**User's Manual of**

**Reactor Monte Carlo Code RMC**



Reactor Engineering Analysis Laboratory (REAL)

Institute of Nuclear Energy Science and Engineering Management (INESEM)

Department of Engineering Physics (DEP), Tsinghua University

Beijing, P.R. China

October, 2013

**FOREWORD**

This manual is a practical guide for the use of Reactor Monte Carlo code RMC, which provides the detailed specifications for RMC input and options, numerous examples, and the general description of the output generated by RMC. The first chapter is a primer on basic RMC installation and running. The second chapter gives a general introduction of the input for the code. The third to eighth chapters provide information on preparing input blocks, including geometry, material, criticality calculation, tally, source acceleration, and burnup calculation. In the end, the ninth chapter explains the output, and the tenth chapter shows the geometry plotting of the code.

RMC (Reactor Monte Carlo code) is a 3-D Monte Carlo neutron transport code being developed by Reactor Engineering Analysis Laboratory (REAL) in Department of Engineering Physics, Tsinghua University. The code RMC intends to solve reactor analysis problems, and is able to deal with complex geometry, using continuous energy point cross sections of different materials and temperatures. RMC beta 1.0 and beta 2.0 have been released in 2012 and 2013 domestically in China mainland, respectively.

The development of RMC started in the year 2001, and has been put into more than 40 person-years of human efforts. The undergoing development has been through three main stages.

* **Stage 1 (2001~2008)** : Researching and developing methods and functions for reactor analysis based on existing Monte Carlo codes;
* **Stage 2 (2008~2009)** : Prototype RMC code was developed;
* **Stage 3 (2009~now)**: New methods research and new functions development based on RMC, and also the standardization of code development based on modern software engineering has been implemented for RMC.

RMC now has the following functions and special techniques: criticality calculation, burnup calculation, CPU parallelism capability, fix source calculationand *kinetics simulation*, *on-the-fly cross-sections processing with temperature*, high efficiency searching methods (for cross-sections and geometry), source convergence acceleration, *full-core hybrid calculation methods (RMMC method), data and domain decomposition, GPU parallelism, continuously varying medium simulation, perturbation and sensitivity analysis, criticality search, N-TH coupling* and so on. Parts of the functions and techniques *(in italic)* have NOT been included in this release.

RMC is coded in C++ language and can be executable for Windows PCs and Linux PCs. The data libraries with this release are processed by the code RXSP mainly from ENDF/B-VII.0.

RMC and this manual are the products of the combined efforts of many people in REAL team, Tsinghua University:

**Main Advisor**

|  |  |  |
| --- | --- | --- |
| Prof. Kan WANG |  |  |

**Monte Carlo Group**

|  |  |  |
| --- | --- | --- |
| Zeguang LI | Ding SHE | Qi XU |
| Jingang LIANG | Jiankai YU | Jialong SUN |
| Yishu QIU | Xiao FAN | Ouwen YEXIN |
| Shichang LIU | Zonghuan CHEN | Linsen LI |
| Peng ZHANG  Ganglin YU | Mancang LI | Gaochen WU |

**Cross-section Data Group**

|  |  |  |
| --- | --- | --- |
| Jiankai YU  Guanbo WANG | Wanlin LI | Songyang LI |

**Software Engineering Group**

|  |  |  |
| --- | --- | --- |
| Prof. Kan WANG  Jingang LIANG | Zeguang LI | Jialong SUN |

The RMC code and manual can be obtained from the Reactor Engineering Analysis Laboratory (REAL), Liuqing Building 906, Tsinghua University, Beijing, 100084, CHINA.

**References:**

[1] K. Wang, Z. Li, D. She, et al., “RMC – A Monte Carlo Code for Reactor Physics Analysis,” International Conference on Mathematics and Computational Methods Applied to Nuclear Science and Engineering, M and C 2013, v 1, p 89-104, Sun Valley, ID, USA, May 5-9, 2013.

[2] D. She, K. Wang, G. Yu, “Development of the point-depletion code DEPTH”. Nuclear Engineering and Design, May, 2013, 258: 235–240.

[3] Y. Qiu, D. She, X. Fan, K. Wang, et al., “3D Neutron Transport PWR Full-core Calculation with RMC code”. Joint International Conference on Supercomputing in Nuclear Applications and Monte Carlo 2013 (SNA + MC 2013), La Cité des Sciences et de l’Industrie, Paris, France, October 27-31, 2013.

[4] J. Yu, S. Li, K. Wang, et al., “The Development and Validation of Nuclear Cross Section Processing Code for Reactor-RXSP”. The 2013 21st INTERNATIONAL CONFERENCE ON NUCLEAR ENGINEERING, July 29-August 2, 2013, Chengdu, China.

[5] D. She, Y. Liu, K. Wang, et al., “Development of Burnup Methods and Capabilities in Monte Carlo Code RMC”. Annals of Nuclear Energy, January 2013, 51: 289–294.

[6] D. She, K. Wang, G. Yu, “Asymptotic Wielandt Method and Superhistory Method for Source Convergence in Monte Carlo Criticality Calculation,” Nuc. Sci. Eng., 172, pp. 127-137 (2012).

[7] Z. Li, K. Wang, “A Hybrid Monte Carlo and Response Matrix Monte Carlo Method in Criticality Calculation,” PHYSOR 2012, Knoxville, Tennessee, USA, April 15-20, 2012.

[8] J. Liang, Y. Cai, K. Wang, et al., “Implementation of Domain Decomposition and Data Decomposition Algorithms in RMC code”. Joint International Conference on Supercomputing in Nuclear Applications and Monte Carlo 2013 (SNA + MC 2013), La Cité des Sciences et de l’Industrie, Paris, France, October 27-31, 2013.

[9] Z. Li, K. Wang, X. Zhang. “Research on Applying Neutron Transport Monte Carlo Method in Materials with Continuously Varying Cross-sections,” M&C 2011, Rio de Janeiro, RJ, Brazil, May 8-12, 2011.

[10] L. Li, H. Yuan, K. Wang. “COUPLING OF RMC AND CFX FOR ANALYSIS OF PEBBLE BED-ADVANCED HIGH TEMPERATURE REACTOR CORE.” Nuclear Engineering and Design, 250: pp 385-391 (2012).

[11] Z. Li, K. Wang, J. Deng, “Implementation of Perturbation Based Monte Carlo Criticality Search in RMC,” 2013 ANS Winter Meeting, Washington, DC, Nov. 10-14, 2013.

**Keywords:**

MONTE CARLO; NEUTRON; REACTOR ANALYSIS; COMPLEX GEOMETRY; CROSS SECTIONS; COUPLED

Kan WANG

Professor, REAL Leader

<http://reallab.ep.tsinghua.edu.cn>

wangkan@tsinghua.edu.cn

Contents

[Chapter 1 Installation and Running of RMC 1](#_Toc369335171)

[1.1 Installation 1](#_Toc369335172)

[1.1.1 Versions classification 1](#_Toc369335173)

[1.1.2 Installation configuration 2](#_Toc369335174)

[1.2 Running RMC 3](#_Toc369335175)

[1.2.1 Serial running 3](#_Toc369335176)

[1.2.2 Parallel running 3](#_Toc369335177)

[1.3 Input/output test 4](#_Toc369335178)

[Chapter 2 Overview of RMC Input 5](#_Toc369335179)

[2.1 Input blocks 5](#_Toc369335180)

[2.2 Input cards 5](#_Toc369335181)

[2.3 Input format 7](#_Toc369335182)

[Chapter 3 Geometry 9](#_Toc369335183)

[3.1 Surface 9](#_Toc369335184)

[3.2 Cell 12](#_Toc369335185)

[3.2.1 Boolean definition for the surface of a cell 12](#_Toc369335186)

[3.2.2 Options of the cells information 14](#_Toc369335187)

[3.3 Universe 15](#_Toc369335188)

[3.3.1 Single-lever universe 15](#_Toc369335189)

[3.3.2 Multi-level universe 16](#_Toc369335190)

[3.3.3 Geometric transformation 16](#_Toc369335191)

[3.4 Lattice 18](#_Toc369335192)

[3.4.1 Rectangular lattice 19](#_Toc369335193)

[3.4.2 Hexagonal lattice 20](#_Toc369335194)

[3.5 Input examples of geometry blocks 22](#_Toc369335195)

[3.5.1 PWR assembly 22](#_Toc369335196)

[3.5.2 PWR reactor core 23](#_Toc369335197)

[3.5.4 Hexagonal assembly 25](#_Toc369335198)

[3.5.5 Hexagonal reactor core 27](#_Toc369335199)

[Chapter 4 Material 30](#_Toc369335200)

[4.1 Input card of material 30](#_Toc369335201)

[4.2 Input card for the continuous-energy ACE cross sections 31](#_Toc369335202)

[4.3 Input card for the multi-group ACE cross sections 32](#_Toc369335203)

[4.4 Input examples of the material block 32](#_Toc369335204)

[4.4.1 Material inputs with continuous-energy ACE libraries 32](#_Toc369335205)

[4.4.2 Material inputs with multi-group ACE libraries 33](#_Toc369335206)

[Chapter 5 Criticality Calculation 34](#_Toc369335207)

[5.1 Parameters of source iteration 34](#_Toc369335208)

[5.2 Initial fission source 35](#_Toc369335209)

[5.3 Random number generator 35](#_Toc369335210)

[5.4 Parallel critical calculation mode 36](#_Toc369335211)

[5.5 Input examples of the critical calculation block 37](#_Toc369335212)

[Chapter 6 Tally 38](#_Toc369335213)

[6.1 Cell tally 38](#_Toc369335214)

[6.1.1 Cell option 39](#_Toc369335215)

[6.1.2 Filter option 41](#_Toc369335216)

[6.1.3 Integral option 42](#_Toc369335217)

[6.2 Mesh tally 43](#_Toc369335218)

[6.3 Cross-section tally 44](#_Toc369335219)

[6.4 Tally acceleration 46](#_Toc369335220)

[6.5 Input examples of the tally block 47](#_Toc369335221)

[6.5.1 Tally in axial sections of PWR fuel pin 47](#_Toc369335222)

[6.5.2 Large-scale tally for Hoogenboom whole core benchmark 48](#_Toc369335223)

[Chapter 7 Diagnosis and Acceleration of Source Convergence 53](#_Toc369335224)

[7.1 Shannon Entropy statistics 53](#_Toc369335225)

[7.2 Fission matrix statistics 54](#_Toc369335226)

[7.3 Source convergence acceleration 55](#_Toc369335227)

[7.3.1 Factor option 55](#_Toc369335228)

[7.3.2 AutoFactor option 56](#_Toc369335229)

[7.3.3 Precautions of the source convergence acceleration 56](#_Toc369335230)

[7.4 Input examples of the source convergence block 57](#_Toc369335231)

[7.4.1 Source convergence acceleration of the OECD benchmark 57](#_Toc369335232)

[7.4.2 Source convergence acceleration of the Hoogenboom whole core benchmark 58](#_Toc369335233)

[Chapter 8 Burnup Calculation 62](#_Toc369335234)

[8.1 Overview of the Monte Carlo burnup calculation 62](#_Toc369335235)

[8.2 Input card of the burnup block 62](#_Toc369335236)

[8.3 Input examples of the burnup block 65](#_Toc369335237)

[8.3.1 Burnup example of PWR cell 65](#_Toc369335238)

[8.3.2 Burnup example of PWR assembly 67](#_Toc369335239)

[8.3.3 Burnup example of PWR core 69](#_Toc369335240)

[Chapter 9 Output Control 73](#_Toc369335241)

[9.1 Input card of the output control block 73](#_Toc369335242)

[9.2 Input example of the output control block 74](#_Toc369335243)

[Chapter 10 Plotting 75](#_Toc369335244)

[10.1 Input card of the plotting block 75](#_Toc369335245)

[10.2 Input example of the plotting block 76](#_Toc369335246)

[Appendixes 78](#_Toc369335247)

[Appendix 1 Lists of the input/output files 78](#_Toc369335248)

[Appendix 2 Continuous-energy ACE libraries 79](#_Toc369335249)

[Appendix 3 Burnup libraries 82](#_Toc369335250)

# Chapter 1 Installation and Running of RMC

## 1.1 Installation

### 1.1.1 Versions classification

This release of RMC program contains multiple versions, including 32/64-bit Windows/Linux operating systems specifically including:

(1) RMC-windows: The version of RMC program operated on Windows systems, supporting 32-bit and 64-bit Windows systems. The suggested operating environments are Windows XP or more advanced versions, with the vc2010 runtime library installed.

(2) RMC-linux32: The version of RMC program operated on 32-bit Linux systems. The suggested operating environments are Red Hat Enterprise Linux 5.6 or more advanced version, with the gcc library installed.

(3) RMC-linux64: The version of RMC program operated on 64-bit Linux systems. The suggested operating environments are Red Hat Enterprise Linux 5.6 or more advanced version, with the gcc library installed.

The above versions cover the versions for serial and parallel computations, and the file names of parallel versions end with "\_mpi". For the large-Scale computations which require the simulations of large amounts of particles (such as whole core and burnup calculations), the 64-bit Linux parallel version operated on the parallel workstation is suggested.

### 1.1.2 Installation configuration

This release of RMC program is an executable program, with no installing packets. The directories of RMC include:

|  |
| --- |
| Bin folder: The executable files, xsdir and DepthMainLib are stored.  Library folder: ACE library is stored.  Example documents: Input/output examples are stored.  User's Manual of Reactor Monte Carlo Program RMC |

Before installing the program, necessary configurations are required, of which the procedures are listed below:

(1) Copy the Bin folder and Library folder in RMC folder of the release package of program to computers respectively. These two folds can be stored in different directories, making no difference to the running of RMC.

(2) Open the xsdir file in the Bin folds with text editor, and then set the full directories of the Library fold, such as “DATAPATH = E:\Library” (windows) or “DATAPATH = /home/username/ Library” (Linux).

(3) Put the input files in the directories of RMC executable program, and then the calculations examples can be run. Notice that index file "xsdir" should be placed in the same directories of the RMC executable program. Before the burnup calculation, burnup library file "DepthMainLib" should also be placed in the same directory of the RMC executable program.

(4) Before running the parallel versions of RMC, the MPI parallel system should be firstly configured in the operation systems. Software such as mpich2, impi, openmpi and mvapich can create the MPI parallel environment. The detailed installing procedures of the corresponding software can refer to their documentations.

## 1.2 Running RMC

### 1.2.1 Serial running

Assuming that the name of executable file of RMC is "RMC", the method to run RMC in serial mode is: Through the windows command console or Linux terminal, users can enter the directory of RMC executable program, and then input the following commands:

|  |
| --- |
| Windows environment: **RMC (space) name of the input file (space) name of output file**  Linux environment: **./RMC(space)name of the input file(space)name of output file** |

Notice:

(1)The input files should be placed in the same directory of RMC executable program, or the full path of files must be used.

(2)The names of output files can be omitted, and the program will use the input files’ names with the suffix ".out" as the output files’ names by default. As for the burnup calculations, the program will also use the specified suffix ".nuc" and ".power" for the additional output files. Users must avoid the conflict of output files’ names.

(3)The suffix should be included if the input files' names have a suffix. In the windows systems, the input files in "txt" text format are not suggested. It's advised to use software such as Ultraedit to convert its format to DOC format.

### 1.2.2 Parallel running

The MPI paralel environments should be firstly configured in the operation systems before the parallel invocations of RMC. Software such as mpich, impi, openmpi and mvapich can provide the MPI parallel environments. The common users are advised to use mpich2, whose latest release can be downloaded on <http://www.mcs.anl.gov/research/projects/m>pich2 for free.

Taking the MPICH2-1.4.1p1 in Windows for example, if the executable program of parallel RMC is “RMC\_mpi”, the command of executing the parallel calculations is:

|  |
| --- |
| **mpiexec (space) –n (space) number of parallel cores (space) RMC\_mpi (space) name of input file (space) name of output file** |

For example, “mpiexec –n 10 RMC\_mpi input output” means 10 processes are used for calculations, and the name of input and output files are “input” and “output” respectively.

## 1.3 Input/output test

The input and output files for the classical calculations examples are attached in the Example folder of release packages. Users can validate whether the program has been installed successfully by running these examples (such as “3\_1\_PWR\_assembly, 8\_1\_Burn\_PWR\_pin”).

# Chapter 2 Overview of RMC Input

## 2.1 Input blocks

The input files of RMC are divided into different **blocks**, whose names and functions are listed as below:

* **SURFACE**: Define the types and equation of surfaces.
* **UNIVERSE**: Describe an integrated geometric space. RMC adopts the universe-based geometry system. Several UNIVERSE blocks can exist in one input file.
* **MATERIAL**: Define the material compositions.
* **CRITICALITY**: Define parameters of critical calculations, including particles number, initial sources and so on.
* **TALLY**: Define counter, including flux, power, reaction rate and so on.
* **CONVERGENCE**: Define diagnosis of source convergence and acceleration parameters.
* **BURNUP**: Define parameters of burnup calculations, including burnup lattice cell, power, time step and so on.
* **PLOT**: Define parameters of 2D sectional geometry/material plotting
* **PRINT**: Define the content of output.

## 2.2 Input cards

Specific input cards for different input blocks:

Table 2-1 Overview of RMC input cards

|  |  |  |  |
| --- | --- | --- | --- |
| Input blocks | Input card | Descriptions of functions | Chapters and sections in the manual |
| Surface | Surf | Define a curved surface, including the type, equation parameters, boundary conditions of the surface, and so on | 3.1 |
| Universe | Cell | Define a lattice cell in the space, including its material, geometry, volume, temperature and so on | 3.2 |
| Universe | Define a integrated geometric space, including coordinate transformations, repeated mesh and so on | 3.3 |
| Material | Mat | Define a kind of material, including density，nuclide composition, and so on | 4.1 |
| Sab | Determine the thermalized cross sections libraries for a kind of material | 4.1 |
| CeAce | Determine the input parameters related to the continuous-energy ACE cross sections | 4.2 |
| MgAce | Determine the input parameters related to the multi-group ACE cross sections | 4.3 |
| Criticality | PowerIter | Determine the initial *keff* and particles numbers for source iterations | 5.1 |
| InitSrc | Determine the initial fission distributions source for source iterations | 5.2 |
| RNG | Determine the type and parameters of the random number generator | 5.3 |
| ParallelBank | Determine the handle mode for the neutron library of fission source of parallel critical calculations. | 5.4 |
| Tally | CellTally | Define the cell tally. Sum up the flux, power, total absorption reaction rate or total fission reaction rate in one or more cells | 6.1 |
| MeshTally | Define mesh tally. According to the pre-defined mesh, sum up the flux, power, total absorption reaction rate or total fission reaction rate in each cell | 6.2 |
| CsTally | Define cross sections tally. Determine one or more reaction types, and sum up the reaction cross sections of all the nuclides in a cell | 6.3 |
| AcceTally | Determine whether the acceleration function of the tally is used. | 6.4 |
| Convergence | SeMesh | Define the Shannon entropy mesh | 7.1 |
| FmMesh | Define the Fission matrix mesh | 7.2 |
| AcceFsc | Determine the acceleration method of source convergence and the corresponding parameters | 7.3 |
| Burnup | BurnCell | Determine burnup area | 8.2 |
| Power | Determine the total power |
| TimeStep | Determine the time step |
| SubStep | Determine inner burnup step |
| Inherent | Determine the inheritance share of some important nuclides |
| AceLib | Determine the matched ACE cross sections library of some important nuclides |
| Strategy | Determine whether the burnup step strategy of predictor-corrector method is used |
| Solver | Determine the solution method of burnup equation |
| Parallel | Determine whether the parallel burnup calculation is used |
| Outputcell | Determine the cells whose nuclides densities need to be output |
| Print | Mat | Determine whether all the information of material will be output | 9.1 |
| Keff | Determine whether the *keff* of each iteration will be output |
| Source | Determine whether the information of fission source in each iteration will be output |
| CellTally | Determine whether the cell tallies will be output |
| MeshTally | Determine whether the mesh tallies will be output |
| CsTally | Determine whether the cross sections tallies will be output |
| Plot | ColorScheme | Determine the color schemes for plotting | 10.1 |
| PlotID | Determine the parameters |

## 2.3 Input format

Several points should be noticed on the format of RMC input files:

1. Each block is identified by the corresponding key word, and separated by blank lines. Such as:

|  |
| --- |
| Universe 0  ……  Universe 1  ……  Surface  ……  Material  ……  Criticality  …… |

(2) The input cards should be written flush, and the tabs of input cards should be separated by spaces. If one of the lines of the input cards has not been finished, it can be continue in a new line after spaces. Such as:

|  |
| --- |
| CellTally 2 type = 1 filter = 1 0 1 energy = 6.25E-7 20.0  cell = 2 > 0 > 1:289 |

(3) “//” can be used for annotations (C++ style)[[1]](#footnote-1).

