3)		Vector describing the time step T (days), burnup B (kWd/kg) and irradiation w (n/kb) currently reached by this cell
ISOTOPESDENS	R(N_I)	(cm b)$^{-1}$	Isotopic densities ρ_i.

3.4 Records in the BURNUP Data Structure

The following records and sub-directories will be found in the /burnup/ data structure:[3]

Table 11: Main records and sub-directories in /burnup/

Name	Type	Units	Comment
SIGNATURE$\sqcup\sqcup\sqcup$	C$*$12		Signature of the data structure (SIGNA)
STATE-VECTOR	I(20)		Vector describing the various parameters associated with this data structure $\mathcal{S}_i^b$
EVOLUTION-R$\sqcup$	R(5)		Vector describing the various parameters associated with the burnup calculation options R_i
ISONAMES-EVO	C($\mathcal{S}_4^b$)$*$12		Name of isotopes.
ISOMIXT-EVO$\sqcup$	I($\mathcal{S}_4^b$)		Mixture associated with each isotopes.
FUELDEN-INIT	R(2)		Vector giving the initial volumic density of heavy element in the fuel ρ_f (g/cm^3) the initial linear density of heavy element in the fuel m_f (g/cm)
DEPL-TIMES$\sqcup\sqcup$	R($\mathcal{S}_3^b$)	10^8 s	Vector describing the various time steps at which burnup information has been saved T_i
{*/depldir/*}	Dir		List of $\mathcal{S}_3^b$ sub-directories which contain the properties associated with each burnup step T_i

The signature variable for this data structure must be SIGNA=L_BURNUP$\sqcup\sqcup\sqcup\sqcup$. The dimensioning parameters for this data structure, that are stored in the state vector $\mathcal{S}_i^b$, represents:

- The type of solution considered $I_s = \mathcal{S}_1^b$ where

$$I_s = \begin{cases} 1 & \text{Fifth order Runge–Kutta method} \\ 2 & \text{Forth order Kaps–Rentrop method} \end{cases}$$

- The type of burnup considered $I_t = \mathcal{S}_2^b$

$$I_t = \begin{cases} 0 & \text{Out of core or zero flux/power depletion} \\ 1 & \text{Constant flux depletion} \\ 2 & \text{Constant power depletion} \end{cases}$$

- Number of time steps for which burnup properties are present in this directory $N_t = \mathcal{S}_3^b$
- Number of isotopes $N_I = \mathcal{S}_4^b$
- Number of groups $G = \mathcal{S}_5^b$ where
- Number of regions $N_r = \mathcal{S}_6^b$ where

Note that the records `ISONAMES-EVO` and `ISOMIXT-EVO␣` appeared on this structure after revision 3.04G of DRAGON. A second observation is that the record `EVOLUTION-R` created by the `HST:` module corresponds to the default values for these parameters defined in DRAGON.

The list of directory {/depldir/} names DEPLDIR will be composed according to the following law. The first eight character (DEPLDIR`(1:8)`) will always be given by `DEPL-DAT`. The last four character (DEPLDIR`(9:12)`) represents time step saved. For the case where N_t time steps were saved we would use the following FORTRAN instructions to create the last four character of each of the directory names:

$$\texttt{WRITE}(\text{DEPLDIR}(\texttt{9:12}),'(\texttt{I4})')\ J$$

for $1 \leq J \leq N_t$ with the time stamp associated with each directory being given by T_J. For the case where $(N_t = 2)$, two such directory would be generated, namely

Table 12: Example of depletion directories

Name	Type	Units	Comment
`DEPL-DAT␣␣␣1`	Dir		Sub-directories which contain the information associated with time step 1
`DEPL-DAT␣␣␣2`	Dir		Sub-directories which contain the information associated with time step 2

Inside each depletion directory the following records and sub-directories will be found:

Table 13: Contents of a depletion sub-directory in /burnup/

Name	Type	Units	Comment
ISOTOPESDENS	R(N_I)	(cm b)$^{-1}$	Isotopic densities ρ_i for each of the isotopes described in the /microlib/ directory where the order of the isotopes is also specified
FLUX-NORM$_{\sqcup\sqcup\sqcup}$	R(1)		Flux normalization constant. It is zero for out of core depletion and represent the normalization of the flux ϕ_r^g that is used to ensure that the cell integrated flux or power is that required when fixed flux or power burnup is requested
DELTA$_{\sqcup\sqcup\sqcup\sqcup\sqcup\sqcup\sqcup}$	R(2)		Fuel burnup (MW d T^{-1}) and irradiation (Kb^{-1}) increments between the current time step and the preceding time step
BURNUP-IRRAD	R(2)		Fuel burnup (MW d T^{-1}) and irradiation (Kb^{-1}) reached at this time step

3.5 Records in the FUEL_MAP Data Structure

On its first level, the following records, found in the main /map/ directory, will be used by the HST: module:

Table 14: Records in /map/ used by HST:

Name	Type	Units	Comment
SIGNATURE$_{\sqcup\sqcup\sqcup}$	C$*$12		Signature of the data structure (SIGNA).
STATE-VECTOR	I(40)		Vector describing the various parameters associated with this data structure $\mathcal{S}_i^M$. This record may be absent from some data structures in which case the information stored in the record RES-DATA must be used.
WEIGHT$_{\sqcup\sqcup\sqcup\sqcup\sqcup\sqcup}$	R(1)	kg	Inital fuel bundle weight.
DEPL-TIME$_{\sqcup\sqcup\sqcup}$	R(1)	days	Next time step ΔT for burnup.
REFUEL-SCH$_{\sqcup\sqcup}$	I($\mathcal{S}_2^M$)		Vector S_i containing the refueling scheme used for each channel i.
REFUEL-CHAN$_{\sqcup}$	R($\mathcal{S}_2^M$)	days	Vector T_i containing the burnup time interval before refueling takes place. In the case where no refueling takes place for channel i a value of $T_i = 0.0$ days is provided. Otherwise $0.0 < T_i \leq \Delta T$ and the channel will be burned for a period T_i, then refueled and then burned for an additional period $\Delta T - T_1$.
POWER-BUN$_{\sqcup\sqcup\sqcup}$	R($\mathcal{S}_2^M, \mathcal{S}_1^M$)	kW	Bundle power $P_{i,j}$ associated with channel i, bundle j.
BURN-DEB$_{\sqcup\sqcup\sqcup}$	R($\mathcal{S}_2^M, \mathcal{S}_1^M$)	kWd/kg	Bundle burnups $B_{i,j}$ associated with channel i, bundle j.

The signature variable for this data structure must be SIGNA=L_MAP␣␣␣␣␣␣␣␣. The dimensioning parameters stored in the state vector $\mathcal{S}_i^M$ and used by the history module are:

- number of bundles per channel $\mathcal{S}_1^{\mathrm{M}}$
- number of channels in the core $\mathcal{S}_2^{\mathrm{M}}$

Here $\mathcal{S}_i^{\mathrm{M}}$ also represents the information stored in the record RES-DATA␣␣␣␣ for the MAP structure[4] of DONJON. Also note that the records REFUEL-CHAN␣ and DEPL-TIME␣␣␣ are not present in the current map structure. We will assume that in the case where these records are not provided, no refueling ever takes place.

4 DOCUMENTATION OF THE ROUTINES IN THE HST: MODULE

4.1 Structure of `HST:`

The `HST:` module can be represented by the following tree:

Structure of the history module: `HST`

```
HST
|------ HSTGDM
|------ HSTUHM
|        |------ HSTGMA
|        |------ HSTREF
|                 |------ HSTGSL
|                 |------ HSTGSD
|        |------ HSTGSL
|------ HSTUMH
|------ HSTGET
|        |------ HSTGSL
|------ HSTUHB
|        |------ HSTGSD
|------ HSTUBH
|        |------ HSTGSD
```

4.2 General Routines Description

4.2.1 HST

Purpose To extract from or save to a HISTORY data structure the information related to various cells in a reactor.

Syntax CALL HST(NENTRY, HENTRY, IENTRY, JENTRY, KENTRY)

Author(s) G. Marleau

Description of input parameters

NENTRY integer scalar variable containing the number of data structures transfered to this module.

HENTRY character*12 array HENTRY(NENTRY) containing the name of the data structures.