(4) The input flies of RMC are case-insensitive.

(5) In the windows systems, the input files in "txt" text format are not suggested. It's advised to use software such as Ultraedit to convert the format to DOC format.

# Chapter 3 Geometry

The capacity of description of complex geometry is one of the most important advantages of Monte Carlo programs compared with the deterministic programs. Like most of the Monte Carlo programs in the world, RMC adopts the universe-based geometry system.

The geometric description of RMC includes 3 kinds of basic units, i.e. surface, cell and universe. Generally, a physical system consists of one or more universe, and each universe has several cells, which are defined by the join and merge operations in the surface direction (sense). The geometry description of RMC generally has the following form:

|  |
| --- |
| UNIVERSE 0 // Describe the topmost Universe block  Cell … // Describe the 1st cell in Universe 0  Cell … // Describe the 2nd cell in Universe 0  ……  UNIVERSE 1 // Describe lower layer Universes  ……  SURFACE // Describe the curved surface block  Surf ... // Describe the 1st curved surface  Surf ... // Describe the 2nd curved surface  …… |

## 3.1 Surface

Surfaces are the most fundamental geometric descriptions in RMC. Considering the widespread users of MCNP, RMC refers to the definition method of surfaces in MCNP. The definition method of input cards for surfaces is:

|  |
| --- |
| Surf <id> <type> {params} [Bc = <flag>] |

In which,

* **Surf** is the key word of the input cards for surfaces
* **Id** is the identifier of surfaces, and unique positive integers are required.
* **Type** is the key word of corresponding surface types, and params is the parameters of the surface equation. Table 3-1 lists the supported surface types in RMC and their surface equations.
* **Params** are the parameters of surface equation, which are shown in following table. The unit of surface parameters is in cm.
* **Bc** is the boundary condition. Bc = 0 (Default) means the vacuum boundary condition, Bc = 1 means the reflective boundary condition.

Table 3-1 Supported surface types in RMC

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Type | key word | Description | Equation | Parameters of surface equation |
| Plane | **P** | General |  |  |
| **PX** | perpendicular to X axis |  |  |
| **PY** | perpendicular to Y axis |  |  |
| **PZ** | perpendicular to Z axis |  |  |
| Sphere | **SO** | Center in origin |  |  |
| **S** | General |  |  |
| **SX** | Center in X axis |  |  |
| **SY** | Center in Y axis |  |  |
| **SZ** | Center in Z axis |  |  |
| Cylinder | **C/X** | Parallel to X axis |  |  |
| **C/Y** | Parallel to Y axis |  |  |
| **C/Z** | Parallel to Z axis |  |  |
| **CX** | Axes in X axis |  |  |
| **CY** | Axes in Y axis |  |  |
| **CZ** | Axes in Z axis |  |  |
| Cone | **K/X** | Parallel to X axis |  |  |
| **K/Y** | Parallel to Y axis |  |  |
| **K/Z** | Parallel to Z axis |  |  |
| **KX** | Axes in X axis |  |  |
| **KY** | Axes in Y axis |  |  |
| **KZ** | Axes in Z axis |  |  |
| Ellipsoid/ Hyperboloid /Paraboloid | **SQ** | Axis parallel to X, Y or Z axis | Not supported in this release |  |
| Cylinder/ Cone / Ellipsoid/ Hyperboloid /Paraboloid | **GQ** | Axis not parallel to X, Y or Z axis | Not supported in this release |  |
| Elliptical/Circular torus | **TX** | Parallel to X axis | Not supported in this release |  |
| **TY** | Parallel to Y axis | Not supported in this release |  |
| **TZ** | Parallel to Z axis | Not supported in this release |  |

## 3.2 Cell

The definition method of input cards for cells is:

|  |
| --- |
| Cell <id> {surf\_bool\_definition} [cell\_info] |

In which,

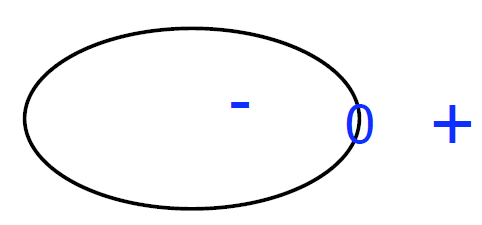
* **Cell** is the key word of the input cards for surfaces
* **id** is the identifier of cells, unique positive integers are required.
* **surf\_bool\_definition** means the bool definition for the surface of a cell, consisting of the surface with direction and boolean operators. **cell\_info** defines the other information related to this cell. Both of **surf\_bool\_definition** and **cell\_info** will be illustrated in the following.

### 3.2.1 Boolean definition for the surface of a cell

The bool definition for the surface of a cell consists of a serial of surfaces and boolean operators, such as

|  |
| --- |
| **<±surf> <boolean> <±surf> <boolean> <±surf> …** |

“Sense” is defined as: If the calculated value of a point (x,y,z) in the surface function is *f* (*x,y,z*) > 0, then this point is regarded as positive to the surface; in the other hand, if *f* (*x,y,z*) < 0, this point is negative. If *f* (*x,y,z*) = 0, it means that the point is on the surface. Figure 3-1 gives the corresponding area related to different directions of a surface.



F(x,y,z) < 0

F(x,y,z) > 0

F(x,y,z) = 0

Figure 3-1 Diagram of the surface direction

The boolean operations in RMC include intersection (&), union (:) and complement (!). The adjusting of operator precedence by parenthesis is also supported. Complement is prior than the intersection and union. The priority of intersection and union is the same, and the boolean operations are according to the sequence of definitions of boolean operators. The parentheses have the topmost priority, and the expressions can also be nested in multiple levels of parentheses, similar to the arithmetic operation. Assuming that the geometric descriptions of cell 1 and cell 2 are:

Cell 1: (1 & -2) : 3

Cell 2: 4 & -5: ! 1

The geometric region of cell 1 means: (positive direction of surface 1 ∩ negative direction of surface 2) ∪ positive direction of surface 3

The geometric region of cell 2 means: (positive direction of surface 4 ∩negative direction of surface 5) ∪ Logical negation of cell 1. Another equivalent expression of cell 2 is: 4 & 5 : !((1 & -2):3). It should be noticed that, if “!” is before numbers, it means the logical negation of cells; if “!” is before parentheses, it means the logical negation of surfaces.

### 3.2.2 Options of the cells information

Options of the cells information consist of a serial of tabs, meanly to describe the physical and geometric parameters, including material, volume, temperature, information of hierarchy filling, and so on.

|  |
| --- |
| Cell … [Mat = <id>] [Vol = <vol>] [Tmp = <tmp>]  **[Void = <flag>] [Fill = <id>] [Inner = <flag>]** |

In which,

* **Mat** option defines the filling material of cells. By default, **Mat = 0** means vacuum.
* V**ol** option defines the volume of cells (in cm3). By default, **Vol = 1.0cm3**。
* **Tmp** option defines the temperature of cells (in K). By default, **Tmp = 293.6 (K)**. If the temperature of the filling material can’t match the temperature of cell, the doppler broadening of nuclides cross sections will be conducted.
* **Void** option determines whether the tracking of neutrons will be stopped after neutrons entering this cell, meanly to describe the region outside the void boundary. By default, **Void = 0** means the tracking continue after neutrons entering this region, while **Void = 1** means stop tracking.
* **Fill** option defines the space for filling inside the cells. More information can refer to section 3.3.2.
* **Inner** option determines whether the cell is an inner cell, i.e., the cell has not been split by the outer boundaries in the progress of filling. By default, **Inner = 0** means non inner cells and no acceleration, while **Inner = 1** means inner cells and acceleration. Determine inner cells can accelerate the geometric processing, while the wrong determinations of inner cells will lead to faults of geometric tracking. Therefore, this option is advised only for advanced users.

## 3.3 Universe

### 3.3.1 Single-level universe

A universe is composed of a serial of cells, while the overlap and undefined areas among these cells are not allowed. The input card of single-lever universe can be written as:

|  |
| --- |
| UNIVERSE <id> [options] |

In which, **id** is the identifier of universe, and **options** are related to geometrical transformation in space and repeated structures (lattices). The form is shown as below and will be introduced in detail in the following.

|  |
| --- |
| **[Move = <params>] [Rotate = <params>] [Lat = <params>]**  **[Pitch = <params>] [Scope = <params>] [Sita = <param>] [Fill = <params>]** |

At least a universe is required to describe an arbitrary physic system. This universe wil be defined as Universe 0 in the input files. For example, the following input file is for the geometry of a common PWR cell.

|  |
| --- |
| /////// PWR pin: defined in single universe /////////////  Universe 0  cell 1 -10 mat = 1 // Fuel Pin  cell 2 10 & -11 mat=2 // Air  cell 3 11 & -12 mat =3 // cladding  cell 4 12 & 13 & -14 & 15 & -16 mat= 4 // water  cell 5 -13 : 14 : -15 : 16 void = 1 // outside  Surface  surf 10 cz 0.4096  surf 11 cz 0.4178  surf 12 cz 0.4750  surf 13 px -0.63 bc = 1  surf 14 px 0.63 bc = 1  surf 15 py -0.63 bc = 1  surf 16 py 0.63 bc = 1 |

### 3.3.2 Multi-level universe

For a complex physics system, the descriptions of the space filling are probably necessary, i.e. filling a universe into a cell of another universe.

The tabs of space filling are embedded in the input cards of cells:

|  |
| --- |
| **Cell ... [Fill = <universe>]** |

Notice that the filling regions should cover the regions to be filled, or an undefined blank region will exist in this region of cell, leading to the error of particles tracking.

For the example of PWR cell above, we can use the method of space filling to describe equivalently, as shown below:

|  |
| --- |
| /////// PWR pin: defined in multilevel universe /////////////  Universe 0  Cell 101 13 & -14 & 15 & -16 Fill = 1 // define a cell filled by a universe  cell 102 -13 : 14 : -15 : 16 void = 1 // outside the box  Universe 1  cell 1 -10 mat = 1 // Fuel Pin  cell 2 10 & -11 mat = 2 // Air  cell 3 11 & -12 mat = 3 // cladding  cell 4 12 mat = 4 // water  Surface  surf 10 cz 0.4096  surf 11 cz 0.4178  surf 12 cz 0.4750  surf 13 px -0.63 bc = 1  surf 14 px 0.63 bc = 1  surf 15 py -0.63 bc = 1  surf 16 py 0.63 bc = 1 |

### 3.3.3 Geometric transformation

RMC supports the translation transformation and rotation transformation. The tabs of geometric transformations are embedded in the input cards of universes:

|  |
| --- |
| **Universe ... [Move = ]**  **[Rotate =** ] |

The expression of translation transformation is:



In which, and resent the position coordinates of an arbitrary point before and after the transformation respectively.  is the vector of translation transformation.

An arbitrary axis can be selected for rotation transformation, with the expression:



In which, is the matric of rotation transformation. The actual requirement for users in RMC is the transposed matrix  of the rotation transformation matric. The parameters of  can be defined as: given a certain rectangular coordinate system, it can be converted to a new coordinate system  by rotation transformation, and  will be:



represents the cosine value of of the angle between axis  and , and the same with rest elements.

Notice that if both rotation transformation and translation transformation are implied on a universe, rotation will go first and then come to translation. For a multi-level universe, the geometric transformation for this universe always comes along the whole transformation including the inner filling structure. In addition, Universe 0 is the reference space, so the geometric transformation for Universe 0 is forbidden.

We can redefine the PWR cell using the geometric transformation, as is shown below. The regions of fuel pin and moderator (Universe 1) are defined as a cylinder parallel to X axis. By translation (move = 0.5 0.5 0) and transformation (rotate = 0 0 -1 0 1 0 1 0 0), it can be filled in the cell (Cell 101).

|  |
| --- |
| // PWR pin: defined in multilevel universe with coordinate transformation //  Universe 0  cell 101 13 & -14 & 15 & -16 Fill = 1 // define a cell filled by a universe  cell 102 -13 : 14 : -15 : 16 void // outside the box  Universe 1 move = 0.5 0.5 0 rotate = 0 0 -1 0 1 0 1 0 0  cell 1 -10 mat = 1 // Fuel Pin  cell 2 10 & -11 mat = 2 // Air  cell 3 11 & -12 mat = 3 // cladding  cell 4 12 mat = 4 // water  Surface  surf 10 c/x -0.5 -0.5 0.4096  surf 11 c/x -0.5 -0.5 0.4178  surf 12 c/x -0.5 -0.5 0.4750  surf 13 px -0.63 bc = 1  surf 14 px 0.63 bc = 1  surf 15 py -0.63 bc = 1  surf 16 py 0.63 bc = 1 |

## 3.4 Lattice

Lattice is a kind of special universe, consisting of meshes in regular array. RMC supports the frequently-used rectangular lattices and hexagonal lattices, which are the most common lattices in reactor core calculation and analysis. Rectangular lattice can be built up in 1, 2 or 3 dimensions, while the hexagonal lattice can be built up in a 2-D plane.

The tags of lattice are embedded in the input cards or universe:

|  |
| --- |
| Universe … [Lat = <type>] |

In which, **Lat = 1** means the rectangular lattice, and **Lat = 2** means the hexagonal lattice. These two kinds of lattices will be illustrated respectively in the following.

### 3.4.1 Rectangular lattice

Figure 3-2 gives the diagram of rectangular lattice. Repeated meshes are built in xyz coordinates, with the origin in the bottom left corner of the 1st mesh (numbered 1).

x

y

z

1

2

3

4

5

6

7

8

9

10

11

12

13

14

15

16

24

25

26

27

28

36

O

Figure 3-2 Diagram of rectangular lattice

The tag of rectangular lattice is:

|  |
| --- |
| Universe … [Lat = 1] [Scope = <xNum yNum zNum>]  [Pitch = <xLen yLen zLen>] [Fill = <U1 U2 … UM>] |

In which,

* **Lat = 1** means the type of lattice is rectangular lattice.
* **Scope** option defines the number of repeated meshed in the x, y, z directions. Particularly, “1” means only a layer of meshes in the direction. For example, the repeated meshes of 2 dimensions PWR assembly can be represented as **Scope = 17 17 1.** It should be pointed out that although the rectangular lattice s in 3D can be define directly in RMC, users are advised to build the 3D lattice using the filling mode combined of 2D lattices and 1D lattices.
* **Pitch** option defines the width of repeated meshed in the x, y, z directions. This parameter should be positive. If there is only a layer of meshes in a certain direction, the parameter of **Pitch** option has no real meaning.
* **Fill** option successively defines the identifiers of the universe filling in the meshes, totally  identifiers. The filling order of **Fill** is: x direction filled at first, then the y direction, at last the z direction. Figure 3-2 shows the numbering of indexes for rectangular lattices, which map the filling order of **Fill** option and the identifiers of tally for lattices (see chapter 5).

### 3.4.2 Hexagonal lattice

x

y

Sita

b1Len

b2Len

b1

b2

O

1

2

3

4

5

6

7

8

9

10

11

12

Figure 3-3 Diagram of hexagonal lattice

Figure 3-3 gives the diagram of hexagonal lattice. It is not difficult to discover that the center of each hexagon is arranged in form of parallelogram. The direction vectors of two edges of the parallelogram b1 and b2 are in x-y plane, and the direction of b1 is the same as x axis. The origin is built in the center of the 1st hexagon (numbered 1)

The tags of hexagonal lattices are:

|  |
| --- |
| Universe … [Lat = 2] [Scope = <b1Num b2Num>] [Sita = <sita>]  **[Pitch = <b1Len b2Len>] [Fill = <U1 U2 … UM>]** |

In which,

* **Lat = 2** means the type of lattice is hexagonal lattice.
* **Scope** option defines the number of lattices in the directions of b1 and b2.
* **Pitch** option defines the width of lattices in the directions of b1 and b2.
* **Sita** option defines the angle (in degree °) between a pair of adjacent edges of hexagonal lattice (as in Figure 3-3).
* **Fill** option successively defines the identifiers of the universe filling in the meshes, totally  identifiers. The filling order of **Fill** is: b1(x) direction filled at first, then the b2 direction, as is shown in Figure 3-3.

It should be pointed out that the hexagonal lattices are built in x-y plane by default in RMC. Through the translation transformation and rotation transformation, it can be converted to x-z or y-z plane.

## 3.5 Input examples of geometry blocks

### 3.5.1 PWR assembly

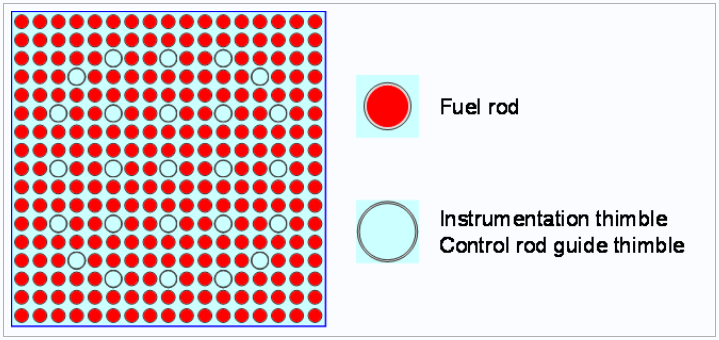


Figure 3-4 PWR 17×17 assembly

Example 3-1 is an input example of PWR 17×17 assembly (Figure 3-4). Universe 1 and Universe 3 are fuel cells and thimbles cells respectively, with origin in (0, 0, 0). Universe 8 is the rectangular lattice, with thecenter of the 1st mesh is (0.63, 0.63, 0), as the origin (0, 0, 0) of the rectangular lattice is built in the bottom left corner. After translating Universe 1 and 3 according to vector (0.63, 0.63, 0), they can be filled in the 1st mesh of Universe 8, and then unfold in the array of rectangular lattices.

Example 3 – 1

|  |
| --- |
| // STANDARD WESTINGHOUSE 17\*17 ASSEMBLY MODEL. SHE DING : 2012-03-08 //  UNIVERSE 0  CELL 1 6 & -7 & 8 & -9 mat = 0 Fill = 8 // Assembly inside  CELL 2 -6 : 7 : -8 : 9 mat = 0 void = 1 // Assembly outside  UNIVERSE 8 lat = 1 pitch = 1.26 1.26 1 scope = 17 17 1 fill =  1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1  1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1  1 1 1 1 1 3 1 1 3 1 1 3 1 1 1 1 1  1 1 1 3 1 1 1 1 1 1 1 1 1 3 1 1 1  1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1  1 1 3 1 1 3 1 1 3 1 1 3 1 1 3 1 1  1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1  1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1  1 1 3 1 1 3 1 1 3 1 1 3 1 1 3 1 1  1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1  1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1  1 1 3 1 1 3 1 1 3 1 1 3 1 1 3 1 1  1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1  1 1 1 3 1 1 1 1 1 1 1 1 1 3 1 1 1  1 1 1 1 1 3 1 1 3 1 1 3 1 1 1 1 1  1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1  1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1  UNIVERSE 1 move = 0.63 0.63 0 // Fuel rod  cell 3 -1 mat = 1 inner = 1 // Fuel  cell 4 1 & -2 mat = 3 inner = 1 // Air  cell 5 2 & -3 mat = 4 inner = 1 // Zr  cell 6 3 mat = 5 // water  UNIVERSE 3 move = 0.63 0.63 0 // Guide tube  cell 11 -4 mat = 5 inner = 1 // water  cell 12 4 & -5 mat = 4 inner = 1 // Air  cell 13 5 mat = 5 // water  SURFACE  surf 1 cz 0.4096  surf 2 cz 0.4178  surf 3 cz 0.4750  surf 4 cz 0.5690  surf 5 cz 0.6147  surf 6 px 0 bc = 1  surf 7 px 21.42 bc = 1  surf 8 py 0 bc = 1  surf 9 py 21.42 bc = 1  MATERIAL  mat 1 -10.196  92235.30c 6.9100E-03  92238.30c 2.2062E-01  8016.30c 4.5510E-01  mat 3 -0.001  8016.30c 3.76622E-5  mat 4 -6.550  40000.60c -98.2  mat 5 9.9977E-02  1001.30c 6.6643E-02  8016.30c 3.3334E-02  sab 5 lwtr.60t  CRITICALITY  PowerIter population = 10000 50 300 // keff0 = 1.0  InitSrc point = 0.63 0.63 0 |

### 3.5.2 PWR reactor core

Example 3-2 is an input example of the PWR reactor core. The input file has only a kind of assembly for simplification. Universe 1 is a 21×21 rectangular lattice, including meshes of assemblies and reflectors. The assembly (Universe 3) is a 17×17 rectangular lattice, filled with fuel cell (Universe 6) and thimble cell (Universe 7).