IENTRY integer array IENTRY(NENTRY) containing the data structure type where:

- IENTRY=1 for LCM memory object;
- IENTRY=2 for XSM file;
- IENTRY=3 for sequential binary file;
- IENTRY=4 for sequential ASCII file.

JENTRY integer array JENTRY(NENTRY) containing the access permission for the data structure where:

- JENTRY=0 for a data structure in creation mode;

- JENTRY=1 for a data structure in modifications mode;
- JENTRY=2 for a data structure in read-only mode.

KENTRY integer array KENTRY(NENTRY) containing the data structure pointer.

Called by

DRAGON routine(s) : DRAGON

4.2.2 HSTGDM

Purpose To read the editing level and general dimensioning parameters for the HISTORY data structure.

Syntax CALL HSTGDM(IPRINT, NGLO, NLOC, NCHA, NBUN, BUNLEN, ITYRED, CARRED)

Author(s) G. Marleau

Description of input/output parameters

IPRINT integer scalar variable containing the print level.

NGLO integer scalar variable containing the number of global parameters.

NLOC integer scalar variable containing the number of local parameters.

NCHA integer scalar variable containing the number of fuel channels.

NBUN integer scalar variable containing the number of bundles per channel.

BUNLEN real scalar variable containing the length (cm) of a bundle.

ITYRED integer scalar variable containing the type of the last variable read.

CARRED character*12 scalar variable containing the last character string read.

Called by

DRAGON routine(s) : HST

4.2.3 HSTGET

Purpose To read from the input file or send to CLE-2000 variables the local and burnup parameters associated with a fuel cell.

Syntax CALL HSTGET(IPHST, IPRINT, MAXG, MAXL, NCHA, NBUN, ITYPRO, ITYRED, CARRED, IUPDC, IUPDB, NAMG, PARAMG, NAML, PARAML, IDCELL, IDFUEL)

Author(s) G. Marleau

Description of input parameters

IPHST integer scalar variable containing the address of the HISTORY data structure.

IPRINT integer scalar variable containing the print level.

MAXG integer scalar variable containing the maximum number of global parameters.

MAXL integer scalar variable containing the maximum number of local parameters.

NCHA integer scalar variable containing the number of fuel channels.

NBUN integer scalar variable containing the number of bundles per channel.

ITYPRO integer scalar variable containing the type of processing where:

- ITYPRO $>$ 0 if the HISTORY is in creation or update mode;
- ITYPRO $<$ 0 if the HISTORY is in read-only mode.

ITYRED integer scalar variable containing the type of the last variable read.

CARRED character*12 scalar variable containing the last character string read.

Description of input/output parameters

NMAG integer array NAMG(3,0:MAXG) containing the global parameter names.

PARAMG real array PARAMG(0:MAXG) containing the values of the global parameters.

NMAG integer array NAML(3,0:MAXL) containing the local parameter names.

PARAML real array PARAML(0:MAXL) containing the values of the local parameters.

IDCELL integer array IDCELL(NBUN,NCHA) containing the cell identifier for each fuel bundle in each channel.

IDFUEL integer array IDCELL(NBUN,NCHA) containing the fuel type identifier for each fuel bundle in each channel.

Description of output parameters

IUPDC integer scalar variable containing the number of the channel to analyze.

IUPDB integer scalar variable containing the number of the bundle to analyze.

Called by

DRAGON routine(s) : HST

4.2.4 HSTUHM

Purpose To update the HISTORY data structure using the information provided on the MAP data structure.

Syntax CALL HSTUHM(IPHST, IPMAP, IPRINT, MAXL, NCHA, NBUN,
MAXI, POWER, BURNP, IREFUS, REFUT, BUNLEN,
IDCELL, IDFUEL, PARAML, DENI, ISHUFF)

Author(s) G. Marleau, E. Varin

Description of input parameters

IPHST integer scalar variable containing the address of the HISTORY data structure.

IPMAP integer scalar variable containing the address of the MAP data structure.

IPRINT integer scalar variable containing the print level.

MAXL integer scalar variable containing the maximum number of local parameters.

NCHA integer scalar variable containing the number of fuel channels.

NBUN integer scalar variable containing the number of bundles per channel.

MAXI integer scalar variable containing the maximum number of isotopes.