Example 3 – 2

|  |
| --- |
| ////////// PWR core. SHE Ding 2012-07-01 ////////////  UNIVERSE 0  CELL 1 -10 mat = 0 Fill = 1 // Core inside  CELL 2 10 mat = 0 void = 1 // Core outside  UNIVERSE 1 lat = 1 pitch = 21.42 21.42 1 scope = 21 21 1 Fill = // core lattice zone  2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2  2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2  2 2 2 2 2 2 2 3 3 3 3 3 3 3 2 2 2 2 2 2 2  2 2 2 2 2 3 3 3 3 3 3 3 3 3 3 3 2 2 2 2 2  2 2 2 2 3 3 3 3 3 3 3 3 3 3 3 3 3 2 2 2 2  2 2 2 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 2 2 2  2 2 2 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 2 2 2  2 2 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 2 2  2 2 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 2 2  2 2 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 2 2  2 2 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 2 2  2 2 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 2 2  2 2 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 2 2  2 2 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 2 2  2 2 2 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 2 2 2  2 2 2 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 2 2 2  2 2 2 2 3 3 3 3 3 3 3 3 3 3 3 3 3 2 2 2 2  2 2 2 2 2 3 3 3 3 3 3 3 3 3 3 3 2 2 2 2 2  2 2 2 2 2 2 2 3 3 3 3 3 3 3 2 2 2 2 2 2 2  2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2  2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2  UNIVERSE 2 // reflector  cell 21 1 mat = 5  cell 22 -1 mat = 5  UNIVERSE 3 lat = 1 pitch = 1.26 1.26 1 scope = 17 17 1 fill = // assembly  6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6  6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6  6 6 6 6 6 7 6 6 7 6 6 7 6 6 6 6 6  6 6 6 7 6 6 6 6 6 6 6 6 6 7 6 6 6  6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6  6 6 7 6 6 7 6 6 7 6 6 7 6 6 7 6 6  6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6  6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6  6 6 7 6 6 7 6 6 7 6 6 7 6 6 7 6 6  6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6  6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6  6 6 7 6 6 7 6 6 7 6 6 7 6 6 7 6 6  6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6  6 6 6 7 6 6 6 6 6 6 6 6 6 7 6 6 6  6 6 6 6 6 7 6 6 7 6 6 7 6 6 6 6 6  6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6  6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6  UNIVERSE 6 move = 0.63 0.63 0 // Fuel rod  cell 3 -1 mat = 1 inner = 1 // Fuel  cell 4 1 & -2 mat = 3 inner = 1 // Air  cell 5 2 & -3 mat = 4 inner = 1 // Zr  cell 6 3 mat = 5 // water  UNIVERSE 7 move = 0.63 0.63 0 // Guide tube  cell 11 -4 mat = 5 inner = 1 // water  cell 12 4 & -5 mat = 4 inner = 1 // Air  cell 13 5 mat = 5 // water  SURFACE  surf 1 cz 0.4096  surf 2 cz 0.4178  surf 3 cz 0.4750  surf 4 cz 0.5690  surf 5 cz 0.6147  surf 10 c/z 224.91 224.91 209 bc = 1 // container  MATERIAL  mat 1 -10.196  92235.30c 6.9100E-03  92238.30c 2.2062E-01  8016.30c 4.5510E-01  mat 3 -0.001  8016.30c 3.76622E-5  mat 4 -6.550  40000.60c -98.2  mat 5 9.9977E-02  1001.30c 6.6643E-02  8016.30c 3.3334E-02  sab 5 lwtr.60t  CRITICALITY  PowerIter population = 100000 250 500 // keff0 = 1.0  InitSrc point = 224.91 226.17 0 |

### 3.5.4 Hexagonal assembly

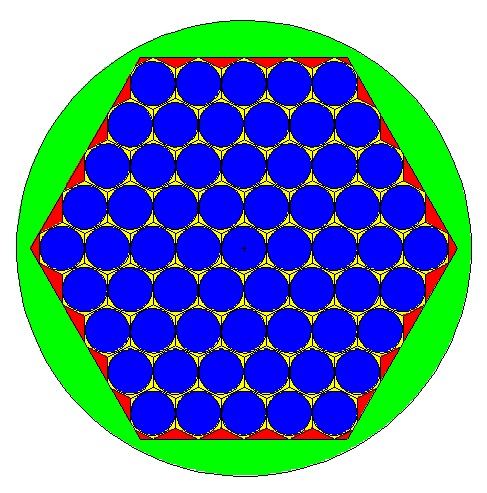


Figure 3-5 Simplified dirgram of hexagonal assembly

Exmaple 3-3 is a simplified input exmaple for a hexagonal assembly, including 61 hexagonal fuel cells. In Figure 3-5, Blue ones represent the fuel, and yellow ones represent the wrapped wires, green ones reflector. Universe1 is hexagonal lattices, nicluding coolant cells (Universe2) and fuel cells (Universe3). From the array of hexagons in Figure 3-3, it can be found that the origin of hexagonal lattices is the center of hexagon in the bottom left corner, therefore, Universe 1 needs to ,ove by -15and -9.05 in x and y direction respectively, so as to coincide the center of Universe 1 woth that of Cell 1.

Example 3 – 3

|  |
| --- |
| ///////////// MFR ASSEMBLY. FAN Xiao 2012-09-17 /////////////  Universe 0  cell 1 -1&-2&-3&4&-5&-6&7&-8 mat=0 fill=1 //Assembly inside  cell 2 16:-17:18 mat=0 void=1 //Assembly outside  cell 3 -16&17&-18&(1:2:3:-4:5:6:-7:8) mat=5 //reflector  Universe 1 move=-15 -9.05 0 lat=2 pitch=2 2.06787 scope=11 11 sita=63.435 fill=  2 2 2 2 2 2 2 2 2 2 2  2 2 2 2 2 3 3 3 3 3 2  2 2 2 2 3 3 3 3 3 3 2  2 2 2 3 3 3 3 3 3 3 2  2 2 3 3 3 3 3 3 3 3 2  2 3 3 3 3 3 3 3 3 3 2  2 3 3 3 3 3 3 3 3 2 2  2 3 3 3 3 3 3 3 2 2 2  2 3 3 3 3 3 3 2 2 2 2  2 3 3 3 3 3 2 2 2 2 2  2 2 2 2 2 2 2 2 2 2 2  Universe 2  cell 21 -15 mat=1  cell 22 15 mat=1  Universe 3  cell 31 -15 mat=2  cell 32 15 mat=3  Surface  surf 1 py 8.4  surf 2 p 1.732 1 0 16.3  surf 3 p 1.732 -1 0 16.3  surf 4 py -8.4  surf 5 p -1.732 -1 0 16.3  surf 6 p -1.732 1 0 16.3  surf 7 pz -30  surf 8 pz 30  surf 15 cz 0.975  surf 16 cz 30  surf 17 pz -35  surf 18 pz 35  Material  mat 1 -0.8139 // Na  11023.30c 1.0  mat 2 -10.41 // UO2  92235.30c -56.5 92238.30c -31.1 8016.30c -12.3 13027.30c -0.02  20000.60c -0.02 12000.60c -0.02 26000.55c -0.02 14000.60c -0.02  mat 3 -0.8355 // wiry  11023.30c 2.132 28000.50c 3.223E-3 24000.50c 4.759E-3 26000.55c 1.634E-2  mat 5 0.1236 // Be9  4009.30c 1  Criticality  PowerIter keff0=1.0 population = 2000 50 300  InitSrc point=0 0 0 |

### 3.5.5 Hexagonal reactor core

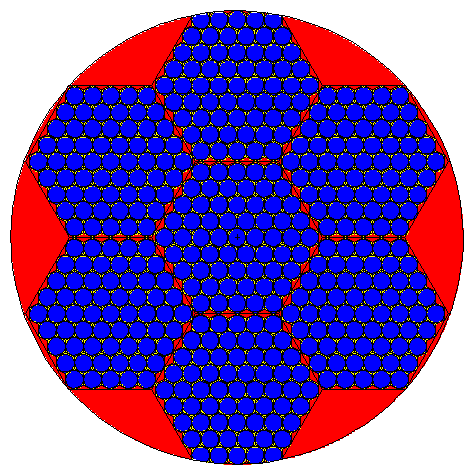


Figure 3-6 Simplified dirgram of hexagonal reactor core

Example 3-4 is the simplified input example of the hexagonal reactor core. In Figure 3-6, blue ones represent fuel, yellow ones represent the wrapped wires, red ones coolant. For simplifications, the input file of reactor core only contains 7 fuel assemblies in the same type. Assemblies can be added freely for a real reactor core, while only the process of geometric transformation and filling will be focused to introduce here. Universe 1 is hexagonal lattices, in which Universe 2 is the coolant and Universe 3 is the hexagonal fuel assembly. Universe 3 itself contains fuel cells in hexagonal lattices (similar to the descriptions of Example 3-3), i.e. two levels of hexagonal lattices have been applied to the whole system. When describing this geometric structure, fuel and coolant (Universe 4 and Universe 5) are filled into the hexagonal lattices (Universe 3) at first, and then rotate the hexagonal lattices (Universe 3) anticlockwise by 90° (rotate = 0 1 0 -1 0 0 0 0 1) in the x-y plane. Then translate it by (move = 9.05, -15, 0), filled in the hexagonal lattices of assemblies (Universe 1). Universe 1 will be translated by (move = -50.4 -27.942 0), and filled in the center of reactor core (Cell 1). It should be noticed that users must rotate the geometry before translating.

Example 3 – 4

|  |
| --- |
| //// MFR CORE.FAN Xiao 2012-09-17 ////  Universe 0  cell 1 -21&7&-8 mat=0 fill=1  cell 2 21:-7:8 mat=0 void=1  Universe 1 move=-50.4 -27.942 0 lat=2 pitch=16.8 16.302 scope=5 5 sita=60 fill=  2 2 2 2 2  2 2 3 3 2  2 3 3 3 2  2 3 3 2 2  2 2 2 2 2  Universe 2  cell 21 -15 mat=1  cell 22 15 mat=1  Universe 3 move=9.05 -15 0 rotate=0 1 0 -1 0 0 0 0 1  lat=2 pitch=2 2.06787 scope=11 11 sita=63.435 fill=  4 4 4 4 4 4 4 4 4 4 4  4 4 4 4 4 5 5 5 5 5 4  4 4 4 4 5 5 5 5 5 5 4  4 4 4 5 5 5 5 5 5 5 4  4 4 5 5 5 5 5 5 5 5 4  4 5 5 5 5 5 5 5 5 5 4  4 5 5 5 5 5 5 5 5 4 4  4 5 5 5 5 5 5 5 4 4 4  4 5 5 5 5 5 5 4 4 4 4  4 5 5 5 5 5 4 4 4 4 4  4 4 4 4 4 4 4 4 4 4 4  Universe 4  cell 41 -15 mat=1  cell 42 15 mat=1  Universe 5  cell 51 -15 mat=2  cell 52 15 mat=3  Surface  surf 5 p 1 1.6632 0 46.474  surf 2 p 1 -1.6632 0 46.474  surf 3 p -1 -1.6632 0 46.474  surf 6 p -1 1.6632 0 46.474  surf 1 px 27.942  surf 4 px -27.942  surf 7 pz -30  surf 8 pz 30  surf 15 cz 0.975  surf 21 cz 25  Material  mat 1 -0.8139 // Na  11023.30c 1.0  mat 2 -10.41 // UO2  92235.30c -56.5 92238.30c -31.1 8016.30c -12.3 13027.30c -0.02  20000.60c -0.02 12000.60c -0.02 26000.55c -0.02 14000.60c -0.02  mat 3 -0.8355 // wiry  11023.30c 2.132 28000.50c 3.223E-3 24000.50c 4.759E-3 26000.55c 1.634E-2  Criticality  PowerIter keff0=1.0 population = 50000 200 1000  InitSrc point=0 0 0 |

# Chapter 4 Material

The input block of material describes the material compositions, including material density, cross section library of nuclides, nuclides shares and the optional parameters related to continuous-energy or multi-group ACE cross section libraries.

## 4.1 Input card of material

The input card of common material in RMC is:

|  |
| --- |
| **Mat <mat\_id> <density>**  **<zaid.xxx> <fraction>**  **<zaid.xxx> <fraction>**  **……** |

In which,

* **Mat** is the key word of the material input card.
* **mat\_id** is the identifiers of material, related to the filling material in Cell input cards.
* **Density** is the total density of material. If **density > 0,** it is the atomic density in 1024 atoms/cm3, while **density < 0,** it is the mass density in g/cm3.
* **zaid.xxx** determines the ACE cross section libraries related to nuclides, in which **zaid** is the ID of a nuclide ,with the suffix “**.xxx**” which determines the type of cross section libraries. For better distinguishing, when creating a library, users are advised to use the suffix “**.xxc**” for continuous-energy ACE cross section libraries, while the suffix “**.xxm**” for multi-group ACE cross section libraries. The detailed nuclides and types related to the libraries can be referred to the index file of libraries **xsdir.**
* **fraction** is the proportion of a nuclide in the material. If **fraction > 0**, it means the relative proportion of atomic density, while **fraction < 0** means the relative proportion of mass density. For the same material, the sign of **fraction** for different nuclides must be the same.

Besides the input cards for common material, input cards for thermalized material are also supported in RMC, providing the thermalized libraries for continuous-energy ACE cross sections.

|  |
| --- |
| **Sab <mat\_id> <zaid.xxx>**  **<zaid.xxt>**  **……** |

In which,

* **Sab** is the key word of the input card for thermalized material.
* **mat\_id** is the identifier of material, coinciding the identifier of material in **Mat** card.
* **zaid.xxx** determines the thermalized cross section libraries for the nuclides, referring to the index file of libraries **xsdir** for detail.

## 4.2 Input card for the continuous-energy ACE cross sections

If the continuous-energy ACE cross sections are adopted for the nuclides in the input cards of material, the optional parameters can be set using the following input card:

|  |
| --- |
| **CeAce [ErgBinHash = <flag>] [pTable = <flag>]** |

In which,

* **CeAce** is the key word of the input card for the continuous-energy ACE cross sections.
* **ErgBinHash** option determines whether the embedded hash tables method is applied to accelerate the searching speed of energy grids. Higher efficiency can be achieved with the acceleration method, while a few additional memory will be consumed as a cost. By default, **ErgBinHash = 1** means hash tables acceleration is on, while **ErgBinHash = 0** means hash tables acceleration is off.
* **pTable** option determines whether the probability table will be used for the indiscernible resonance regions. Only when the UNR block is contained in the ACE cross sections libraries for the nuclides, this option is effective. **pTable = 1** means the probability table is on, while **pTable = 0** (be default) means the probability table is off.

## 4.3 Input card for the multi-group ACE cross sections

If the multi-group ACE cross sections are adopted for the nuclides in the input card of material, the users must determine the optional parameters for multi-group cross sections with the following input card:

|  |
| --- |
| **MgAce [ErgGrp = <grp>]** |

In which,

* **MgAce** is the key word of input cards for the multi-group ACE cross sections.
* **ErgGrp** option determines the number of groups of multi-group ACE cross sections.

It should be pointed out that the multi-group cross sections libraries depend heavily on the actual physical problems. Therefore, the multi-group cross sections libraries are not supported in this RMC release. Users can produce the multi-group ACE cross sections libraries related to the practical problems with other program for libraries processing.

## 4.4 Input examples of the material block

### 4.4.1 Material inputs with continuous-energy ACE libraries

|  |
| --- |
| MATERIAL  mat 1 -10.196  92235.30c 0.03  92238.30c 0.97  8016.30c 2.0  mat 2 0.9997  1001.30c 2.0  8016.30c 1.0  Sab 2 lwtr.60t  CEACE pTable = 0 ErgBinHash = 1 |

In the above material block, UO2 and H2O have been defined with **Mat** input card firstly. The mass density of UO2 is -10.196 g/cm3, and the atomic ratios of U235、U238 and O16 are 0.03 : 0.97 : 2.0. The atomic density of H2O is 0.9997 bar-1cm-1, and the atomic ratios of H1 and O16 is 2 : 1. The thermalized library (lwtr.60t) is determined for H1 (1001.30c) of the H2O by **Sab** option. In the **CeAce** option, **pTable = 0** means the probability table is not used, and **ErgBinHash = 1** means the hash table acceleration is applied to energy searching.

### 4.4.2 Material inputs with multi-group ACE libraries

|  |
| --- |
| MATERIAL  mat 1 -10.198  92235.50m 6.9100E-03  92238.50m 2.2062E-01  8016.50m 4.5510E-01  mat 2 -0.001  8016.50m 3.76622E-5  mat 3 -6.550  40000.50m -98.2  mat 4 -0.997  1001.50m 6.6643E-02  8016.50m 3.3334E-02  MgAce ErgGrp = 30 |

In the above material block, “.50m” means the 30 groups multi-group ACE cross section libraries. The number of energy groups for multi-group cross section is determined by the **ErgGrp** option**.**

# Chapter 5 Criticality Calculation

The method of fission source iteration is applied to the critical calculation in Monte Carlo, i.e. the fission source produced in the current generation will be the initial source of the next generation. The basic parameters for source iteration must be defined in the input files by users, including initial effective multiplication factor (keff), the number of neutrons in each generation and the total number of generations, as well as the initial distribution of fission source.

In the block of critical calculation, users can choose the types and parameters of random number generator, and choose the parallel mode for parallel critical calculation.

## 5.1 Parameters of source iteration

|  |
| --- |
| **PowerIter [Keff0 = <keff0>] [Population = <N Mi Mt >] [Batch=< Mb >]** |

In which,

* **PowerIter** is the key word of the source iteration input card.
* **Keff0** option sets the initial effective multiplication factor, **keff0 = 1** by default. The **keff0** setbyusers must be in the range of: 0.1 < keff0 < 10.
* **Population** option sets the number of neutrons per generation (**N**), number of inactive generations (**Mi**) as well as the total generation (**Mt**). Accordingly, the number of active generations **Ma=(Mt-Mi)**, the total number of active neutrons is **M=N×Ma**.
* **Batch** option is used to reduce the merging operations in the tally of parallel critical calculations, condensing **Ma** active generations to **Mb** generations, with every **Ma/ Mb** active generations per data merging. **Ma= Mb** by default.

## 5.2 Initial fission source

|  |
| --- |
| **InitSrc [<type> = {params}]** |

In which,

* **InitSrc** is the key word of the input card of initial fission source.
* **type** is the key word of the types of fission source, and **params** is corresponding parameters. Two types of initial fission sources are supported in RMC: point source and uniform solid source, seeing Table 5-1 for details.

Table 5-1 Initial fission source distribution supported by RMC

|  |  |  |
| --- | --- | --- |
| key word | Description | Parameters |
| **Point** | Isolated point source | x1 y1 z1 x2 y2 z2 … |
| **Slab** | Uniform cuboid source parallel to the coordinate axis | xmin xmax ymin ymax zmin zmax |
| **Sph** | uniform sphere source | x y z R |
| **Cyl/x** | uniform cylindrical source parallel to x axis | y z R xmin xmax |
| **Cyl/y** | uniform cylindrical source parallel to y axis | x z R ymin ymax |
| **Cyl/z** | uniform cylindrical source parallel to z axis | x y R zmin zmax |

## 5.3 Random number generator

RMC contains 3 different random number generators. For the general users, it is advised to use the default type and parameters of the random number generator. The advanced users can use the specific parameters of random number, such as using different seeds for random number generator to acquire independent results for the same calculation test. The input card of custom-made random number generators is supported in RMC:

|  |
| --- |
| **RNG [Type = <type>] [Seed = <seed>] [Stride = <stride>]** |

In which,

* **RNG** is the key word of the input card of random number generator.
* **Type** option defines the type of random number generator. **Type = 1** means the generator of 48-bit multiplicative linear congruential method, **Type =2** (by default) means the generator of 64-bit multiplicative linear congruential method, and **Type = 3** is the generator of the 64-bit hybrid linear congruential method.
* **Seed** option defines the initial seed of random number generator, which can be an arbitrary positive odd number, **Seed = 1** by default.
* **Stride** option defines the length of segmented random number assigned for each neutron history when parallel calculation, which only the advanced used are advised to use. The default value is **Stride = 10000**.

## 5.4 Parallel critical calculation mode

In parallel critical calculation, considering the load balancing, the neutrons of fission source in each progress need to be collected and redistributed. The “master-slave” mode is commonly adopted by the traditional Monte Carlo program, whose efficiency of collecting and redistributing is relatively low. Different with traditional method, RMC uses the “peer-peer” mode to improve the parallel efficiency. The input card to select the mode of parallel critical calculation is:

|  |
| --- |
| **ParallelBank <flag>** |

In which,

* **ParallelBank** is the key word of input card for the mode of parallel critical calculation.
* **flag** option determines the parallel mode, as **flag = 0** means “master-slave” and **flag = 1** (by default) means “slave-slave”.

## 5.5 Input examples of the critical calculation block

|  |
| --- |
| CRITICALITY  PowerIter Population = 5000 30 100 // keff0 = 1.0  InitSrc point = 0.0 0.0 0.0  0.5 0.5 0.0  -0.5 -0.5 0.0  RNG type = 3 seed = 12345 stride = 10000  ParallelBank 1 |

In the above critical calculation block, **PowerIter** determines the number of particle per generation is 5000, and total generations are 100 after skipping 30 generations. The initial ***keff*** is 1.0 by default. **InitSrc** card set the initial source as a point source, locating in (0, 0, 0), (0.5, 0.5, 0) and (0.5,-0.5, 0). The fission source will be generated randomly in these three positions. **RNG** card sets the type of random number generator to be that of the 64-bit hybrid linear congruential method, and the initial seed of generator is 12345. The length of random number assigned for each particle is 10000. **ParallelBank** card means the collecting and redistributing of fission source using the slave-slave mode is applied to parallel calculations.

# Chapter 6 Tally

There are 3 types of tallies in RMC, including cell tally, mesh tally and cross-section tally. The cell tally is used to sum up the macroscopic physical quantities, including integral flux, power, fission rate and abortion rate. The Cell-Mapping method is adopted by RMC, so as to manage the large-scale cell tally efficiently. The mesh tally is similar to the cell tally, while the physical quantities it counts are not based on cells but the predefined meshes. The cross-section tally sums up the cross sections (micro reaction rate) of specified reaction types for specified nuclides in the cells. All these three tallies support the statistics of divisive energy groups i.e. statistics according to the predefined energy interval.

## 6.1 Cell tally

The input card of cell tally is

|  |
| --- |
| **CellTally <id> [Type = <type>] [Energy = <erg\_bin>] [Filter = <params>] [Integral = <params>] [Cell = <cell\_vector\_group>]** |

In which,

* **CellTally** is the key word of the input card of cell tally.
* **id** is the identifier of the cell tally, for easy consulting and output.
* **Type** card determines the type of counting. **Type = 1** means flux of cell, **Type = 2** means power of cell, **Type = 3** means fission rate of cell, **Type = 4** means absorption rate of cell. Notice that the flux, power, reaction rates are the integral quantities for the volume of cells.
* **Energy** card defines the energy interval for counting in the divisive energy groups. The parameter is the interval points (Mev) between energy groups. For example, “**Energy = 6.25E-7 20**” means the counting intervals are from 0 to 0.625ev, 0.625ev to 20Mev and 20Mev to the positive infinity, totally 3 intervals. Meanwhile, the total counting will be given. Especially, for multi-group critical calculations, “**Energy = -1**” means the cross section libraries are adopted for the division of energy structure. Only the total counting rate will be counted, if there is no **Energy** optionin the input card.
* **Cell、Filter、Integral** options are for the descriptions of the cell tally, which will be introduced in detail in the following.

### 6.1.1 Cell option

In the system of “universe”, an arbitrary region of cell is determined uniquely by a serial of cell identifiers and the relationship between “universes”. We can use the form of “Cell vector” to describe the cells with hierarchical relationship:

**Cell\_vector = C[1] > C[2] > … > C[n]**

In which, “>” means the inclusion relation of upper level to lower levels. C[i] means the identifiers of cells and lattices (when the ith level is in lattices). In the progress of Monte Carlo tracking, the particles always need to be located to the cells in the bottom layer (i.e. C[n] should not be filled by any other universes or cells). Then the corresponding information such as material and temperature can be acquired for transports simulations.

Similar to the descriptions of cells for particles locations (refer to section 3.5), the cell for tally also adopts the vector descriptions as following:

**C[1] > C[2] > … > C[n]**

The case that C[n] is the cell in basic level will be considered firstly, i.e. C[n] will no longer be filled by lower structures. The general implementation procedure of the tally is: In the progress of Monte Carlo simulations, check whether the located cell coincides with the counting cell (both two are cells in basic level). If yes, the information about counts will accumulate. It should be pointed out that the repetitive cell vectors are forbidden in the cell table of the same **Cell** card.

In order to improve the flexibility for the descriptions of counting cell, two auxiliary notations “: ” and “\*” have been introduced in RMC.

Taking the example of the PWR reactor core, the whole core (Cell No.1) is assumed to contain 21×21=441 repeated meshes, which are fuel assemblies or reflectors. Each fuel assembly will be further divided into 17×17=289 repeated meshes, which will be filled with fuel pins (Cell No.35) and moderator (Cell No.36) in each mesh.

In order to count the flux of the fuel pin in the center cell (repeated mesh No.145) of the center assembly (repeated mesh No. 221) of the core, the input of the counting cell is:

**1 > 221 > 145 > 35**

By that analogy, if users want to count the flux in the center fuel pins of other assemblies, the input should be:

**1 > 1 > 145 > 35**

**1 > 2 > 145 > 35**

**1 > 3 > 145 > 35**

**…**

**1 > 441 > 145 > 35**

With the unfolded notation “: ”, the above input can be simplified as:

**1 > 1:441 > 145 > 35**

RMC also support the multi-level unfolded input mode, such as “1 > 1:441 > 1:289 > 35”, unfolding level by level from right to left:

**1 > 1 > 1 > 35**

**…**

**1 > 1 > 289 > 35**

**1 > 2 > 1 > 35**

**…**

**1 > 2 > 289 > 35**

**…**

**1 > 441 > 1 > 35**

**…**

**1 > 441 > 289 > 35**

The global unfolded notation “\*” is a special case of the unfolded notation “: ”, as it will automatically search all the regions whose basic cell has a specific identifier, to count respectively. In the above example, users can input:

**\*36**

Thus the flux in every region of moderator (Cell No.36) in every assembly will be counted respectively.

### 6.1.2 Filter option

In 6.1.1, the description of counting cell “C[1] > C[2] > … > C[n]” only considers C[n] as the cell in bottom level (i.e. C[n] can no longer be filled by the lower structure). Sometimes user may need to count the flux distribution of cells which are not in the bottom level or composite cells, thus **Filter** option is needed.

**Filter** option’s parameter is the array made up of 0 and 1, and the length of the array is equal to the level of counting cell. By default, the elements of the array are 1. If the “0” wildcard character exists in the counting cell (see the following example), the corresponding positions in the **Filter** vector will be replaced by “0”.

One of the functions of **Filter** option is to count the flux of the non-bottomed cells. Taking the example of the case in 6.1.1, the objects of flux counting are the assemblies which are the meshes of repeated structures in the 1st level. The input card of cell tally is:

|  |
| --- |
| CellTally 1 Type = 1 Filter = 1 1  Cell = 1 > 1:441 |

In which, “1 > 1:441” is equal to “1 > 1 1 > 2 …… 1 > 441”. “Filter = 1 1” indicates that each counting cell of this tally has only two levels. The tally will give out 441 counts, which coincide with the flux of meshes in the level of 441 assemblies (including the meshes of reflector)

The other function of **Filter** option is to count the tally of the composite cell, as is shown below:

|  |
| --- |
| CellTally 1 Type = 1 Filter = 1 1 0 1  Cell = 1 > 1:441 > 0 > 35 |

Notice that the “0 in ” “1 > 1:441> 0 > 35” is a wildcard character, which means that the cells identifiers or meshes identifiers in the corresponding level will be neglected in the progress of counting matching. The corresponding level will be identified by “0” in the **Filter** option. The tally will give out 441 flux counting, among which the ist counting coincides with the total flux of fuel pins in the ist assemblies.

RMC adopt the Cell mapping method to handle the large-scale cell counting quickly. Users are suggested to put the counting cell in the same type (with the same Filter) in the same CellTally, to reduce the total number of CellTally (increase the counting scale per CellTally), so as to improve the counting efficiency.

### 6.1.3 Integral option

**Integral** option’ function is to merge the counting cells in the tally piece by piece, treating them as a whole for counting. Such as:

|  |
| --- |
| CellTally 1 Type = 1 Filter = 1 1 0 1  Integral = 100\*3 141 （namely Integral = 100 100 100 141）  Cell = 1 > 1:441 > 0 > 35 |

The tally will give out 4 counting: total counting in 1 > 1:100 > 0 > 35, total counting in 1 > 101:200 > 0 > 35, total counting in 1 > 201:300 > 0 > 35 and total counting in 1 > 301:441 > 0 > 35. With the **Integral** option, any number of cells can be counted as a whole theoretically (even though they are not next to each other physically).

## 6.2 Mesh tally

The input card of mesh tally is:

|  |
| --- |
| **MeshTally <id> [Type = <type>] [Energy = <erg\_bin>]**  **[Scope = <params>] [Bound = <params>]** |

In which,

* **MeshTally** is the key word of the input card of mesh tally.
* **Id** is the identifier of the mesh tally, for easy consulting and output.
* **Type** option determines the type of counting. **Type = 1** means flux of cell, **Type = 2** means power of cell, **Type = 3** means fission rate of cell, **Type = 4** means absorption rate of cell.
* **Energy** card determines the energy intervals for counting in the divisive energy groups. The parameter is the interval points (Mev) between energy groups. For example, “**Energy = 6.25E-7 20**” means the counting intervals are from 0 to 0.625ev, 0.625ev to 20Mev and 20Mev to the positive infinity, totally 3 intervals. Meanwhile, the total counting will be given. Especially, for multi-group critical calculations, “**Energy = -1**” means the cross section libraries are adopted for the division of energy structure. Only the total counting rate will be counted, if there is no **Energy** optionin the input card.
* **Scope** option determines the numbers of meshes in the x, y, z directions. Especially, the parameter “-1” means a single level of infinite meshes in this direction[[2]](#footnote-2).
* **Bound** option determines the boundary of meshes in the x, y, z directions, such as “Bound = x\_min x\_max y\_min y\_max z\_min z\_max”. If there’s only a level of meshes in a certain direction, the parameters of **Bound** option have no actual meaning.

## 6.3 Cross-section tally

The cross-section tally counts the one-group cross-sections or multi-group cross-sections (the tally for multi-group cross-sections is not supported in this release of RMC) of the specified reaction types of all the nuclides of specified material in the specified cells. The input card of cross-section tally is:

|  |
| --- |
| CsTally <id> [Cell = <cell\_vector>] [Mat = <mat>] [Energy = <erg\_bin>]  [MT = <mt\_list\_1, mt\_list\_2, …>] |

In which,

* **CsTally** is the key word of the input card of the cross-section tally.
* **Id** is the identifier of the tally.
* **Cell** option determines the cell to be counted. Notice that different from the cell tally, the input of the cross-section tally is a single cell vector which has to be the bottomed cell. Moreover, **Cell** option must be different in different **CsTally** card.
* **Mat** option determines the material to be counted. This material can be different from the actual filling material in the tally cell. If users need to count the cross-sections of different nuclides in the same cell, these nuclides can be defined in the same material.
* **Energy** option determines the energy intervals for counting in the divisive energy groups. The parameter is the interval points (Mev) between energy groups. For example, “**Energy = 6.25E-7 20**” means the counting intervals are from 0 to 0.625ev, 0.625ev to 20Mev and 20Mev to the positive infinity, totally 3 intervals. Meanwhile, the total counting will be given. Especially, for multi-group critical calculations, “**Energy = -1**” means the cross section libraries are adopted for the division of energy structure. Only the total counting rate will be counted, if there is no **Energy** optionin the input card. The tally for multi-group cross-sections is not supported in this release of RMC, thus the **Energy** group is closed.
* **MT** option determines the reaction types of each nuclide. A nuclide can have multiple reaction types, which are separated by spaces, while commas are used to separate different nuclides, such as “MT = 16 17, 102, -6, 107”. The corresponding relations between reaction types and identifiers can be referred to manuals of ENDF/B. Table 6-1 has given out the identifiers of some common reaction types.

Table 6-1 Relationships between reaction types and identifiers

(Only parts of the ENDF reaction types)

|  |  |  |
| --- | --- | --- |
| MT number | Reaction types | Comments |
| **-1** | Total cross-section | For continuous-energy ACE cross-sections, the doppler broadening is adopted to adjust the elastic scattering cross-sections and the total cross-sections, when the cross-section temperature does not match the cell temperature.  The cross-sections to be counted here is the one after adjusting. |
| **-2** | Absorption | Fission not included |
| **-3** | Elastic scattering |  |
| **-6** | Fission |  |
| **16** | (n，2n) | Only for continuous-energy ACE cross-sections |
| **17** | (n，3n) |
| **102** | (n，γ) |
| **103** | (n，p) |
| **107** | (n，α) |

The following input card counts the one-group cross-sections of 3 nuclides in a certain cell (1 > 221 > 145 > 35), including the fission cross-sections of U235, absorption and fission cross-sections of U238 and the (n，γ) cross-sections of O16.

|  |
| --- |
| MATERIAL  mat 2 -10.196  92235.30c 0.03  92238.30c 0.97  8016.30c 2.0  CsTally 1 Cell = 1 > 221 > 145 > 35 Mat = 2 MT = -6 , -2 -6 , 102 |

## 6.4 Tally acceleration

Aiming at the mesh tally with large sum of meshes and cross-section tally with a large sum of nuclides, the acceleration function is provided by RMC, whose input card is:

|  |
| --- |
| **AcceTally [Map = <flag>] [Union = <flag>]** |

In which,

* **AcceTally** is the key word of the input card of the tally acceleration
* **Map** option determines whether the fast location method of cells is used to handle the cell tally. **Map = 1** (by default) means the fast location method is on, while **Map = 0** means it is off. If the cell of the cell tally has large sum of meshes, the calculation time can be significantly reduced when the **Map** function is on.
* **Union** option determines whether the method of uniform energy frame is used to handle the cell tally. **Union= 1** means the method is on, while **Union= 0** (by default)means it is off.

If the cell of the cross-section tally has a large sum of nuclides, the calculation time can be reduced with the method of uniform energy frame, while at the expense of missing of variance information and additional consumption of memory.

## 6.5 Input examples of the tally block

### 6.5.1 Tally in axial sections of PWR fuel pin

Example 6-1 is a PWR fuel pin, which is divided into 10 sections in the axial direction. Two cell tallies, a mesh tally and a cross-section tally have been defined in the tally block respectively.

The 1st cell tally (CellTally 1) counts the flux of different energy groups in the fuel regions and moderator regions of each axial section, while the 2nd cell tally (CellTally 1) counts the total fission rate in the fuel regions of each axial section. The mesh tally (MeshTally 1) counts the flux distribution of different energy groups in the 100 axial sections. The cross-section (CsTally 1) tally counts the one-group cross-sections of each nuclide in the 5th section of fuel region: the fission cross-section (-6) and (n，γ) cross-section (102) of U235, the fission cross-section (-6), n-2n cross-section (16) and (n，γ) cross-section (102) of U238, as well as the n-a cross-section (107) of O16.

Example 6-1

|  |
| --- |
| ///// PWR pin divided into 10 nodes in axial. Qiu Yishu 2012-09-15 //////  UNIVERSE 0  cell 1 6 & -7 & 8 & -9 & 10 & -11 Fill = 8 // Pin inside  cell 2 -6 : 7 : -8 : 9 : -10 : 11 void = 1 // Pin outside  UNIVERSE 8 lat = 1 pitch = 1 1 0.5 scope = 1 1 10  fill = 1 \* 10  UNIVERSE 1 move = 0.63 0.63 0 // Fuel rod  cell 3 -1 mat = 1 // Fuel  cell 4 1 & -2 mat = 3 // Air  cell 5 2 & -3 mat = 4 // Zr  cell 6 3 mat = 5 // water  SURFACE  surf 1 cz 0.4096  surf 2 cz 0.4178  surf 3 cz 0.4750  surf 6 px 0 bc = 1  surf 7 px 1.26 bc = 1  surf 8 py 0 bc = 1  surf 9 py 1.26 bc = 1  surf 10 pz 0 bc = 1  surf 11 pz 5 bc = 1  MATERIAL  mat 1 -10.196  92235.30c 6.9100E-03  92238.30c 2.2062E-01  8016.30c 4.5510E-01  mat 3 -0.001  8016.30c 3.76622E-5  mat 4 -6.550  40000.60c -98.2  mat 5 9.9977E-02  1001.30c 6.6643E-02  8016.30c 3.3334E-02  sab 5 lwtr.60t  CeAce ErgBinHash = 0 pTable = 0  CRITICALITY  PowerIter population = 1000 30 200 // keff0 = 1.0  InitSrc point = 0.63 0.63 2.75  Tally  CellTally 1 type = 1 energy = 6.25E-7 20  cell = 1 > 1: 10 > 3  1 > 1: 10 > 6  CellTally 2 type = 3 integral = 10  cell = 1 > 1: 10 > 3  MeshTally 1 type = 1 energy = 6.25E-7 20  Scope = 1 1 100  Bound = 0 1.26 0 1.26 0 5  CsTally 1 cell = 1 > 5 > 3  mat = 1  mt = -6 102 , -6 16 102, 102 |

### 6.5.2 Large-scale tally for Hoogenboom whole core benchmark

This example is a PWR whole core benchmark. The reactor core contains 241 same fuel assemblies, which contain 17×17 cell per assembly. Each cell is divided into 100 axial levels. 5 cell tallies and 2 cross-section tallies have been defined in the tally block. The 1st cell tally counts the flux in the fuel region of the whole core, and the 2nd one counts the flux of assemblies in 3 different positions (0, 0), (3, 2) and (-3, 2). The 3rd cell tally counts the power in the fuel regions of fuel assemblies in 3 different positions (0, 0), (3, 2) and (-3, 2). The 4th cell tally counts the fission rate of two different fuel pins. The 5th cell tally counts the absorption rate of different energy groups in three different axial nodes of a certain fuel pin. The 1st cross-section tally counts the one-group cross-sections of each nuclide in a certain axial node: the elastic scattering cross-section (-3) of H1, the total cross-section (-1) and absorption cross-section (-2) of O16, the elastic scattering cross-section (-3) of B10 and the (n，γ) cross-section (102) of B11. This material is one kind of the material which is used for the practical problem. The 2nd cross-section tally counts the one-group cross-sections of each nuclide in a certain axial node: the total cross-section (-1), absorption cross-section (-2), elastic scattering cross-section (-3), fission cross-section (-6), n-2n cross-section (16), (n，γ) cross-section (102) and n-a cross-section (107) of N14. This material is a “virtual” material, which will not be used for the critical calculation in practice.