POWER real array POWER(NCHA,NBUN) containing the power for each fuel bundle in each channel.

BURNP real array BURNUP(NCHA,NBUN) containing the burnup for each fuel bundle in each channel.

IREFUS integer array IREFUS(NCHA) containing the refueling strategy for each channel. A channel is refueled using a NBS bundle shift procedure if IREFUS(I)=NBS. In the case where NBS > 0, bundles 1 to NBUN-NBS are displaced to position NBS+1 to NBUN while locations 1 to NBS are filled with new fuel. In the case where NBS < 0, bundles -NBS+1 to NBUN are displaced to position 1 to NBUN+NBS while locations NBUN+NBS+1 to NBUN are filled with new fuel.

REFUT real array REFUT(NCHA) containing the refueling time for each channel.

BUNLEN real scalar variable containing the length (cm) of a bundle.

Description of input/output parameters

IDCELL integer array IDCELL(NBUN,NCHA) containing the cell identifier for each fuel bundle in each channel.

IDFUEL integer array IDCELL(NBUN,NCHA) containing the fuel type identifier for each fuel bundle in each channel.

Description of work parameters

PARAML real array IDCELL(0:MAXL,2) containing the local parameters.

DENI real array DENI(0:MAXI) containing the isotopic concentrations.

ISHUFF integer array IDCELL(NBUN) containing the fuel shuffling index for a channel.

Called by

DRAGON routine(s) : HST

4.2.5 HSTGMA

Purpose To read from the MAP data structure the power and burnup distribution for each cell as well as the refueling option for each channel.

Syntax CALL HSTGMA(IPMAP, IPRINT, NCHA, NBUN, DELTAT, POWER, IREFUS, REFUT, NBFUEL)

Author(s) G. Marleau

Description of input parameters

IPMAP integer scalar variable containing the address of the MAP data structure.

IPRINT integer scalar variable containing the print level.

NCHA integer scalar variable containing the number of fuel channels.

NBUN integer scalar variable containing the number of bundles per channel.

Description of input/output parameters

DELTAT real scalar variable containing the last character string read.

POWER real array POWER(NCHA,NBUN) containing the burnup power for each fuel bundle in each channel.

IREFUS integer array IREFUS(NCHA) containing the refueling strategy for each channel. A channel is refueled using a NBS bundle shift procedure if IREFUS(I)=NBS. In the case where NBS > 0, bundles 1 to NBUN-NBS are displaced to position NBS+1 to NBUN while locations 1 to NBS are filled with new fuel. In the case where NBS < 0, bundles -NBS+1 to NBUN are displaced to position 1 to NBUN+NBS while locations NBUN+NBS+1 to NBUN are filled with new fuel.

REFUT real array REFUT(NCHA) containing the refueling time for each channel.

NBFUEL integer scalar variable containing the number of fueled channels.

Called by

DRAGON routine(s) : HSTUHM

4.2.6 HSTREF

Purpose To refuel the reactor according to the information provided in a MAP data structure.

Syntax CALL HSTREF(IPHST, IPRINT, MAXL, NCHA, NBUN, MAXI, NBFUEL, DELTAT, POWER, IREFUS, REFUT, IDCELL, IDFUEL, PARAML, DENI, ISHUFF)

Author(s) G. Marleau

Description of input parameters

IPHST integer scalar variable containing the address of the HISTORY data structure.

IPRINT integer scalar variable containing the print level.

MAXL integer scalar variable containing the maximum number of local parameters.

NCHA integer scalar variable containing the number of fuel channels.

NBUN integer scalar variable containing the number of bundles per channel.

MAXI integer scalar variable containing the maximum number of isotopes.

NBFUEL integer scalar variable containing the number of fueled channels.

DELTAT real scalar variable containing the last character string read.

POWER real array POWER(NCHA,NBUN) containing the burnup power for each fuel bundle in each channel.

IREFUS integer array IREFUS(NCHA) containing the refueling strategy for each channel. A channel is refueled using a NBS bundle shift procedure if IREFUS(I)=NBS. In the case where NBS > 0, bundles 1 to NBUN-NBS are displaced to position NBS+1 to NBUN while locations 1 to NBS are filled with new fuel. In the case where NBS < 0, bundles -NBS+1 to NBUN are displaced to position 1 to NBUN+NBS while locations NBUN+NBS+1 to NBUN are filled with new fuel.