Example 6-2

|  |
| --- |
| ///// Tally of MC full-core benchmark. Qiu Yishu 2012-09-15 /////  universe 0  cell 1 -11 : 19 : 9 mat = 0 void = 1 // outside core  cell 2 11 & -19 & 8 & -9 mat = 1 vol = 1.3575E+07 // reactor vessel  cell 3 12 & -18 & 7 & -8 mat = 2 vol = 1.1393E+07 // downcomer  cell 6 18 & -19 & -8 mat = 3 vol = 1.3180E+06 // upper core plate region  cell 7 11 & -12 & -8 mat = 4 vol = 4.9424E+06 // lower core plate region  cell 8 17 & -18 & -6 mat = 5 vol = 1.3268E+06 // top nozzle region  cell 9 12 & -13 & -6 mat = 6 vol = 6.6339E+05 // bottom nozzle region  cell 10 16 & -17 & -6 mat = 7 vol = 2.2113E+06 // top FA region  cell 11 13 & -14 & -6 mat = 8 vol = 1.1056E+06 // bottom FA region  cell 12 16 & -18 & 6 & -7 mat = 9 vol = 8.5323E+05 // radial hot water  cell 13 12 & -14 & 6 & -7 mat = 10 vol = 4.2662E+05 // radial cold water  cell 14 14 & -16 & -7 fill= 1 vol = 5.0225E+07  // assembly zone  universe 1 move = -224.91 -224.91 -183 lat = 1 pitch = 21.42 21.42 1 scope = 21 21 1 fill=  2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2  2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2  2 2 2 2 2 2 2 3 3 3 3 3 3 3 2 2 2 2 2 2 2  2 2 2 2 2 3 3 3 3 3 3 3 3 3 3 3 2 2 2 2 2  2 2 2 2 3 3 3 3 3 3 3 3 3 3 3 3 3 2 2 2 2  2 2 2 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 2 2 2  2 2 2 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 2 2 2  2 2 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 2 2  2 2 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 2 2  2 2 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 2 2  2 2 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 2 2  2 2 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 2 2  2 2 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 2 2  2 2 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 2 2  2 2 2 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 2 2 2  2 2 2 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 2 2 2  2 2 2 2 3 3 3 3 3 3 3 3 3 3 3 3 3 2 2 2 2  2 2 2 2 2 3 3 3 3 3 3 3 3 3 3 3 2 2 2 2 2  2 2 2 2 2 2 2 3 3 3 3 3 3 3 2 2 2 2 2 2 2  2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2  2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2  universe 2 fill = // single reflector lattice  cell 21 16 mat=9 // upper radial reflector  cell 22 -16 mat=10 // lower radial reflector  universe 3 lat = 1 pitch = 1.26 1.26 1 scope = 17 17 1 fill =  4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4  4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4  4 4 4 4 4 5 4 4 5 4 4 5 4 4 4 4 4  4 4 4 4 5 4 4 4 4 4 4 4 5 4 4 4 4  4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4  4 4 5 4 4 5 4 4 5 4 4 5 4 4 5 4 4  4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4  4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4  4 4 5 4 4 5 4 4 5 4 4 5 4 4 5 4 4  4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4  4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4  4 4 5 4 4 5 4 4 5 4 4 5 4 4 5 4 4  4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4  4 4 4 4 5 4 4 4 4 4 4 4 5 4 4 4 4  4 4 4 4 4 5 4 4 5 4 4 5 4 4 4 4 4  4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4  4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4  universe 4 lat=1 pitch = 1 1 3.66 scope = 1 1 100 fill =  6\*50 7\*50  universe 5 lat=1 pitch = 1 1 3.66 scope = 1 1 100 fill =  8\*50 9\*50  universe 6 move = 0.63 0.63 1.83  cell 100 -1 mat=11  cell 24 1 & -2 mat=12  cell 25 2 mat=2  universe 7 move = 0.63 0.63 1.83  cell 101 -1 mat =11  cell 27 1 & -2 mat =12  cell 28 2 mat =22  universe 8 move = 0.63 0.63 1.83  cell 29 -3 mat =2  cell 30 3 & -4 mat =12  cell 31 4 mat =2  universe 9 move = 0.63 0.63 1.83  cell 32 -3 mat =22  cell 33 3 & -4 mat =12  cell 34 4 mat =22  SURFACE  surf 1 cz 0.41  surf 2 cz 0.475  surf 3 cz 0.56  surf 4 cz 0.62  surf 5 cz 1.26  surf 6 cz 187.6  surf 7 cz 209  surf 8 cz 229  surf 9 cz 249 bc =1 // radial boundary  surf 11 pz -229 bc =1 // bottom boundary  surf 12 pz -199  surf 13 pz -193  surf 14 pz -183  surf 15 pz 0  surf 16 pz 183  surf 17 pz 203  surf 18 pz 215  surf 19 pz 223 bc =1 // upper boundary  MATERIAL  mat 1 -7.9 // reactor vessel  26054.30c -5.4371E-02 26056.30c -8.8501E-01 26057.30c -2.0801E-02  26058.30c -2.8216E-03 28058.30c -6.7198E-03 28060.30c -2.6776E-03  28061.30c -1.1830E-04 28062.30c -3.8350E-04 28064.30c -1.0080E-04  25055.30c -1.0000E-02 42000.60c -6.0000E-03 28058.30c -3.6746E-03  28060.30c -1.9336E-04 28061.30c -1.3200E-04 24050.30c -1.0435E-04  24052.30c -2.0925E-03 24053.30c -2.4185E-04 24054.30c -6.1325E-05  6000.30c -2.5000E-03 29063.30c -1.3696E-03 29065.30c -6.3040E-04  mat 2 -0.74 // Borated water below midplane  1001.30c 2.0000E+00 8016.30c 1.0000E+00 5010.30c 6.4900E-04  5011.30c 2.6890E-03  sab 2 lwtr.60t  mat 22 -0.66 // Borated water above midplane  1001.30c 2.0000E+00 8016.30c 1.0000E+00 5010.30c 6.4900E-04  5011.30c 2.6890E-03  sab 22 lwtr.60t  mat 3 -4.28 // top core plate region  1001.30c -8.6117E-03 8016.30c -6.8337E-02 5010.30c -2.7764E-05  5011.30c -1.2648E-04 26054.30c -3.5954E-02 26056.30c -5.8522E-01  26057.30c -1.3755E-02 26058.30c -1.8658E-03 28058.30c -5.5815E-02  28060.30c -2.2240E-02 28061.30c -9.8261E-04 28062.30c -3.1854E-03  28064.30c -8.3725E-04 25055.30c -1.8458E-02 28058.30c -8.4783E-03  28060.30c -4.4613E-04 28061.30c -3.0456E-04 24050.30c -7.3191E-03  24052.30c -1.4677E-01 24053.30c -1.6963E-02 24054.30c -4.3013E-03  mat 4 -7.184 // bottom plate region  1001.30c -1.1505E-03 8016.30c -9.1296E-03 5010.30c -3.7092E-06  5011.30c -1.6897E-05 26054.30c -3.8556E-02 26056.30c -6.2759E-01  26057.30c -1.4750E-02 26058.30c -2.0009E-03 28058.30c -5.9855E-02  28060.30c -2.3850E-02 28061.30c -1.0537E-03 28062.30c -3.4159E-03  28064.30c -8.9785E-04 25055.30c -1.9794E-02 28058.30c -9.0920E-03  28060.30c -4.7842E-04 28061.30c -3.2660E-04 24050.30c -7.8489E-03  24052.30c -1.5739E-01 24053.30c -1.8191E-02 24054.30c -4.6127E-03  mat 5 -1.746 // top nozzle region  1001.30c -3.5887E-02 8016.30c -2.8478E-01 5010.30c -1.1570E-04  5011.30c -5.2708E-04 26054.30c -2.6440E-02 26056.30c -4.3037E-01  26057.30c -1.0115E-02 26058.30c -1.3721E-03 28058.30c -4.1046E-02  28060.30c -1.6355E-02 28061.30c -7.2261E-04 28062.30c -2.3425E-03  28064.30c -6.1571E-04 25055.30c -1.3574E-02 28058.30c -6.2349E-03  28060.30c -3.2808E-04 28061.30c -2.2397E-04 24050.30c -5.3825E-03  24052.30c -1.0793E-01 24053.30c -1.2475E-02 24054.30c -3.1632E-03  mat 6 -2.53 // bottom nozzle region  1001.30c -2.4501E-02 8016.30c -1.9443E-01 5010.30c -7.8992E-05  5011.30c -3.5985E-04 26054.30c -3.0411E-02 26056.30c -4.9501E-01  26057.30c -1.1635E-02 26058.30c -1.5782E-03 28058.30c -4.7211E-02  28060.30c -1.8812E-02 28061.30c -8.3114E-04 28062.30c -2.6944E-03  28064.30c -7.0819E-04 25055.30c -1.5613E-02 28058.30c -7.1713E-03  28060.30c -3.7736E-04 28061.30c -2.5761E-04 24050.30c -6.1909E-03  24052.30c -1.2414E-01 24053.30c -1.4348E-02 24054.30c -3.6383E-03  mat 7 -1.762 // top FA region  1001.30c -2.9286E-02 8016.30c -2.3239E-01 5010.30c -9.4416E-05  5011.30c -4.3012E-04 40000.60c -7.3780E-01  mat 8 -3.044 // bottom FA region  1001.30c -1.6291E-02 8016.30c -1.2928E-01 5010.30c -5.2523E-05  5011.30c -2.3927E-04 40000.60c -7.3780E-01  mat 9 -4.28 // upper radial reflector  1001.30c -8.6117E-03 8016.30c -6.8337E-02 5010.30c -2.7764E-05  5011.30c -1.2648E-04 26054.30c -3.5954E-02 26056.30c -5.8522E-01  26057.30c -1.3755E-02 26058.30c -1.8658E-03 28058.30c -5.5815E-02  28060.30c -2.2240E-02 28061.30c -9.8261E-04 28062.30c -3.1854E-03  28064.30c -8.3725E-04 25055.30c -1.8458E-02 28058.30c -8.4783E-03  28060.30c -4.4613E-04 28061.30c -3.0456E-04 24050.30c -7.3191E-03  24052.30c -1.4677E-01 24053.30c -1.6963E-02 24054.30c -4.3013E-03  mat 10 -4.32 // lower radial reflector  1001.30c -9.5661E-03 8016.30c -7.5911E-02 5010.30c -3.0841E-05  5011.30c -1.4050E-04 26054.30c -3.5621E-02 26056.30c -5.7981E-01  26057.30c -1.3628E-02 26058.30c -1.8485E-03 28058.30c -5.5298E-02  28060.30c -2.2034E-02 28061.30c -9.7351E-04 28062.30c -3.1559E-03  28064.30c -8.2950E-04 25055.30c -1.8287E-02 28058.30c -8.3998E-03  28060.30c -4.4200E-04 28061.30c -3.0174E-04 24050.30c -7.2514E-03  24052.30c -1.4541E-01  24053.30c -1.6806E-02 24054.30c -4.2615E-03  mat 11 -10.062 // fuel  92234.30c 4.9476E-06 92235.30c 4.8218E-04 92236.30c 9.0402E-05  92238.30c 2.1504E-02 93237.30c 7.3733E-06 94238.30c 1.5148E-06  94239.30c 1.3955E-04 94240.30c 3.4405E-05 94241.30c 2.1439E-05  94242.30c 3.7422E-06 95241.30c 4.5041E-07 95242.30c 9.2300E-09  96243.30c 4.7878E-07 96242.30c 1.0485E-07 96243.30c 1.4300E-09  96244.30c 8.8760E-08 96245.30c 3.5300E-09 42095.30c 2.6497E-05  43099.30c 3.2772E-05 44101.30c 3.0742E-05 44103.30c 2.3505E-06  47109.30c 2.0009E-06 54135.30c 1.0800E-08 55133.30c 3.4612E-05  60143.30c 2.6078E-05 60145.30c 1.9898E-05 62147.30c 1.6128E-06  62149.30c 1.1627E-07 62150.30c 7.1727E-06 62151.30c 5.4947E-07  62152.30c 3.0221E-06 63153.30c 2.6209E-06 64155.30c 1.5400E-09  8016.30c 4.5737E-02  mat 12 -5.77 // cladding composition also the guide tube ma  40000.60c -7.3780E-01  mat 13 1.0 // a material which is not used in the problem  7014.30c 1.0  CRITICALITY  PowerIter population = 100000 250 1250 // keff0 = 1.0  InitSrc point = 1.26 0 0.1  ParallelBank 1  Tally  celltally 1 Type = 1 filter = 1 0 0 0 1 integral = 2  cell = 14 > 0 > 0 > 0 > 100:101  celltally 2 Type = 1 filter = 1 1  cell = 14 > 221  14 > 266  14 > 260  celltally 3 Type = 2 filter = 1 1 0 0 1 integral = 2\*3  cell = 14 > 221 > 0 > 0 > 100:101  14 > 266 > 0 > 0 > 100:101  14 > 260 > 0 > 0 > 100:101  celltally 4 Type = 3 filter = 1 1 1  Cell = 14 > 266 > 1  14 > 266 > 164  celltally 5 Type = 4 Energy=6.25E-07  Cell = 14 > 266 > 164 > 1 > 100  14 > 266 > 164 > 50 > 100  14 > 266 > 164 > 100 > 101  CsTally 6 Cell = 14 > 266 > 164 > 49 > 100  Mat = 2 MT = -3, -1 -2, -3, 102  csTally 7 Cell = 14 > 266 > 164 > 51 > 101  Mat = 13 MT = -1 -2 -3 -6 16 17 102 103 107 |

# Chapter 7 Diagnosis and Acceleration of Source Convergence

The fission source iteration method is adopted in the critical calculation of MC program. The convergence speed of fission source distribution depends on the dominance ratio of the system. For the systems with large dominance ratio (such as whole core, spent fuel pool), hundreds or even thousands of generations have to be spent on the fission source convergence, leading to large sum of time spent on the inactive generations which have no direct contributions to the statistical results.

## 7.1 Shannon Entropy statistics

The source convergence block of RMC provides the function of Shannon Entropy statistics, qualitatively reflecting the convergence trend of the fission source distribution, so as to help users select the number of neutrons in the inactive generations reasonably.

Shannon Entropy can be defined as:



In which, is the proportion of fission source in the ith mesh.

The divisions of the Shannon Entropy meshes can be defined by the following input card:

|  |
| --- |
| SeMesh [Scope = < xNum yNum zNum >]  [Bound = <xMin xMax yMin yMax zMin zMax>] |

In which,

* **SeMesh** is the key word of the input card of Shannon Entropy mesh.
* **Scope** option determines the number of Shannon Entropy meshes in the x, y, z directions. **xNum**, **yNum**, **zNum** must be positive integers.
* **Bound** option determines the boundary of Shannon Entropy meshes in the x, y, z directions. If the number of meshes in a certain direction of the **Scope** option is “-1”, the default boundary of meshes will be (-∞, +∞). In this case, the boundary in a certain direction of the **Bound** option has no practical meaning. Users should notice that the regions of meshes must cover the regions of neutron tracking, or the statistical results of Shannon Entropy will go wrong.

## 7.2 Fission matrix statistics

The fission matrix statistics for the source convergence block will also be provided in RMC. Based on the fission matrix, the undersampling of Monte Carlo can be diagnosed, so as to figure out the dominance ratio. Besides the calculation of the dominance ratio with fission matrix calculation, RMC also supports the method based on the error propagation matrix, which will not be provided in this release of RMC.

The fission matrix can be defined as: Each element  of the fission matrix *F* means the probability that the neutrons produced in region *j* produce the fission neutrons in region *i*.



The input card of fission matrix statistics is similar to that of Shannon Entropy statistics:

|  |
| --- |
| FmMesh [Scope = < xNum yNum zNum >]  [Bound = <xMin xMax yMin yMax zMin zMax>] |

In which,

* **FmMesh** is the key word of the input card of the fission matrix meshes.
* **Scope** option determines the number of fission matrix meshes in the x, y, z directions. **xNum**, **yNum**, **zNum** must be positive integers. The size of the fission matrix=( **xNum**\***yNum\*zNum**) 2, while the meshes in vary small scale will reduce the calculation efficiency.
* **Bound** option determines the boundary of fission matrix meshes in the x, y, z directions. If the number of meshes in a certain direction of the **Scope** option is “-1”, the default boundary of meshes will be (-∞, +∞). In this case, the boundary in a certain direction of the **Bound** option has no practical meaning. Users should notice that the regions of meshes must cover the regions of neutron tracking, or the statistical results of fission matrix will go wrong.

## 7.3 Source convergence acceleration

RMC adopts the asymptotical super-history method and asymptotical Wielandt method to accelerate the source convergence, reducing the inactive generations. This release of RMC only supports the super-history method for acceleration.

The input card of the source convergence acceleration is:

|  |
| --- |
| **AcceFsc [Factor = <f(1) p(1) f(2) p(2) …f(n) p(n)>]**  **[AutoFactor = <inactive\_cycle>]** |

In which,

* **AcceFsc** is the key word of the input card of the Shannon Entropy meshes.
* **Factor** option and **AutoFactor** option are used to determine the parameters of asymptotical super-history acceleration. Both two methods will be discussed in the following respectively.

### 7.3.1 Factor option

**Factor** option is for the user-defined acceleration parameters. f(i) in the input card is the acceleration factor, and p(i) is the acceleration period, which means: use the acceleration factor f(1) in first generation p(1), and use f(2) in the next generation p(2), and so on. The theory of the asymptotical super-history method for acceleration will not be introduced in detail, while only the two parameters of acceleration factor and acceleration period will be introduced in the following:

The effect of acceleration is more obvious, when the acceleration factor f(i) is bigger, while with a bigger statistical fluctuation probably. Users need to define a group of asymptotical decreased acceleration factor {f(i)}, such as “16 → 8 → 4 → 2”. Notice that the acceleration factor should not be too large (less than 20 for suggestion), or it will lead to the unstability.

The acceleration period p (i > 1) is normally set to 5-10 generations. The first acceleration period is normally set bigger, for its acceleration factor is the largest one which plays the major role of acceleration.

The asymptotical super-history method for acceleration influences on the first  inactive generations, whose acceleration effect is roughly equivalent to the  generations without acceleration. For example, 200 inactive generations are needed for a whole core critical calculation without acceleration. The equivalent effect can be achieved by “factor = 16 10 8 5 4 5 2 5” to accelerate the first 10+5+5+5=25 inactive generations.

### 7.3.2 AutoFactor option

As the alternative of **Factor** option, RMC supports the **AutoFactor** option which can generate the parameters of source convergence acceleration automatically. For the common users, it is advised to replace the user-defined input of **Factor** option with **AutoFactor** option. In this option, users can set the number of inactive generations **inactive\_cycle** without the acceleration, then the parameters array will be generated automatically in the program. Assuming the number of inactive generations without the acceleration is N, the inactive generations with the **AutoFactor** option is about:



For example, if 300 generations are needed before convergence for the whole core calculation, then with the automatic source convergence acceleration, the required inactive generations can be set as: . If the inactive generations N set in the **AutoFactor** option are less than 30, the source convergence acceleration will be closed.

### 7.3.3 Precautions of the source convergence acceleration

When using the source convergence acceleration, the matched parameters of the PowerIter card in the criticality calculation block must be reasonable, including:

1. Set a reasonable initial effective multiplication factor in the Keff0 option, which is close to the actual Keff.
2. Set the number of particle big enough in the Population option. For the whole core critical calculation, the particles per generation are advised to be more than 100,000.
3. Set a reasonable number of inactive generations in the Population option. The number of inactive generations should be not less than the generations of source convergence acceleration.

## 7.4 Input examples of the source convergence block

### 7.4.1 Source convergence acceleration of the OECD benchmark

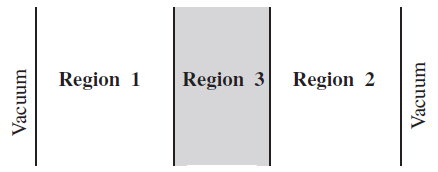


Figure 7-1 benchmark of OECD MC source convergence

Figure 7-1 describes a benchmark of OECD MC source convergence problem. This benchmark is a weak coupling system of three 1D plates, i.e. two 20cm thick region of fuel in the both sides and a 30cm region of water in the middle. The initial source locates in the middle of the left fuel region, with about 1000 inactive generations for the normal source iteration convergence. With the source convergence acceleration of RMC, the inactive generations can be reduced less than 100 generations. In the source convergence block of example 7-1, the number of Shannon entropy meshes is set to 70, which are 1.0cm width. As large sums of particles are required for the simulation of this example, the parallel computers are advised.

Example 7-1

|  |
| --- |
| ///// OECD MC convergence benchmark 3 . SHE Ding 2012-09-12 /////  UNIVERSE 0  cell 1 1 & -2 mat = 1  cell 2 2 & -3 mat = 2  cell 3 3 & -4 mat = 1  cell 4 -1 : 4 void = 1  SURFACE  surf 1 px 0  surf 2 px 20  surf 3 px 50  surf 4 px 70  MATERIAL  mat 1 9.9487E-02  92235.30c 7.6864E-05  92238.30c 6.8303E-04  8016.30c 3.7258E-02  1001.30c 5.9347E-02  7014.30c 2.1220E-03  mat 2 1.0006E-01  1001.30c 6.6706E-02  8016.30c 3.3353E-02  CRITICALITY  PowerIter population = 500000 100 1000  InitSrc point = 10 0 0  Tally  Celltally 1 type = 1 cell = 1 3  CONVERGENCE  SeMesh Scope = 70 -1 -1 Bound = 0 70 0 1 0 1  FmMesh Scope = 70 -1 -1 Bound = 0 70 0 1 0 1  AcceFsc Autofactor = 1000 |

### 7.4.2 Source convergence acceleration of the Hoogenboom whole core benchmark

Example 7-2 describes the Hoogenboom whole core benchmark. The number of inactive generations for normal source iteration convergence is 250, while it can be set to 35 inactive generations with the parameters of the automatic source convergence acceleration. The Shannon entropy meshes (21×21) built in the assemblies have been defined in the input block of source convergence, to help users diagnose the trend of source convergence. As large sums of particles are required for the simulation of this example, the parallel computers are advised.

Example 7-2

|  |
| --- |
| /// Convergence Acceleration of MC full core Benchmark. SHE Ding 2012-09-12 ///  Universe 0  cell 1 -11 : 19 : 9 mat = 0 void = 1 // outside core  cell 2 11 & -19 & 8 & -9 mat = 1 vol = 1.3575E+07 // reactor vessel  cell 3 12 & -18 & 7 & -8 mat = 2 vol = 1.1393E+07 // downcomer  cell 6 18 & -19 & -8 mat = 3 vol = 1.3180E+06 // upper core plate region  cell 7 11 & -12 & -8 mat = 4 vol = 4.9424E+06 // lower core plate region  cell 8 17 & -18 & -6 mat = 5 vol = 1.3268E+06 // top nozzle region  cell 9 12 & -13 & -6 mat = 6 vol = 6.6339E+05 // bottom nozzle region  cell 10 16 & -17 & -6 mat = 7 vol = 2.2113E+06 // top FA region  cell 11 13 & -14 & -6 mat = 8 vol = 1.1056E+06 // bottom FA region  cell 12 16 & -18 & 6 & -7 mat = 9 vol = 8.5323E+05 // radial hot water  cell 13 12 & -14 & 6 & -7 mat = 10 vol = 4.2662E+05 // radial cold water  cell 14 14 & -16 & -7 fill= 1 vol = 5.0225E+07  // assembly zone  Universe 1 move = -224.91 -224.91 -183 lat = 1 pitch = 21.42 21.42 1 scope = 21 21 1 fill=  2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2  2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2  2 2 2 2 2 2 2 3 3 3 3 3 3 3 2 2 2 2 2 2 2  2 2 2 2 2 3 3 3 3 3 3 3 3 3 3 3 2 2 2 2 2  2 2 2 2 3 3 3 3 3 3 3 3 3 3 3 3 3 2 2 2 2  2 2 2 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 2 2 2  2 2 2 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 2 2 2  2 2 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 2 2  2 2 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 2 2  2 2 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 2 2  2 2 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 2 2  2 2 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 2 2  2 2 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 2 2  2 2 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 2 2  2 2 2 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 2 2 2  2 2 2 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 2 2 2  2 2 2 2 3 3 3 3 3 3 3 3 3 3 3 3 3 2 2 2 2  2 2 2 2 2 3 3 3 3 3 3 3 3 3 3 3 2 2 2 2 2  2 2 2 2 2 2 2 3 3 3 3 3 3 3 2 2 2 2 2 2 2  2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2  2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2  Universe 2 fill = // single reflector lattice  cell 21 16 mat=9 // upper radial reflector  cell 22 -16 mat=10 // lower radial reflector  Universe 3 lat = 1 pitch = 1.26 1.26 1 scope = 17 17 1 fill =  4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4  4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4  4 4 4 4 4 5 4 4 5 4 4 5 4 4 4 4 4  4 4 4 4 5 4 4 4 4 4 4 4 5 4 4 4 4  4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4  4 4 5 4 4 5 4 4 5 4 4 5 4 4 5 4 4  4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4  4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4  4 4 5 4 4 5 4 4 5 4 4 5 4 4 5 4 4  4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4  4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4  4 4 5 4 4 5 4 4 5 4 4 5 4 4 5 4 4  4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4  4 4 4 4 5 4 4 4 4 4 4 4 5 4 4 4 4  4 4 4 4 4 5 4 4 5 4 4 5 4 4 4 4 4  4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4  4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4  Universe 4 lat=1 pitch = 1 1 3.66 scope = 1 1 100 fill =  6\*50 7\*50  Universe 5 lat=1 pitch = 1 1 3.66 scope = 1 1 100 fill =  8\*50 9\*50  Universe 6 move = 0.63 0.63 1.83  cell 100 -1 mat=11  cell 24 1 & -2 mat=12  cell 25 2 mat=2  Universe 7 move = 0.63 0.63 1.83  cell 101 -1 mat =11  cell 27 1 & -2 mat =12  cell 28 2 mat =22  Universe 8 move = 0.63 0.63 1.83  cell 29 -3 mat =2  cell 30 3 & -4 mat =12  cell 31 4 mat =2  Universe 9 move = 0.63 0.63 1.83  cell 32 -3 mat =22  cell 33 3 & -4 mat =12  cell 34 4 mat =22  SURFACE  surf 1 cz 0.41  surf 2 cz 0.475  surf 3 cz 0.56  surf 4 cz 0.62  surf 5 cz 1.26  surf 6 cz 187.6  surf 7 cz 209  surf 8 cz 229  surf 9 cz 249 bc =1 // radial boundary  surf 11 pz -229 bc =1 // bottom boundary  surf 12 pz -199  surf 13 pz -193  surf 14 pz -183  surf 15 pz 0  surf 16 pz 183  surf 17 pz 203  surf 18 pz 215  surf 19 pz 223 bc =1 // upper boundary  MATERIAL  mat 1 -7.9 // reactor vessel  26054.30c -5.4371E-02 26056.30c -8.8501E-01 26057.30c -2.0801E-02  26058.30c -2.8216E-03 28058.30c -6.7198E-03 28060.30c -2.6776E-03  28061.30c -1.1830E-04 28062.30c -3.8350E-04 28064.30c -1.0080E-04  25055.30c -1.0000E-02 42000.60c -6.0000E-03 28058.30c -3.6746E-03  28060.30c -1.9336E-04 28061.30c -1.3200E-04 24050.30c -1.0435E-04  24052.30c -2.0925E-03 24053.30c -2.4185E-04 24054.30c -6.1325E-05  6000.30c -2.5000E-03 29063.30c -1.3696E-03 29065.30c -6.3040E-04  mat 2 -0.74 // Borated water below midplane  1001.30c 2.0000E+00 8016.30c 1.0000E+00 5010.30c 6.4900E-04  5011.30c 2.6890E-03  sab 2 lwtr.60t  mat 22 -0.66 // Borated water above midplane  1001.30c 2.0000E+00 8016.30c 1.0000E+00 5010.30c 6.4900E-04  5011.30c 2.6890E-03  sab 22 lwtr.60t  mat 3 -4.28 // top core plate region  1001.30c -8.6117E-03 8016.30c -6.8337E-02 5010.30c -2.7764E-05  5011.30c -1.2648E-04 26054.30c -3.5954E-02 26056.30c -5.8522E-01  26057.30c -1.3755E-02 26058.30c -1.8658E-03 28058.30c -5.5815E-02  28060.30c -2.2240E-02 28061.30c -9.8261E-04 28062.30c -3.1854E-03  28064.30c -8.3725E-04 25055.30c -1.8458E-02 28058.30c -8.4783E-03  28060.30c -4.4613E-04 28061.30c -3.0456E-04 24050.30c -7.3191E-03  24052.30c -1.4677E-01 24053.30c -1.6963E-02 24054.30c -4.3013E-03  mat 4 -7.184 // bottom plate region  1001.30c -1.1505E-03 8016.30c -9.1296E-03 5010.30c -3.7092E-06  5011.30c -1.6897E-05 26054.30c -3.8556E-02 26056.30c -6.2759E-01  26057.30c -1.4750E-02 26058.30c -2.0009E-03 28058.30c -5.9855E-02  28060.30c -2.3850E-02 28061.30c -1.0537E-03 28062.30c -3.4159E-03  28064.30c -8.9785E-04 25055.30c -1.9794E-02 28058.30c -9.0920E-03  28060.30c -4.7842E-04 28061.30c -3.2660E-04 24050.30c -7.8489E-03  24052.30c -1.5739E-01 24053.30c -1.8191E-02 24054.30c -4.6127E-03  mat 5 -1.746 // top nozzle region  1001.30c -3.5887E-02 8016.30c -2.8478E-01 5010.30c -1.1570E-04  5011.30c -5.2708E-04 26054.30c -2.6440E-02 26056.30c -4.3037E-01  26057.30c -1.0115E-02 26058.30c -1.3721E-03 28058.30c -4.1046E-02  28060.30c -1.6355E-02 28061.30c -7.2261E-04 28062.30c -2.3425E-03  28064.30c -6.1571E-04 25055.30c -1.3574E-02 28058.30c -6.2349E-03  28060.30c -3.2808E-04 28061.30c -2.2397E-04 24050.30c -5.3825E-03  24052.30c -1.0793E-01 24053.30c -1.2475E-02 24054.30c -3.1632E-03  mat 6 -2.53 // bottom nozzle region  1001.30c -2.4501E-02 8016.30c -1.9443E-01 5010.30c -7.8992E-05  5011.30c -3.5985E-04 26054.30c -3.0411E-02 26056.30c -4.9501E-01  26057.30c -1.1635E-02 26058.30c -1.5782E-03 28058.30c -4.7211E-02  28060.30c -1.8812E-02 28061.30c -8.3114E-04 28062.30c -2.6944E-03  28064.30c -7.0819E-04 25055.30c -1.5613E-02 28058.30c -7.1713E-03  28060.30c -3.7736E-04 28061.30c -2.5761E-04 24050.30c -6.1909E-03  24052.30c -1.2414E-01 24053.30c -1.4348E-02 24054.30c -3.6383E-03  mat 7 -1.762 // top FA region  1001.30c -2.9286E-02 8016.30c -2.3239E-01 5010.30c -9.4416E-05  5011.30c -4.3012E-04 40000.60c -7.3780E-01  mat 8 -3.044 // bottom FA region  1001.30c -1.6291E-02 8016.30c -1.2928E-01 5010.30c -5.2523E-05  5011.30c -2.3927E-04 40000.60c -7.3780E-01  mat 9 -4.28 // upper radial reflector  1001.30c -8.6117E-03 8016.30c -6.8337E-02 5010.30c -2.7764E-05  5011.30c -1.2648E-04 26054.30c -3.5954E-02 26056.30c -5.8522E-01  26057.30c -1.3755E-02 26058.30c -1.8658E-03 28058.30c -5.5815E-02  28060.30c -2.2240E-02 28061.30c -9.8261E-04 28062.30c -3.1854E-03  28064.30c -8.3725E-04 25055.30c -1.8458E-02 28058.30c -8.4783E-03  28060.30c -4.4613E-04 28061.30c -3.0456E-04 24050.30c -7.3191E-03  24052.30c -1.4677E-01 24053.30c -1.6963E-02 24054.30c -4.3013E-03  mat 10 -4.32 // lower radial reflector  1001.30c -9.5661E-03 8016.30c -7.5911E-02 5010.30c -3.0841E-05  5011.30c -1.4050E-04 26054.30c -3.5621E-02 26056.30c -5.7981E-01  26057.30c -1.3628E-02 26058.30c -1.8485E-03 28058.30c -5.5298E-02  28060.30c -2.2034E-02 28061.30c -9.7351E-04 28062.30c -3.1559E-03  28064.30c -8.2950E-04 25055.30c -1.8287E-02 28058.30c -8.3998E-03  28060.30c -4.4200E-04 28061.30c -3.0174E-04 24050.30c -7.2514E-03  24052.30c -1.4541E-01  24053.30c -1.6806E-02 24054.30c -4.2615E-03  mat 11 -10.062 // fuel  92234.30c 4.9476E-06 92235.30c 4.8218E-04 92236.30c 9.0402E-05  92238.30c 2.1504E-02 93237.30c 7.3733E-06 94238.30c 1.5148E-06  94239.30c 1.3955E-04 94240.30c 3.4405E-05 94241.30c 2.1439E-05  94242.30c 3.7422E-06 95241.30c 4.5041E-07 95242.30c 9.2300E-09  96243.30c 4.7878E-07 96242.30c 1.0485E-07 96243.30c 1.4300E-09  96244.30c 8.8760E-08 96245.30c 3.5300E-09 42095.30c 2.6497E-05  43099.30c 3.2772E-05 44101.30c 3.0742E-05 44103.30c 2.3505E-06  47109.30c 2.0009E-06 54135.30c 1.0800E-08 55133.30c 3.4612E-05  60143.30c 2.6078E-05 60145.30c 1.9898E-05 62147.30c 1.6128E-06  62149.30c 1.1627E-07 62150.30c 7.1727E-06 62151.30c 5.4947E-07  62152.30c 3.0221E-06 63153.30c 2.6209E-06 64155.30c 1.5400E-09  8016.30c 4.5737E-02  mat 12 -5.77 // cladding composition also the guide tube ma  40000.60c -7.3780E-01  CRITICALITY  PowerIter population = 1000000 35 535 // keff0 = 1.0  InitSrc point = 1.26 0 0.1  ParallelBank 1  CONVERGENCE  SeMesh Scope = 21 21 1 Bound = -224.91 224.91 -224.91 224.91 -229 223  AcceFsc Autofactor = 250 |

# Chapter 8 Burnup Calculation

## 8.1 Overview of the Monte Carlo burnup calculation

The Monte Carlo burnup calculation is the coupling of Monte Carlo critical calculations and point burnup calculation. The traditional MC burnup programs (such as MCBurn、MCODE) normally adopt the third-party interface, calling the MC transportation program (such as MCNP) and point burnup program (such as ORIGEN-2) circularly by the method of external coupling .

The self-developed point burnup calculation code **DEPTH** is embedded in RMC. **DEPTH** adopts the matrix exponent method, which can solve the fine burnup chains containing about 1500 nuclides precisely and efficiently. The general procedure of burnup calculation in RMC is: firstly, the data such as neutron flux and one-group reaction cross-sections are acquired by the module of critical calculation (continuous-energy), and then passing these data to **DEPTH**. Secondly, **DEPTH** completesthe point burnup calculation to acquire the new nuclides density, and then passing them to the module of critical calculation. The total process of burnup calculation will be finished by the round-trip passing of data.

Compared with the traditional MC burnup programs (such as MCBurn、MCODE), the basic features of the burnup calculation function in RMC includes:

(1) The inner coupling burnup calculation module is included, which can handle the fine burnup chains containing about 1500 nuclides; the latest point burnup libraries of ORIGEN-S and ORIGEN-2 have been integrated.

(2) The burnup calculations containing the repeated structures are supported. Users do not need to determine the initial material in each burnup region, reducing the users’ input greatly.

(3) The “parallel critical and parallel point burnup” calculations for the large-scale burnup regions are supported. In the mode of parallel point burnup calculation, the burnup regions are equally assigned to each progress, which can execute the point burnup calculation separately.

## 8.2 Input card of the burnup block

The input card of burnup block includes:

|  |
| --- |
| **BURNUP**  **BurnCell < cell\_1 cell\_2 cell\_3 … >**  **TimeStep <time\_1 time\_2 … time\_n>**  **Power <power\_1 power\_2 … power\_n>**  **SubStep <sub\_num>**  **Inherent <fraction>**  **AceLib <lib\_type>**  **Strategy <flag>**  **Solver <flag>**  **Parallel <flag>**  **OutputCell <cell\_vector>** |

In which,

* **BURNUP** is the key word of burnup calculation block.
* **BurnCell** option determines the cells for burnup. The input parameters are the simple cell identifiers cell\_i (not the vectors of hierarchical cells). The program will automatically search all cell vectors whose bottomed cells is cell\_i, regarding them as independent burnup region, i.e. each burnup cell will be unfolded automatically to create the independent filling material, even though their initial material are the same. The function to unfold the independent burnup regions is very important for the burnup calculations of the repeated structures, reducing the users’ input burnden sgreatly.
* **TimeStep** option determines the time step (in day) of each burnup step. Notice that these inputs are time intervals in each step, but not the accumulated time.
* **Power** option determines the power density (in W/gHM) in each burnup step. The program acquires the actual total power according to the initial total density of the heavy metals. The actual power of each burnup region can also be acquired according to the power distribution of each burnup region in each burnup step. Users must input the volumes of the burnup cells correctly (see the input card of cells in section 3.2), to calculate the initial total density of the heavy metals and the power distribution.
* **SubStep** option determines the number of steps in the point burnup calculations (from 1 to 9999), 10 steps by default. For common users, it is not advised to change the default parameter by this option.
* **Inherent** option determines the absorption proportions and the mass proportions for inheriting the important nuclides, 0.99 and 0.999 by default respectively. About 1500 kinds of nuclides need to been handled for the point burnup calculations, while the critical calculations can (also just need to) inherit a part of these important nuclides. The way to inherit is: firstly, nuclides in the material card will always be inherit; then other nuclides will be inherited according to their proportions of absorption and mass, from high to low until the total proportion reaches to the absorption defined by users. If some important nuclides need to be focused on by users, they are advised to be written into the material cards, to guarantee that these nuclides will always be inherited. The more nuclides to be inherited, the more accuracy calculation results, while the larger consuming of time and memory.
* **AceLib** option determines the libraries for the inherited nuclides, such as “AceLib = .30c”. For the nuclides in the material cards, they will inherit their original libraries. To guarantee the accuracy of burnup calculation, users should determine the ACE libraries coinciding with the cell temperatures. Moreover, the library of this temperature should contain the nuclides lists as complete as possible. In the burnup calculation containing multiple temperatures, RMC can match the libraries according to the cell temperatures, while this function is not supported in this release of RMC.
* **Strategy** option determines the strategy of burnup step. **Strategy = 0** (by default) means using the starting point approximation burnup strategy, while **Strategy = 0** means using the predictor-corrector burnup strategy. Normally if the reasonable burnup steps are set, the starting point approximation burnup strategy can satisfy most of the requirement of burnup calculation.
* **Solver** option determines the solving method of the point burnup equation. **Solver = 1** means the transmutation trajectory method, **Solver = 2** (by default) means the Chebyshev rational approximation method, **Solver = 3** means the spline rational approximation method, and **Solver = 4** means the Laguerre polynomial approximation method. For common users, the default parameter is advised.
* **Parallel** option determines whether the parallel burnup calculations are used for the parallel critical calculations, which is only effective to the parallel version of RMC. **Parallel = 0** (by default) means the parallel burnup calculations are off, while **Parallel = 1** means it is on. In the mode of parallel burnup calculation, the burnup regions are equally assigned to each progress, which can execute the point burnup calculation separately. For the large-scale burnup calculation which contains large sums of burnup regions the parallel burnup calculation can reduce the calculation time significantly.
* **Outputcell** option determines the nuclides densities of cells, storing them in the files with suffix “.den\_tot”.

It should be pointed out that large sum of reactions rates need to be counted in the MC critical calculations for burnup calculations, consuming a lot of computing time. For the large-scale burnup calculations with hundreds or even thousands of burnup regions, the time consuming on the point burnup itself is aslo very large. Therefore, the users are advised to use the parallel version of RMC for the burnup calculations, and uses the parallel burnup mode (**Parallel = 1**) when it is necessary.

Theoretically, any number of burnup regions is supported in RMC, while actually it is limited to the computer hardware. The actual testing shows that the burnup calculations containing 10000 burnup regions need to consume about 1.5G memories. Therefore, on this basis, about 1.5G memories increase for every 10000 burnup regions increase.

## 8.3 Input examples of the burnup block

### 8.3.1 Burnup example of PWR cell

The burnup example of PWR cell has only one burnup region, i.e. Cell 3 whose temperature is 293.6K. The burnup history totally has 72 burnup steps, with the power density of 30W/gHM per burnup history. The burnup steps are: “3.333333 13.333333 16.666667 33.333333\*69”, coinciding with the accumulated burnup of “0.1 0.5 1.0 2.0 … 70.0 ” MWD/KgHM.