REFUT real array REFUT(NCHA) containing the refueling time for each channel.

Description of input/output parameters

IDCELL integer array IDCELL(NBUN,NCHA) containing the cell identifier for each fuel bundle in each channel.

IDFUEL integer array IDCELL(NBUN,NCHA) containing the fuel type identifier for each fuel bundle in each channel.

Description of work parameters

PARAML real array IDCELL(0:MAXL,2) containing the local parameters.

DENI real array DENI(0:MAXI) containing the isotopic concentrations.

ISHUFF integer array IDCELL(NBUN) containing the fuel shuffling index for a channel.

Called by

DRAGON routine(s) : HSTUHM

4.2.7 HSTUMH

Purpose To update the MAP data structure using the information provided on the HISTORY data structure.

Syntax CALL HSTUMH(IPHST, IPMAP, IPRINT, NCHA, NBUN, IDCELL, BURNUP)

Author(s) G. Marleau, E. Varin

Description of input parameters

IPHST integer scalar variable containing the address of the HISTORY data structure.

IPMAP integer scalar variable containing the address of the MAP data structure.

IPRINT integer scalar variable containing the print level.

NCHA integer scalar variable containing the number of fuel channels.

NBUN integer scalar variable containing the number of bundles per channel.

IDCELL integer array IDCELL(NBUN,NCHA) containing the cell identifier for each fuel bundle in each channel.

Description of input/output parameters

IPMAP integer scalar variable containing the address of the MAP data structure.

Description of work parameters

BURNUP real array BURNUP(NCHA,NBUN) containing the burnup for each fuel bundle in each channel.

Called by

DRAGON routine(s) : HST

4.2.8 HSTUHB

Purpose To update the HISTORY data structure using the information provided on the BURNUP data structure.

Syntax CALL HSTUHB(IPHST, IPEVO, IPRINT, MAXI, NBBTS, NCHA, NBUN, IUPDC, IUPDB, IDCELL, IDFUEL, NAMIH, MIXIH, NAMIB, MIXIB, DENI, DEPLT, MAXL, PARAML)

Author(s) G. Marleau

Description of input parameters

IPHST integer scalar variable containing the address of the HISTORY data structure.

IPEVO integer scalar variable containing the address of the BURNUP data structure.

IPRINT integer scalar variable containing the print level.

MAXI integer scalar variable containing the maximum number of isotopes.

NBBTS integer scalar variable containing the number of depletion steps.

NCHA integer scalar variable containing the number of fuel channels.

NBUN integer scalar variable containing the number of bundles per channel.

IUPDC integer scalar variable containing the number of the channel to analyze.

IUPDB integer scalar variable containing the number of the bundle to analyze.

IDCELL integer array IDCELL(NBUN,NCHA) containing the cell identifier for each fuel bundle in each channel.

IDFUEL integer array IDCELL(NBUN,NCHA) containing the fuel type identifier for each fuel bundle in each channel.

Description of work parameters

NAMIH integer array NAMIH(3,0:MAXI) containing the name of isotopes on the HISTORY structure.

MIXIH integer array MIXIH(0:MAXI) containing the mixture number associated with the isotopes on the HISTORY structure.

NAMIB integer array NAMIB(3,0:MAXI) containing the name of isotopes on the BURNUP structure.

MIXIB integer array MIXIB(0:MAXI) containing the mixture number associated with the isotopes on the BURNUP structure.

DENI real array DENI(0:MAXI) containing the isotopic concentrations of the isotopes on the BURNUP or HISTORY structure.

DEPLT real array DEPLT(NBBTS) containing the time associated with each depletion step on the BURNUP structure.

MAXL integer scalar variable containing the maximum number of local parameters.

PARAML real array PARAML(0:MAXL) containing the values of the local parameters.

Called by

DRAGON routine(s) : HST

4.2.9 HSTUBH

Purpose To update the BURNUP data structure using the information provided on the HISTORY data structure.

Syntax CALL HSTUBH(IPEVO, IPHST, IPRINT, MAXI, NBBTS, NCHA, NBUN, IUPDC, IUPDB, IDCELL, IDFUEL, NAMIH, MIXIH, DENI, DEPLT)

Author(s) G. Marleau

Description of input parameters

IPEVO integer scalar variable containing the address of the BURNUP data structure.