Example 8-1

|  |
| --- |
| ////// Burnup calculation of PWR pin. SHE Ding 2012-07-01 //////  UNIVERSE 0  cell 3 -1 mat = 1 vol = 1.0 // Fuel  cell 4 1 & -2 mat = 3 // Air  cell 5 2 & -3 mat = 4 // Zr  cell 6 3 & 4 & -5 & 6 & -7 mat = 5 // water  SURFACE  surf 1 cz 0.4096  surf 2 cz 0.4178  surf 3 cz 0.4750  surf 4 px -0.63 bc = 1  surf 5 px 0.63 bc = 1  surf 6 py -0.63 bc = 1  surf 7 py 0.63 bc = 1  MATERIAL  mat 1 -10.196  92235.30c 6.9100E-03  92238.30c 2.2062E-01  8016.30c 4.5510E-01  34079.30c 1.0E-25 36083.30c 1.0E-25 36085.30c 1.0E-25  38089.30c 1.0E-25 38090.30c 1.0E-25 39091.30c 1.0E-25  40093.30c 1.0E-25 40094.30c 1.0E-25 40095.30c 1.0E-25  40096.30c 1.0E-25 42095.30c 1.0E-25 42098.30c 1.0E-25  42099.30c 1.0E-25 42100.30c 1.0E-25 43099.30c 1.0E-25  44101.30c 1.0E-25 44102.30c 1.0E-25 44103.30c 1.0E-25  44105.30c 1.0E-25 44106.30c 1.0E-25 45103.30c 1.0E-25  45105.30c 1.0E-25 47109.30c 1.0E-25 47510.30c 1.0E-25  47111.30c 1.0E-25 50126.30c 1.0E-25 51125.30c 1.0E-25  51126.30c 1.0E-25 52527.30c 1.0E-25 52529.30c 1.0E-25  53127.30c 1.0E-25 53129.30c 1.0E-25 53131.30c 1.0E-25  53135.30c 1.0E-25 54131.30c 1.0E-25 54133.30c 1.0E-25  54134.30c 1.0E-25 54135.30c 1.0E-25 54136.30c 1.0E-25  55133.30c 1.0E-25 55134.30c 1.0E-25 55135.30c 1.0E-25  55136.30c 1.0E-25 55137.30c 1.0E-25 56138.30c 1.0E-25  56140.30c 1.0E-25 57139.30c 1.0E-25 57140.30c 1.0E-25  58141.30c 1.0E-25 58142.30c 1.0E-25 58143.30c 1.0E-25  58144.30c 1.0E-25 59143.30c 1.0E-25 60143.30c 1.0E-25  60144.30c 1.0E-25 60145.30c 1.0E-25 60147.30c 1.0E-25  60148.30c 1.0E-25 61147.30c 1.0E-25 61148.30c 1.0E-25  61548.30c 1.0E-25 61149.30c 1.0E-25 62147.30c 1.0E-25  62148.30c 1.0E-25 62149.30c 1.0E-25 62150.30c 1.0E-25  62151.30c 1.0E-25 62152.30c 1.0E-25 63153.30c 1.0E-25  63154.30c 1.0E-25 63155.30c 1.0E-25 63156.30c 1.0E-25  64155.30c 1.0E-25 64157.30c 1.0E-25 92234.30c 1.0E-25  92236.30c 1.0E-25 92237.30c 1.0E-25 92239.30c 1.0E-25  92240.30c 1.0E-25 93236.30c 1.0E-25 93237.30c 1.0E-25  93238.30c 1.0E-25 93239.30c 1.0E-25 94238.30c 1.0E-25  94239.30c 1.0E-25 94240.30c 1.0E-25 94241.30c 1.0E-25  94242.30c 1.0E-25 94243.30c 1.0E-25 94244.30c 1.0E-25  95241.30c 1.0E-25 95242.30c 1.0E-25 95642.30c 1.0E-25  95243.30c 1.0E-25 95244.30c 1.0E-25 96242.30c 1.0E-25  96243.30c 1.0E-25 96244.30c 1.0E-25 96245.30c 1.0E-25  96246.30c 1.0E-25 96247.30c 1.0E-25 96248.30c 1.0E-25  96249.30c 1.0E-25 97249.30c 1.0E-25 97250.30c 1.0E-25  98249.30c 1.0E-25 98250.30c 1.0E-25 98251.30c 1.0E-25  mat 3 -0.001  8016.30c 3.76622E-5  mat 4 -6.550  40000.60c -98.2  50000.42c -1.5  26000.55c -0.12  24000.50c -0.1  28000.50c -0.05  8016.30c -0.13  mat 5 9.9977E-02  1001.30c 6.6643E-02  8016.30c 3.3334E-02  sab 5 lwtr.60t  CRITICALITY  PowerIter population = 5000 30 230 // keff0 = 1.0  InitSrc point = 0 0 0  BURNUP  BurnCell 3  TimeStep 3.333333 13.333333 16.666667 33.333333\*69  Power 30\*72  Substep 10  Inherent 0.9999  AceLib .30c  outputcell 3 |

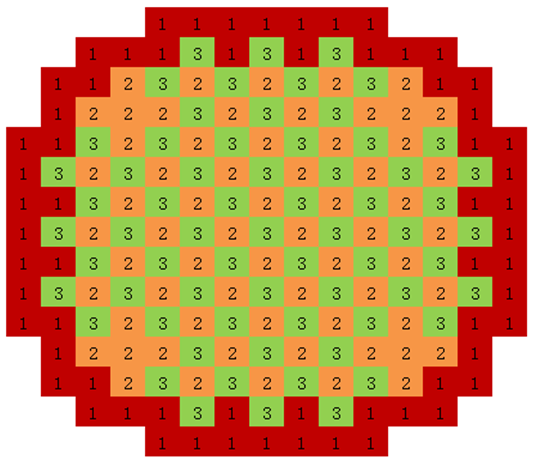
### 8.3.2 Burnup example of PWR assembly

Example 8-2 is an example of a PWR17×17 assembly which contains 264 burnup regions, adopting the mode of parallel burnup calculations (effective in the conditions of parallel calling). The nuclides densities of the fuel cells in 4 corners of the assembly will be output. The amount of computations needed is so large that the parallel computers are advised for calculations.

Example 8-2

|  |
| --- |
| ////// Burnup calculation of PWR 17\*17 assembly. SHE Ding 2012-07-01 //////  UNIVERSE 0  CELL 1 6 & -7 & 8 & -9 mat = 0 Fill = 8 // Assembly inside  CELL 2 -6 : 7 : -8 : 9 mat = 0 void = 1 // Assembly outside  UNIVERSE 8 lat = 1 pitch = 1.26 1.26 1 scope = 17 17 1 fill =  1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1  1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1  1 1 1 1 1 5 1 1 5 1 1 5 1 1 1 1 1  1 1 1 5 1 1 1 1 1 1 1 1 1 5 1 1 1  1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1  1 1 5 1 1 5 1 1 5 1 1 5 1 1 5 1 1  1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1  1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1  1 1 5 1 1 5 1 1 5 1 1 5 1 1 5 1 1  1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1  1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1  1 1 5 1 1 5 1 1 5 1 1 5 1 1 5 1 1  1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1  1 1 1 5 1 1 1 1 1 1 1 1 1 5 1 1 1  1 1 1 1 1 5 1 1 5 1 1 5 1 1 1 1 1  1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1  1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1  UNIVERSE 1 move = 0.63 0.63 0 // Fuel rod  cell 3 -1 mat = 1 inner = 1 tmp = 300 // Fuel  cell 4 1 & -2 mat = 3 inner = 1 // Air  cell 5 2 & -3 mat = 4 inner = 1 // Zr  cell 6 3 mat = 5 // water  UNIVERSE 5 move = 0.63 0.63 0 // Guide tube  cell 7 -4 mat = 5 inner = 1 // water  cell 8 4 & -5 mat = 4 inner = 1 // Air  cell 9 5 mat = 5 // water  SURFACE  surf 1 cz 0.4096  surf 2 cz 0.4178  surf 3 cz 0.4750  surf 4 cz 0.5690  surf 5 cz 0.6147  surf 6 px 0 bc = 1  surf 7 px 21.42 bc = 1  surf 8 py 0 bc = 1  surf 9 py 21.42 bc = 1  MATERIAL  mat 1 -10.196  92235.30c 6.9100E-03  92238.30c 2.2062E-01  8016.30c 4.5510E-01  34079.30c 1.0E-25 36083.30c 1.0E-25 36085.30c 1.0E-25  38089.30c 1.0E-25 38090.30c 1.0E-25 39091.30c 1.0E-25  40093.30c 1.0E-25 40094.30c 1.0E-25 40095.30c 1.0E-25  40096.30c 1.0E-25 42095.30c 1.0E-25 42098.30c 1.0E-25  42099.30c 1.0E-25 42100.30c 1.0E-25 43099.30c 1.0E-25  44101.30c 1.0E-25 44102.30c 1.0E-25 44103.30c 1.0E-25  44105.30c 1.0E-25 44106.30c 1.0E-25 45103.30c 1.0E-25  45105.30c 1.0E-25 47109.30c 1.0E-25 47510.30c 1.0E-25  47111.30c 1.0E-25 50126.30c 1.0E-25 51125.30c 1.0E-25  51126.30c 1.0E-25 52527.30c 1.0E-25 52529.30c 1.0E-25  53127.30c 1.0E-25 53129.30c 1.0E-25 53131.30c 1.0E-25  53135.30c 1.0E-25 54131.30c 1.0E-25 54133.30c 1.0E-25  54134.30c 1.0E-25 54135.30c 1.0E-25 54136.30c 1.0E-25  55133.30c 1.0E-25 55134.30c 1.0E-25 55135.30c 1.0E-25  55136.30c 1.0E-25 55137.30c 1.0E-25 56138.30c 1.0E-25  56140.30c 1.0E-25 57139.30c 1.0E-25 57140.30c 1.0E-25  58141.30c 1.0E-25 58142.30c 1.0E-25 58143.30c 1.0E-25  58144.30c 1.0E-25 59143.30c 1.0E-25 60143.30c 1.0E-25  60144.30c 1.0E-25 60145.30c 1.0E-25 60147.30c 1.0E-25  60148.30c 1.0E-25 61147.30c 1.0E-25 61148.30c 1.0E-25  61548.30c 1.0E-25 61149.30c 1.0E-25 62147.30c 1.0E-25  62148.30c 1.0E-25 62149.30c 1.0E-25 62150.30c 1.0E-25  62151.30c 1.0E-25 62152.30c 1.0E-25 63153.30c 1.0E-25  63154.30c 1.0E-25 63155.30c 1.0E-25 63156.30c 1.0E-25  64155.30c 1.0E-25 64157.30c 1.0E-25 92234.30c 1.0E-25  92236.30c 1.0E-25 92237.30c 1.0E-25 92239.30c 1.0E-25  92240.30c 1.0E-25 93236.30c 1.0E-25 93237.30c 1.0E-25  93238.30c 1.0E-25 93239.30c 1.0E-25 94238.30c 1.0E-25  94239.30c 1.0E-25 94240.30c 1.0E-25 94241.30c 1.0E-25  94242.30c 1.0E-25 94243.30c 1.0E-25 94244.30c 1.0E-25  95241.30c 1.0E-25 95242.30c 1.0E-25 95642.30c 1.0E-25  95243.30c 1.0E-25 95244.30c 1.0E-25 96242.30c 1.0E-25  96243.30c 1.0E-25 96244.30c 1.0E-25 96245.30c 1.0E-25  96246.30c 1.0E-25 96247.30c 1.0E-25 96248.30c 1.0E-25  96249.30c 1.0E-25 97249.30c 1.0E-25 97250.30c 1.0E-25  98249.30c 1.0E-25 98250.30c 1.0E-25 98251.30c 1.0E-25  mat 3 -0.001  8016.30c 3.76622E-5  mat 4 -6.550  40000.60c -98.2  mat 5 9.9977E-02  1001.30c 6.6643E-02  8016.30c 3.3334E-02  sab 5 lwtr.60t  CRITICALITY  PowerIter population = 2000 50 300 // keff0 = 1.0  InitSrc point = 0.63 0.63 0  BURNUP  BurnCell 3  TimeStep 3.333333 13.333333 16.666667 33.333333\*69  Power 30\*72  Substep 10  Inherent 0.9999  AceLib .30c  Strategy 0  Parallel 1  Solver 2  outputcell 1 > 1 > 3  1 > 17 > 3  1 > 273 > 3  1 > 289 > 3  PRINT  mat 0  csTally 0 |

### 8.3.3 Burnup example of PWR core



|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 1 | Assembly of 3.1% enrichment | 2 | Assembly of 2.6% enrichment | 3 | Assembly of 2.1% enrichment |

Figure 8-1 Layout of the PWR core

The 2D PWR core contains 193 fuel assemblies, outside which are the water reflector. The radius of core (with reflector) is 187.6 cm. The fuel assembly is in 17×17 array, containing 264 fuel pins and 25 water channels. These assemblies are divided into 3 types: 3.1%, 2.6% and 2.1% according to the enrichment of UO2 fuel. The fuel assemblies are arranged symmetrically in the core, as is shown in Figure 8-1. The burnup history contain 41 burnup steps totally, with the power density of 30W/gHM per burnup step, and the accumulated burnup is 20 MWD/KgHM.

Example 8-3

|  |
| --- |
| /////// PWR core burnup calculation SHE Ding 2013-07-01 /////////////  Universe 0  cell 1 -9 fill = 11 //core inside  cell 2 9 mat=0 void = 1 //core outside  UNIVERSE 11 move= -224.91 -224.91 0 lat=1 pitch=21.42 21.42 1 scope=21 21 1 fill=  8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8  8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8  8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8  8 8 8 8 8 8 8 1 1 1 1 1 1 1 8 8 8 8 8 8 8  8 8 8 8 8 1 1 1 3 1 3 1 3 1 1 1 8 8 8 8 8  8 8 8 8 1 1 2 3 2 3 2 3 2 3 2 1 1 8 8 8 8  8 8 8 8 1 2 2 2 3 2 3 2 3 2 2 2 1 8 8 8 8  8 8 8 1 1 3 2 3 2 3 2 3 2 3 2 3 1 1 8 8 8  8 8 8 1 3 2 3 2 3 2 3 2 3 2 3 2 3 1 8 8 8  8 8 8 1 1 3 2 3 2 3 2 3 2 3 2 3 1 1 8 8 8  8 8 8 1 3 2 3 2 3 2 3 2 3 2 3 2 3 1 8 8 8  8 8 8 1 1 3 2 3 2 3 2 3 2 3 2 3 1 1 8 8 8  8 8 8 1 3 2 3 2 3 2 3 2 3 2 3 2 3 1 8 8 8  8 8 8 1 1 3 2 3 2 3 2 3 2 3 2 3 1 1 8 8 8  8 8 8 8 1 2 2 2 3 2 3 2 3 2 2 2 1 8 8 8 8  8 8 8 8 1 1 2 3 2 3 2 3 2 3 2 1 1 8 8 8 8  8 8 8 8 8 1 1 1 3 1 3 1 3 1 1 1 8 8 8 8 8  8 8 8 8 8 8 8 1 1 1 1 1 1 1 8 8 8 8 8 8 8  8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8  8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8  8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8    UNIVERSE 1 lat=1 pitch=1.26 1.26 1 scope=17 17 1 fill=  10 10 10 10 10 10 10 10 10 10 10 10 10 10 10 10 10  10 10 10 10 10 10 10 10 10 10 10 10 10 10 10 10 10  10 10 10 10 10 40 10 10 40 10 10 40 10 10 10 10 10  10 10 10 40 10 10 10 10 10 10 10 10 10 40 10 10 10  10 10 10 10 10 10 10 10 10 10 10 10 10 10 10 10 10  10 10 40 10 10 40 10 10 40 10 10 40 10 10 40 10 10  10 10 10 10 10 10 10 10 10 10 10 10 10 10 10 10 10  10 10 10 10 10 10 10 10 10 10 10 10 10 10 10 10 10  10 10 40 10 10 40 10 10 40 10 10 40 10 10 40 10 10  10 10 10 10 10 10 10 10 10 10 10 10 10 10 10 10 10  10 10 10 10 10 10 10 10 10 10 10 10 10 10 10 10 10  10 10 40 10 10 40 10 10 40 10 10 40 10 10 40 10 10  10 10 10 10 10 10 10 10 10 10 10 10 10 10 10 10 10  10 10 10 40 10 10 10 10 10 10 10 10 10 40 10 10 10  10 10 10 10 10 40 10 10 40 10 10 40 10 10 10 10 10  10 10 10 10 10 10 10 10 10 10 10 10 10 10 10 10 10  10 10 10 10 10 10 10 10 10 10 10 10 10 10 10 10 10      UNIVERSE 2 lat=1 pitch=1.26 1.26 1 scope=17 17 1 fill=  20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20  20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20  20 20 20 20 20 40 20 20 40 20 20 40 20 20 20 20 20  20 20 20 40 20 20 20 20 20 20 20 20 20 40 20 20 20  20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20  20 20 40 20 20 40 20 20 40 20 20 40 20 20 40 20 20  20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20  20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20  20 20 40 20 20 40 20 20 40 20 20 40 20 20 40 20 20  20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20  20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20  20 20 40 20 20 40 20 20 40 20 20 40 20 20 40 20 20  20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20  20 20 20 40 20 20 20 20 20 20 20 20 20 40 20 20 20  20 20 20 20 20 40 20 20 40 20 20 40 20 20 20 20 20  20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20  20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20  UNIVERSE 3 lat=1 pitch=1.26 1.26 1 scope=17 17 1 fill=  30 30 30 30 30 30 30 30 30 30 30 30 30 30 30 30 30  30 30 30 30 30 30 30 30 30 30 30 30 30 30 30 30 30  30 30 30 30 30 40 30 30 40 30 30 40 30 30 30 30 30  30 30 30 40 30 30 30 30 30 30 30 30 30 40 30 30 30  30 30 30 30 30 30 30 30 30 30 30 30 30 30 30 30 30  30 30 40 30 30 40 30 30 40 30 30 40 30 30 40 30 30  30 30 30 30 30 30 30 30 30 30 30 30 30 30 30 30 30  30 30 30 30 30 30 30 30 30 30 30 30 30 30 30 30 30  30 30 40 30 30 40 30 30 40 30 30 40 30 30 40 30 30  30 30 30 30 30 30 30 30 30 30 30 30 30 30 30 30 30  30 30 30 30 30 30 30 30 30 30 30 30 30 30 30 30 30  30 30 40 30 30 40 30 30 40 30 30 40 30 30 40 30 30  30 30 30 30 30 30 30 30 30 30 30 30 30 30 30 30 30  30 30 30 40 30 30 30 30 30 30 30 30 30 40 30 30 30  30 30 30 30 30 40 30 30 40 30 30 40 30 30 30 30 30  30 30 30 30 30 30 30 30 30 30 30 30 30 30 30 30 30  30 30 30 30 30 30 30 30 30 30 30 30 30 30 30 30 30  UNIVERSE 8  cell 3 -6 mat = 5 tmp = 300  cell 4 6 mat = 5 tmp = 300  UNIVERSE 10 move = 0.63 0.63 0 // 3.1% Fuel rod  cell 13 -1 mat = 10 inner = 1 tmp = 300 // Fuel  cell 14 1 & -2 mat = 3 inner = 1 // Air  cell 15 2 & -3 mat = 4 inner = 1 // Zr  cell 16 3 mat = 5 tmp = 300 // water  UNIVERSE 20 move = 0.63 0.63 0 // 2.6% Fuel rod  cell 23 -1 mat = 20 inner = 1 tmp = 300 // Fuel  cell 24 1 & -2 mat = 3 inner = 1 // Air  cell 25 2 & -3 mat = 4 inner = 1 // Zr  cell 26 3 mat = 5 tmp = 300 // water  UNIVERSE 30 move = 0.63 0.63 0 // 2.1% Fuel rod  cell 33 -1 mat = 30 inner = 1 tmp = 300 // Fuel  cell 34 1 & -2 mat = 3 inner = 1 // Air  cell 35 2 & -3 mat = 4 inner = 1 // Zr  cell 36 3 mat = 5 tmp = 300 // water  UNIVERSE 40 move = 0.63 0.63 0 // Guide tube  cell 7 -4 mat = 5 inner = 1 tmp = 300 // water  cell 8 4 & -5 mat = 4 inner = 1 // Air  cell 9 5 mat = 5 tmp = 300 // water  Surface  surf 1 cz 0.4096  surf 2 cz 0.4178  surf 3 cz 0.4750  surf 4 cz 0.5690  surf 5 cz 0.6147  surf 6 cz 900  surf 9 cz 187.6 bc = 1  Material  mat 10 -10.2 // 3.1%  92235.30c 7.1421E-04  92238.30c 2.2044E-02  8016.30c 4.5515E-02  54134.30c 1.0E-25  54135.30c 1.0E-25  54136.30c 1.0E-25  mat 20 -10.2 // 2.6%  92235.30c 5.9902E-04  92238.30c 2.2157E-02  8016.30c 4.5513E-02  54134.30c 1.0E-25  54135.30c 1.0E-25  54136.30c 1.0E-25  mat 30 -10.2 // 2.1 %  92235.30c 4.8383E-04  92238.30c 2.2271E-02  8016.30c 4.5510E-02  54134.30c 1.0E-25  54135.30c 1.0E-25  54136.30c 1.0E-25  mat 3 -0.001  8016.30c 3.76622E-5  mat 4 -6.550  40000.60c -98.2  mat 5 -1.0034  1001.30c 6.66E-02  8016.30c 3.33E-02  sab 5 lwtr.60t  Criticality  poweriter keff0=1.0 population = 500000 200 500 batch = 10  initsrc cyl/z = 0 0 166 -1 1  BURNUP  BurnCell 13 23 33  TimeStep 3.333333 13.333333 16.666667\*39  Power 30 \*41  Substep 2  Inherent 0.999 0.999  AceLib .30c  Strategy 1  Parallel 1  Solver 2  PRINT  cstally 0  mat 0 |

# Chapter 9 Output Control

The output control block of RMC is used to define the output by the users. Especially for the large-scale burnup calculations in which large sums of output information are produced, the size of output files can be effectively reduced by the output control block.

## 9.1 Input card of the output control block

The input of the output control block is:

|  |
| --- |
| **PRINT**  **Mat <flag>**  **Keff <flag>**  **Source <flag>**  **CellTally <flag>**  **MeshTally <flag>**  **CsTally <flag>** |

In which,

* **PRINT** is the key word of the output control block.
* **Mat**, **Keff** and the other options determine whether the corresponding information will be output. **flag = 0** means the specific information will not be output, while **flag = 1** means they will be output.

Table 9-1 Input cards of output control block

|  |  |  |
| --- | --- | --- |
| Input card | Output resluts | Default options |
| **Mat** | Nuclides densities lists of all the material | flag = 1, output |
| **Keff** | *Keff* per generation | flag = 1, output |
| **Source** | Fission source information per generation | flag = 0, no output |
| **CellTally** | Results of the cell tallies | flag = 1, output |
| **MeshTally** | Results of the mesh tallies | flag = 1, output |
| **CsTally** | Results of the cross-section tallies, including the statistical results of one-group cross-sections in burnup calculations | flag = 1, output |

## 9.2 Input example of the output control block

For the burnup calculations containing large sums of burnup regions, it is advised to mask the output results of material and one-group cross-sections by the following input card, to avoid the productions of large sums of data files.

|  |
| --- |
| PRINT  Mat 0  CsTally 0 |

# Chapter 10 Plotting

The 2D sectional geometry/material plotting of models are supported in RMC. The images containing the geometry/material information (in PNG format) are produced according to the rectangular regions and image sizes defined by the users. This release of RMC Beta 2.0 only supports the plotting in rectangular regions which are vertical to the coordinate axises.

## 10.1 Input card of the plotting block

The input of the plotting block is:

|  |
| --- |
| **PLOT**  **[ColorScheme = <scheme number>] [Continue-calculation = <flag>]**  **PlotID <id> <ColorType> Pixels= <** **width height > Vertexes= < x1 y1 z1 x2 y2 z2>** |

In which,

* **PLOT** is the key word of theplotting block.
* **ColorScheme** option sets the color schemes of plotting (only for the colorized filled material). **Scheme number** can be an arbitrary positive integer, and the collocations of materials and colors can be changed randomly with different **Scheme number**. This option can adjust the color contrast of images, “**1**” by default.
* **Continue-calculation** option controls whether the calculations will be executed. “**0**” (buy default) means skipping the calculations and stopping the program after plotting, while “**1**” means the calculations will continue after plotting.
* **PlotID** option determines the parameters for plotting in which the **<id>** option sets the plotting identifiers which are used for identifying the images. The file names related to the images is “*inputfilename*\_plot\_*id*.png”.
* **ColorType** sets the color patterns of plotting. According to whether the cell boundaries will be plotted and whether to fill the material in colors, the plotting patterns of RMC include three types: material color pattern (**MatColor**), cell boundary pattern (**CellSurf**) and material cell pattern (**MatCell**). The **MatColor** pattern means filling the material in colors, **CellSurf** pattern means plotting the cell boundary with only the black line (one pixel wide), **MatCell** pattern means filling the material in colors, as well as plotting the cell boundary with the black line.
* **Pixels** sets the image sizes that **width** is the width of images and **height** is the height of images. The image will be more detailed with a larger image size, while larger memories as well. The image sizes are advised to be matched the rectangular size of plotting.
* **Vertexes** set the two vertexes of the rectangle for plotting to determine the rectangular region. **Vertex1(x1,y1,z1), Vertex2(x2,y2,z2)** are the left top vertex and right bottom vertex of the rectangle respectively (as is shown in Figure10-1). This release of RMC Beta 2.0 only supports the plotting in rectangular regions vertical to the coordinate axis, such as **Vertexes**= -15 70 20 65 10 20 represents a rectangle vertical to the z axis (in the plane of z=20). Note that the surface of plotting area is not recommended to be overlapping with any surface predefined in geometry.

**Vertex1(x1,y1,z1)**

**Vertex2(x2,y2,z2)**

Figure10-1 Diagram of the rectangle vertex method for plotting

## 10.2 Input example of the plotting block

Multiple images can be plotted per running, and the information such as the image sizes and colours will be stored in file “*inputfilename*.plot” which is output with the images.

|  |
| --- |
| PLOT ColorScheme=9  PlotID 1 MatColor Pixels=900 900 Vertexes=0 20 0 20 0 0  //PlotID 3 CellSurf Pixels=600 600 Vertexes=0 20 20 0 60 60  //PlotID 4 MatCell Pixels=1600 1200 Vertexes=0 0 60 80 0 0  PlotID 6 MatColor Pixels=9000 9000 vertexes=-167 167 0 167 -167 0 |

# Appendixes

## Appendix 1 Lists of the input/output files

The input files and output results of classical eamples are attached in the release package of RMC (see Appendix table-1). All the examples have been tested on the “Exploration 100” clustered systems of hundred trillion operations per second (Red Hat Enterprise Linux 5.6) of Tsinghua University.

Appendix table-1 Lists of the input files

|  |  |
| --- | --- |
| Input files | Examples descriptions |
| 3\_1\_PWR\_assembly | Example of geometric repeated structures, PWR assembly |
| 3\_2\_PWR\_core | Example of geometric repeated structures, PWR core |
| 3\_3\_MFR\_assembly | Example of geometric repeated structures,  hexagonal assembly |
| 3\_4\_MFR\_core | Example of geometric repeated structures,  hexagonal core |
| 6\_1\_Tally\_PWR\_pin | Example of tally, 轴向分段的PWR fuel pin |
| 6\_2\_Tally\_Hoogenboom\_core | Example of tally, MC whole core benchmark |
| 7\_1\_FSC\_slab | Example of source convergence acceleration, OECD source convergence benchmark |
| 7\_2\_FSC\_Hoogenboom\_core | Example of source convergence acceleration, MC whole core benchmark |
| 8\_1\_Burn\_PWR\_pin | Burnup example, PWR fuel pin |
| 8\_2\_Burn\_PWR\_assembly | Burnup example, PWR assembly |
| 8\_3\_Burn\_SCWR | Burnup example, SCWR of European Union |

## Appendix 2 Continuous-energy ACE libraries

A series of continuous-energy ACE libraries (with the suffix “.30c”) in 300K based on the ENDF7.0 are attached in the release package of RMC, containing about 400 nuclides which are shown in Appendix table-2. Moreover, parts of the thermalized libraries are also attached in the release package, as well as the libraries of nature nuclides (see Appendix table-3). The detailed information can be referred to the file “xdir”.

Appendix table-2 Lists of the continuous-energy ACE libraries

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| ZAID | Nuclide | ZAID | Nuclide | ZAID | Nuclide | ZAID | Nuclide |
| 1001 | H1 | 1002 | H2 | 1003 | H3 | 2003 | He3 |
| 2004 | He4 | 3006 | Li6 | 3007 | Li7 | 4007 | Be7 |
| 4009 | Be9 | 5010 | B10 | 5011 | B11 | 6000 | C |
| 7014 | N14 | 7015 | N15 | 8016 | O16 | 8017 | O17 |
| 9019 | F19 | 11022 | Na22 | 11023 | Na23 | 12024 | Mg24 |
| 12025 | Mg25 | 12026 | Mg26 | 13027 | Al27 | 14028 | Si28 |
| 14029 | Si29 | 14030 | Si30 | 15031 | P31 | 16032 | S32 |
| 16033 | S33 | 16034 | S34 | 16036 | S36 | 17035 | Cl35 |
| 17037 | Cl37 | 18036 | Ar36 | 18038 | Ar38 | 18040 | Ar40 |
| 19039 | K39 | 19040 | K40 | 19041 | K41 | 20040 | Ca40 |
| 20042 | Ca42 | 20043 | Ca43 | 20044 | Ca44 | 20046 | Ca46 |
| 20048 | Ca48 | 21045 | Sc45 | 22046 | Ti46 | 22047 | Ti47 |
| 22048 | Ti48 | 22049 | Ti49 | 22050 | Ti50 | 23000 | V |
| 24050 | Cr50 | 24052 | Cr52 | 24053 | Cr53 | 24054 | Cr54 |
| 25055 | Mn55 | 26054 | Fe54 | 26056 | Fe56 | 26057 | Fe57 |
| 26058 | Fe58 | 27058 | Co58 | 27458 | Co58m1 | 27059 | Co59 |
| 28058 | Ni58 | 28059 | Ni59 | 28060 | Ni60 | 28061 | Ni61 |
| 28062 | Ni62 | 28064 | Ni64 | 29063 | Cu63 | 29065 | Cu65 |
| 30000 | Zn | 31069 | Ga69 | 31071 | Ga71 | 32070 | Ge70 |
| 32072 | Ge72 | 32073 | Ge73 | 32074 | Ge74 | 32076 | Ge76 |
| 33074 | As74 | 33075 | As75 | 34074 | Se74 | 34076 | Se76 |
| 34077 | Se77 | 34078 | Se78 | 34079 | Se79 | 34080 | Se80 |
| 34082 | Se82 | 35079 | Br79 | 35081 | Br81 | 36078 | Kr78 |
| 36080 | Kr80 | 36082 | Kr82 | 36083 | Kr83 | 36084 | Kr84 |
| 36085 | Kr85 | 36086 | Kr86 | 37085 | Rb85 | 37086 | Rb86 |
| 37087 | Rb87 | 38084 | Sr84 | 38086 | Sr86 | 38087 | Sr87 |
| 38088 | Sr88 | 38089 | Sr89 | 38090 | Sr90 | 39089 | Y89 |
| 39090 | Y90 | 39091 | Y91 | 40090 | Zr90 | 40091 | Zr91 |
| 40092 | Zr92 | 40093 | Zr93 | 40094 | Zr94 | 40095 | Zr95 |
| 40096 | Zr96 | 41093 | Nb93 | 41094 | Nb94 | 41095 | Nb95 |
| 42092 | Mo92 | 42094 | Mo94 | 42095 | Mo95 | 42096 | Mo96 |
| 42097 | Mo97 | 42098 | Mo98 | 42099 | Mo99 | 42100 | Mo100 |
| 43099 | Tc99 | 44096 | Ru96 | 44098 | Ru98 | 44099 | Ru99 |
| 44100 | Ru100 | 44101 | Ru101 | 44102 | Ru102 | 44103 | Ru103 |
| 44104 | Ru104 | 44105 | Ru105 | 44106 | Ru106 | 45103 | Rh103 |
| 45105 | Rh105 | 46102 | Pd102 | 46104 | Pd104 | 46105 | Pd105 |
| 46106 | Pd106 | 46107 | Pd107 | 46108 | Pd108 | 46110 | Pd110 |
| 47107 | Ag107 | 47109 | Ag109 | 47510 | Ag110m1 | 47111 | Ag111 |
| 48106 | Cd106 | 48108 | Cd108 | 48110 | Cd110 | 48111 | Cd111 |
| 48112 | Cd112 | 48113 | Cd113 | 48114 | Cd114 | 48515 | Cd115m1 |
| 48116 | Cd116 | 49113 | In113 | 49115 | In115 | 50112 | Sn112 |
| 50113 | Sn113 | 50114 | Sn114 | 50115 | Sn115 | 50116 | Sn116 |
| 50117 | Sn117 | 50118 | Sn118 | 50119 | Sn119 | 50120 | Sn120 |
| 50122 | Sn122 | 50123 | Sn123 | 50124 | Sn124 | 50125 | Sn125 |
| 50126 | Sn126 | 51121 | Sb121 | 51123 | Sb123 | 51124 | Sb124 |
| 51125 | Sb125 | 51126 | Sb126 | 52120 | Te120 | 52122 | Te122 |
| 52123 | Te123 | 52124 | Te124 | 52125 | Te125 | 52126 | Te126 |
| 52527 | Te127m1 | 52128 | Te128 | 52529 | Te129m1 | 52130 | Te130 |
| 52132 | Te132 | 53127 | I127 | 53129 | I129 | 53130 | I130 |
| 53131 | I131 | 53135 | I135 | 54123 | Xe123 | 54124 | Xe124 |
| 54126 | Xe126 | 54128 | Xe128 | 54129 | Xe129 | 54130 | Xe130 |
| 54131 | Xe131 | 54132 | Xe132 | 54133 | Xe133 | 54134 | Xe134 |
| 54135 | Xe135 | 54136 | Xe136 | 55133 | Cs133 | 55134 | Cs134 |
| 55135 | Cs135 | 55136 | Cs136 | 55137 | Cs137 | 56130 | Ba130 |
| 56132 | Ba132 | 56133 | Ba133 | 56134 | Ba134 | 56135 | Ba135 |
| 56136 | Ba136 | 56137 | Ba137 | 56138 | Ba138 | 56140 | Ba140 |
| 57138 | La138 | 57139 | La139 | 57140 | La140 | 58136 | Ce136 |
| 58138 | Ce138 | 58139 | Ce139 | 58140 | Ce140 | 58141 | Ce141 |
| 58142 | Ce142 | 58143 | Ce143 | 58144 | Ce144 | 59141 | Pr141 |
| 59142 | Pr142 | 59143 | Pr143 | 60142 | Nd142 | 60143 | Nd143 |
| 60144 | Nd144 | 60145 | Nd145 | 60146 | Nd146 | 60147 | Nd147 |
| 60148 | Nd148 | 60150 | Nd150 | 61147 | Pm147 | 61148 | Pm148 |
| 61548 | Pm148m1 | 61149 | Pm149 | 61151 | Pm151 | 62144 | Sm144 |
| 62147 | Sm147 | 62148 | Sm148 | 62149 | Sm149 | 62150 | Sm150 |
| 62151 | Sm151 | 62152 | Sm152 | 62153 | Sm153 | 62154 | Sm154 |
| 63151 | Eu151 | 63152 | Eu152 | 63153 | Eu153 | 63154 | Eu154 |
| 63155 | Eu155 | 63156 | Eu156 | 63157 | Eu157 | 64152 | Gd152 |
| 64153 | Gd153 | 64154 | Gd154 | 64155 | Gd155 | 64156 | Gd156 |
| 64157 | Gd157 | 64158 | Gd158 | 64160 | Gd160 | 65159 | Tb159 |
| 65160 | Tb160 | 66156 | Dy156 | 66158 | Dy158 | 66160 | Dy160 |
| 66161 | Dy161 | 66162 | Dy162 | 66163 | Dy163 | 66164 | Dy164 |
| 67165 | Ho165 | 67566 | Ho166m1 | 68162 | Er162 | 68164 | Er164 |
| 68166 | Er166 | 68167 | Er167 | 68168 | Er168 | 68170 | Er170 |
| 71175 | Lu175 | 71176 | Lu176 | 72174 | Hf174 | 72176 | Hf176 |
| 72177 | Hf177 | 72178 | Hf178 | 72179 | Hf179 | 72180 | Hf180 |
| 73181 | Ta181 | 73182 | Ta182 | 74182 | W182 | 74183 | W183 |
| 74184 | W184 | 74186 | W186 | 75185 | Re185 | 75187 | Re187 |
| 77191 | Ir191 | 77193 | Ir193 | 79197 | Au197 | 80196 | Hg196 |
| 80198 | Hg198 | 80199 | Hg199 | 80200 | Hg200 | 80201 | Hg201 |
| 80202 | Hg202 | 80204 | Hg204 | 82204 | Pb204 | 82206 | Pb206 |
| 82207 | Pb207 | 82208 | Pb208 | 83209 | Bi209 | 88223 | Ra223 |
| 88224 | Ra224 | 88225 | Ra225 | 88226 | Ra226 | 89225 | Ac225 |
| 89226 | Ac226 | 89227 | Ac227 | 90227 | Th227 | 90228 | Th228 |
| 90229 | Th229 | 90230 | Th230 | 90232 | Th232 | 90233 | Th233 |
| 90234 | Th234 | 91231 | Pa231 | 91232 | Pa232 | 91233 | Pa233 |
| 92232 | U232 | 92233 | U233 | 92234 | U234 | 92235 | U235 |
| 92236 | U236 | 92237 | U237 | 92238 | U238 | 92239 | U239 |
| 92240 | U240 | 92241 | U241 | 93235 | Np235 | 93236 | Np236 |
| 93237 | Np237 | 93238 | Np238 | 93239 | Np239 | 94236 | Pu236 |
| 94237 | Pu237 | 94238 | Pu238 | 94239 | Pu239 | 94240 | Pu240 |
| 94241 | Pu241 | 94242 | Pu242 | 94243 | Pu243 | 94244 | Pu244 |
| 94246 | Pu246 | 95241 | Am241 | 95242 | Am242 | 95642 | Am242m1 |
| 95243 | Am243 | 95244 | Am244 | 95644 | Am244m1 | 96241 | Cm241 |
| 96242 | Cm242 | 96243 | Cm243 | 96244 | Cm244 | 96245 | Cm245 |
| 96246 | Cm246 | 96247 | Cm247 | 96248 | Cm248 | 96249 | Cm249 |
| 96250 | Cm250 | 97249 | Bk249 | 97250 | Bk250 | 98249 | Cf249 |
| 98250 | Cf250 | 98251 | Cf251 | 98252 | Cf252 | 98253 | Cf253 |
| 98254 | Cf254 | 99253 | Es253 | 99254 | Es254 | 99255 | Es255 |
| 100255 | Fm255 |  |  |  |  |  |  |

Appendix table -3 Lists of the libraries of nature nuclides

|  |  |  |  |
| --- | --- | --- | --- |
| 6000.60c | 12000.60c | 14000.60c | 16000.60c |
| 17000.60c | 18000.59c | 19000.60c | 20000.60c |
| 22000.60c | 23000.60c | 24000.50c | 26000.55c |
| 28000.50c | 29000.50c | 30000.42c | 31000.60c |
| 40000.60c | 42000.60c | 47000.55c | 48000.51c |
| 49000.60c | 50000.42c | 51000.42c | 54000.42c |
| 63000.42c | 64000.35c | 72000.60c | 74000.55c |
| 77000.55c | 78000.42c | 82000.50c |  |

## Appendix 3 Burnup libraries

A series of burnup libraries are attached in the release package, containing the one-group cross-sections of 1487 nuclides and the fission yield of 30 nuclides. These burnup libraries are adapted to PWR, and the burnup libraries for fast reactors will be provided in the following release of RMC.

1. Annotation: In the version of RMC-Beta1.0, “!” can be used as annotations. In RMC-Beta2.0 and future versions, “!” will be used as“logical negation”in boolen expressions for the surface of cells, and no longer be used as annotations. [↑](#footnote-ref-1)
2. Notice：In the **Scope** option of “Universe”, the parameter “-1” means only a single level of infinite meshes in this direction. [↑](#footnote-ref-2)