IPHST integer scalar variable containing the address of the HISTORY data structure.

IPRINT integer scalar variable containing the print level.

MAXI integer scalar variable containing the maximum number of isotopes.

NBBTS integer scalar variable containing the number of depletion steps.

NCHA integer scalar variable containing the number of fuel channels.

NBUN integer scalar variable containing the number of bundles per channel.

IUPDC integer scalar variable containing the number of the channel to analyze.

IUPDB integer scalar variable containing the number of the bundle to analyze.

IDCELL integer array IDCELL(NBUN,NCHA) containing the cell identifier for each fuel bundle in each channel.

IDFUEL integer array IDCELL(NBUN,NCHA) containing the fuel type identifier for each fuel bundle in each channel.

Description of work parameters

NAMIH integer array NAMIH(3,0:MAXI) containing the name of isotopes on the HISTORY or BURNUP structure.

MIXIH integer array MIXIH(0:MAXI) containing the mixture number associated with the isotopes on the HISTORY or BURNUP structure.

DENI real array DENI(0:MAXI) containing the isotopic concentrations of the isotopes on the HISTORY or BURNUP structure.

DEPLT real array DEPLT(NBBTS) containing the time associated with each depletion step on the BURNUP structure.

Called by

DRAGON routine(s) : HST

4.2.10 HSTGSL

Purpose To read from or write to the HISTORY data structure the local parameters and burnup power.

Syntax CALL HSTGSL(IPHST, MAXL, IOK, TIMPOW, PARAML)

Author(s) G. Marleau

Description of input parameters

IPHST integer scalar variable containing the address of the HISTORY data structure.

MAXL integer scalar variable containing the maximum number of local parameters.

Description of input/output parameters

IOK integer scalar variable containing the processing option where:

- on input, a negative value indicates that the information is to be extracted from the HISTORY data structure and a positive value indicates that the information is to be stored on the HISTORY data structure (-1 and 1 for before refueling and -2, 2 for after refueling);
- on output, a value of 0 indicates that the required processing took place successfully while a negative value indicates a failure of the processing.

TIMPOW real array TIMPOW(2) containing the burnup time and power density.

PARAML real array IDCELL(0:MAXL) containing the local parameters.

Called by

DRAGON routine(s) : HSTGET, HSTREF

4.2.11 HSTGSD

Purpose To read from or write to the HISTORY data structure the isotopic and fuel densities.

Syntax CALL HSTGSD(IPHST, MAXI, IOK, DENI, FDEN)

Author(s) G. Marleau

Description of input parameters

IPHST integer scalar variable containing the address of the HISTORY data structure.

MAXI integer scalar variable containing the maximum number of isotopes.

Description of input/output parameters

IOK integer scalar variable containing the processing option where:

- on input, a negative value indicates that the information is to be extracted from the HISTORY data structure and a positive value indicates that the information is to be stored on the HISTORY data structure;

- on output, a value of 0 indicates that the required processing took place successfully while a negative value indicates a failure of the processing.

DENI real array DENI(0:MAXI) containing the isotopic concentration.

FDEN real array FDEN(2) containing the average fuel density and weight.

Called by

DRAGON routine(s) : HSTREF, HSTUHB, HSTUBH

REFERENCES

[1] G. Marleau, A. Hébert and R. Roy, *A User's Guide for DRAGON*, Report IGE-174 Rev. 5, École Polytechnique de Montréal, Institut de Génie Nucléaire (2000).

[2] E. Varin, A. Hébert, R. Roy and J. Koclas, *A User's Guide for DONJON*, Report IGE-208 Rev 2, École Polytechnique de Montréal, Institut de Génie Nucléaire (2004).

[3] A. Hébert, G. Marleau and R. Roy, *A Description of the DRAGON Data Structures*, Report IGE-232 Rev.3, École Polytechnique de Montréal, Institut de Génie Nucléaire (2000).

[4] E. Varin, A. Hébert *Linked list specifications for DONJON*, Report IGE-226 Rev. 2, École Polytechnique de Montréal, Institut de Génie Nucléaire (2004).